Advances in Industrial Control

David H. Owens

Iterative Learning Control

An Optimization Paradigm

Springer

Advances in Industrial Control

More information about this series at http://www.springer.com/series/1412

David H. Owens

Iterative Learning Control

An Optimization Paradigm

Springer

David H. Owens
University of Sheffield
Sheffield
UK

ISSN 1430-9491 ISSN 2193-1577 (electronic)
Advances in Industrial Control
ISBN 978-1-4471-6770-9 ISBN 978-1-4471-6772-3 (eBook)
DOI 10.1007/978-1-4471-6772-3

Library of Congress Control Number: 2015950040

Springer London Heidelberg New York Dordrecht

Printed on acid-free paper

Springer-Verlag London Ltd. is part of Springer Science+Business Media
(www.springer.com)

To my wife Rose for her patience and support during the writing of this text

Series Editors' Foreword

The series *Advances in Industrial Control* aims to report and encourage technology transfer in control engineering. The rapid development of control technology has an impact on all areas of the control discipline. New theory, new controllers, actuators, sensors, new industrial processes, computer methods, new applications, new philosophies,…, new challenges. Much of this development work resides in industrial reports, feasibility study papers and the reports of advanced collaborative projects. The series offers an opportunity for researchers to present an extended exposition of such new work in all aspects of industrial control for wider and rapid dissemination.

In the 1980s, "Emerging Technologies" was the theme of many conferences and the published control literature. Techniques such as fuzzy-logic control (FLC), genetic algorithms for design optimization and iterative learning control (ILC) were just some of the avenues explored. Iterative learning control was viewed as an important development owing to the technological revolution involving the large-scale introduction of repetitive robotic manipulators in manufacturing processes. An iconic picture of this change was the car-body paint shop containing only robots making repetitive paint-spraying motions. It took some years before the control literature caught up with technological developments and the *Advances in Industrial Control* series was fortunate to publish a seminal monograph in the field: Kevin L. Moore's text *Iterative Learning Control for Deterministic Systems* (ISBN 978-1-4471-1914-2, 1992).

For some years, Professor David Owens has researched the modelling and analysis of systems with repetitive dynamics, and has stimulated interest in the topic. This work arose from study of the control of underground coal-cutters and culminated in a theory of "multipass processes" that was developed in 1976 with follow-on applications introduced by Professor John Edwards. Subsequent research and applications experience led to substantial contributions (with collaborator Professor Eric Rogers, and others) in the area of repetitive control systems, but more specifically, since 1996, in the field of iterative learning control.

Professor Owens's seminal contribution was the introduction of optimization to the ILC community in the form of "norm optimal iterative learning control

(NOILC)". This idea has now evolved into a fully fledged theory and the *Advances in Industrial Control* series is fortunate to include this important monograph: *Iterative Learning Control: An Optimization Paradigm* by Professor Owens, describing the full optimization-based theory.

In the 1970s, postgraduates in control engineering and engineers were reading a control literature that was influenced by the mathematics of "functional analysis". Textbooks such as *The Calculus of Variations, Functional Analysis and Optimal Control Problems* by E.K. Blum (1967), *Optimization by Vector Space Methods* by D.G. Luenberger (1969), *Functional Analysis and Numerical Mathematics* by J.M. Ortega and W.C. Rheinboldt (1970), and *Functional Analysis in Modern Applied Mathematics* by R.F. Curtain and A.J. Prichard (1977) were all typical inspirational textbooks of the period.

One of the key attractions of functional analysis was the potential for abstraction to generalize problem description, analysis and solution. This is an aspect that Professor Owens has exploited fully in formulating the new optimization paradigm presented for iterative learning control. It is for this reason that, after an introductory Chap. 1, there is a substantial Chap. 2 on "Mathematical Methods" that presents the functional analysis topics of Banach, Hilbert and real Hilbert spaces. These presentations are followed by Chaps. 3 and 4 on state-space models (both continuous and discrete systems) and matrix models, respectively. The analysis of iterative learning control begins in Chap. 5 and continues through to Chap. 11.

The presentations of ideas begin with general ILC problem formulations that evolve through precisely specified issues and algorithms. In these chapters the analytical power of the functional analysis techniques is fully revealed. The monograph is self-contained and will be essential reading for those researchers and engineers working in the field of iterative learning control. Postgraduates and academics in the mechanical engineering, control engineering and applied mathematical communities will also find the monograph of interest. Professor Owens's text nicely complements two other recent entries to the *Advances in Industrial Control* monograph series on the ILC topic:

- *Real-time Iterative Learning Control* by Jian-Xin Xu, Sanjib K. Panda and Tong H. Lee (ISBN 978-1-84882-174-3, 2009)
- *Practical Iterative Learning Control with Frequency Domain Design and Sampled Data Implementation* by Danwei Wang, Yongqiang Ye, and Bin Zhang (ISBN 978-981-4585-59-0, 2014)

About the Author

Professor Owens's academic and engineering career has spanned some four decades. He was Head of the School of Engineering and Computer Science at the University of Exeter (1990–1999) and Head of Dynamics and Control in the Department of Mechanical Engineering at the University of Strathclyde (1985–1990). At the

University of Sheffield he has been Head of the Department of Automatic Control and Systems Engineering (1999–2008) and Dean of the Faculty of Engineering (2002–2006). Since 2009, he has been an Emeritus Professor of Control and Systems Engineering in the Department of Automatic Control and Systems Engineering at the University of Sheffield, UK. He is a Visiting Professor at The University of Southampton, UK, and a scientific collaborator with the Italian Institute of Technology, Genova, Italy. Professor Owens is a Fellow of the Royal Academy of Engineering, UK. He is also a Fellow of the Institution of Engineering and Technology (IET) and of the Institution of Mechanical Engineering (IMechE).

Scotland, UK

Michael J. Grimble
Michael A. Johnson

Preface

Control Theory and Control Systems Design has a history of over sixty years and covers a wide range of topics, many of which have relevance to Control Engineering practice and the process of education and training. Iterative Learning Control (ILC for short) is an area of control theory and applications that has risen in its relevance and popularity over the past 30 years. It is seen, for example, in applications of control where a machine is required to complete a specific tracking task over a finite time interval [0, T] to a high degree of accuracy. The accuracy required is taken to be beyond the practical capabilities of feedback control. The additional features of the system that makes improved accuracy possible are the assumptions that the task is repeated an indefinite number of times, so that (possibly large) errors are tolerable in the first few repetitions and that information on previously seen tracking errors, inputs and other signals can be measured and stored.

Systems with these properties include manufacturing and rehabilitation robotics and systems for mechanical testing of components and component assemblies in the automotive and related industries. In the automotive arena, this includes the testing of whole car performance using laboratory test rigs to replicate accurately road or track conditions. The words Iterative Control are used to describe the repetitive nature of the system behaviour and the need for control action. The word iterative is chosen by analogy with iteration in numerical analysis where the repetition is used to sequentially (iteration to iteration) reduce the error in obtaining the solution of, say, a set of algebraic equations. The natural choice of the name "repetitive control" was rejected by the community as this has been used previously to describe feedback control design for the tracking of periodic reference signals of known period.

ILC is naturally a branch of control theory and control systems' design but differs from its more traditional counterparts owing to the need to include both times during the repetition (iteration or trial) and the iteration index in data descriptions. This is done by setting $f_k(t)$ to be the value of a signal f at time "t" on iteration "k". The control problem therefore has two independent variables (t, k).

Control of the system not only requires proper and effective control during an iteration (using normal classical criteria for control performance on [0, T]) but also includes the need to control performance over successive iterations with indices k = 0,1,2,3,4,…. The aim is to improve performance from iteration to iteration. The ultimate aim is to provide highly accurate tracking which, in the mathematical limit as "k" tends to infinity becomes *perfect tracking*. Ideally, this property will be retained in the presence of modelling errors.

The mechanism for achieving this perfect tracking is the transfer of data from iteration to iteration. This transfer of data is similar to the process of "learning" from experience. Hence the subject title "Iterative Learning Control". Control design in ILC is the choice of control algorithm to generate an input signal magnitude $u_k(t)$ at time t on iteration k such that the application of this signal over the interval $t \in [0, T]$ leads to improved tracking as described above. In more details, design requirements include:

1. Stability during each iteration.
2. Asymptotic convergence of the tracking error to zero as $k \to \infty$.
3. Acceptable behaviour of the tracking error from iteration to iteration by,
    a. achieving reasonable improvement within a few iterations and
    b. ensuring acceptable convergence and performance despite plant modelling errors.
4. Implementation options that range from advanced control to simpler forms.

A control theory for ILC is hence at least as rich as that seen in classical control with opportunities for feedback and feedforward structures, consideration of output feedback and state feedback strategies, the possibilities for time domain and frequency domain design tools, adaptive and predictive algorithms, optimal control and many more. This is not to mention the need for linear, nonlinear and hybrid systems versions of the theories! A single text covering the whole spectrum of possibilities is not feasible and, indeed, many of these problems are not yet fully analysed and solved to produce design strategies. For this reason the text examines one particular coherent body of knowledge within the current ILC spectrum with the aim of, first, bringing the work together in an integrated whole for researchers, students and interested users and, second, presenting new results, algorithms and insights to add value to the literature.

More precisely, this text concentrates on one particular paradigm and focusses on linear systems, although it is noted that extensions to cover the case of some nonlinear systems are possible using, for example, linearization-based methodologies. Motivated by the natural desire to systematically reduce the tracking error from iteration to iteration, the chosen paradigm is that of the use of optimization as a tool for algorithm design. The text covers several model types including both single-input, single-output (SISO) and multi-input, multi-output (MIMO)

continuous and discrete time, state-space systems. There is some emphasis on discrete systems as the essentials of the theory and algorithm development are more easily derived and understood in this format. It is also consistent with the likely implementation using digital control hardware and software and notionally provides an insight into the continuous time case by letting the sampling rate become infinite. Those reading the text will understand that the continuous time case introduces many additional mathematical complexities that need to be addressed for technical completeness. Much of this "infinite dimensional" material is included but the author believes that it is neither necessary for an understanding of the basic concepts nor, indeed, relevant to many applications.

Optimization is used as the paradigm for algorithm development because of its proven ability to guarantee, in the absence of modelling errors, monotonic reductions in a norm of the tracking error time series from each iteration k to the next iteration $k+1$. This reduction has many interpretations, the simplest being the reduction of the "energy" in the tracking error. The energy interpretation is related to the use of the mean square error as the norm. The consequences of this simple observation are the subject of this text which, following a discussion of some of the relevant history and applications of ILC algorithms in the introductory chapter, explores the known consequences and design options available. The rest of the text is divided into several parts:

1. The design of algorithms might be expected to be model dependent. This is true but the construction of algorithms and the derivation of their properties is most easily seen using an operator description of the model and regarding input, output and other signals as elements of appropriate Hilbert spaces. This level of abstraction is analogous to the use of transfer function descriptions of linear systems. As an aid to study, the reader is provided with a chapter of mathematical methods which provides a summary of the essential properties of operators between Hilbert spaces and the geometrical interpretation of signals and their relationships. This is followed by material that looks closely at the structure of discrete time, state space models and the use of the (matrix) *supervector description*.
2. Many of the ideas and algorithms apply widely. The overview and formulation of ILC is constructed using the *language of operator theory* and recursive relationships of the typical form $e_{k+1} = Le_k, k \geq 0$ in a Hilbert space $\mathscr{Y}$. Convergence of the solution sequence $\{e_k\}_{k\geq 0}$ to a limit is related to the spectrum of the operator $L : \mathscr{Y} \to \mathscr{Y}$. If $\mathscr{Y}$ is finite dimensional, the results are familiar from matrix theory but the case when $\mathscr{Y}$ is infinite dimensional is more complex. Fortunately, for the purposes of optimization, only the case of L being self adjoint is relevant to this text. This case is however important for continuous time systems for example.

3. Although not obviously connected to optimization concepts, a fairly detailed examination of so-called *inverse model algorithms* is then provided. This approach assumes the existence of a left or right inverse to the plant operator and motivates a set of Iterative Control algorithms that provide simple monotonic error convergence at a rate described by a single gain parameter. This has practical value but its real value in the following optimization approaches lies in robustness analysis. The robustness theory presented in this text underpins the need for *positive real multiplicative modelling errors*. The results can be converted into frequency domain tests when the model has a discrete time, state-space form. By the end of the text, the reader will have made the surprising observation that this robustness theory applies, with suitable modification, to many of the algorithms described.
4. The first step towards the optimization paradigm is made in the consideration of *gradient* or *steepest descent* algorithms motivated by familiar numerical optimization methods. The gradient is naturally described by the adjoint plant operator and provides a link to the *co-state* equations that appear in optimal control theory. Similar algorithm analyses are provided together with a robustness analysis which again has a frequency domain form for discrete time state space systems that reappears for more complex algorithms later in the text.
5. The central section of the text contains the basic concepts of *Norm Optimal Iterative Learning Control* (NOILC) This algorithm has strong connections to linear quadratic optimal control theory and proceeds by minimization of a sequence of quadratic objective functions. For discrete state-space systems, it has a realization using either feedforward computations or familiar feedback structures based on solutions of time dependent matrix Riccati equations. The algorithm is a benchmark algorithm in the sense that monotonic error convergence is always guaranteed with convergence rates being influenced by the relative weighting of error and input terms. Greater insight into the behaviour is obtained using eigenvalue/singular value analysis and approximate eigenvectors constructed from frequency domain considerations. A robustness analysis is presented that has close links to inverse model and gradient results, probably because the NOILC paradigm can be regarded as being both a descent algorithm and an approximation to an inversion process.
6. The power of the NOILC philosophy and the use of operator descriptions are demonstrated by the descriptions of natural extensions of the algorithm. These include *intermediate point control problems, tracking for multi-rate sampled systems and systems where the initial condition on each iteration can be varied*. The concept of *Multi-task Algorithms* unifies these control problems as a mix of these and similar variations. All of these variations are described by the NOILC relationships but differ in form when converted from the operator description to more familiar state-space equations.
7. In all of these cases, the performance of the control can be enhanced by including one or both of two important features:

a. by searching for a solution to the tracking problem that also minimizes an additional *Auxiliary Objective Function* representing additional control objectives and/or
b. improving convergence rates by basing computations on a *multi-model* that enhances performance using optimization problems incorporating predictions of future errors. This resultant *Predictive Norm Optimal Iterative Learning Control* algorithm, when applied to state-space systems, can again be implemented using either feedforward computations or feedback realizations using the Riccati matrix and states of the multi-model.

In all cases, the robustness theory needed has a very similar structure to that of inverse model and gradient algorithms.

8. The apparent "perfection" of the NOILC approach is, however, misleading as proofs of convergence do not provide full information about rates of convergence. This is convincingly demonstrated by a consideration of *non-minimum-phase discrete, state-space systems*. For inverse model control, practical problems will then occur as the inverse system is unstable. For gradient and NOILC algorithms, the problem appears as a plateauing/flatlining effect where, after a period of fast error norm reduction, a long period of extremely slow convergence (represented by infinitesimal changes in error and error norm) can occur. In practical terms, the algorithm may fail to achieve acceptable tracking accuracy in the desired number of iterations despite the theoretical proof of ultimate convergence. The magnitude of the problem is assessed and shown to depend on the time interval length and the structure of the initial error e_0 .
9. The operator theory approach has great power but does not easily make possible, for example, the inclusion of constraints. The chapters on *Successive Projection* methodologies provide ways forward. They are equivalent to NOILC and its variants in the absence of constraints but, being based on projection onto closed, convex sets, allow many *convex constraints to be included* whether they be input and output constraints or constraint objectives for auxiliary variables. The geometry of the approach is used to *create accelerated algorithms* using extrapolation mechanisms. An interesting interpretation of the ideas is given in the section on *Loop Management Algorithms* where, rather than using ILC as an automated control algorithm, it can be used as a decision support aid for the human operators who are supervising the iteration process.
10. One, particularly interesting, algorithm that further demonstrates the potential of successive projection is the introduction of the new *Notch Algorithm* which uses successive projection onto sets defined by modified plant dynamics. The modification is parameterized by a single parameter σ^2 that can be related to a property of *Approximate Annihilation* of spectral components in the error close to that value. By varying σ^2 from iteration to iteration, different parts of the spectrum are reduced substantially producing rapid convergence in a systematic

way that resembles an approximate inverse algorithm. Robustness may however be reduced if the choices of σ^2 are not constrained.

11. The final chapter examines the idea of *Parameter Optimal Iterative Learning Control* (POILC) by replacing optimization over input signals by optimization over the parameters in a linear input update equation containing only known error data and a finite number of free parameters. This reduces the dimension of the optimization problem and defines the required parameter values using formulae and off-line computation. The approach has close links to NOILC (when the number of parameters is "large") but, more generally, retains the property of monotonic error norm reduction and simplifies both the control implementation and its associated computations. The choice of parameterization is free for the user to choose but can be suggested by the previous chapters in the form of approximations to inverses, gradients and norm optimizers. An examination of the inverse model and other cases suggests that the approach will be robust provided that gains are "low" and a positive real condition is again satisfied.

The material in the text has been chosen and ordered to *tell a story* rather than reflect the historical development of the ideas. Emphasis is placed on providing rigour with understanding with the aim of informing and preparing readers for their own studies, research or applications. The text has many sections that hopefully help readers find, quickly, the issues that interest them. The reader should note the following declarations:

1. Although based on optimization concepts, many of the results and algorithms presented are new to the literature. For those algorithms that have been published in the open literature, the text provides a more detailed analysis of their properties and includes new, previously unreported robustness characterizations and methodologies.
2. The author has aimed for consistency in notation but has had to deal with the finite nature of the alphabet and the need for extensive use of subscripts and superscripts. For example, the subscript y_j will typically denote the signal on iteration j but it may also denote the *jth* element in the column matrix y. It is the author's view that the context of the use of the symbol will provide the right interpretation.
3. The reader should understand that the text is motivated by the perceived practical needs of the subject when used in applications but also has the important objective of addressing what are thought to be important scientific issues associated with the topic, some of which are speculative. Applications of the ideas in controlled laboratory conditions has, to date, provided evidence that the ideas can translate into practically convergent procedures and this positive outcome has been supported by a number of industrial tests (subject to commercial, in confidence, agreements). However, it must be recognized that real-world systems present a whole spectrum of problems that the theory has not considered. No attempt will be made to list these problems exhaustively but they clearly include

the presence of severe nonlinearity, substantial noise, unrepeatable/unpredictable disturbances and time variation of parameters. In addition, hardware issues of a poor experimental setup or combinations of plant dynamics, parameter choices and reference signals that maximize the sensitivity of the approaches to such problems will also often need to be addressed. In view of these comments, the reader uses the ideas and algorithms at his or her own risk. The author has made every effort to ensure that the text is both accurate and understandable. He believes that the methods have both scientific and engineering merit but, given the uncertainties, he accepts no responsibility for any unacceptable outcomes arising from the use of the ideas or algorithms described in this text.

Sheffield, UK David H. Owens
April 2015

Acknowledgments

The author wishes to acknowledge students and colleagues who have played some role in the work described in this text. His interest in repetition in physical engineering systems was sparked early in his career in a collaboration, on multipass processes in mining systems, with Professor John Edwards of the Department of Control Engineering at the University of Sheffield, UK.

Professor Eric Rogers of Southampton University was the first of many research students in multipass/repetitive systems with his work forming the basis of a long and very productive research collaboration. Over the past 22 years, over fourteen research students have contributed to the Iterative Control research outcomes. Particular mention should be made of Dr. Notker Amann (who was part of the early development of Norm Optimal Iterative Learning Control), Dr. Bing Chu (who played a role in developing the ideas of Successive Projection) and Dr. Kevin Feng (who supported the development of the initial ideas of Parameter Optimal Iterative Learning Control).

Other senior research staff who made a substantial contribution include Dr. Jari Hätönen whilst the collaborations with Professor Steve Daley, Professor Paul Lewin, Dr. Chris Freeman and Dr. Bing Chu of the University of Southampton have been both valuable and enjoyable, particularly in the areas of identifying and evaluating applications in mechanical test, manufacturing robotics and rehabilitation robotics.

The author is an Emeritus Professor in the Department of Automatic Control and Systems Engineering at the University of Sheffield, a Visiting Professor in Electronics and Computer Science at the University of Southampton and a Visiting Researcher in the Robotics Division at the Italian Institute of Technology in Genova (Italy). Thanks are due to these organizations for providing the facilities to make this work possible and for their help in funding research and travel during the period of writing this text.

Finally, every researcher has had his or her role models and mentors. I am in debt to Professor Brian Knowles, Professor David Mayne FRS and the late Professor Jan Willems for setting me on my path and showing me what research is and should be.

Contents

Chapter 1
Introduction

Iterative Learning Control sits on the shoulders of the subject of Control Engineering, adding to its diversity and capabilities in a very specific way. Control Engineering is now a long established discipline that has worked with other areas of Engineering and Science in the design and working efficiency of almost every machine used by mankind including household devices, transportation systems, power generation plant and equipment that makes satellite and spacecraft technology feasible, operationally reliable and safe. Its enthusiasts think of it as the *unsung hero* of modern technology as few complex, multi-component systems work effectively without active control intervention either through algorithms embedded in electro-mechanical devices, computer control systems or simpler rule-based operator guidelines. Yet much of the hardware and software of control technology is hidden from the user who, as a consequence, can be forgiven for taking the beneficial consequences of good control for granted.

Given a desired system behaviour, control systems are essentially objects driven by input data streams and creating output data streams. This is because effective control requires measurement or estimation of relevant output and other signals as the basis for computation of a suitable intervention using available actuators. Although simply stated, the reality is that control systems take a wide variety of forms and design is a skilled process. The complexities of control systems design are reflected in a number of challenges including

1. The system to be controlled, often called the *plant*, typically has its own temporal and/or spatial dynamics. Dynamics can be linear or nonlinear. Behaviours of apparently complex plant can be quite simple to characterize and control whilst those of even apparently simple systems can exhibit more complexity ranging from instability to oscillation to counter-intuitive non-minimum phase characteristics to time variation or structural change.
2. The accuracy and capabilities of sensors and actuators available to the designer may be limited by cost, availability and/or the need for a natural simplicity to make commissioning and maintenance easy.

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_1

As a consequence, the need for general theoretical principles to underpin and guide the design process has been essential. Traditionally, the two guiding design principles have been the use of *feedback* and *feedforward* loops arising from the simple intuitions that effective control intervention at a given time can only be undertaken based on, and with a knowledge of, current and past behaviour of the system (i.e. data feedback) and a knowledge of the desired behaviour (i.e. feedforward elements). The general design challenge is to choose control elements and parameters to achieve the desired plant behaviour.

1.1 Control Systems, Models and Algorithms

The wide need for control technology and the complexities briefly stated above have led to the development of an underlying *Control Science* that can be used in the education of practitioners, to underpin the understanding of possibilities, to identify the source of limitations to achievable controlled performance, to develop and analyse established control principles and to extend the range of controlled behaviour using new design paradigms. The search for general principles and processes for control systems analysis and design for dynamic processes has inevitably led to the extensive use of mathematical techniques. These have not only included matrix theory, ordinary differential equations and the Laplace transform but also many other branches of mathematics including aspects of optimization theory, statistical estimation theory, functional analysis and operator theory and, for highly nonlinear systems, differential geometry.

To make use of the power of this mathematics, a *mathematical model of the plant* is placed at the centre of the discussion. This model can have many forms but, for the purposes of this text, it will typically take the form of a set of linear or nonlinear differential or difference equations with a defined initial condition. These "multi-input multi-output" (MIMO) models relate the behaviour of m defined *output signals* $y_1(t), y_2(t), \ldots, y_m(t)$ over a time interval $t \geq t_0$ to a set of ℓ *input (actuation) signals* $u_1(t), u_2(t), \ldots, u_\ell(t)$. Control objectives are typically, but not exclusively, described as the need to track a set of *demand* or *reference signals* $r_1(t), r_2(t), \ldots, r_m(t)$ defined on $t \geq t_0$. Without loss of generality, it is usual to take $t_0 = 0$ by simply "re-calibrating the clock".

Good tracking in this context means that the controlled system is *stable* and that the tracking error signals $e_j(t) = r_j(t) - y_j(t), 1 \leq j \leq m$, have acceptable magnitude and form. The notion of acceptable varies from application to application but errors are inevitably non-zero because of the control system's feedback structure and the reality that the behaviour predicted by the model does not match the real, physical plant behaviour exactly. This *plant-model mismatch* or *modelling error* is unavoidable in the real world. If a property of the controlled system predicted by the model is retained by the real plant then that property is said to be *robust*. The most important property of a controlled system is that its stability is robust but the idea applies more generally.

Plant mathematical models are the core of the design process, being used not only for analysis purposes but also for simulation of uncontrolled (*open-loop*) and controlled (*closed-loop*) behaviours. Design is model-based and the nature of the model has an inevitable impact, in off-lines design studies, on the control system chosen. An important observation, however, is that the implemented controller can also be model based in the sense that the plant model can form a part of the control scheme itself. In this sense, the idea of a controller as the manifestation of an *algorithm* (embedded in hardware and/or software) manipulating measured data from $t' \leq t$ at time t to construct an input signal value $u(t)$ is a core component of control science philosophy and control engineering practice. Algorithms are, at their most general, rule-based with rules expressed in a variety of formats including "IF X THEN Y" type rules, formulae such as the well known ideas of proportional plus integral plus derivative (PID) control

$$u_j(t) = \sum_{p=1}^{m} \left[K_{jp}^P e_p(t) + K_{jp}^I \int_0^t e_p(t')dt' + K_{jp}^D \frac{de_p(t)}{dt} \right], \quad 1 \leq j \leq \ell, \quad (1.1)$$

and/or computational rules that are supported by real-time, on-line simulation of a plant model. Parameters used in the computations may be specified and kept constant or may themselves vary with time if the control structure contains *adaptive* elements or on-line model building tools such as *system identification* or *parameter estimation* techniques are used.

In most classical control topics, tracking control is constructed as the need to track $r(t)$ on the notionally infinite interval $0 \leq t < \infty$. There are however situations where control on a *finite* time interval $0 \leq t \leq T$ is the essence of the problem. This is the case for the area of *Iterative Learning Control*.

1.2 Repetition and Iteration

The structure of a good control system design will reflect both the plant dynamics and the nature of the control objective. For example, in classical feedback control systems design, the requirement to track step demands accurately leads to the use of integral action in the control element. In the context of this text, the special property of the plant is the repetitive nature of its operation. Repetition is a common feature in control applications and takes several forms.

1.2.1 Periodic Demand Signals

The simplest notion of repetition is evident in the situation where the demand signal $r(t)$ is periodic. That is, for some period T, it satisfies the condition $f(t) = f(t+T)$

for all t. Periodicity can be interpreted as repetition of the basic signal on the first interval $0 \leq t < T$ on subsequent intervals $kT \leq t < (k+1)T$ for all integers $k \geq 1$. Control Design for such systems lies within classical control and asymptotic tracking is achieved using feedback control with compensators that include the control element (in transfer function notation)

$$K(s) = \frac{1}{1 - e^{-sT}} \tag{1.2}$$

based on the idea of the *Internal Model Principle*. This area of study is of relevance to many applications where disturbances to dynamics have a known source and period (for example, the vibrations generated by a power generator in a ship or aircraft) but whose form is uncertain. Periodic reference signals also apply to rotating systems of known angular frequency. The subject is not considered in this text.

1.2.2 Repetitive Control and Multipass Systems

A related area, often called *Repetitive Control*, grew out of the area of *Multipass Systems Theory*. It considers systems that evolve on a finite interval $0 \leq t \leq T$ after which both the system and the clock are reset and the operation repeated. This procedure is then repeated indefinitely. *Repetitions are often called trials, passes or iterations* and the special dynamic characteristic is that the output signal from each pass has an effect on the next, essentially acting as a correlated disturbance disrupting the dynamics. This intuitive idea can be illustrated by the simple example of a recursive first order differential equation

$$\frac{dy_{k+1}(t)}{dt} = -y_{k+1}(t) + 2y_k(t) + 1, \qquad y_k(0) = 0 \quad k \geq 0, \tag{1.3}$$

with starting condition $y_0(t) \equiv 0$. Here, $y_k(t)$ represents the output on pass k and the term $2y_k(t)$ represents the effect of pass k on the dynamics of the $(k+1)^{th}$ pass. The presence of the term $-y_{k+1}(t)$ indicates the necessary stable response on each pass but note that problems accumulate after a large number of passes. More precisely, assuming that the outputs converge to a limit, $\lim_{k\to\infty} y_k(t) = y_\infty(t)$, this limit is the solution of the equation

$$\frac{dy_\infty(t)}{dt} = y_\infty(t) + 1, \qquad y(0) = 0. \tag{1.4}$$

The conclusion is that a series of stable iterations on a dynamical system can converge but to an unstable, and hence unacceptable, dynamical behaviour. Effective control is, therefore, a necessary part of good plant operation in a repetitive environment.

1.2.2.1 Example: Automated Ploughing

A conceptual model of the idea of automated agricultural ploughing is the creation of a sequence of parallel furrows and approached by setting up an initial furrow as a reference signal for further furrow creation. Let $k \geq 1$ be a counter for the furrows created. After k furrows, furrow $k + 1$ is created by using sensors to detect the form/position of the furrow k and using control signals and systems to recreate its form at a set distance (the furrow width). The process is subject to disturbance and errors implicit in the feedback process. The objective of control is to ensure that the repetitions "settle down" to a sequence of furrows of acceptable form that replicate themselves from pass to pass with little error. A feature of the system is that its dynamics not only occur during each pass, creating a new furrow, but also from pass to pass as the form of the new furrow is influenced, through the control system detectors, by the data generated from the previous furrow.

1.2.2.2 Example: Automated Coal Cutting

Control of underground longwall coal cutting equipment is an example of a multipass process. Automated coal cutting machinery is designed to "move" through the subterranean world removing coal from a coal seam. Once cut, the coal is transported from the coal face and, from there, to the surface for distribution. The machinery takes the form of rotary cutting tools transported along the coal face extracting material from the rock strata at the point of contact. The coal seam is worked over a fixed distance spanning a section of the coal deposit. The coal is ideally removed by the cutter moving along the seam with dynamics affected by the nature of the rock bed on which it moves and by control systems that attempt to ensure that the cutters cut coal rather than the surrounding rock. Once a pass along the coal face has been completed, the machinery is moved forward to sit on the bed of rock revealed by the cut. The process is then repeated and, in this repetitive way, the coal cutter moves through the earth. Note that the physical position of the cutter is influenced by the previous cut as the machinery sits on that cut. As a consequence, the two issues here are, firstly, control of the cutter tool position during each pass and, secondly, the control of the dynamic behaviour tracking the coal seam from pass to pass. Instability is manifested by the divergence of the cutting trajectory from the actual coal seam—a situation when the cutters "cut rock" and the coal mine would become a "rock mine" of no commercial value.

1.2.2.3 Example: Metal Rolling

Metal rolling has the structure of a repetitive/multipass process. The strip metal is passed sequentially though the rolls which apply forces to squeeze the metal and reduce the metal thickness (guage) in a manner dependent on many factors including the physical shape of the initial metal strip, the force applied by the rolls, the strip

temperature and the internal stresses within the metal strip. The repetitive nature of the process is self-evident. The control problem, in essence, is to adjust applied forces on each pass through the rolls in such a manner as to ensure the correct physical shape of the final strip after many passes. The complexity of the control problem lies not only in the need to ensure good performance within each pass but also to cope with the fact that the outcome of each pass has an effect on the dynamics and outcome of the next.

1.2.3 Iterative Control Examples

In many ways, Iterative Learning Control (ILC) is a special case of repetitive process control. It is relevant to applications that consider trajectory tracking control on a finite interval $[0, T]$ and focusses on problems where interaction between passes is normally zero but where the repetition of a specified task creates the possibility of improving performance from task to task. A simple analogy is the repetitive practice of a piece of music on piano until your performance is as desired.

The mechanism for improvement in engineering systems terms is the systematic use of data collected from previous repetitions to create structured interactions that benefit controlled performance. The combined plant plus control system is hence a multipass process by design rather than by physical necessity. The repetitive/iterative nature of the control schemes proposed, the use of measured data from past executions of the task to update/improve control action and the asymptotic *learning* of the required control signals put the topic in the area of classical control design with the added complexity of controlling pass to pass dynamics.

Note: The use of the word "learning" in ILC is now established in the published literature and has a precise interpretation. It reflects the perception that the iterative structure creates a desired and previously unknown control signal based on the observed behaviour of previous trials. In this sense, the iteration "learns" the desired signal using "past experience".

The analysis and form of control design methodologies for ILC problems is the central topic of this text. The following subsections present simplified examples of application areas. The aim is to provide easily visualized physical examples that could help the reader relate the more abstract, mathematical problem formulations and algorithms to a physical context.

1.2.3.1 Iteration and Robotics

The simplest visualization of Iterative learning Control can be found in the area of robotic assembly where a robotic device is used to complete a specified task such as "pick and place". In such situations, the objective is to transfer a sequence of objects (delivered to the robot by conveyor) from a specified position to a new position after travelling along a well-defined spatial trajectory. The time taken to complete each

task is assumed to be fixed at a specified value. It is important to note, however, that the tracking of the trajectory during each transfer is essentially a classical control problem that can be approached using feedback and feedforward methodologies. This approach inevitably leads to non-zero tracking errors due to the very nature of feedback control and the imperfections in the success of feedforward elements arising from the uncertainties in the model of the system. It is hence natural to ask the general questions:

1. *Can the repetitive nature of the task provide a means of improving tracking accuracies despite the model uncertainties?*
2. *How can this objective be achieved and what are the properties of the plant dynamics that aid or inhibit the process?*

The questions are quite general and have many potential solutions. However, the general approach, taken here, to sequential improvement in tracking accuracy is to make use of the data recorded during each execution of the task. Let k be the index of the repetition. Conceptually, the use, on repetition $k+1$, of data recorded on previous repetitions with index $k' \leq k$ will contain data on actual input-output behaviour and the success of previously used control signals. If used intelligently, this "intelligence" could provide a means of error reduction in future repetitions so that, as k increases indefinitely, the tracking error will, ideally, be reduced to zero. The combination of any such control scheme with the robot is an Iterative Learning Control scheme but it also fits into the repetitive control framework as interaction between task executions has been introduced through the use of past data. In general,

1. a good Iterative Learning Controller will ensure the reasonable trajectory tracking behaviour during initial iterations but, ultimately,
2. it will also ensure the *convergence property* that this tracking accuracy will become perfect as $k \to \infty$. This statement also leads to the notion of *rate of convergence* as the designer will naturally only tolerate a small number of iterations (pick and place operations) where large tracking errors will be observed.
3. Consideration should also be given to the *initialization of the iterative process*. That is, the choice of the control signals used on the "zeroth" iteration when the robot is set up.

1.2.3.2 Iteration and Mechanical Test

Mechanical test procedures have a structure very similar to the robotic example described above. Test procedures are widely used as a means of either

1. exposing a mechanical specimen to a repeated sequence of defined force profiles over a specified time interval or
2. replicating, in the laboratory, either-user defined conditions representing typical operational events or physical conditions recorded during actual operation of the plant.

For example, a four poster automotive test system is specifically designed for the testing of vehicles (cars, trucks). Such test systems consist of four hydraulic actuators on top of which the wheels of the vehicle are placed. Movements of the actuators aim to simulate movement of the vehicle over the road surface, the forces exerted by the road on the wheels and the physical displacements of components or subassemblies. To achieve this, the movements of the system need to be tightly controlled by a digital test controller. During the development phase, a 4-poster system is often used to test newly designed suspension systems and their durability. Different kinds of testing are possible including sine tests (including frequency sweeps) and tests aimed at reproducing the conditions met by the vehicle in an actual road environment (represented by data coming from an outdoor acquisition session/road test).

Successful control during mechanical test is not as straightforward as might initially appear. For visualization purposes, consider the control of a test facility that is required to take a road vehicle through a laboratory test that simulates *very accurately* the conditions recorded on a chosen road surface. The vehicle is assumed to have been fully instrumented and driven at the desired speed over the surface with measurements of useful variables made and recorded as time series over a specified time period. Measurements could include measured forces, accelerations and displacements at chosen points of the vehicle including points on the suspension and subassembly. These signals form the reference signal for the following process of iteration in the laboratory.

For commercial or research reasons, suppose that it is required that the vehicle be subjected to the same "road" conditions but in a laboratory setting. Typically, high accuracy replication of road conditions is required. The forces are recreated by suitable controlled operation of the rig hydraulics but, as the rig is operating dynamically, it interacts with the vehicle and responds to its motions. As a consequence, errors in reproducing the desired vehicle response are inevitable in any one test. For this reason, it is essential that the process is repeated through a sequence of repetitions or iterations designed to reduce errors by changing the hydraulic actuation signals. Physically, the iterations consist of repeated operation of the rig using improved actuation signals. The aim of the iterative process is to reduce the errors between signals observed on road tests and those observed in the laboratory tests. When sufficiently small to satisfy the needs of the test, the road conditions have been faithfully replicated and the user is able to take the vehicle through the same conditions, whenever it is required, without the time and expense of more road tests.

The technological key to achieving improved replication of road conditions in the laboratory is some form of Iterative Learning Control which, in a similar manner to the robotics example outlined above, uses previously recorded test data to improve accuracy from iteration to iteration. More precisely, data recorded on previous iterations with index $k' \leq k$ will contain data on actual vehicle behaviour whilst on the rig and, if used intelligently, this could provide a means of generating actuation signal updates that guarantee error reductions. As k increases indefinitely, the tracking error, representing the mismatch between road and laboratory conditions, will hopefully be reduced, asymptotically, to zero.

1.3 Dynamical Properties of Iteration: A Review of Ideas

In the context of the material in this text, the examples of the previous section indicate that Iterative Learning Control (ILC) has two dynamical characteristics, namely

1. *the dynamics within each iteration, as exemplified by the time response of the vehicle to hydraulic actuation forces during each iteration of the mechanical test procedure, and*
2. *the dynamics induced by the iterative control interaction between repetitions. This interaction produces a systematic dynamic behaviour in the sequence of outputs from the iterations. These dynamics are dependent on both the plant dynamics and the chosen data transfer mechanisms between iterations.*

In general, the dynamics within each iteration will interact with the dynamics induced by the iterations. This interaction can be positive (that is, beneficial) or negative (that is, performance degrading) and it is certainly not the case that good control within each iteration implies that error reduction will be achieved from iteration to iteration. These ideas will form part of the following chapters. The following observations use a simple matrix iteration to provide an insight into the issues that can arise:

Matrix Iteration: The dynamic nature of iteration can be illustrated by the analysis of iteration in matrix theory in the form of iterative solution of linear simultaneous sets of algebraic equations. Iterative Control is analogous to the iterative computation of the solution vector u, for a given choice of r, of a matrix equation

$$\begin{aligned} & r = y && (\textit{The "Control Objective"}) \\ \textit{where} \quad & y = Gu + d && (\textit{The "Input/Output Relation"}). \end{aligned} \tag{1.5}$$

The dimensions of the column vectors r, u, d and matrix G are not relevant to the discussion but are required to be consistent with the matrix operations indicated.

An Approach to Iterative Solution: Let $u_0, u_1, u_2, \ldots$ be a sequence of estimated solutions of the equation. Let $y_0, y_1, \ldots$ be the corresponding values of y. Choose an initial guess u_0 and construct the remainder of the sequence by the update relation

$$u_{k+1} = u_k + K_0 e_k + K_1 e_{k+1}, \quad k \geq 0, \tag{1.6}$$

where, for any $j \geq 0$, $e_j = r - y_j = r - Gu_j - d$ represents the error in satisfying the equation $r = y$ using u_j. The matrices K_0, K_1 are chosen by the user and $K_0 e_k + K_1 e_{k+1}$ represents a proposed correction to the iterate u_k to produce a new iterate u_{k+1}. The iterates change every iteration stopping only if $u_{k+1} = u_k$ (when, necessarily, $e_{k+1} = e_k$).

Error Dynamics: The update relationship is implicit as e_{k+1} depends on u_{k+1}. In this sense the update relationship is an algebraic parallel of the use of feedback

(represented by $K_1 e_{k+1}$) and feedforward (represented by $K_0 e_k$) control. An explicit formula is obtained by noting that

$$e_{k+1} = e_k - GK_0 e_k - GK_1 e_{k+1}, \quad k \geq 0, \tag{1.7}$$

so that, denoting the unit matrix by I and assuming that $I + GK_1$ is nonsingular,

$$e_{k+1} = Le_k \quad where \quad L = (I + GK_1)^{-1}(I - GK_0). \tag{1.8}$$

It follows that the iteration update formula can be written in the equivalent "feedforward form"

$$u_{k+1} = u_k + (K_0 + K_1 L)e_k \tag{1.9}$$

and the evolution of the iteration errors has the dynamic behaviour

$$e_k = L^k e_0, \quad for \quad k \geq 0. \tag{1.10}$$

That is, the errors observed in the iterative process evolve in a dynamic way dependent on the initial error e_0, G and both K_0 and K_1.

Error Convergence to Zero: Error convergence to zero is the primary objective of the iterative process. It clearly depends upon the nature of G, K_0, K_1 and e_0. In particular, e_k converges to zero for all possible initial errors e_0 if, and only if, all eigenvalues of L have modulus strictly less than unity. This condition, in particular, means that, after a sufficiently large number of iterations, the error in satisfying the equation is arbitrarily small.

Technical Note: This eigenvalue condition may need refinement if the subspace of initial errors is restricted but these details are left for later in the text. To illustrate the refinements, suppose that $K_1 = 0$ so that $L = I - GK_0$. Suppose that G has full column rank but is non-square so that the best that can be achieved by choice of K_0 is all eigenvalues with modulus less than, or equal to, unity. Unfortunately, under these conditions, there will always be at least one eigenvalue with the value of one and hence convergence to zero for all initial errors e_0 cannot be achieved. If however, attention is focussed only on initial errors that can be written as $e_0 = Gw_0$ for some column vector w_0 the convergence condition will change. To understand the possibilities, suppose that G has full column rank and suppose that $e_0 = Gw_0$ for some column vector w_0. Write, $e_k = Gw_k, k \geq 0$, where the sequence $w_0, w_1, \ldots$ is uniquely defined. It follows that $w_{k+1} = (I - K_0 G)w_k, k \geq 0$, and hence that w_k (and, equivalently, e_k) converges to zero if, and only if, the eigenvalues of $I - K_0 G$ have modulus strictly less than unity.

Nature of the Convergence: The eigenvalue condition describes asymptotic convergence precisely and, in simple terms, the closer the eigenvalues are to zero, the faster the convergence to the solution will be. This simplicity is misleading however as the precise nature of this convergence depends critically on the detail of the matrices involved. For example, if

$$L = \begin{bmatrix} 0.5 & \alpha \\ 0 & 0.7 \end{bmatrix}, \tag{1.11}$$

then errors will always converge to zero as the eigenvalues are 0.5 and 0.7. In particular, error magnitudes will reduce from iteration to iteration if $\alpha = 0$. In contrast, if $\alpha \neq 0$, errors from the defined initial condition take the form

$$e_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad e_1 = \begin{bmatrix} \alpha \\ 0.7 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 1.2\alpha \\ (0.7)^2 \end{bmatrix}, \quad e_3 = \begin{bmatrix} (0.6 + (0.7)^2)\alpha \\ (0.7)^3 \end{bmatrix}, \cdots \tag{1.12}$$

which will ultimately reduce to zero but, before that happens, can take arbitrarily large values (dependent on the parameter α) as the iteration index k increases. As a consequence, it can be expected that, in any iterative process, convergence to a desired solution will not necessarily imply that the convergence will be accompanied by acceptable magnitudes and forms of the iterates. In Iterative Learning Control involving experimental work on the physical system, the magnitudes of signals are normally subject to constraints due to safety boundaries and/or the physical range of actuators. Ensuring acceptable signal magnitudes is crucial in this case.

Sensitivity of the Convergence: If the matrix G is such that L only has eigenvalues with modulus less than unity, then convergence of iterations will be guaranteed. However, if G is perturbed by a small amount, convergence may be lost. In general terms, the effect of a small perturbation will be to perturb the eigenvalues also by a small amount. It follows that those eigenvalues with modulus closest to unity will be the most sensitive to perturbations to G as perturbations with only small magnitudes may be all that is required to move the eigenvalue to a position with modulus greater than unity. Divergence of the iterates will then occur.

The Sign of the Modelling Error: It is interesting to note that the nature of the perturbation is crucial to the consequences for convergence. More precisely, for an eigenvalue close to unity, small perturbations could increase the eigenvalue modulus (potentially causing divergence) whilst the same perturbation but with opposite sign will tend to reduce the modulus (hence increasing convergence speeds). The ability of an iterative process to tolerate perturbations and retain convergence therefore can be expected to depend on its eigenvalue magnitudes and some concept of the "sign" of the perturbation. This will be reflected in the following text where, for dynamical systems, the concept of *positive real* operators plays a role in understanding this issue more fully.

Design Factors: The factors K_0 and K_1 are chosen as part of the update relationship by the user whilst the factor G is analogous to the plant dynamical model. The interplay between these three objects is represented mathematically by the form of L which is the key object in ensuring algorithm convergence. Clearly the primary requirement is convergence and hence eigenvalue positioning. However, an important secondary objective is to control the magnitudes of the iterates as k increases. To achieve this requires a careful design of K_0 and K_1, an issue that dominates this text and the area of Iterative Control. Last but not least, the choices

made will have an impact on the sensitivity/robustness of the convergence properties and choices that provide good robustness might be preferred.

An insight into possible guidelines to design choices is suggested by the form of L. More precisely,

1. If K_1 is a right inverse of G in the sense that $GK_0 = I$, then $L = 0$ and the iterations converge in one step with $e_1 = 0$ independent of the starting condition e_0. This choice requires inversion which may not be regarded as suitable but, using some notion of K_0 as an approximate right inverse might have the effect of reducing eigenvalues of L.
2. The factor $(I + GK_1)^{-1}$ in L may tend to reduce in magnitude if K_1 is "large" in some sense. It may therefore tend to reduce the eigenvalue magnitudes.

The observations made above have a great similarity to the use of inverse systems in classical feedforward control and the notion of high gain feedback being beneficial for control performance. The discussion does not prove that the ideas carry over to Iterative Learning Control without change but they do suggest that inverse systems and gain may play central roles in design decisions.

1.4 So What Do We Need?

Iterative Learning Control is a model-based control design paradigm that has a structure very similar to the matrix iteration discussed in the previous section. A good understanding of matrix theory and computational techniques is still essential but, as Iterative Learning Control is applied to dynamical systems, there is also a need to be able to use, develop and analyse problems where plant dynamics are described by ordinary differential or difference equations.

For linear systems, the techniques used traditionally in control systems design have included Laplace and $\mathscr{Z}$-transforms. For cases where the Laplace transform is relevant and $m = \ell = 1$, Eq. (1.5) could be replaced by the *frequency domain* input/output model

$$\begin{aligned} r(s) &= y(s) \\ y(s) &= G(s)u(s) + d(s) \end{aligned} \tag{1.13}$$

where $f(s)$ denotes the Laplace transform of a function $f(t)$ on $t \geq 0$. The symbol $G(s)$ represents the system transfer function and $d(s)$ is the term generated by possibly non-zero initial conditions at $t = 0$. This equation can be returned to the *time domain* using the convolution description

$$\begin{aligned} r(t) &= y(t) \\ y(t) &= \int_0^t h(t - t')u(t')dt' + d(t), \quad t \geq 0 \end{aligned} \tag{1.14}$$

where $h(t)$ is the impulse response of the system (the inverse Laplace transform of $G(s)$). The conclusion reached from this discussion is that, for a wide class of linear dynamical systems, the simple representation (1.5) has meaning if r, y, u and d are interpreted as signals in defined vector spaces and G is a linear operator from the space containing the input signal into the space containing the output signal. The advantage of this interpretation is that the reader can, in many ways, think of the problem in matrix or transfer function terms as the algebraic manipulation of matrices, transfer functions and operators are very similar. There are differences however which, from time to time, play a crucial role in the theoretical development but ultimately, the computational procedures needed returns to matrix computations and experimental work combined with numerical simulation of dynamic models.

1.4.1 An Overview of Mathematical Techniques

So what do we need to develop a theory of design for Iterative Learning Control? The structure of the following development is based on the following mathematical ideas and needs:

1. For completeness, the essential mathematical background in matrix methods is summarized in a form that meets traditional engineering mathematics course requirements but provides an interface with the following work on vector spaces and operator theory.
2. Control algorithms manipulate signals in the form of functions or time series which can be regarded as being elements of vector spaces. The relevant concepts, constructions and relationships needed for analysis in finite or infinite dimensional vector spaces are therefore introduced with emphasis on the ideas of norms as measures of magnitude, convergence of infinite sequences and the definition and properties of bounded linear operators. Banach spaces are presented initially but emphasis is placed on the properties of Hilbert spaces as the intuitively "familiar" geometry of Hilbert spaces and the notions of adjoint operators, orthogonal projection and convexity provide the mathematical methodologies that underpin the use of the optimization methods that follow.
3. Optimization methodologies in Hilbert spaces are introduced early in the text as a means of providing a structured improvement in error magnitudes from iteration to iteration. The focus is on the minimization of objective functions that are quadratic in the relevant signals and/or parameters and subject to linear constraints. The solution of these problems using either the simple notion of completing the square or, equivalently, using the projection theorem is described in full detail. Projection onto closed convex sets forms the core concept in the text. It is used in many guises to cover algorithm development, the inclusion of constraints and formal solution techniques with detailed application to linear systems of differential or difference equations (in state space form).

Note: The advantage of the use of operator theory is that, in principle, the results and algorithms apply to any situation where the mathematical assumptions of linearity and boundedness apply. These include multi-rate sampled digital dynamical systems, models consisting of differential equations where signals may be delayed (differential-delay systems), models containing both algebraic and differential equations and models containing a wide variety of integral equations. The computational aspects and challenges arising from these applications may be much more complex than those met in familiar state space systems.

4. With a focus on state space models, the relevant properties of these models, the associated transfer function matrices and the ideas of inverse dynamical systems are central to the content. The link to optimal control with quadratic objective functions is established and relates naturally to the practical use of optimal control and state feedback methodologies based on the Riccati equation. The link to optimal control theory is crucial in providing the step that takes the analysis from the high level (but abstract) simplicity of operator descriptions nearer to practical implementations using state feedback and feedforward strategies.
5. There is a natural difference between continuous time systems and discrete systems on finite time intervals. The difference is quite technical in that continuous time systems require signal spaces that are infinite dimensional whereas discrete systems are finite dimensional. The finite dimensionality of discrete systems makes it possible to write dynamics in the form of a matrix equation of the form of (1.5) with signal time series replaced by so-called "supervectors" and the matrix G constructed using the impulse response of the system. This interpretation releases many familiar matrix techniques for application to analysis and algorithm development. In particular,

   a. eigenvalue methods underpin the analysis of error convergence and, through inequalities relating signals on finite intervals to the $\mathscr{Z}$-transform, provide frequency domain sensitivity and robustness tests.
   b. In addition, a number of techniques can be used to condition or redefine output signals as a means of guaranteeing useful range and kernel properties of G and hence the existence of left and right matrix inverses.

6. Matrix theory aids the development of ideas in the case of discrete time dynamics. For continuous time systems, the use of more sophisticated mathematical tools is necessary. This complexity arises as some of the underlying signal spaces are necessarily infinite dimensional. The operators in such cases have identical algebraic properties to matrices but notions of convergence require a more detailed consideration. Eigenvalues are replaced by the concept of the spectrum of the operator but the link between the spectrum and convergence is obtained through theorems from Functional Analysis.

Whilst representing the dominant specific needs of the theory, the general ideas within classical (non-repetitive) control engineering also play a strong role in interpretation of models, control structures and behaviours.

1.4.2 The Conceptual Basis for Algorithms

The mathematical and modelling techniques summarized above form the toolbox proposed to develop the ideas and algorithms for optimization-based Iterative Learning Control presented in this text. The topics contain a number of threads and discussion items including the following:

1. General mathematical models of Iterative Learning Control using operator-based representations form the basis of the approach. The simplicity of the presentation is enhanced by this choice of approach but interpretation of the equations needs to be carefully considered. Readers who are inexperienced in the area should find the material accessible by interpreting the operators and vectors as matrices and the systems as being discrete time.
2. Formal definitions of the problem of control algorithm design are used throughout. The primary objective is always error convergence but other issues include,
    a. identifying the influence of user defined parameters such as gains and the weights used in optimization criteria,
    b. characterizing the robustness of the convergence to errors in the plant model,
    c. controlling the nature of the error convergence in terms of ensuring monotonic reductions of defined error measures (norms) from iteration to iteration and/or
    d. using available degrees of freedom to ensure error convergence to zero whilst achieving other performance requirements such as minimizing an auxiliary objective function representing other desirable dynamical objectives.
3. For linear systems, the system dynamics and the control objectives can be regarded as defining separate convex sets in suitable Hilbert spaces. This interpretation forms the basis of algorithm construction using sequential projection of signals onto these sets. The many different ways of choosing the Hilbert space topology and choosing the order of the sequence of projections provides a wide range of different convergent algorithms including:
    a. *Inverse Model* and *Gradient* algorithms that play a useful role in defining simple approaches and typical properties,
    b. *Norm-optimal Iterative Learning Control* (the basic algorithm that forms the core of the work),
    c. *Predictive Norm-optimal Iterative Learning Control* (where the effect on future iterations is taken into account in input selection),
    d. the *inclusion of convex constraints on inputs and output signals* (to allow for applications where such limits are present),
    e. *algorithms whose objective is limited* to, for example, ensuring that an output signal $y(t)$ takes specified values at times $t_1, t_2, \ldots, t_M$—the so-called *Intermediate Point Control Problem*. Such problems typically have degrees of freedom that permit auxiliary optimization objectives to be achieved,
    f. *Multi-task Algorithms* that unify some of these ideas under one algorithmic structure,

g. so-called *Notch Algorithms* where the projections are selected carefully to eliminate, or almost eliminate, errors in output subspaces characterized by a single parameter or
h. algorithms whose objective is *Control Loop Management*.

All such algorithms ensure error convergence but the nature of each convergence differs in substantive, but useful, ways that give flexibility to the control design process. In general terms, the computational basis of each iteration is optimal control related and, for state space systems, provides an implementation in terms of a state feedback and an iteration to iteration feedforward structure.

4. Convergence rates depend in a simple way upon the choice of different topologies in the input and output spaces but also upon plant dynamical characteristics. The biggest problems are met when controlling systems which are *non-minimum-phase* in the sense that the system transfer function has zeros in the unstable region of the complex plane. The presence of such zeros leads to rapid error reduction in the first few iterations but, from that point onwards, the convergence rate is infinitesimally slow. An analysis of this phenomena is possible and leads, in practical terms, to design guidelines aimed at improving the situation.
5. Finally, some engineers might baulk at the complexities of optimal control implementations of the algorithms. Fortunately, the beneficial properties of optimization can be retained using simpler implementations and the idea of *Parameter Optimal Iterative Learning Control*. Details are contained in later chapters of this text but can be summarized as replacing control laws based on function or time series optimization by control laws expressed, in a linear way, in terms of known data and unknown parameters deduced from a low order optimization problem. The results of function optimization and parameter optimization are shown to be identical under certain well-defined conditions. They can then be regarded as realistic alternative approaches. Parameter optimization can however introduce a number of new dynamical phenomena in the iterative process, including convergence but to non-zero errors. The reasons for this are related to ideas of positivity of certain matrices and techniques for assessing the problem and avoiding its occurrence using suitable parameterizations and/or switching strategies are presented.

1.5 Discussion and Further Background Reading

Control Engineering is a topic of considerable depth and scope with many texts and published papers. An overview is available in the Control Handbook [65]. The idea of iteration in control probably emerged from unpublished empirical techniques used in industry. There was little theoretical support or analysis. The first academic paper on Iterative Learning Control written in English was published by Arimoto and colleagues [10] under the title of a *"betterment process"* and has received considerable attention since that time. Readers wishing to read alternative and more recent moti-

vations for Iterative Control problems can find additional material in survey papers such as [2, 19, 66, 68, 88, 93, 100, 111, 113] and published texts including the first in the area [74] together with [3, 17, 25, 78, 114]. The range of applications is growing with examples to be found in [13, 29, 40, 42, 97, 100] and elsewhere.

The internal model principle is explained in [38]. Readers may also find it useful to refresh their knowledge of classical control techniques using texts such as [39, 48, 49, 63]. The extensions to state space and multi-input/multi-output (multivariable) control systems can be found in a number of texts including [4, 43, 71, 81]. The well-established ideas of optimal control seen in texts such as [8, 9, 11, 24] are central to what follows. An undergraduate level introduction can be found in [30]. In particular, the geometric approach used in [69] is particularly suited to the optimization content of this text.

The ideas of multipass systems theory arose out of the pioneering work of Edwards and Owens and can be explored in some detail in [34] with related work on the mathematical analysis of repetitive dynamical systems in [99]. The main preoccupation of this text is linear systems but many of the ideas apply to nonlinear systems with little conceptual change but with considerable changes needed in detail and methodology. The computational load increases greatly and problems such as convergence to local minima cannot be excluded. The overall nature of the iterative paradigm remains unchanged however.

Finally, readers might find it useful to relate the material to the linear and nonlinear iterative techniques found in the mathematical texts such as Collatz [28] and Ortega and Rheinboldt [79] where the general ideas of fixed point or Newton-like iteration have potential application to Iterative Control (as illustrated by many nonlinear Iterative Control methods [113] and in Lin et al. [67]).

Chapter 2
Mathematical Methods

A study of control systems dynamics and optimization has always needed a suitable mathematical language to formulate the problems, analyse possible behaviours and difficulties, develop algorithms and prove that these algorithms have the desired properties. This is also true of Iterative Control which exhibits all the properties and challenges of classical control problems with the added need to consider the effect of iteration on behaviours. The material in this chapter acts to remind readers of the mathematics needed for analysis of state space models in control theory and the essential structure of quadratic optimization problems. To this mix is added an introduction to the essentials of Hilbert spaces as a representation of signals, as a means of representing dynamical systems as operators on such spaces and as a means of creating a geometric approach to iterative optimization based control. The presentation aims to define both notation and explain concepts. Fuller details and proofs of the statements can be found in the references.

2.1 Elements of Matrix Theory

A $p \times q$ real (or complex) matrix A is an array of real (or complex) numbers of the form

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1q} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2q} \\ \vdots & & & & \vdots \\ A_{p1} & A_{p2} & A_{p3} & \cdots & A_{pq} \end{bmatrix} \tag{2.1}$$

The element in the ith row and jth column is denoted A_{ij}. If $q = 1$, the matrix is often called a vector. Block matrices can also be defined where the A_{ij} are $p_i \times q_j$ sub-matrices. In this case the dimensions of the matrix A are $\sum_{i=1}^{p} p_j \times \sum_{j=1}^{q} q_j$.

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_2

The following is essential for control theoretical purposes

1. The set of $p \times 1$ real (respectively complex) vectors is given the symbol $\mathscr{R}^p$ (respectively $\mathscr{C}^p$).
2. A is said to be *square* if the number of rows is equal to the number of columns.
3. Addition of two $p \times q$ matrices A, B to form a $p \times q$ matrix C written $C = A + B$ is defined by the elements

$$C_{ij} = A_{ij} + B_{ij}, \quad for \quad 1 \leq i \leq p, \quad 1 \leq j \leq q. \tag{2.2}$$

   The $p \times q$ *zero matrix* is the matrix with all elements equal to zero.
4. Multiplication of A by a scalar λ produces a matrix $C = \lambda A$ where

$$C_{ij} = \lambda A_{ij}, \quad for \quad 1 \leq i \leq p, \quad 1 \leq j \leq q. \tag{2.3}$$

5. Multiplication of a $p \times q$ matrix A by a $q \times r$ matrix B to produce a $p \times r$ matrix $C = AB$ is defined by the following computation of elements of C

$$C_{ij} = \sum_{k=1}^{q} A_{ik} B_{kj}, \quad for \quad 1 \leq i \leq p, \quad 1 \leq j \leq r. \tag{2.4}$$

6. The *transpose* of a $p \times q$ matrix A is the $q \times p$ matrix A^T with

$$(A^T)_{ji} = A_{ij}, \quad for \quad 1 \leq i \leq p, \quad 1 \leq j \leq q. \tag{2.5}$$

   The act of taking the transpose of a product satisfies the rule $(AB)^T = B^T A^T$. If $A = A^T$ then A is said to be *symmetric*.
7. The *conjugate transpose* of a complex $p \times q$ matrix A is the $q \times p$ matrix A^* with

$$(A^*)_{ji} = \overline{A}_{ij}, \quad for \quad 1 \leq i \leq p, \quad 1 \leq j \leq q, \tag{2.6}$$

   where $\overline{a}$ denotes the complex conjugate of a. The act of taking the conjugate transpose of a product satisfies the rule $(AB)^* = B^* A^*$. If $A = A^*$ then A is said to be *Hermitian*. If A is real then the conjugate transpose is simply the transpose and, if A is Hermitian, it is symmetric.
8. The determinant of a square $p \times p$ matrix is denoted $\det[A]$, $|A|$ or,

$$\det[A] = \begin{vmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1p} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2p} \\ \vdots & & & & \vdots \\ A_{p1} & A_{p2} & A_{p3} & \cdots & A_{pp} \end{vmatrix} \tag{2.7}$$

The determinant has many properties including

a. the properties $\det[A] = \det[A^T]$ and $\det[\overline{A}] = \overline{det[A]}$.
b. If both A and B are square $p \times p$ matrices, then $\det[AB] = \det[A]\det[B] = \det[BA]$.

9. A is said to be an *injection* or *one-to-one* if the only $p \times 1$ vector v satisfying the equation $Av = 0$ is the vector $v = 0$. In general, the set of vectors v such that $Av = 0$ is a vector subspace of $\mathscr{R}^q$ (or $\mathscr{C}^q$) and is called the *kernel* or *null space* of A and denoted
$$ker[A] = \{v : Av = 0\} \tag{2.8}$$
The subspace $ker[A]$ is always $\neq \{0\}$ when $q > p$.
10. If, for every $p \times 1$ vector w, there exists a vector v such that $w = Av$, then A is said to be *onto* or a *surjection*. More generally, the set of all vectors w for which there exists a vector v such that $w = Av$ is called the *range* of A. It is a vector subspace of $\mathscr{R}^p$ (or $\mathscr{C}^p$) and is denoted by
$$\mathscr{R}[A] = \{w : w = Av \quad for\, some\, vector\, v\} \tag{2.9}$$
A necessary condition for the range to be equal to $\mathscr{R}^p$ (or $\mathscr{C}^p$ as appropriate) is that $q \geq p$.
11. If A is both a surjection and an injection, it is said to be a *bijection* (or simply *nonsingular*). If A is a $p \times p$ square matrix, then it is a bijection if, and only if, it has non-zero determinant. If $\det[A] = 0$ then A is said to be *singular*.
12. A $p \times q$ matrix A is *invertible* if, and only if, it is a bijection. In particular, this requires that it is square ($p = q$) and it is equivalent to the statement that, for every vector w, there exists a unique vector v such that $w = Av$. The *inverse* of A is denoted by A^{-1}. It is a square $p \times p$ matrix of the form
$$A^{-1} = \frac{adj[A]}{\det[A]} \tag{2.10}$$
where the *adjugate* matrix $adj[A]$ has elements that are well defined polynomials in the elements of A.
13. For all invertible $p \times p$ matrices A, the inverse A^{-1} satisfies the equations
$$AA^{-1} = A^{-1}A = I_p \tag{2.11}$$
where I_p denotes the $p \times p$ *unit matrix* or *identity*
$$I_p = \begin{bmatrix} 1\,0\,0 \cdots 0 \\ 0\,1\,0 \cdots \\ \vdots \qquad\quad \vdots \\ 0\,0\,0 \cdots 1 \end{bmatrix} \tag{2.12}$$

For any $p \times q$ matrix B and $q \times p$ matrix C, the properties $I_p B = B$ and $C I_p = C$ hold true. Also $\det[I_p] = 1$ and hence $\det[A^{-1}] = (\det[A])^{-1}$.
Note: for notational simplicity, the subscript p on I_p is sometimes dropped to leave the symbol I. This should cause no confusion as matrix dimensions are usually clear from the context.

14. If A and B are $p \times p$ nonsingular matrices, then $(AB)^{-1} = B^{-1}A^{-1}$.
15. If T is square and nonsingular, then the mapping $A \mapsto T^{-1}AT$ is a *similarity transformation* Both A and $T^{-1}AT$ have the same eigenvalues and $\det[A] = \det[T^{-1}AT]$.
16. For non-square $p \times q$ matrices A, other definitions of inverse play a role in matrix analysis. In particular, if $p \geq q$, a *left inverse* B of A is any matrix satisfying the condition $BA = I_q$. In a similar manner, if $p \leq q$, a *right inverse* B of A is any matrix satisfying the condition $AB = I_p$. A left inverse of A exists if, and only if, A has kernel $\{0\}$ and a right inverse exists, if and only if, $\mathscr{R}[A] = \mathscr{R}^p$ (or $\mathscr{C}^p$). If $p \neq q$, any left or right inverse is non-unique. If $p = q$, then they are unique and equal to the inverse A^{-1}. Specific examples of left, respectively right, inverses are given by, respectively,

$$B = (A^*A)^{-1}A^*, \quad B = A^*(AA^*)^{-1}. \tag{2.13}$$

17. Given the definition of the unit matrix, two useful relationships are as follows

    a. If A and B are, respectively, $p \times q$ and $q \times p$, then

    $$\det[I_p + AB] = \det[I_q + BA]. \tag{2.14}$$

    b. If A has the partitioned form

    $$A = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{2.15}$$

    with M_{11} square and nonsingular, then *Schur's Formula* is valid,

    $$\det[A] = \det[M_{11}] \det[M_{22} - M_{21}M_{11}^{-1}M_{12}]. \tag{2.16}$$

The above algebraic properties of matrices are the basis of manipulation. For analysis purposes, a number of other properties and concepts are required and are summarized as follows

1. A finite set $\{H_j\}_{1 \leq j \leq M}$ of real (respectively complex) $p \times q$ matrices is said to be *linearly independent* if, and only if, the only real (respectively complex) scalars $\{a_j\}_{1 \leq j \leq M}$ satisfying the condition $\sum_{j=1}^{M} a_j H_j = 0$ are $a_j = 0,\ 1 \leq j \leq M$.
2. The *rank* of a $p \times q$ matrix A is the maximum number of linearly independent columns of A regarded as $p \times 1$ vectors. A $p \times p$ matrix A is nonsingular if, and only if, it has rank equal to its dimension p.

3. The *characteristic polynomial* of a square $p \times p$ matrix A is defined by the determinant
$$\rho(s) = \left|sI_p - A\right| = \sum_{j=0}^{p} a_{p-j} s^j \quad with\ a_0 = 1. \tag{2.17}$$
It is a polynomial of degree p in the complex variable s with p, possibly complex, roots $\lambda_j,\ 1 \le j \le p$ called the *eigenvalues* of A. If A is a real matrix, then the eigenvalues are either real or exist in complex conjugate pairs. More precisely, if λ is an eigenvalue, then its complex conjugate $\overline{\lambda}$ is also an eigenvalue. The *spectral radius* of A is defined by
$$r(A) = \max_{1 \le j \le p} |\lambda_j|. \tag{2.18}$$
4. A complex number λ is an eigenvalue of A if, and only if, there exists a non-zero solution vector $v \in \mathscr{C}^p$ solving the equation
$$Av = \lambda v \tag{2.19}$$
Such an *eigenvector* is not uniquely defined as, for example, it can be multiplied by any scalar and still be an eigenvector. If A has p linearly independent eigenvectors $\{v_j\}_{1 \le j \le p}$ then an *eigenvector matrix* E of A is defined to be the block matrix $E = \left[v_1, v_2, \ldots, v_p\right]$. It is nonsingular and can be used to *diagonalize* A using the *similarity transformation*
$$E^{-1}AE = \begin{bmatrix} \lambda_1 & 0 & 0 \cdots & 0 \\ 0 & \lambda_2 & 0 \cdots & \\ \vdots & & & \vdots \\ 0 & 0 & 0 \cdots & \lambda_p \end{bmatrix} = diag\left[\lambda_1, \lambda_2, \ldots, \lambda_p\right] \tag{2.20}$$
The diagonal matrix produced is often called the *diagonal canonical form* of A. A always has linearly independent eigenvectors if its p eigenvalues are distinct.
5. As $|sI - A| = \overline{|\bar{s}I - A^*|}$, the eigenvalues of the conjugate transpose matrix are exactly the complex conjugates of the eigenvalues $\{\lambda_j\}$ of A. Suppose that the eigenvectors of A^* are denoted $w_j,\ 1 \le j \le p$ and that $A^* w_j = \bar{\lambda}_j w_j$. This can be rewritten in the form $w_j^* A = \lambda_j w_j^*$ and w_j^* is termed a *left eigenvector* of A. If A has p linearly independent eigenvectors and associated eigenvector matrix E, then
$$E^{-1}A = diag[\lambda_1, \lambda_2, \ldots, \lambda_p]E^{-1}. \tag{2.21}$$
Equating rows of the two sides of the equation indicates that the rows of E^{-1} are left eigenvectors of A and, as $E^{-1}E = I$, these left eigenvectors satisfy the conditions
$$w_i^* v_j = \delta_{ij}, \tag{2.22}$$

the *Kronecker delta* defined as $\delta_{ij} = 0$ whenever $i \neq j$ and unity otherwise.

6. If A is Hermitian, then its eigenvalues are all real valued. Its eigenvectors v_j form a set of p linearly independent vectors. It is always possible to scale these vectors so that they satisfy the *orthogonality condition*

$$v_i^* v_j = \delta_{ij} \tag{2.23}$$

Under these circumstances, $E^{-1} = E^*$ so that $E^* E = I$, an example of a *unitary matrix*. If E is real then it is an *orthogonal* matrix.

7. Almost every square matrix A can be diagonalized in the manner shown above. In some cases, diagonalization is not possible but, in such cases, a *Jordan canonical form* can be produced. More precisely, there exists an integer q and a nonsingular $p \times p$ matrix J with the property that

$$J^{-1} A J = \begin{bmatrix} J_1 & 0 & 0 \cdots & 0 \\ 0 & J_2 & 0 \cdots & \\ \vdots & & & \vdots \\ 0 & 0 & 0 \cdots & J_q \end{bmatrix} = blockdiag\left[J_1, J_2, \ldots, J_q\right] \tag{2.24}$$

where each *Jordan block* J_j, $1 \leq j \leq q$ has the structure of a $q_j \times q_j$ matrix as follows

$$J_j = \begin{bmatrix} \gamma_j & 1 & 0 \cdots & 0 & 0 \\ 0 & \gamma_j & 1 \cdots & 0 & 0 \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & 0 \cdots & \gamma_j & 1 \\ 0 & 0 & 0 \cdots & 0 & \gamma_j \end{bmatrix} \tag{2.25}$$

where each γ_j is an eigenvalue of A.

8. In all cases the determinant of A is computed from the product of all p eigenvalues

$$\det[A] = \prod_{j=1}^{p} \lambda_j. \tag{2.26}$$

A is singular if, and only if, it has a zero eigenvalue and hence $ker[A] \neq \{0\}$.

9. In all cases, a square matrix A "satisfies its own characteristic equation"

$$\rho(A) = \sum_{j=0}^{p} a_{p-j} A^j = 0. \tag{2.27}$$

This statement is normally known as the *Cayley-Hamilton Theorem*. The result is the basis of many theoretical simplifications and insight exemplified by the easily proven fact that, if A is nonsingular, then $a_p \neq 0$ and

$$A^{-1} = -a_p^{-1} \sum_{j=1}^{p} a_{p-j} A^{j-1}. \tag{2.28}$$

That is, the inverse can be expressed in terms of powers of A and the coefficients in the characteristic polynomial.
Note: A related polynomial is the *minimum polynomial* of A which is the uniquely defined polynomial $\rho_{min}(s)$ of minimum degree that has the property $\rho_{min}(A) = 0$. The degree of the minimum polynomial is always less than or equal to p and is always equal to p if the eigenvalues of A are distinct.

10. More generally, the Cayley-Hamilton theorem implies useful facts about *functions of matrices*. If $f(s)$ is an analytic function of the complex variable s expressible as a power series $\sum_{j=0}^{\infty} f_j s^j$ with radius of convergence R, then the symbol $f(A)$ denotes the associated function of A defined by

$$f(A) = \sum_{j=0}^{\infty} f_j A^j \tag{2.29}$$

This series converges whenever the spectral radius $r(A) < R$. For example,

a. the exponential function e^s has a power series expansion with $f_j = \frac{1}{j!}$. The corresponding matrix exponential is

$$e^A = \sum_{j=0}^{\infty} \frac{1}{j!} A^j = I + A + \frac{1}{2!} A^2 + \frac{1}{3!} A^3 + \cdots. \tag{2.30}$$

b. The function $(1 - s)^{-1} = \sum_{j=0}^{\infty} s^j$ has a radius of convergence $R = 1$. It follows that, if the spectral radius $r(A) < 1$, the matrix inverse $(I - A)^{-1}$ exists and has the convergent series expansion

$$(I - A)^{-1} = \sum_{j=0}^{\infty} A^j. \tag{2.31}$$

If A has a nonsingular eigenvector matrix E, then $A^j = E\ diag[\lambda_1^j, \ldots, \lambda_p^j] E^{-1}$ and

$$f(A) = \sum_{j=0}^{\infty} f_j A^j = E\ diag[f(\lambda_1), \ldots, f(\lambda_m)] E^{-1} \tag{2.32}$$

11. From the Cayley-Hamilton theorem it is easily seen that all powers A^j with $j \geq p$ can be expressed as a polynomial in A of degree less than or equal to $p - 1$. It follow that all functions $f(A)$ can be expressed as a polynomial in A of degree less than or equal to $p - 1$ by suitable choice of coefficients.

12. The *Spectral Mapping Theorem* states that, if A has eigenvalues $\lambda_j,\ 1 \leq j \leq p$, and $r(A) < R$, then the eigenvalues of $f(A)$ are precisely $f(\lambda_j),\ 1 \leq j \leq p$.

The final group of useful properties are associated with the idea of positivity of *quadratic forms*

1. Suppose that A is a square, real, symmetric, $p \times p$ matrix and x an arbitrary $p \times 1$ vector in $\mathscr{R}^p$. Then the quadratic function $x^T Ax$ is a *quadratic form*. If A is not symmetric, it can always be replaced by a symmetric matrix as $x^T Ax \equiv x^T \left(\frac{A+A^T}{2}\right) x$.
2. If A is complex then the quadratic form is defined on $\mathscr{C}^p$ as $x^* Ax$. If A is Hermitian, then $x^* Ax$ takes only real values.
3. A real matrix A is said to be *positive* if, and only if, $x^T Ax \geq 0$ for all vectors $x \in \mathscr{R}^p$. If $x^T Ax > 0$ whenever $x \neq 0$, then A is said to be *positive definite* and written in the form $A > 0$. If A is positive but not positive definite, it is *positive semi-definite* and written in the form $A \geq 0$. The expression $A \geq B$ (respectively $A > B$) is equivalent to $A - B \geq 0$ (respectively $A - B > 0$). Similar definitions are used for complex matrices and their associated quadratic forms.
4. Conditions for positivity for real, symmetric matrices include the following

   a. A real, symmetric matrix A is positive if, and only if, all its eigenvalues satisfy the inequalities $\lambda_j \geq 0,\ 1 \leq j \leq p$. It is positive definite if, and only if, all eigenvalues are strictly positive. Positive definite, symmetric matrices are hence always invertible.
   b. If A and B are matrices and $A = B^T B$, then A is positive. A is positive definite if, and only if, $ker[B] = \{0\}$.
   c. A real, symmetric $p \times p$ matrix A is positive definite if and only if the *Principal Minors*

$$\begin{vmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1q} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2q} \\ \vdots & & & & \vdots \\ A_{q1} & A_{q2} & A_{q3} & \cdots & A_{qq} \end{vmatrix} > 0, \qquad for\ 1 \leq q \leq p. \tag{2.33}$$

5. Positive, symmetric, real matrices A always have an orthogonal eigenvector matrix E and can be expressed in the form $A = Ediag[\lambda_1, \ldots, \lambda_p]E^T$ with $E^T E = I$.
6. A positive, symmetric, real matrix A has a unique symmetric, positive *square root* B such that $B^2 = A$. As with scalar square roots, B is usually denoted by the symbol $A^{\frac{1}{2}}$. It can be computed from the formula

$$A = Ediag[\lambda_1^{\frac{1}{2}}, \ldots, \lambda_p^{\frac{1}{2}}]E^T \tag{2.34}$$

2.2 Quadratic Optimization and Quadratic Forms

2.2.1 Completing the Square

The conceptual basis of much of optimization theory used in control systems algorithms has its origins in the simple ideas of minimization of quadratic functions of vectors in $\mathscr{R}^p$. This short section explains the basic ideas using a simple example and without the need for advanced mathematical methodologies more complicated than the matrix theory described above. The problem used to illustrate the ideas is the problem of minimizing the quadratic objective function

$$J(x) = x^T Ax + 2b^T x + c. \tag{2.35}$$

where the $p \times p$ matrix A is real, symmetric and positive definite, b is real and $p \times 1$ and c is a real number. The solution is easily found by *completing the square* and verifying that

$$J(x) = (x + A^{-1}b)^T A(x + A^{-1}b) - b^T A^{-1}b + c. \tag{2.36}$$

The second two terms are independent of x. The fact that A is positive definite immediately yields the fact that the minimum value occurs when the first term is zero. The unique minimizing solution is hence

$$x_\infty = -A^{-1}b \quad \textit{and} \quad J(x_\infty) = -b^T A^{-1}b + c. \tag{2.37}$$

Both can be computed using standard software if the matrices involved are of reasonable dimension and not ill-conditioned. Factors causing problems include:

1. Suppose that the eigenvalues of A are listed in order of ascending value $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_p$ and A is written in its diagonal form $A = Ediag[\lambda_1, \ldots, \lambda_p]E^T$ with $E^{-1} = E^T$. The *condition number* of A is defined to be $c(A) = \frac{\lambda_p}{\lambda_1}$. It follows that the inverse of A has the structure

$$A^{-1} = Ediag[\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_p^{-1}]E^T \tag{2.38}$$

   The situation where the spread of the eigenvalues of A is large (that is, $c(A)$ is large) can be discussed by considering the case where λ_1 is very small. In such situations, small errors in characterizing this eigenvalue can lead to large changes in the computed solution $x = -A^{-1}b$.
2. These problems are exacerbated if the dimension p is large due to the number of floating point operations necessary in computer computation of A^{-1}.

In the quadratic problems considered in this text, similar quadratic objective functions will be considered but the "matrices" involved are replaced by operators associated with dynamical systems models and, in intuitive terms, have very high

(even infinite) dimensions and extremely large (possibly infinite) condition numbers. Formal solutions paralleling the algebraic constructions illustrated above hence have no immediate computational value, the core of the theoretical problem being that of developing control algorithms that use only feasible computational procedures that can be implemented using well-conditioned off-line algorithms and on-line feedback controllers.

2.2.2 Singular Values, Lagrangians and Matrix Norms

The *singular values* $0 \leq \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_q$ of a real (respectively complex) $p \times q$ matrix A are real, positive numbers computed from the eigenvalues $0 \leq \lambda_1 \leq \cdots \leq \lambda_q$ of the symmetric (respectively Hermitian) matrix $A^T A$ (respectively $A^* A$) by writing $\lambda_j = \sigma_j^2,\ 1 \leq j \leq q$. The corresponding eigenvectors are often called *singular vectors*.

Associated with the matrix A is the notion of a matrix *norm*. As will be seen throughout this text, the idea of a norm is non-unique. What follows, therefore, is only an example that builds on the idea of singular values and illustrates the use of Lagrangian methods in optimization problems. The first step is the definition of a particular vector norm, the Euclidean norm, defined on vectors $x \in \mathscr{R}^q$ (respectively $\mathscr{C}^q$) by $\|x\| = \sqrt{x^T x}$ (respectively $\sqrt{x^* x}$). The Euclidean norm induces a norm $\|A\|$ on the matrix A by defining

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup\{\|Ax\| : \|x\| = 1\} \tag{2.39}$$

In particular, it follows that, for all vectors $x \in \mathscr{R}^q$ (respectively $\mathscr{C}^q$),

$$\|Ax\| \leq \|A\|\ \|x\|. \tag{2.40}$$

As a consequence, if A and B are $p \times q$ and $q \times r$ matrices, it follows that $\|ABx\| \leq \|A\|\ \|B\|\ \|x\|$ and hence

$$\|AB\| \leq \|A\| \|B\|. \tag{2.41}$$

Suppose now that A is real and vectors are in $\mathscr{R}^q$. From the above, the induced norm is the solution of an optimization problem with an equality constraint, namely

$$\|A\|^2 = \sup\{\|Ax\|^2 : \|x\| = 1\} = \sup\{x^T A^T Ax : x^T x = 1\}. \tag{2.42}$$

The solution of this problem is computed by solving for the unique stationary point of the *Lagrangian*

$$\mathscr{L}[x, \lambda] = x^T A^T Ax + 2\lambda^T \left(1 - x^T x\right) \tag{2.43}$$

where λ is the scalar *Lagrange Multiplier* for the single constraint $1 - x^T x = 0$. The stationary point of the Lagrangian is the solution of the two equations

$$1 - x^T x = 0 \quad \textit{and} \quad A^T A x = \lambda x, \tag{2.44}$$

That is, λ is an eigenvalue of $A^T A$ and the largest value of the optimization objective function $\|Ax\|^2 = x^T A^T A x$ is simply the largest eigenvalue of $A^T A$. Hence

$$\|A\| = \sigma_q. \tag{2.45}$$

which provides a simple link between matrix norms and singular values. This relationship also holds for complex matrices operating on $\mathscr{C}^q$. Finally,

1. If A is $p \times p$ and nonsingular, then $\|A\| = \sigma_p$ and $\|A^{-1}\| = \sigma_1^{-1}$.
2. The smallest and largest singular values are denoted by $\underline{\sigma}(A) = \sigma_1$ and $\overline{\sigma}(A) = \sigma_p$ respectively.
3. The spectral radius is linked to matrix norms by the formula

$$r(A) = \lim_{k \to \infty} \|A^k\|^{1/k} \tag{2.46}$$

from which, for all $\varepsilon > 0$, there exists a real number $M_\varepsilon \geq 1$ such that

$$\|A^k\| \leq M_\varepsilon (r(A) + \varepsilon)^k \tag{2.47}$$

If A can be diagonalized by a nonsingular eigenvector matrix E, then it is possible to choose $\varepsilon = 0$ and $M_0 = \|E^{-1}\| \|E\|$.

2.3 Banach Spaces, Operators, Norms and Convergent Sequences

2.3.1 Vector Spaces

Matrices are just part of a more general approach to signal analysis based on *vector spaces* which are a mathematical generalization of the familiar three dimensional world that we live in. A real (respectively complex) vector space $\mathscr{V}$ is a collection of objects (called vectors) with defined properties of vector addition and multiplication by real (respectively complex) scalars that satisfy the familiar relations

$$\begin{aligned} v_1 + v_2 &= v_2 + v_1 \\ v_1 + (v_2 + v_3) &= (v_1 + v_2) + v_3 \\ (\lambda_1 + \lambda_2)v &= \lambda_1 v + \lambda_2 v \\ \lambda(v_1 + v_2) &= \lambda v_1 + \lambda v_2 \end{aligned} \tag{2.48}$$

for all v, v_1, v_2, v_3 in $\mathscr{V}$ and all scalars λ, λ_1, λ_2. The zero vector in $\mathscr{V}$ is denoted by the symbol 0. A *vector subspace* (or, more simply, a subspace) $\mathscr{U} \subset \mathscr{V}$ is any subset of $\mathscr{V}$ that satisfies the properties defined above.

It is easily seen that $\mathscr{R}^p$ (respectively $\mathscr{C}^p$) is a real (respectively complex) vector space. Also the set of real (respectively complex) $p \times q$ matrices is a real (respectively complex) vector space. Other examples and constructs of relevance to this text include

1. If $\mathscr{V}$ is any real vector space, then its *complexification* $\mathscr{V}^c$ is defined to be the complex vector space of all complex vectors $v = v_1 + iv_2$ with both v_1 and v_2 elements of $\mathscr{V}$. $\mathscr{V}^c$ is sometimes written in the form

$$\mathscr{V}^c = \mathscr{V} \oplus i\mathscr{V}. \tag{2.49}$$

2. The space of infinite sequences (or time series) $\alpha = \{\alpha_0, \alpha_1, \alpha_2, \ldots\}$ with $\alpha_j \in \mathscr{R}^p$ (or $\mathscr{C}^p$) is a vector space with addition $\gamma = \alpha + \beta$ and multiplication by scalars $\gamma = \lambda\alpha$ defined by the equations, $\gamma_j = \alpha_j + \beta_j$ and $\gamma_j = \lambda\alpha_j$ for $j = 0, 1, 2, 3, \ldots$. A number of subspaces are of particular relevance here including ℓ_∞ (the subspace of bounded sequences of scalars satisfying $\sup_{j\geq 0} |\alpha_j| < +\infty$) and ℓ_2 (the subspace of sequences of scalars satisfying $\sum_{j=0}^{\infty} |\alpha_j|^2 < +\infty$)
3. The space of real or complex valued continuous functions of a real variable t on an interval $a \leq t \leq b$ (denoted $[a, b] \subset \mathscr{R}$) is denoted by the symbol $C[a, b]$ and is a vector space with the usual definitions of addition and multiplication.
4. The space of all functions defined on $[a, b]$ and taking values in $\mathscr{R}^p$ (respectively $\mathscr{C}^p$) is a real (respectively complex) vector space with the usual definitions of addition and multiplication. The real vector space $L_2^p[a, b]$ is the set of all real $p \times 1$ vector-valued functions f such that the Lebesgue integral

$$\|f\|^2 = \int_a^b \|f(t)\|^2 dt \tag{2.50}$$

is well defined and finite. If Q is any symmetric, positive definite $p \times p$ matrix, then an equivalent statement is that

$$\|f\|_Q^2 = \int_a^b f^T(t) Q f(t) dt \tag{2.51}$$

is well defined and finite. If $p = 1$, then the space is written $L_2[a, b]$.

The notion of *linear independence* of a set of vectors follows the example of matrix theory. More precisely, a set of vector $\{v_j\}_{1\leq j\leq M}$ is linearly independent if, and only if, $\sum_{j=1}^{M} a_j v_j = 0$ implies that all scalars a_j are zero. A *basis* for $\mathscr{V}$ is a linearly independent set $\{v_j\}_{1\leq j\leq M}$ such that all elements $v \in \mathscr{V}$ can be written as a unique linear combination of the $\{v_j\}_{1\leq j\leq M}$. If M is finite, then the space is said to be *finite dimensional* of *dimension* M. Otherwise it is infinite dimensional. The spaces $\mathscr{R}^p$ and $\mathscr{C}^p$ have dimension p whilst $C[a, b]$ and $L_2[a, b]$ are infinite dimensional. For infinite dimensional spaces, the statement is more clearly stated by saying that the basis set

$\{v_j\}_{j\geq 0}$ has the property that all finite subsets of this set are linearly independent and the set of all finite linear combinations is *dense* in $\mathscr{V}$. The concept of denseness is more fully described in Sect. 2.3.4.

Finally,

1. The space $\mathscr{V}$ is said to be the sum of the vector subspaces $\{\mathscr{V}_j\}_{1\leq j\leq q}$ of $\mathscr{V}$, written

$$\mathscr{V} = \mathscr{V}_1 + \mathscr{V}_2 + \cdots + \mathscr{V}_q, \tag{2.52}$$

   if any vector $v \in \mathscr{V}$ can be written as a linear combination $v = \sum_{j=1}^{q} \alpha_j v_j$ with $v_j \in \mathscr{V}_j,\ 1 \leq j \leq q$ and suitable choice of scalars $\{\alpha_j\}_{1\leq j\leq q}$. It is a *direct sum decomposition* written

$$\mathscr{V} = \mathscr{V}_1 \oplus \mathscr{V}_2 \oplus \cdots \oplus \mathscr{V}_q \tag{2.53}$$

   if, and only if, each $v \in \mathscr{V}$ can be written as a unique linear combination of elements of the subspaces.
2. A *product vector space* constructed from vector spaces $\{\mathscr{V}_j\}_{1\leq j\leq q}$ is denoted by the *Cartesian Product* notation

$$\mathscr{V} = \mathscr{V}_1 \times \mathscr{V}_2 \times \cdots \times \mathscr{V}_q \tag{2.54}$$

   and consists of the set of p-tuples $v = \{v_1, v_2, \ldots, v_q\}$ with $v_j \in \mathscr{V}_j, 1 \leq j \leq q$ and the same laws of composition as those defined for (finite) times series. An example of this notation is the real product space $L_2^p[a, b]$ defined by

$$L_2^p[a, b] = \underbrace{L_2[a, b] \times L_2[a, b] \times \cdots \times L_2[a, b]}_{p - copies.} \tag{2.55}$$

   It is sometimes convenient to identify $L_2^p[a, b]$ with the space of $p \times 1$ vectors f with elements consisting of real valued functions $f_j \in L_2[a, b],\ 1 \leq j \leq p$.

2.3.2 Normed Spaces

Measures of magnitude are important in applications of mathematics and are used extensively in this text as a means of algorithm design and analysis. For the vector space $\mathscr{R}^p$, the familiar measure is the Euclidean length of the vector defined as $\|v\| = \sqrt{v^T v}$. This is just an example of the more general concept of a *vector norm*. More precisely, if $\mathscr{V}$ is a finite or infinite dimensional, real or complex vector space, then a norm on $\mathscr{V}$ is a mapping from vectors v into real numbers $\|v\|$ with the properties that

$$\begin{gathered} \|v\| \geq 0 \\ \|v\| = 0 \quad \textit{if, and only if,} \quad v = 0 \\ \|\lambda v\| = |\lambda| \|v\| \\ \|v_1 + v_2\| \leq \|v_1\| + \|v_2\| \end{gathered} \tag{2.56}$$

for all vectors v, v_1, v_2 in $\mathscr{V}$ and scalars λ. Note that the space $\mathscr{V}$ possessing the norm is normally understood from the context but it is often useful to identify the space using a subscript such as $\|v\|_{\mathscr{V}}$ or by other means.

An example of a norm in $L_2^p[a, b]$ is

$$\|f\| = \left(\int_a^b e^{2\alpha t} f^T(t) Q f(t) dt \right)^{1/2} \tag{2.57}$$

where Q is a symmetric, real, positive-definite $p \times p$ matrix and α is any real scalar. Also, a norm in $C[a, b]$ can be defined by

$$\|f\| = \sup_{a \leq t \leq b} \left(e^{\alpha t} |f(t)| \right). \tag{2.58}$$

The $L_2^p[a, b]$ norm is also a norm on $C[a, b]$.

For real (or complex) $p \times q$ matrices, A, one choice of norm is the maximum singular value $\overline{\sigma}(A)$ of A whilst, if A is real, another is the so-called *Frobenius Norm* defined by the trace formula

$$\|A\| = \sqrt{tr[A^T A]} \quad (\textit{the Frobenius Norm}). \tag{2.59}$$

or, more generally,

$$\|A\| = \sqrt{tr[W A^T Q A]} \quad (\textit{the weighted Frobenius Norm}). \tag{2.60}$$

where Q and W are symmetric and positive definite matrices.

When endowed with a chosen norm $\|\cdot\|$, the space $\mathscr{V}$ is called a *normed space*. The choice of norm is non-unique and the same underlying vector space, when given different norms, generates a new normed space. For $\mathscr{R}^p$, the following are norms for $v = [v_1, v_2, \ldots, v_p]^T$,

$$\|v\| = \sqrt{v^T Q v} \ \ (Q = Q^T > 0), \quad \|v\| = \max |v_j|, \quad \|v\| = (\sum_{j=1}^{p} |v_j|^q)^{1/q}, q \geq 1. \tag{2.61}$$

Two norms $\|\cdot\|_1$ and $\|\cdot\|_2$ on the same underlying vector space are said to be *topologically equivalent* if, and only if, there exists scalars $0 < \beta_1 \leq \beta_2$ such that, for all $v \in \mathscr{V}$,

$$\beta_1 \|v\|_1 \leq \|v\|_2 \leq \beta_2 \|v\|_1 \tag{2.62}$$

All norms on a given finite dimensional space are topologically equivalent.

2.3.3 Convergence, Closure, Completeness and Banach Spaces

Given a normed space $\mathscr{V}$, an infinite sequence $\{v_j\}_{j\geq 0} = \{v_0, v_1, v_2, \ldots\}$ is said to *converge in the norm topology* (or, more simply, to *converge*) to a limit vector $v \in \mathscr{V}$ if, and only if,

$$\lim_{j\to\infty} \|v - v_j\|_{\mathscr{V}} = 0 \quad (\textit{written} \lim_{j\to\infty} v_j = v). \tag{2.63}$$

The nature of this convergence is defined by the norm used but it is easily seen that convergence with respect to one norm implies convergence with respect to any other topologically equivalent norm.

A subset $S \subset \mathscr{V}$ is said to be an *open* subset if, for every point $v \in \mathscr{V}$, the *Open Ball* $B(v; \delta)$ defined by

$$B(v; \delta) = \{w : w \in \mathscr{V}, \|v - w\| < \delta\} \tag{2.64}$$

lies in S for some choice of $\delta > 0$. S is said to be *closed* if it contains the limit points of all convergent sequences with elements in S. The *closure* of a subset S (denoted $\overline{S}$) is the set consisting of points in S plus the limits of all convergent sequences in S. One consequence of this is that the *Closed Ball* $B_c(v; \delta)$ defined by

$$B_c(v; \delta) = \{w : w \in \mathscr{V}, \|v - w\| \leq \delta\} \tag{2.65}$$

is the closure of the open ball $B(v; \delta)$. Subsets can be neither open nor closed. For real numbers, the symbols $[a, b]$, $[a, b)$, $(a, b]$ and (a, b) are used to denote the intervals, respectively,

$$\begin{array}{l} \{t : a \leq t \leq b\} \quad (\textit{a closed interval}), \\ \{t : a \leq t < b\} \quad (\textit{a half open interval}), \\ \{t : a < t \leq b\} \quad (\textit{a half open interval}), \\ \{t : a < t < b\} \quad (\textit{an open interval}), \end{array} \tag{2.66}$$

A *Cauchy Sequence* $\{v_j\}_{j\geq 0}$ in $\mathscr{V}$ is a sequence with the property that, for all $\varepsilon > 0$, there exists an integer n_ε such that $\|v_j - v_k\| < \varepsilon$ for all $j \geq n_\varepsilon$ and $k \geq n_\varepsilon$. That is, all points v_j in the sequence get "closer and closer together" as the index j increases. In general, not all Cauchy sequences converge. An example of this is the space $C[a, b]$ with the $L_2[a, b]$ norm. It is a simple matter to construct a sequence of continuous functions that converge in norm to a discontinuous function which, by definition, is not in $C[a, b]$. A normed space where all Cauchy sequences converge is said to be *complete* and it is said to be a *Banach Space*. For the purposes of this text, note that all normed spaces used are Banach Spaces unless otherwise stated including $\mathscr{R}^p$, $\mathscr{C}^p$, ℓ_2 and $L_2^p[a, b]$ and their Cartesian products for any p and $-\infty < a < b < +\infty$.

2.3.4 Linear Operators and Dense Subsets

All $p \times q$ real matrices are examples of linear operators between the real spaces $\mathscr{R}^q$ and $\mathscr{R}^p$. The more general concepts of a linear operator/linear operators $\Gamma : \mathscr{V} \mapsto \mathscr{W}$ mapping a vector space $\mathscr{V}$ into another vector space $\mathscr{W}$ follows similar lines by satisfying the linearity assumptions that, for all v, v_1, v_2 in $\mathscr{V}$ and scalars λ,

$$\begin{aligned} \Gamma(v_1 + v_2) &= \Gamma v_1 + \Gamma v_2 \\ \Gamma(\lambda v) &= \lambda \Gamma v \end{aligned} \tag{2.67}$$

For example, if $\mathscr{V} = L_2^q[0, T]$ and $\mathscr{W} = L_2^p[0, T]$, T is finite and $H(t)$ is a $p \times q$ matrix with elements that are continuous in t, then the mapping $v \mapsto \Gamma v$ defined by the *Convolution Integral*

$$(\Gamma v)(t) = \int_0^t H(t - t')v(t')dt', \quad 0 \le t \le T, \tag{2.68}$$

is a well-defined linear operator. The *identity or unit operator* in $\mathscr{V}$ is the linear operator $I : \mathscr{V} \mapsto \mathscr{V}$ defined by $Iv = v$ for all $v \in \mathscr{V}$. If both $\mathscr{V}$ and $\mathscr{W}$ are real vector spaces then a linear operator $\Gamma : \mathscr{V} \mapsto \mathscr{W}$ can be extended to a linear operator (again denoted by Γ) mapping the complexification $\mathscr{V}^c$ into the complexification $\mathscr{W}^c$ by the relation $\Gamma(u + iv) = \Gamma u + i\Gamma v$. Also two operators $\Gamma_1 : \mathscr{V} \mapsto \mathscr{V}$ and $\Gamma_2 : \mathscr{V} \mapsto \mathscr{V}$ are said to *commute* if

$$\Gamma_1 \Gamma_2 = \Gamma_2 \Gamma_1. \tag{2.69}$$

Linear operators can be associated with norms quite easily. More precisely, with the notation used above, suppose that both $\mathscr{V}$ and $\mathscr{W}$ are normed spaces, then the *operator norm of* Γ (induced by the norms in $\mathscr{V}$ and $\mathscr{W}$) is defined to be

$$\|A\| = \sup_{v \neq 0} \frac{\|\Gamma v\|_{\mathscr{W}}}{\|v\|_{\mathscr{V}}} = \sup_{\|v\|=1} \frac{\|\Gamma v\|_{\mathscr{W}}}{\|v\|_{\mathscr{V}}} \tag{2.70}$$

If the norm is finite, the operator is said to be *bounded*. In all other cases, it is *unbounded*. The identity operator is bounded with induced norm $\|I\| = 1$.

The definition of the operator norm implies that, for all $v \in \mathscr{V}$,

$$\|\Gamma v\|_{\mathscr{W}} \le \|\Gamma\| \|v\|_{\mathscr{V}} \tag{2.71}$$

which implies the fact that boundedness of an operator is equivalent to its continuity. In addition, it is easily shown that, if $\Gamma_2 : \mathscr{V} \mapsto \mathscr{W}$ and $\Gamma_1 : \mathscr{W} \mapsto \mathscr{Z}$ are two bounded linear operators between normed spaces, then the composite operator $\Gamma_1 \Gamma_2 : \mathscr{V} \mapsto \mathscr{Z}$ defined by $(\Gamma_1 \Gamma_2)v = \Gamma_1(\Gamma_2 v)$ has a norm bound

$$\|\Gamma_1 \Gamma_2\| \le \|\Gamma_1\| \|\Gamma_2\|. \tag{2.72}$$

In a similar manner, the operator sum $\Gamma_1+\Gamma_2$ defined by $(\Gamma_1+\Gamma_2)v = \Gamma_1 v+\Gamma_2 v$ and multiplication by scalars $(\lambda\Gamma)v = \lambda(\Gamma v)$ rules make the *set* $\mathscr{L}(\mathscr{V};\mathscr{W})$ *of bounded linear operators from* $\mathscr{V}$ *into* $\mathscr{W}$ into a normed vector space in its own right. If $\mathscr{V}$ and $\mathscr{W}$ are Banach spaces, then so is $\mathscr{L}(\mathscr{V};\mathscr{W})$.

The *kernel* and *range* of an operator $\Gamma : \mathscr{V} \to \mathscr{W}$ play a vital role in analysis and are defined (as for matrices) using

$$\begin{aligned} ker[\Gamma] &= \{v \in \mathscr{V} : \Gamma v = 0\} \\ \mathscr{R}[\Gamma] &= \{w \in \mathscr{W} : w = \Gamma v \, for \, some \, v \in \mathscr{V}\} \end{aligned} \tag{2.73}$$

The operator is *injective* (or *one-to-one*) if its kernel is the single point $\{0\}$ and it is *surjective* (or *onto*) if its range is $\mathscr{W}$. It is *bijective* if it is both injective and surjective. If Γ is bijective and $\mathscr{W}$ is finite dimensional, then it has a bounded, linear inverse $\Gamma^{-1} : \mathscr{W} \mapsto \mathscr{V}$ defined by the relation $\Gamma^{-1}(\Gamma v) = v$ for all $v \in \mathscr{V}$ or, more simply $\Gamma^{-1}\Gamma = I$ where I is the identity in $\mathscr{V}$. If $\mathscr{V} = \mathscr{W}$, then, in addition, $\Gamma\Gamma^{-1} = I$.

The notion of inverse familiar in matrix theory also has relevance to the interpretation of the idea of *inverse operators* but with more technical complexity if $\mathscr{V}$ and $\mathscr{W}$ are infinite dimensional. One concept that is central to the discussion is that of a *dense subset* S_1 of a subset S of a normed space $\mathscr{V}$. More precisely, S_1 is dense in S if, and only if, for every point $v \in S$ and for every $\varepsilon > 0$, there exists a point $v_\varepsilon \in S_1$ such that $\|v - v_\varepsilon\| < \varepsilon$. In effect, every point in S has a point in S_1 arbitrarily close to it. Three observations are related to this

1. If Γ is bounded, then its kernel is closed.
2. Γ being bounded does not necessarily imply that its range is closed.
3. If $\mathscr{W}$ is finite dimensional and Γ is bounded, Γ has a closed range.

The second observation can be illustrated by the case of a convolution operator (2.68) mapping $L_2^q[0, T]$ into $L_2^p[0, T]$. The range of Γ is, at best, the set of continuous $p \times 1$ matrix valued functions which is known to be dense in $L_2^p[0, T]$.

The important point that follows from the above is that the range of the operator in infinite dimensional spaces has more complex properties than those observed for matrices. The consequences of this fact are many in number and include the possibility that the inverse of a bounded operator may exist but be unbounded. If Γ has a bounded (and hence continuous) inverse, it is said to be a *Homeomorphism*. In such cases it is easily seen that $1 \leq \|\Gamma\|\|\Gamma^{-1}\|$.

For matrices, the eigenvalues of a matrix A are defined to be scalars λ which ensure that $\lambda I - A$ has no inverse. The idea of eigenvalues requires careful generalization to the case of linear operators. More precisely, suppose that $\mathscr{V}$ is a complex Banach space, then the *Resolvent Set* of a linear operator $\Gamma : \mathscr{V} \mapsto \mathscr{V}$ is defined to be the set of complex numbers λ where $\lambda I - \Gamma$ is bijective. As a consequence of the *Open Mapping Theorem*, for any such λ, the *Resolvent Operator* $(\lambda I - \Gamma)^{-1}$ is bounded. Using this construction, *the spectrum* (denoted by $spec[\Gamma]$) of Γ is defined to be the complement of the Resolvent Set and hence is the set of complex numbers λ where $\lambda I - \Gamma$ does not have a bounded inverse. This definition includes the *eigenvalues* of Γ (the so-called *Point Spectrum* defined by the existence of non-zero

eigenvectors/eigenfunctions $v \in \mathscr{V}$ such that $\Gamma v = \lambda v$) but also other points in what are termed the *Continuous and Residual Spectrum*. In finite dimensional spaces, the residual and continuous spectra are empty and the results of matrix algebra describe the spectrum completely in terms of eigenstructure. Finally,

1. the *spectral radius* of a bounded operator $\Gamma : \mathscr{V} \mapsto \mathscr{V}$ is defined by

$$r(\Gamma) = \sup\{|\lambda| : \lambda \in spec[\Gamma]\}, \tag{2.74}$$

   a definition that reduces to that for matrices if $\mathscr{V} = \mathscr{C}^p$. In particular, if $\mathscr{V}$ is a Banach space, then

$$r(\Gamma) = \lim_{k \to \infty} \|\Gamma^k\|^{1/k} \tag{2.75}$$

   which relates the spectral radius to powers of Γ. As $\|\Gamma^k\| \leq \|\Gamma\|^k$ for all $k \geq 1$, this implies that

$$r(\Gamma) \leq \|\Gamma\| \tag{2.76}$$

   and, for all $\varepsilon > 0$, there exists a real number $M_\varepsilon \geq 1$ such that

$$\|\Gamma^k\| \leq M_\varepsilon (r(\Gamma) + \varepsilon)^k. \tag{2.77}$$

2. The ideas of *functions of operators* and the *Spectral Mapping Theorem*, easily proven for matrices, can be extended to bounded operators from Banach spaces into Banach spaces. More precisely, if $f(z)$ has a power series expansion $\sum_{j=0}^{\infty} f_j z^j$ with radius of convergence R, then $f(\Gamma)$ is defined to be the operator $\sum_{j=0}^{\infty} f_j \Gamma^j$ which is convergent if $r(\Gamma) < R$. The spectrum of $f(\Gamma)$ is simply $\{z : z = f(\eta),\ \eta \in spec[\Gamma]\}$ or, more compactly,

$$spec[f(\Gamma)] = f(spec[\Gamma]). \tag{2.78}$$

   For example, the *Resolvent* $(\lambda I - \Gamma)^{-1}$ has the power series representation

$$(\lambda I - \Gamma)^{-1} = \sum_{j=0}^{\infty} \lambda^{-(j+1)} \Gamma^j \tag{2.79}$$

   which is convergent if Γ has spectral radius strictly less than $|\lambda|$. A sufficient condition for this is that $\|\Gamma\| < |\lambda|$. The spectrum of the Resolvent is $\{z : z = (\lambda - \eta)^{-1},\ \eta \in spec[\Gamma]\}$.

The following result relates the spectral radius to the iterative learning control studies in this text. More precisely, the spectral radius describes convergence in norm of a simple, but typical, iteration formula.

Theorem 2.1 (Convergence in Norm and the Spectral Radius) *Let $\mathscr{V}$ be a Banach space. Then, given an arbitrary starting vector $v_0 \in \mathscr{V}$ and a bounded linear operator $\Gamma : \mathscr{V} \mapsto \mathscr{V}$, the sequence $\{v_j\}_{j\geq 0}$ generated by the iteration $v_{j+1} = \Gamma v_j, j \geq 0$, converges (in norm) to zero if (a sufficient condition)*

$$r(\Gamma) < 1. \tag{2.80}$$

A sufficient condition for this to be true is that $\|\Gamma\| < 1$.

Proof Note, using induction, $v_j = \Gamma^j v_0$ for all $j \geq 0$. Using the notation above, the assumptions make possible the selection of $\varepsilon > 0$ such that $r(\Gamma) + \varepsilon < 1$. It follows that, as required,

$$\|v_j\| = \|\Gamma^j v_0\| \leq \|\Gamma^j\| \|v_0\| \leq M_\varepsilon (r(\Gamma) + \varepsilon)^j \to 0 \quad as\ j \to \infty. \tag{2.81}$$

The theorem is proved as the norm condition is sufficient to ensure the required condition on the spectral radius. □

Note the conceptual similarity of this results to the familiar results from discrete time, sampled data systems control where asymptotic stability of $x_{j+1} = Ax_j$ is equivalent to the condition $r(A) < 1$ and hence is equivalent to the poles of the systems transfer function being inside the unit circle of the complex plane.

2.4 Hilbert Spaces

2.4.1 Inner Products and Norms

Although Banach spaces play a role is some areas of Control Theory and Optimization, the addition of geometrical structures plays an important role in algorithm design. The relevant structure is that of a *Hilbert Space*. More precisely, let $\mathscr{V}$ be a real (respectively complex) Banach space endowed with an associated *inner product* $\langle \cdot, \cdot \rangle : \mathscr{V} \times \mathscr{V} \mapsto \mathscr{R}$ (respectively $\mathscr{C}$) possessing the properties that, for all u, v, w in $\mathscr{V}$ and real (respectively complex) scalars λ,

$$\begin{aligned} \langle u, v \rangle &= \langle v, u \rangle \ \ (\textit{respectively } \langle u, v \rangle = \overline{\langle v, u \rangle}), \\ \langle u, v + w \rangle &= \langle u, v \rangle + \langle u, w \rangle, \\ \langle u, \lambda v \rangle &= \lambda \langle u, v \rangle, \\ \langle v, v \rangle &\geq 0 \ \ \textit{and} \\ \langle v, v \rangle &= 0 \quad \textit{if, and only if,} \ \ v = 0. \end{aligned} \tag{2.82}$$

Suppose also that the norm in $\mathscr{V}$ can be computed from the inner product using the formula

$$\|v\| = \sqrt{\langle v, v \rangle}, \tag{2.83}$$

then $\mathscr{V}$ is said to be a real (respectively complex) Hilbert Space. Note that, if it is necessary to identify the space or some other aspect of the formulae arising in the theory, the norm and inner product may be given subscripts as an aide memoire to the reader. For example, to identify the space being considered, both $\|v\|$ and $\langle u, v\rangle$ can be written in the form $\|v\|_{\mathscr{V}}$ and $\langle u, v\rangle_{\mathscr{V}}$.

Examples of Hilbert spaces include:

1. The space $\mathscr{R}^p$ is a Hilbert space with inner product $\langle u, v\rangle = u^T Qv$ where Q is any symmetric, positive definite $p \times p$ matrix.
2. If $-\infty < a < b < +\infty$, then $L_2^p[a, b]$ is a Hilbert space with inner product

$$\langle u, v\rangle = \int_a^b u^T(t)Q(t)v(t)dt \tag{2.84}$$

where $Q(t)$ is any piecewise continuous $p \times p$ matrix satisfying an inequality of the form

$$\alpha_1 I_p \leq Q(t) \leq \alpha_2 I_p, \quad \textit{for all } t \in [a, b] \tag{2.85}$$

and some scalars $0 < \alpha_1 \leq \alpha_2$. For example, if $\alpha \geq 0$ and $Q(t) = e^{2\alpha t}Q$ with Q a constant, symmetric, positive definite matrix with eigenvalues $0 < q_1 \leq q_2 \leq \cdots \leq q_p$, then the conditions are satisfied with $\alpha_1 = q_1 e^{\alpha a}$ and $\alpha_2 = q_p e^{\alpha b}$.

Finally, the inner product has a number of useful additional properties including the *Cauchy-Schwarz Inequality* which takes the form, for any u, v in $\mathscr{V}$

$$|\langle u, v\rangle| \leq \|u\|\|v\| \tag{2.86}$$

with equality holding if, and only if, v is a multiple of u. That is, if and only if, $v = \lambda u$ for some scalar λ.

Also the inner product allows the introduction of ideas of orthogonality. More precisely, two vectors u, v in $\mathscr{V}$ are said to be *orthogonal* if, and only if, $\langle u, v\rangle = 0$, a definition that is consistent with that used for Euclidean geometry in $\mathscr{R}^p$. The *orthogonal complement* of a vector subspace S of $\mathscr{V}$ is denoted $S^\perp$ where

$$S^\perp = \{v \in \mathscr{V} : \langle v, u\rangle = 0 \quad \textit{for all } u \in S\} \tag{2.87}$$

$S^\perp$ is a closed subspace. If, in addition, S is a closed subspace, then $\mathscr{V}$ has the direct sum decomposition

$$\mathscr{V} = S \oplus S^\perp = \{w = u + v : u \in S, \quad v \in S^\perp\}. \tag{2.88}$$

If $\{v_j\}_{j\geq 1}$ is a basis for $\mathscr{V}$ and $\langle v_j, v_k\rangle = 0$ whenever $j \neq k$, then the basis set is an *orthogonal basis*. If, by suitable scaling (notably replacing each v_j by the normalized vector $v_j/\|v_j\|$), the basis set is said to be an *orthonormal basis* with the defining property that, for all $j \geq 1$ and $k \geq 1$,

$$\langle v_j, v_k \rangle = \delta_{jk} \quad where \tag{2.89}$$

the symbol δ_{jk} is the *Kronecker Delta* defined by $\delta_{jk} = 0$ if $j \neq k$ and is unity otherwise. Under these conditions, any vector $v \in \mathscr{V}$ has the form

$$v = \sum_{j=1}^{\infty} \alpha_j v_j \quad with \quad \alpha_j = \langle v_j, v \rangle, \, for \, j \geq 1, \, and \, \|v\|^2 = \sum_{j=1}^{\infty} |\alpha_j|^2 < \infty. \tag{2.90}$$

Finally, the ideas of inner products can be applied to some normed spaces that are not complete. Examples include the space $C[a, b]$ endowed with the inner product (and induced norm) used for the Hilbert space $L_2[a, b]$ or, more generally, a dense, but not complete, subspace of a Hilbert space. Such spaces are said to be *Pre-Hilbert Spaces*. The geometry of such spaces is identical to that of Hilbert spaces but results that rely on the convergence of Cauchy sequences (and hence the existence of limits) need to be carefully considered.

2.4.2 Norm and Weak Convergence

The convergence of sequences in Hilbert spaces is defined as in any normed space but it is often called *convergence in norm*, *convergence in the norm topology* or, more simply, *norm convergence*. This is because, in Hilbert spaces, another useful definition of convergence is that of *Weak Convergence*. More precisely, a sequence $\{v_j\}_{j \geq 0}$ in a Hilbert space $\mathscr{V}$ is said to *converge weakly* to a vector $v_\infty \in \mathscr{V}$ if, and only if,

$$\lim_{j \to \infty} \langle f, v_\infty - v_j \rangle = 0 \quad for \, all \, f \in \mathscr{V}. \tag{2.91}$$

The Cauchy-Schwarz inequality immediately indicates that convergence in norm to v_∞ implies weak convergence to that vector. However, weak convergence of a sequence does not imply, necessarily, its convergence in norm.

The limit need only be valid on a dense subset of $\mathscr{V}$ as,

Theorem 2.2 *Using the notation above, suppose that the sequence $\{v_j\}_{j \geq 0}$ is bounded in the sense that there exists a real scalar M such that $\|v_j\| \leq M$ for all $j \geq 0$. Suppose also that, for some dense subset $S \subset \mathscr{V}$*

$$\lim_{j \to \infty} \langle f, v_\infty - v_j \rangle = 0 \quad for \, all \, f \in S. \tag{2.92}$$

Then, $\{v_j\}_{j \geq 0}$ converges weakly to v_∞.

Proof First note that $\|v_\infty - v_j\| \leq \|v_\infty\| + \|v_j\| \leq \|v_\infty\| + M$. Next write, for any $f_\varepsilon \in \mathscr{V}$,

$$\langle f, v_\infty - v_j \rangle = \langle f - f_\varepsilon, v_\infty - v_j \rangle + \langle f_\varepsilon, v_\infty - v_j \rangle. \tag{2.93}$$

Let $\varepsilon > 0$ be arbitrary and choose $f_\varepsilon \in S$ so that $\|f - f_\varepsilon\| < \varepsilon$, then the inequality

$$|\langle f, v_\infty - v_j\rangle| \leq \|(f - f_\varepsilon\|\|v_\infty - v_j\| + |\langle f_\varepsilon, v_\infty - v_j\rangle| \tag{2.94}$$

indicates that

$$\limsup_{j\to\infty} |\langle f, v_\infty - v_j\rangle| \leq \varepsilon(\|v_\infty\| + M) \tag{2.95}$$

which proves the result as the left hand side is independent of $\varepsilon > 0$ which is arbitrary. □

Another useful property of weak convergence is that of guaranteed convergence of subsequences. This is stated as follows and is usually associated with the notion of *weak compactness of the closed unit ball* $B(0; 1)$ in $\mathscr{V}$.

Theorem 2.3 (Ascoli's Theorem) *If $\mathscr{V}$ is a Hilbert space and $S = \{v_j\}_{j\geq 0}$ is an infinite but bounded sequence of vectors, then S has a subsequence $S_1 = \{v_j\}_{k_j\geq 0}$ that converges weakly to some vector v_∞ in $\mathscr{V}$.*

The result states that *all* bounded sequences in any Hilbert space contain weakly convergent subsequences. If the convergent subsequence is removed from the original sequence, it will leave either a finite set (a situation which implies that the sequence itself converges weakly to v_∞) or an infinite sequence. In the second case, the remaining sequence is also bounded and hence (by Ascoli's Theorem) it, too, has a subsequence converging weakly to some (possibly different) weak limit $\hat{v}_\infty \in \mathscr{V}$. It is concluded that there is a possibility that S has many subsequences with different weak limits.

The essential property needed for weak convergence is boundedness of the sequence. The following result provides some insight into possibilities.

Theorem 2.4 (Weak Convergence and Operator Norms) *Any iteration $v_{j+1} = \Gamma v_j$, $j \geq 0$, in a real Hilbert space $\mathscr{V}$ where $\Gamma : \mathscr{V} \to \mathscr{V}$ is a bounded linear operator with norm $\|\Gamma\| \leq 1$ generates a sequence $\{v_j\}_{k_j\geq 0}$ that is bounded in norm and has weakly convergent subsequences. If $\|\Gamma\| < 1$, then the sequence converges in norm to zero.*

Proof As $v_j = \Gamma^j v_0$, it follows that $\|v_j\| \leq \|\Gamma\|^j\|v_0\|$ which proves the result as the sequence is always bounded by $\|v_j\| \leq \|v_0\|$. Ascoli's Theorem then indicates the existence of a weak limit of some subsequence. Convergence in norm if $\|\Gamma\| < 1$ follows from the definitions. □

Iterations of the form $v_{j+1} = \Gamma v_j$ appear regularly in this text. Theorems 2.1 and 2.4 above provide two conditions for some form of convergence. It is worth noting that the values of the spectral radius or norm of the operator Γ are central to the stated results. As $r(\Gamma) \leq \|\Gamma\|$, the use of the spectral radius will produce the best prediction of some form of convergence. This is particularly apparent in the case when $r(\Gamma) < 1 < \|\Gamma\|$ when the norm cannot be used to prove convergence

but the use of the spectral radius indicates convergence in norm to zero. The more difficult, but important, case that also plays a role in iterative control is the case when $r(\Gamma) = \|\Gamma\| = 1$ when weak convergence of subsequences is guaranteed by Theorem 2.4 but convergence in norm is not covered by either result.

2.4.3 Adjoint and Self-adjoint Operators in Hilbert Space

Suppose that $\Gamma : \mathscr{V} \mapsto \mathscr{W}$ is a bounded linear operator mapping a real or complex Hilbert space $\mathscr{V}$ into a real or complex Hilbert space $\mathscr{W}$. The *Adjoint Operator* $\Gamma^* : \mathscr{W} \mapsto \mathscr{V}$ is the uniquely defined bounded linear operator mapping $\mathscr{W}$ into $\mathscr{V}$ and satisfying the identity, for all $u \in \mathscr{W}$ and $v \in \mathscr{V}$,

$$\langle u, \Gamma v\rangle_{\mathscr{W}} = \langle \Gamma^* u, v\rangle_{\mathscr{V}} \tag{2.96}$$

There are many general links between an operator and its adjoint. These include the additive, multiplicative and inversion rules

$$\begin{aligned} (\Gamma_1 + \Gamma_2)^* &= \Gamma_1^* + \Gamma_2^*, \\ (\Gamma_1\Gamma_2)^* &= \Gamma_2^*\Gamma_1^* \textit{and, if } \mathscr{V} = \mathscr{W}, \quad (A^{-1})^* = (A^*)^{-1} \end{aligned} \tag{2.97}$$

(when the inverse exists). The cases of real and complex Hilbert spaces need a little care as, for $\Gamma : \mathscr{V} \mapsto \mathscr{W}$ and any scalar λ, the adjoint $(\lambda\Gamma)^* = \lambda\Gamma^*$ if $\mathscr{V}$ is a real Hilbert space but equal to $\overline{\lambda}\Gamma^*$ if $\mathscr{V}$ is a complex Hilbert space. Also

$$(\Gamma^*)^* = \Gamma. \tag{2.98}$$

A result that plays a role in the following text expresses the adjoint of a map into a product space in terms of adjoints of operators associated with each component.

Theorem 2.5 (The Adjoint of a Map into a Product Hilbert Space) *Let $\mathscr{V}, \mathscr{W}_1, \ldots, \mathscr{W}_p$ be real Hilbert spaces and define the product Hilbert space $\mathscr{W}_1 \times \cdots \times \mathscr{W}_p$ to be the product space with inner product and induced norm defined by*

$$\begin{aligned} \langle (w_1, \ldots, w_p), (z_1, \ldots, z_p)\rangle_{\mathscr{W}_1\times\cdots\times\mathscr{W}_p} &= \textstyle\sum_{j=1}^{p} \langle w_j, z_j\rangle_{\mathscr{W}_j} \\ \textit{and} \qquad \|(w_1, \ldots, w_p)\|^2_{\mathscr{W}_1\times\cdots\times\mathscr{W}_p} &= \textstyle\sum_{j=1}^{p} \|w_j\|^2_{\mathscr{W}_j}. \end{aligned} \tag{2.99}$$

Let the operator $G : \mathscr{V} \to \mathscr{W}_1 \times \cdots \times \mathscr{W}_p$ be linear and bounded. Then G can be represented by the mapping, for all $v \in \mathscr{V}$,

$$Gv = (G_1 v, G_2 v, \ldots, G_p v) \tag{2.100}$$

where $G_j : \mathscr{V} \to \mathscr{W}_j$ is linear and bounded. The adjoint map $G^* : \mathscr{W}_1 \times \cdots \times \mathscr{W}_p \to \mathscr{V}$ is the bounded linear operator defined by the relation,

$$G^*(w_1, w_2, \ldots, w_p) = G_1^* w_1 + G_2^* w_2 + \cdots + G_p^* w_p, \tag{2.101}$$

where, for $1 \le j \le p$, $G_j^* : \mathscr{W}_j \to \mathscr{V}$ is the adjoint of G_j.

Proof The characterization of G in terms of the G_j follows easily from the linearity of G. The adjoint of G is identified from the equation

$$\begin{aligned} \langle (w_1, \ldots, w_p), Gv \rangle_{\mathscr{W}_1 \times \cdots \times \mathscr{W}_p} &= \textstyle\sum_{j=1}^p \langle w_j, G_j v \rangle_{\mathscr{W}_j} \\ &= \textstyle\sum_{j=1}^p \langle G_j^* w_j, v \rangle_{\mathscr{V}} = \langle \sum_{j=1}^p G_j^* w_j, v \rangle_{\mathscr{V}}. \end{aligned} \tag{2.102}$$

The theorem is proved by comparing this with $\langle G^*(w_1, w_2, \ldots, w_p), v \rangle_{\mathscr{V}}$. □

An operator $\Gamma : \mathscr{V} \mapsto \mathscr{V}$ is *self adjoint* if, and only if, $\Gamma = \Gamma^*$. If Γ is self adjoint then $\langle u, \Gamma u \rangle_{\mathscr{V}}$ is always real. Γ is then said to be *positive* if $\langle u, \Gamma u \rangle \ge 0$ for all $u \in \mathscr{V}$, *positive definite* if it is positive and $\langle u, \Gamma u \rangle = 0$ if, and only if, $u = 0$ and *positive semi-definite* if it is positive but there exits a non-zero u such that $\langle u, \Gamma u \rangle = 0$. Positive commuting operators have special properties as follows:

Theorem 2.6 *If Γ_1, Γ_2 and Γ_3 are linear, bounded, self-adjoint, positive, commuting operators mapping a Hilbert space $\mathscr{V}$ into itself, then*

$$\begin{aligned} \Gamma_1 \ge 0 \;\&\; \Gamma_2 \ge 0 &\Rightarrow \Gamma_1 \Gamma_2 \ge 0 \\ \Gamma_1 \ge \Gamma_2 &\Rightarrow \Gamma_1 \Gamma_3 \ge \Gamma_2 \Gamma_3 \end{aligned} \tag{2.103}$$

The form of the adjoint operator depends on the spaces used and, in particular, on the form of inner product used. For example, matrix algebra proves that,

Theorem 2.7 (Adjoint of a Matrix Operator) *let $\mathscr{V}$ be $\mathscr{R}^p$ with inner product $\langle \hat{v}, v \rangle_{\mathscr{V}} = \hat{v}^T R v$ (where $R = R^T > 0$) and $\mathscr{W}$ be $\mathscr{R}^q$ with inner product $\langle \hat{w}, w \rangle_{\mathscr{W}} = \hat{w}^T Q w$ (where $Q = Q^T > 0$). Γ is a $p \times q$ real matrix with adjoint Γ^* satisfying, for all u and w,*

$$w^T Q \Gamma v = (\Gamma^* w)^T R v \quad \text{and hence} \quad \Gamma^* = R^{-1} \Gamma^T Q. \tag{2.104}$$

In particular, when $R = I_p$ and $Q = I_q$, the adjoint is simply the transpose of the matrix Γ.

Note: it is conventional to use the * notation to denote the adjoint operator but it is also often used to denote the complex conjugate transpose of a matrix. There is a possibility of confusion from time to time but careful attention to the context of the analysis should easily resolve any ambiguity.

The operator $\Gamma^* \Gamma$ is self adjoint. It is also positive as $\langle u, \Gamma^* \Gamma u \rangle = \langle \Gamma u, \Gamma u \rangle = \| \Gamma u \|^2 \ge 0$. As a consequence,

Theorem 2.8 (Invertibility and Positivity of Operators) *With the above notation,*

1. *The operator $\Gamma^*\Gamma : \mathscr{V} \to \mathscr{V}$ is positive definite if, and only if, $ker[\Gamma] = \{0\}$.*
2. *The operator $\Gamma : \mathscr{V} \to \mathscr{W}$ has a bounded inverse if, and only if, there exists a constant $\alpha > 0$ such that $\Gamma^*\Gamma \geq \alpha I_{\mathscr{V}}$ and $\Gamma\Gamma^* \geq \alpha I_{\mathscr{W}}$.*

These properties link invertibility to positivity as follows,

Theorem 2.9 (Invertibility and Positivity of Operators) *An operator $\Gamma : \mathscr{V} \to \mathscr{V}$ has a bounded inverse on the Hilbert space $\mathscr{V}$ if there exists a real number $\varepsilon_0 > 0$ such that*

$$\Gamma + \Gamma^* \geq \varepsilon_0^2 I. \tag{2.105}$$

Proof Noting that, for any real scalar λ,

$$\begin{aligned} 0 &\leq (I - \lambda\Gamma^*)(I - \lambda\Gamma) = I - \lambda(\Gamma + \Gamma^*) + \lambda^2\Gamma^*\Gamma \\ &\leq (1 - \lambda\varepsilon_0^2)I + \lambda^2\Gamma^*\Gamma. \end{aligned} \tag{2.106}$$

Exactly the same relation ship for $\Gamma\Gamma^*$ is obtained using $(I - \lambda\Gamma)(I - \lambda\Gamma^*)$ so that the positivity condition of the previous result holds by choosing λ so that $\alpha = \lambda^{-2}(\lambda\varepsilon_0^2 - 1) > 0$. □

The operator norms $\|\Gamma\|$ and $\|\Gamma^*\|$ are related by the expression,

$$\|\Gamma\| = \|\Gamma^*\| \tag{2.107}$$

and the range and kernels of the operators satisfy the orthogonality relations

$$\begin{array}{ll} (a) & \mathscr{R}[\Gamma^*]^\perp = ker[\Gamma] \text{ and hence } \quad \mathscr{R}[\Gamma]^\perp = ker[\Gamma^*] \\ (b) & \overline{\mathscr{R}[\Gamma^*]} = ker[\Gamma]^\perp \text{ and hence } \quad \overline{\mathscr{R}[\Gamma]} = ker[\Gamma^*]^\perp \end{array} \tag{2.108}$$

from which the *Projection Theorem in Hilbert space* (Theorem 2.17) gives

$$\mathscr{V} = \overline{\mathscr{R}[\Gamma^*]} \oplus ker[\Gamma] \quad \text{and} \quad \mathscr{W} = \overline{\mathscr{R}[\Gamma]} \oplus ker[\Gamma^*] \tag{2.109}$$

A following result provides an important property of a Hilbert space in terms of the range of an operator and and the kernel of its adjoint. The result has close links to the above but, more formally,

Theorem 2.10 (Denseness and the Orthogonal Complement of the Kernel) *Suppose that $\Gamma : \mathscr{V} \mapsto \mathscr{W}$ where $\mathscr{V}$ and $\mathscr{W}$ are Hilbert spaces. Then the range space $\mathscr{R}[\Gamma^*]$ is dense in $\mathscr{V}$ if, and only if, $ker[\Gamma] = \{0\}$.*

Proof If $ker[\Gamma] = \{0\}$, suppose that $\mathscr{R}[\Gamma^*]$ is not dense. It follows that its closure is a proper closed subspace S of $\mathscr{V}$ with an orthogonal complement $S^\perp$ containing non-zero vectors. Let $v \in S^\perp$ be non-zero and write $\langle v, \Gamma^* w\rangle = 0$ for all $w \in \mathscr{W}$. It follows that $\langle \Gamma v, w\rangle = 0$ for all $w \in \mathscr{W}$ so that (choosing $w = \Gamma v$) $\Gamma v = 0$

which contradicts the assumption that $ker[\Gamma] = \{0\}$. Next suppose that $\mathscr{R}[\Gamma^*]$ is dense. It follows that the condition $\langle v, \Gamma^* w\rangle = 0$ for all $w \in \mathscr{W}$ implies that $v = 0$ which trivially leads to $\mathscr{R}[\Gamma^*]^\perp = ker[\Gamma] = \{0\}$ as required. □

The properties of the range spaces of an operator and its adjoint are also connected as follows:

Theorem 2.11 (Closed Range Theorem) *If $\Gamma : \mathscr{V} \to \mathscr{W}$ is a bounded linear operator between Hilbert spaces, then Γ has a closed range in $\mathscr{W}$ if, and only if, the range of the adjoint Γ^* is closed in $\mathscr{V}$.*

The norm of a self adjoint operator $\Gamma : \mathscr{V} \to \mathscr{V}$ is related to the values taken by an associated quadratic form. More precisely its norm can be computed from the parameters

$$\begin{aligned} a &= \inf\{\langle u, \Gamma u\rangle : u \in \mathscr{V} \text{ and } \|u\| = 1\} \\ \text{and} \quad b &= \sup\{\langle u, \Gamma u\rangle : u \in \mathscr{V} \text{ and } \|u\| = 1\} \\ \text{to be} \quad \|\Gamma\| &= \max\{|a|, |b|\} \\ \text{and, in particular,} \quad \|\Gamma\| &= r(\Gamma). \end{aligned} \tag{2.110}$$

This expression can be written in the form, where I is the identity operator,

$$aI \leq \Gamma \leq bI \tag{2.111}$$

which forms the basis of the theorem

Theorem 2.12 (Invertibility of Self Adjoint Operators) *With the notation used above, suppose that $\Gamma : \mathscr{V} \to \mathscr{V}$ is self adjoint. Then the spectrum of Γ contains only real numbers in the closed interval $[a, b]$. In particular, Γ has a bounded inverse if $ab > 0$.*

A useful relationship valid when Γ is self adjoint and positive is

$$\|\Gamma\| = \sup\{\langle u, \Gamma u\rangle : u \in \mathscr{V} \text{ and } \|u\| = 1\} \tag{2.112}$$

which immediately yields the result that, for any bounded $\Gamma : \mathscr{V} \to \mathscr{W}$,

$$\|\Gamma^* \Gamma\| = \|\Gamma\|^2. \tag{2.113}$$

A useful consequence of this is that

Theorem 2.13 (Norm of a Matrix Operator) *Using the notation and assumptions of Theorem 2.7,*

$$\|\Gamma\|^2 = \sup\{\langle u, \Gamma^* \Gamma u\rangle : u \in \mathscr{V} \text{ and } \|u\| = 1\} = r(\Gamma^* \Gamma) \tag{2.114}$$

where $r(\Gamma^*\Gamma)$ is the spectral radius of $\Gamma^*\Gamma = R^{-1}\Gamma^T Q\Gamma$. Moreover, for all choices of Q and R, $\|\Gamma\|$ is the largest singular value of $\Gamma_{QR} = Q^{\frac{1}{2}}\Gamma R^{-\frac{1}{2}}$ and

$$\|\Gamma^*\| = \|\Gamma\|. \tag{2.115}$$

Proof The proof follows from Theorem 2.7 using Lagrange multiplier techniques to solve 2.112 regarded as a function minimization problem. Next, it is easy to see that $\det(\lambda I_p - \Gamma\Gamma^*) = \lambda^{p-q}\det(\lambda I_q - \Gamma^*\Gamma)$ so that the eigenvalues of $\Gamma^*\Gamma$ and $\Gamma\Gamma^*$ differ, at most, only by a number of zero eigenvalues. Finally, the eigenvalues of $\Gamma^*\Gamma$ are the squares of the singular values of $Q^{\frac{1}{2}}\Gamma R^{\frac{1}{2}}$ as $\Gamma^*\Gamma = R^{-\frac{1}{2}}\left[\left(Q^{\frac{1}{2}}\Gamma R^{-\frac{1}{2}}\right)^T Q^{\frac{1}{2}}\Gamma R^{-\frac{1}{2}}\right] R^{\frac{1}{2}}$. That is $\Gamma^*\Gamma$ and $\Gamma_{QR}^T\Gamma_{QR}$ are related by a similarity transformation. □

If $\Gamma : \mathscr{V} \mapsto \mathscr{V}$ is positive and self adjoint there exists a unique positive, self adjoint operator $\hat{\Gamma} : \mathscr{V} \mapsto \mathscr{V}$ with the property that $\Gamma = \hat{\Gamma}\hat{\Gamma}$. For this reason, $\hat{\Gamma}$ is said to be the unique *positive square root* of Γ and is written $\hat{\Gamma} = \Gamma^{1/2}$. The bounded, positive, self-adjoint linear operator $\Gamma^{1/2}$ has the properties that it commutes with every operator that commutes with Γ and

$$\Gamma^{1/2} = (\Gamma^{1/2})^* \geq 0, \quad ker[\Gamma^{1/2}] = ker[\Gamma] \quad and \quad \mathscr{R}[\Gamma] \subset \mathscr{R}[\Gamma^{1/2}] \tag{2.116}$$

so that, in particular, $\Gamma^{1/2}$ is positive definite if, and only if, Γ is positive definite.

The spectrum of a self adjoint operator Γ lies in the closed ball $B_c(0; r(\Gamma)) \subset B_c(0; \|\Gamma\|)$. Using (2.110), the spectrum of $\Gamma - \frac{b+a}{2}I$ lies in the closed ball $B_c(0, \frac{b-a}{2})$ and hence, using the spectral mapping Theorem, the spectrum of Γ lies in the shifted closed ball $\frac{b+a}{2} + B_c(0, \frac{b-a}{2})$. In particular,

Theorem 2.14 (Invertibility of $(I+\Gamma)^{-1}$) *Suppose that $\Gamma : \mathscr{Y} \to \mathscr{Y}$ where $\mathscr{Y}$ is a real Hilbert space is bounded, self adjoint and positive. Then $I + \Gamma$ is a bijection and the inverse operator $(I+\Gamma)^{-1}$ is well-defined and bounded.*

Proof Using the discussion preceding this result, $a = 0$ and $b = \|\Gamma\|$ and hence

$$spec[\Gamma] \subset \frac{\|\Gamma\|}{2} + B_c(0, \frac{\|\Gamma\|}{2}) . \tag{2.117}$$

The proof is now complete as -1 is not in the spectrum of Γ. □

Note: Operators of this type play a central role in Iterative Algorithms.

Finally, useful conditions for $\Gamma_1\Gamma_2$ to be self adjoint can be stated as follows

Theorem 2.15 (When is $\Gamma_1\Gamma_2$ self adjoint?) *Suppose that the two self-adjoint operators Γ_1 and Γ_2 map a real Hilbert space $\mathscr{Y}$ into itself and that Γ_2 is positive definite. Then the product $\Gamma_1\Gamma_2$ is self-adjoint if the inner product in $\mathscr{Y}$ is replaced by the new inner product*

$$\langle u, v\rangle_0 = \langle u, \Gamma_2 v\rangle_{\mathscr{Y}}. \tag{2.118}$$

The two topologies are equivalent if there exists a real scalar $\varepsilon_0^2 > 0$ *such that* $\Gamma_2 \geq \varepsilon_0^2 I$.

Proof The bilinear form $\langle u, v\rangle_0$ satisfies all the requirements of an inner product and its associated norm $\|\cdot\|_0$. The self-adjoint property follows as

$$\langle u, \Gamma_1\Gamma_2 v\rangle_0 = \langle u, \Gamma_2\Gamma_1\Gamma_2 v\rangle_{\mathscr{Y}} = \langle \Gamma_1\Gamma_2 u, \Gamma_2 v\rangle_{\mathscr{Y}} = \langle \Gamma_1\Gamma_2 u, v\rangle_0 \tag{2.119}$$

Finally, the existence of $\varepsilon_0^2 > 0$ ensures the topological equivalence of the two norms follows as

$$\varepsilon_0^2\|u\|_{\mathscr{Y}}^2 \leq \langle u, \Gamma_2 u\rangle_{\mathscr{Y}} = \|u\|_0^2 \leq \|\Gamma_0\|\|u\|_{\mathscr{Y}}^2. \tag{2.120}$$

□

Note: This result plays a role in the analysis of the convergence and robustness of many of the algorithms in the following chapters.

2.5 Real Hilbert Spaces, Convex Sets and Projections

The structure of real Hilbert spaces provides a powerful set of results related to optimization. These results are expressed in terms of *Projection onto Convex Sets*. A *convex set* $S \subset \mathscr{V}$ in a real Hilbert space $\mathscr{V}$ is any set satisfying the condition that, for any two points u, v in S, the vector

$$w = \lambda u + (1 - \lambda)v \in S \quad \textit{for all } \lambda \in [0, 1]. \tag{2.121}$$

(where λ is a real number). The vector w is said to be a *convex combination* of u and v. The *Convex Hull* of a set $S \subset \mathscr{V}$ is the smallest convex set containing S.

Suppose that v_0 is an arbitrary point of $\mathscr{V}$ and consider the problem of finding the *point in a convex set S that is closest to* v_0. This problem can be written formally as the solution (if it exists) of the optimization problem

$$v_1 = \arg\min\{\|v_0 - v\| : v \in S\} \tag{2.122}$$

That is, v_1 is the vector $v \in S$ that minimizes the norm $\|v_0 - v\|$ and hence is the nearest point in S to v_0. For visualization purposes, v_1 can be thought of as the projection of v_0 onto the set S. In general, it is possible that no such point exists but, for many problems in practice, a solution does exist. The most useful theorem characterizing the existence of v_1 and its relationship to v_0 is as follows:

Theorem 2.16 (Minimum Distance to a Closed Convex Set) *Suppose that S is a closed, convex set in the real Hilbert space* $\mathscr{V}$. *If* $v_0 \in \mathscr{V}$, *then the optimization Problem* (2.122) *has a unique solution* $v_1 \in S$. *A necessary and sufficient condition for* v_1 *to be that solution is that*

$$\langle v - v_1, v_1 - v_0 \rangle_{\mathscr{V}} \geq 0 \quad \textit{for all } v \in S. \tag{2.123}$$

A particular case of interest is when S is a closed vector subspace of $\mathscr{V}$.

Theorem 2.17 (The Projection Theorem in Hilbert Space) *Suppose that S is a closed vector subspace in the real Hilbert space $\mathscr{V}$. If $v_0 \in \mathscr{V}$, then the optimization Problem (2.122) has a unique solution $v_1 \in S$. A necessary and sufficient condition for v_1 to be that solution is that the following orthogonality condition is true,*

$$\langle v - v_1, v_1 - v_0 \rangle_{\mathscr{V}} = 0 \quad \textit{for all } v \in S. \tag{2.124}$$

In particular, as $v - v_1 \in S$ is arbitrary, the condition reduces to

$$\langle v, v_1 - v_0 \rangle_{\mathscr{V}} = 0 \quad \textit{for all } v \in S \tag{2.125}$$

which is simply the requirement that $v_1 - v_0$ is orthogonal to every vector in S.

Proof The existence and uniqueness of v_1 follows from the previous theorem as does the requirement that $\langle v - v_1, v_1 - v_0 \rangle_{\mathscr{V}} \geq 0$ for all $v \in S$. Suppose that there exists a vector $v \in S$ such that $\langle v - v_1, v_1 - v_0 \rangle_{\mathscr{V}} > 0$, then, noting that $\hat{v} = -v + 2v_1 \in S$, a simple computation indicates that $\langle \hat{v} - v_1, v_1 - v_0 \rangle_{\mathscr{V}} < 0$ contradicting the assumption that v_1 solves the problem. □

The case of S being a vector subspace gives rise to the notion of a *Projection Operator*. More precisely, using the notation of the Projection Theorem 2.17, the computation $v_0 \mapsto v_1$ defines a mapping $P_S : \mathscr{V} \to S$. The orthogonality condition also indicates that the mapping is linear and hence P_S is a linear operator called the *Orthogonal Projection Operator onto S*. It is bounded as, writing $v_1 = P_S v_0$, using the orthogonality condition, and noting that $0 \in S$,

$$\begin{aligned} \|v_0\|^2 = \|(v_0 - v_1) + v_1\|^2 &= \|v_0 - v_1\|^2 + 2\langle v_0 - v_1, v_1 \rangle + \|v_1\|^2 \\ &= \|v_0 - v_1\|^2 + \|v_1\|^2 \geq \|v_1\|^2 \end{aligned} \tag{2.126}$$

so that, together with the observation that $P_S v_0 = v_0$ if, and only if, $v_0 \in S$, gives

$$\|P_S\| = 1 \,. \tag{2.127}$$

From the definitions ${P_S}^2 = P_S$ and hence $ker[I - P_S] = S$. In addition, any vector $v_0 \in \mathscr{V}$ has a unique decomposition of the form $v_0 = v_1 + (v_0 - v_1) = P_S v_0 + (I - P_S)v_0$ where $v_1 = P_S v_0$ is orthogonal to S and hence $(I - P_S)v_0$. As a consequence, $\mathscr{V}$ has a direct sum decomposition of the form

$$\mathscr{V} = S \oplus S^{\perp} \quad \textit{where } S = ker[I - P_S] \textit{ and } S^{\perp} = ker[P_S]. \tag{2.128}$$

In particular, for any u and v in $\mathscr{V}$, it follows that $\langle P_S u, v \rangle = \langle P_S u, P_S v \rangle = \langle u, P_S v \rangle$ so that P_S is self adjoint and positive (but not strictly positive).

Finally, another form of convex set that plays a useful role in the analysis of linear control systems is that of a *Linear Variety*, namely a convex set S that is constructed by a translation of a subspace $\mathscr{W}$. The resultant set is denoted by $S = a + \mathscr{W}$ where $a \in \mathscr{V}$ defines the translation and

$$S = \{v : v = a + w \quad \textit{for some } w \in \mathscr{W}\ \}. \tag{2.129}$$

Note that the choice of a is not unique as it can be replaced by any vector $a + w_0$ with $w_0 \in \mathscr{W}$. As $v - v_0 = (v - a) - (v_0 - a)$ and $v - a \in \mathscr{V}$, the solution of the problem in this case can be expressed in the form

$$v_1 = a + P_{\mathscr{W}}(v_0 - a) \tag{2.130}$$

A useful example of a closed linear variety is the set

$$S = \{u : r = Gu + d\} = u_0 + \mathscr{W}, \quad \textit{with } \mathscr{W} = ker[G] \tag{2.131}$$

where $G : \mathscr{U} \to \mathscr{Y}$ is linear and bounded, $\mathscr{U}$ and $\mathscr{Y}$ are real Hilbert spaces, $r \in \mathscr{Y}$, $d \in \mathscr{Y}$ and $u_0 \in \mathscr{U}$ is any point (assumed to exist) satisfying $r = Gu_0 + d$. Another example of a closed linear variety is that of a *closed Hyperplane* defined by taking $\mathscr{Y} = \mathscr{R}$ and G as the map $G : u \mapsto \langle \alpha, u \rangle$ (for some $\alpha \in \mathscr{U}$) and setting

$$S = \{u : \langle \alpha, u \rangle = c\} \tag{2.132}$$

where both $\alpha \in \mathscr{U}$ and the real number c are specified. If $u_0 \in S$, then $S = \{u : \langle \alpha, u - u_0 \rangle = 0\}$ which identifies the set of vectors $u - u_0$ as that of all vectors orthogonal to α. A *Separating Hyperplane* separating two sets S_1 and S_2 in a real Hilbert space $\mathscr{V}$ is a hyperplane of the above type where $\langle \alpha, u \rangle \geq c$ for all $u \in S_1$ and $\langle \alpha, u \rangle \leq c$ for all $u \in S_2$ or vice versa. An example of a separating hyperplane is obtained from the result describing the minimum distance from $v_0 \in \mathscr{V}$ to a closed convex set S. More precisely, suppose that v_0 is not in S. From Theorem 2.122, the hyperplane $\langle v - v_1, v_1 - v_0 \rangle = 0$ is a hyperplane that separates the point set $\{v_0\}$ from S. Separating hyperplanes are not unique as the hyperplane $\langle v - \lambda v_1 - (1 - \lambda)v_0, v_1 - v_0 \rangle = 0$ is also a separating hyperplane when $\lambda \in [0, 1]$.

2.6 Optimal Control Problems in Hilbert Space

Quadratic optimization problems for linear systems play a central role in this text. In general, signals will be regarded as being vectors in suitable real Hilbert spaces. Operators will be used to represent systems behaviour by providing a linear relationship between system output signals and its input signals. The minimization of

objective functions created using quadratic functions of signal norms is the chosen mechanism for creating new control algorithms with known properties. The use of this level of abstraction will be seen to provide solutions that cover the solution of a wide range of problems of interest ranging from the case of continuous dynamics to sampled data systems to multi-rate systems and many other useful situations. The details are provided in the following chapters but the general form of the solution can be derived using the mathematical methods already described in this chapter.

A system will be described as a mapping between a set $\mathscr{U}$ of *input signals* u and a set $\mathscr{Y}$ of resultant *output signals* y. Both $\mathscr{U}$ and $\mathscr{Y}$ are taken to be normed vector spaces. A *Linear System* is characterized by a bounded, linear operator $G : \mathscr{U} \to \mathscr{Y}$ and the input to output mapping is defined by a relation of the form

$$y = Gu + d \tag{2.133}$$

where d represents the output behaviour when the input $u = 0$ (and hence, typically, behaviours due to initial condition and disturbances). A general form of *Linear, Quadratic, Optimal Control Problem* can be defined as the computation of the input that minimizes the *Objective Function* (often called a *Performance Index* or *Performance Criterion*)

$$J(u) = \|r - y\|^2_{\mathscr{Y}} + \|u_0 - u\|^2_{\mathscr{U}} \tag{2.134}$$

subject to the constraint that y and u are linked by $y = Gu + d$. The vectors $u_0 \in \mathscr{U}$ and $r \in \mathscr{Y}$ are assumed to be known and the problem is interpreted as an attempt to reduce the variation of the output from the specified signal r whilst not using input signals that deviate too much from u_0. The relative weighting of these two objectives is reflected in the choice of norms in $\mathscr{Y}$ and $\mathscr{U}$.

The solution to this problem when both $\mathscr{U}$ and $\mathscr{Y}$ are real Hilbert spaces is particularly valuable. Denote the adjoint operator of G by G^* and use the notation

$$\begin{aligned} e = r - y = r - Gu - d \quad & and \quad e_0 = r - y_0 \\ & where \quad y_0 = Gu_0 + d \end{aligned} \tag{2.135}$$

is the output response to the input u_0.

Theorem 2.18 (Solution of the Optimal Control Problem) *With the problem definition given above, the input-output pair* (y, u) *minimizing the objective function* (2.134) *subject to the constraint* (2.133) *is given by the implicit formulae*

$$u = u_0 + G^* e \tag{2.136}$$

As a consequence,

$$e = (I + GG^*)^{-1} e_0 \quad and\ hence \quad u = u_0 + G^*(I + GG^*)^{-1} e_0. \tag{2.137}$$

In particular, the minimum value of $J(u)$ can be computed to be

$$\min J(u) = \langle e_0, (I + GG^*)^{-1} e_0 \rangle_{\mathscr{Y}}. \tag{2.138}$$

Proof Two alternative proofs are provided in the next two subsections. □

The proofs depend on the material in the previous sections. In particular, they depend on the algebraic properties of inner products, the properties of the adjoint operator, the Projection Theorem and the identities

$$\begin{gathered} G^*(I + GG^*)^{-1} = (I + G^*G)^{-1} G^*, \quad G(I + G^*G)^{-1} G^* = GG^*(I + GG^*)^{-1}, \\ \textit{and } (I + GG^*)^{-1} + GG^*(I + GG^*)^{-1} = I. \end{gathered} \tag{2.139}$$

The inverses $(I + GG^*)^{-1}$ and $(I + GG^*)^{-1}$ exist and are bounded due to the positivity of GG^* and G^*G and the resultant lower bounds $I + GG^* \geq I > 0$ and $I + G^*G \geq I > 0$.

2.6.1 Proof by Completing the Square

Note that $G(u - u_0) = -(e - e_0)$ and consider the following inner product

$$\begin{aligned} \gamma &= \langle u - u_0 - G^*(I + GG^*)^{-1} e_0, (I + G^*G)(u - u_0 - G^*(I + GG^*)^{-1} e_0) \rangle_{\mathscr{U}} \\ &\geq \| u - u_0 - G^*(I + GG^*)^{-1} e_0 \|^2_{\mathscr{U}} \ \geq \ 0, \end{aligned} \tag{2.140}$$

noting that it is equal to zero if, and only if $u - u_0 - G^*(I + GG^*)^{-1} e_0 = 0$. If this condition is valid then, operating on the equation with G gives $e = e_0 - GG^*(I + GG^*)^{-1} e_0 = (I + GG^*)^{-1} e_0$ which would prove (2.137) and hence (2.136). It remains to prove therefore that $\gamma = 0$.

The inner product can be written as

$$\begin{aligned} \gamma &= \langle u - u_0 - (I + G^*G)^{-1} G^* e_0, (I + G^*G)(u - u_0 - (I + G^*G)^{-1} G^* e_0) \rangle_{\mathscr{U}} \\ &= \langle u - u_0, (I + G^*G)(u - u_0) \rangle_{\mathscr{U}} - 2\langle u - u_0, G^* e_0 \rangle_{\mathscr{U}} + \langle G^* e_0, (I + G^*G)^{-1} G^* e_0 \rangle_{\mathscr{U}} \\ &= \| u - u_0 \|^2_{\mathscr{U}} + \| e - e_0 \|^2_{\mathscr{Y}} + 2\langle e - e_0, e_0 \rangle_{\mathscr{Y}} + \langle G^* e_0, (I + G^*G)^{-1} G^* e_0 \rangle_{\mathscr{U}} \\ &= \| u - u_0 \|^2_{\mathscr{U}} + \| e \|^2_{\mathscr{Y}} + \| e_0 \|^2_{\mathscr{Y}} - 2\langle e, e_0 \rangle_{\mathscr{Y}} + 2\langle e - e_0, e_0 \rangle_{\mathscr{Y}} \\ &\qquad + \langle G^* e_0, (I + G^*G)^{-1} G^* e_0 \rangle_{\mathscr{U}} \\ &= J(u) + \langle G^* e_0, (I + G^*G)^{-1} G^* e_0 \rangle_{\mathscr{U}} - \| e_0 \|^2_{\mathscr{Y}} \\ &= J(u) + \langle e_0, G(I + G^*G)^{-1} G^* e_0 \rangle_{\mathscr{Y}} - \| e_0 \|^2_{\mathscr{Y}} \\ &= J(u) + \langle e_0, \left(GG^*(I + G^*G)^{-1} - I \right) e_0 \rangle_{\mathscr{Y}} \\ &= J(u) - \langle e_0, (I + GG^*)^{-1} e_0 \rangle_{\mathscr{Y}} \end{aligned} \tag{2.141}$$

the second term being independent of (y, u). It follows that $J(u)$ is minimized if, and only if, $\gamma = 0$. This first proof of Theorem 2.18 is now complete. □

2.6.2 Proof Using the Projection Theorem

An alternative derivation of the solution uses the Projection Theorem in the product space $\mathscr{Y} \times \mathscr{U}$ of input/output pairs (y, u) regarded as a real Hilbert space with inner product (and associated norm) defined using

$$\begin{aligned} \langle (z, v), (y, u) \rangle_{\mathscr{Y} \times \mathscr{U}} &= \langle z, y \rangle_{\mathscr{Y}} + \langle v, u \rangle_{\mathscr{U}} \\ \|(y, u)\|_{\mathscr{Y} \times \mathscr{U}} &= \sqrt{\|y\|_{\mathscr{Y}}^2 + \|u\|_{\mathscr{U}}^2}. \end{aligned} \tag{2.142}$$

With this notation, $J(u) = \|(r, u_0) - (y, u)\|_{\mathscr{Y} \times \mathscr{U}}^2$ and the optimal control problem is that of finding the pair (y_1, u_1) that solves the minimum norm problem

$$\begin{aligned} &(y_1, u_1) = \arg\min\{\|(r, u_0) - (y, u)\|^2 : (y, u) \in S\} \\ &\textit{where} \quad S = \{(y, u) : y = Gu + d\} \end{aligned} \tag{2.143}$$

is a linear variety in $\mathscr{Y} \times \mathscr{U}$. It is closed as any sequence $\{(y_j, u_j)\}_{j \geq 0}$ in S converging (in the norm topology) to a point (y, u) has the property that $\{y_j\}_{j \geq 0}$ converges to y in $\mathscr{Y}$ and $\{u_j\}_{j \geq 0}$ converges to u in $\mathscr{U}$. Also, as $y_j = Gu_j + d,\ j \geq 0$,

$$\begin{aligned} 0 \leq \|y - Gu - d\|_{\mathscr{Y}} &= \|(y - y_j) - G(u - u_j) + (y_j - Gu_j - d)\|_{\mathscr{Y}} \\ &\leq \|(y - y_j) - G(u - u_j)\|_{\mathscr{Y}} + \|y_j - Gu_j - d\|_{\mathscr{Y}} \\ &= \|(y - y_j) - G(u - u_j)\|_{\mathscr{Y}} \\ &\leq \|(y - y_j)\|_{\mathscr{Y}} + \|G\|\|(u - u_j)\|_{\mathscr{U}} \end{aligned} \tag{2.144}$$

which tends to zero as $j \to \infty$. Hence $y = Gu + d$ which proves that $(y, u) \in S$.

Applying the projection theorem, the solution (y_1, u_1) of the optimal control problem satisfies

$$\langle (z, v) - (y_1, u_1), (y_1, u_1) - (r, u_0) \rangle_{\mathscr{Y} \times \mathscr{U}} = 0 \quad \textit{for all } (z, v) \in S\,. \tag{2.145}$$

This equation is just

$$\langle z - y_1, y_1 - r \rangle_{\mathscr{Y}} + \langle v - u_1, u_1 - u_0 \rangle_{\mathscr{U}} = 0 \tag{2.146}$$

Using the equations $y_1 = Gu_1 + d$, $z = Gv + d$ and $e = r - y_1$ then gives

$$\langle v - u_1, -G^* e \rangle_{\mathscr{Y}} + \langle v - u_1, u_1 - u_0 \rangle_{\mathscr{U}} = \langle v - u_1, u_1 - u_0 - G^* e \rangle_{\mathscr{U}} = 0 \tag{2.147}$$

for all $v \in \mathscr{U}$. Choosing $v = 2u_1 - u_0 - G^* e$, it follows that $\|u_1 - u_0 - G^* e\|^2 = 0$ which proves the result using the same algebraic manipulations as those used in the previous subsection and the computation of the minimum value of the objective function as follows,

$$\|r - y_1\|^2_{\mathscr{Y}} + \|u_0 - u_1\|^2_{\mathscr{U}} = \langle e, (I + GG^*)e\rangle_{\mathscr{Y}} = \langle e_0, (I + GG^*)^{-1}e_0\rangle_{\mathscr{Y}}. \tag{2.148}$$

This completes the second proof of Theorem 2.18. □

2.6.3 Discussion

The solution of the optimal control problem described above provides a formal approach to the solution of a wide class of problems following the process summarized as the steps,

1. Identify the vector space $\mathscr{U}$ from which the inputs signals are to be chosen.
2. Choose an inner product and norm for $\mathscr{U}$ that ensures that it is a real Hilbert space and also reflects the physical importance of signals.
3. Identify the vector space $\mathscr{Y}$ containing the outputs signals.
4. Choose an inner product and norm for $\mathscr{Y}$ that ensures that it is a real Hilbert space and also reflects the physical importance of signals.
5. Characterize the system as a bounded linear mapping G from $\mathscr{U}$ into $\mathscr{Y}$ and identify the form of its adjoint operator G^*.
6. Write the defining relationship for the optimal solution in the implicit form $u = u_0 + G^*e$ with $e = r - y$ and find a causal representation of this controller that can be implemented in real life.

This process could apply to any problem satisfying the assumptions but the devil is in the detail. The main problem is that expressed in the last step, namely the conversion of the implicit relationship between u and e into a useable computation. In later chapters (see for example, Sects. 3.10 and 4.7), this idea will be linked, in the special cases of linear state space models, to *Two Point Boundary Value Problems* and the use of *Riccati equations*. More generally, the computations suffer from additional complexities and high dimensionality. Even the simplest cases present challenges. For example, let $\mathscr{U} = \mathscr{R}^q$ with inner product $\langle \hat{v}, v\rangle_{\mathscr{U}} = \hat{v}^T Rv$ (where $R = R^T > 0$) and $\mathscr{Y} = \mathscr{R}^p$ with inner product $\langle \hat{w}, w\rangle_{\mathscr{Y}} = \hat{w}^T Qw$ (where $Q = Q^T > 0$). The operator G is a $p \times q$ real matrix with adjoint (Eq. (2.104)) defined by the $q \times p$ matrix $G^* = R^{-1}G^T Q$. Rather than using the implicit relationship, the direct computation of u can be undertaken using

$$u = u_0 + G^*(I + GG^*)^{-1}e_0 = u_0 + R^{-1}G^T Q(I_p + GR^{-1}G^T Q)^{-1}e_0 \tag{2.149}$$

This is a feasible approach to finding the solution and may work well in many cases but, if the dimensions p and q are large, the calculation of the inverse matrix could be challenging particularly if $I + GR^{-1}G^T Q$ is ill-conditioned. This example does have relevance to the topics in the text associated with the description of discrete time systems in supervector form (see Sect. 4.7). Simplifications are possible in this case

as the elements in G have a structural pattern that makes the inversion implicit in $G^*(I + GG^*)^{-1}$ equivalent to the solution of a Two Point Boundary Value Problem which is solved using Riccati equation and associated simulation methods.

2.7 Further Discussion and Bibliography

The chapter has reviewed material that plays a role in the development of the techniques used in this text. Matrices form the computational core of the algorithms and, although many readers will be familiar with basic matrix algebra, an understanding of the structures and the analysis tools available for matrix methodologies is helpful as is an understanding of the way that matrices are useful in the interpretation of high (but finite) dimensional problems using simple geometrical insights generated from the familiar three-dimensional world. There are many texts that cover the material required ranging from undergraduate engineering texts such as [60, 105, 106] to essentially mathematical texts that approach the topics using both algebraic concepts and the ideas of finite dimensional vector spaces [45, 46, 53] within which matrices are representations of operators using a specified basis set. Many teaching texts on control theory and control engineering also have a summary of the necessary material [4, 39, 43, 63, 71, 81]. The material is essentially the same but differing perspectives and different levels of abstraction are used. It is useful to note that there are links between matrices and transfer function descriptions using Toeplitz matrices, details of which can be found in [51]. For the purposes of this text, an understanding of, and fluency with, the algebraic structures and analysis tools will help the reader to "see through the symbols" and concentrate more usefully on the form and meaning of the system properties used and the nature of the algorithms described. An understanding of the algebraic and computational aspects of matrix theory will form the basis for any computational software required for the exploitation of the material and also in ensuring that data formats fit the necessary matrix structures.

Relevant techniques from functional analysis are also summarized in the chapter. This material will be less familiar to many readers but, in its simplest form, it can be regarded as a generalization of matrix theory to cover a wider range of problems. In particular, matrices are replaced by operators between, possibly infinite dimensional, signal spaces and the geometry of the three dimensional world is generalized to higher, possibly infinite, dimensions. The underpinning ideas of vector spaces endowed with norms to measure signal magnitude and the notion of bounded linear operators between such spaces mirror the familiar notion of a system as a device that maps input signals into output signals and the measurement of signal magnitude using measures such as least square values or maximum magnitudes. Although much of this work can be viewed at the algebraic level as being very similar to matrix (or even transfer function) methodologies, the technical details associated with the ideas do contain many subtle issues that need to be considered at the mathematical level. These take many forms but, perhaps the most important are those of existence of solutions to defined problems, the convergence of infinite sequences, the

introduction of the notion of adjoint operators and their properties, convexity and the idea of projection onto convex sets. There are many texts that provide the mathematical background for these topics including general texts on analysis [101] and functional analysis such as [12, 31, 52, 107] and more specialist texts on Hilbert space theory such as [54], operator theory [32, 33] and optimization algorithms using functional analytic methods [69]. A reference to Open Mapping and Closed Graph Theorems is found in [73] and an extensive analysis of projection methodologies is found in [36]. In the author's experience, the choice of text to suit the needs of a particular reader depends upon that reader and his or her background and preferred way of thinking.

Finally, the content of the text is mathematical in its chosen language and much of the supporting mathematics of Laplace and Z-transforms is used extensively in control engineering texts. Some of the more advanced tools needed can be found in texts on classical and multivariable control (see Sect. 1.5), mathematical systems theory [59, 104] and specialist texts [112] and papers [75] on geometric systems theory and decoupling theory [37, 47].

Chapter 3
State Space Models

In applications of any form of control, there is a system to be controlled, a means of influencing its behaviour (system *input signals*), measurements of performance outcomes (system *output signals*) plus, in some cases, auxiliary measurements made for purposes of monitoring, diagnostics or enhancement of control signals. The *system* to be controlled is sometimes called the *plant* to be controlled. The construction of control systems for a given plant inevitably required the addition, via feedback or some other form of interconnection, of designed systems called *control systems* or, more simply, *controllers*. Controllers process current and available past data as a means of constructing current input values. The computational process is designed to achieve the desired performance and is often called the *control algorithm*. The general aim of control systems analysis and design is to ensure that the composite, interconnected system consisting of the combined plant plus controller has a behaviour that is acceptable to the application and achieves defined performance objectives to a satisfactory standard.

The use of verbal descriptions to define problems in controller design is necessary to define context and aid in communication between design engineers but it fails to produce the means of finding design solutions in the form of control algorithms. To find systematic approaches to successful design, the history of control engineering has indicated the central role of a *mathematical model* of plant behaviour as a means of investigating a range of behaviours using off-line simulation or other tools. Such models provide understanding but also permit the investigation of behaviours that financial or safety considerations might preclude in experiments. Such models can have many forms including algebraic equations, differential equations, difference equations, partial differential equations, differential delay equations, frequency domain relationships and mixtures of such descriptions. To this complexity must be added the properties of *linearity* and *nonlinearity*. The level of detail possible in the analysis of linear systems tends to be much greater than that possible for nonlinear systems, probably because the mathematics of linear systems is more fully understood and has a structure that gives engineers both computational and conceptual

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_3

links between plant behaviour and the structure and parameters appearing in the plant model e.g. the links between eigenvalues/poles and stability, frequency domain analysis and parameter choice using root-locus methodologies and transfer function matrix analysis.

The systems considered in this text will be almost exclusively *linear dynamical systems* with behaviours described by differential or difference equations but the presentation does not preclude the application to more complex systems containing, for example, transport delays. In much of the text the use of operator notation in the construction of controllers can be used to bring these complexities together and, as a by-product, indicate the essential unity of the work despite substantial differences in the detail. In all cases the construction of the actual relationships needed to build the hardware and software used in the physical control system will be based on the (not necessarily easy) challenge of converting operator relationships into feasible, and preferably familiar, model and measurement-based computations.

The aim of this text is to describe the elements of a wide ranging and applicable theory and algorithms for iterative control processes. The development can be based on an underlying representation of plant dynamics by *linear state space models* and, from time to time, the related *transfer function matrix* representations. These assumptions make possible a level of detailed analysis that links the work closely to engineering practice and reveals much of the details needed for computational studies and the properties that add confidence that successful applications are, indeed, possible. The underlying mathematics is that of matrix theory and and an operational representation that unifies the ideas and links the analysis to the geometry of appropriate signal spaces.

In general terms, state space models describe system dynamics by mathematical relationships between inputs signals u, output signals y and a *system state* x representing the internal (possibly unmeasurable) variables needed to fully describe the dynamical behaviour. The two most familiar forms of relationship are those describing continuous time state space models and those describing discrete time (sampled data) state space models.

Consider a dynamical system with m measured outputs $y_1(t), y_2(t), \ldots, y_m(t)$ influenced by ℓ input signals $u_1(t), u_2(t), \ldots, u_\ell(t)$. The symbol t is used to represent a measure of time. Assume also that a full description of the dynamics can be obtained using n *state variables* $x_1(t), x_2(t), \ldots, x_n(t)$. If one or both of m and ℓ are greater than one, then the system is said to be *multivariable, multi-input multi-output, multi-loop or multi-channel* or MIMO for short. If $m = \ell = 1$, the system is said to be *single-input single output* (or SISO for short).

One concept of note is that of *causality*. That is, for physical systems, for any time t, the state of a system at a time t is a consequence of its past history and current influences (defined as relevant data on the interval $t' \leq t$) and is independent of what may happen or be done to it at future times $t' > t$.

3.1 Models of Continuous State Space Systems

A general *nonlinear* continuous time, state space model expressed in terms of these signals can be written in the following form

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t), \quad x(0) = x_0 \\ y(t) &= h(x(t), u(t), t), \quad 0 \le t \le T \end{aligned} \tag{3.1}$$

where

1. T is the (finite or infinite) time interval of interest an
2. the column matrices

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix}, \quad u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_\ell(t) \end{bmatrix} \quad \textit{and} \quad x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \tag{3.2}$$

   are the $m \times 1$ *output vector*, $\ell \times 1$ *input vector* and $n \times 1$ *state vector* respectively.
3. If H is a $p \times q$ matrix H with elements that depend on time and letting $H_{ij}(t)$ denote the element of H on row i and column j, then the symbol "$\dot{H}$" denotes simply the $p \times q$ matrix with (i, j) element equal to $\frac{dH_{ij}(t)}{dt}$ i.e. the matrix of derivatives of the elements of H. The jth element of $\dot{x}(t)$ is hence $\frac{dx_j(t)}{dt}$.
4. The $n \times 1$ vector x_0 is the column matrix of *initial conditions* $x_1(0), x_2(0), \ldots, x_n(0)$ for the state variables $x_1(t), x_2(t), \ldots, x_n(t)$.
5. The vector valued function $f(x, u, t)$ (respectively $h(x, u, t)$) describes the functional dependency of the state derivatives (respectively outputs) at time t on the values of the state vector $x(t)$ at time t, the input vector $u(t)$ at time t and the value of t itself.
6. If both $f(x, u, t)$ and $h(x, u, t)$ have no explicit dependence on t, the system is said to be *time invariant*. Otherwise it is said to be *time varying*.

If both f and g are linear in x and u, the state space model is said to be linear. The linear model used to describe this situation takes the general form

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t), \quad x(0) = x_0 \\ y(t) &= C(t)x(t) + D(t)u(t), \quad 0 \le t \le T \end{aligned} \tag{3.3}$$

where the matrices A, B, C, D are of dimension $n \times n, n \times \ell, m \times n$ and $m \times \ell$ respectively. The following terminology and notation will be adopted here:

1. If $D(t)$ is identically zero (written $D(t) \equiv 0$), the system is said to be *strictly proper*. Otherwise it is said to be *proper*.
2. If any of the matrices A, B, C, D vary with time, then the system is said to be *time varying*. Otherwise it is said to be *time-invariant*.

3. As the linear state space model can be defined by the four matrices A, B, C, D, the system will be referred to as the system $S(A, B, C, D)$. If it is strictly proper, then the notation $S(A, B, C, 0)$ is simplified to $S(A, B, C)$.

3.1.1 Solution of the State Equations

For a given initial state x_0 and input signal $u(t)$, the solutions ($x(t)$ and $y(t)$) of the state equations can be obtained using computational simulation software. For the purposes of theory and algorithm development however, a theoretical description of the solution is extremely useful. This is only possible for linear state space models and takes the form

$$\begin{aligned} x(t) &= \Phi(t,0)x_0 + \int_0^t \Phi(t,t')B(t')u(t')dt' \ , \quad 0 \le t \le T, \\ \textit{with} \ \ y(t) &= C(t)x(t) + D(t)u(t), \quad 0 \le t \le T. \end{aligned} \tag{3.4}$$

Here

1. If $H(t)$ is a $p \times q$ time varying matrix, then $\int_a^b H(t)dt$ denotes the $p \times q$ matrix with (i,j)th element $\int_a^b H_{ij}(t)dt$
2. The *$n \times n$ state transition matrix* $\Phi(t,t')$ is the solution of the family of differential equations

$$\frac{\partial \Phi(t,t')}{\partial t} = A(t)\Phi(t,t'), \quad \textit{with} \ \ \Phi(t',t') = I_n, \ \ 0 \le t' \le t \le T \tag{3.5}$$

   where I_n denotes the $n \times n$ unit matrix. A simple formula for $\Phi(t,t')$ is obtained by solving the equation for $t' = 0$ and then setting

$$\Phi(t,t') = \Phi(t,0)(\Phi(t',0))^{-1} \tag{3.6}$$

3. The state transition matrix hence satisfies the relation

$$\Phi(t',t)\Phi(t,t') = I_n \quad \textit{so that} \quad \Phi(t',t) = (\Phi(t,t'))^{-1} \tag{3.7}$$

   In particular,

$$\frac{\partial \Phi(t',t)}{\partial t}\Phi(t,t') + \Phi(t',t)\frac{\partial \Phi(t,t')}{\partial t} = 0 \tag{3.8}$$

so that

$$\frac{\partial \Phi(t', t)}{\partial t} = -\Phi(t', t)A(t) \quad \textit{and hence} \quad \frac{\partial \Phi^T(t', t)}{\partial t} = -A^T(t)\Phi^T(t', t). \tag{3.9}$$

4. If $A(t)$ is a constant matrix A, then

$$\Phi(t, t') = e^{A(t-t')} \tag{3.10}$$

5. If $S(A, B, C, D)$ is linear and time-invariant, then it is asymptotically stable if, and only if, all eigenvalues of A have strictly negative real parts. That is, all eigenvalues take values in the open left half of the complex plane.

3.1.2 The Convolution Operator and the Impulse Response

Using the above solution in the form, for $0 \leq t \leq T$,

$$y(t) = C(t)\Phi(t, 0)x_0 + \int_0^t \underbrace{C(t)\Phi(t, t')B(t')}\, u(t')dt' + D(t)u(t),$$

or

$$y(t) = C(t)\Phi(t, 0)x_0 + \int_0^t H(t, t')u(t')dt' + D(t)u(t), \tag{3.11}$$

with the obvious identification of the kernel $H(t, t')$. In the linear time-invariant, strictly proper case

$$H(t, t') \;=\; Ce^{A(t-t')}B \tag{3.12}$$

which, for SISO strictly proper systems reduces to the familiar impulse response function. For the case of linear, time-invariant, strictly proper MIMO systems, the $m \times \ell$ matrix $H(t) = Ce^{At}B$ will be termed the *impulse response matrix* of $S(A, B, C)$.

3.1.3 The System as an Operator Between Function Spaces

It will be useful to consider the solution $y(t)$ on the interval $0 \leq t \leq T$ as the response to the signal $u(t)$ on the interval $0 \leq t \leq T$ when the initial condition $x_0 = 0$ plus the response from the initial condition x_0 with $u(t) \equiv 0$ on $0 \leq t \leq T$. In this form, the solution can be written as an operational relationship

$$y = Gu + d = z + d \tag{3.13}$$

between signals $u \in \mathscr{U}$ and $y \in \mathscr{Y}$ in suitable function spaces $\mathscr{U}$ and $\mathscr{Y}$. Here, $d \in \mathscr{Y}$ represents the function $d(t) = C(t)\Phi(t,0)x_0,\ 0 \le t \le T$, and $G : \mathscr{U} \mapsto \mathscr{Y}$ is a linear operator mapping $u(t),\ 0 \le t \le T$, into $z(t)$ where

$$z(t) \ = \ \int_0^t H(t,t')u(t')dt' \ + \ D(t)u(t), \quad 0 \le t \le T. \tag{3.14}$$

The choice of function space is non-unique. Consider the case of T finite and the choices of Cartesian product Hilbert spaces

$$\begin{aligned} \mathscr{U} &= L_2^\ell[0,T] = \underbrace{L_2[0,T] \times L_2[0,T] \times \cdots \times L_2[0,T]}_{\ell - times} \\ \mathscr{Y} &= L_2^m[0,T] = \underbrace{L_2[0,T] \times L_2[0,T] \times \cdots \times L_2[0,T]}_{m - times} \end{aligned} \tag{3.15}$$

with inner products defined by symmetric positive definite matrices R and Q as follows

$$\langle u, v \rangle_{\mathscr{U}} = \int_0^T u^T(t)Rv(t)dt \quad \langle y, w \rangle_{\mathscr{Y}} = \int_0^T y^T(t)Qw(t)dt. \tag{3.16}$$

Note: For notational simplicity, the inner products $\langle u, v \rangle_{\mathscr{U}}$ and $\langle y, w \rangle_{\mathscr{Y}}$ may be denoted by $\langle u, v \rangle_R$ and $\langle y, w \rangle_Q$ to identify the associated matrices.

This abstract description has an algebraic similarity to the use of transfer function matrices for analysis and design but does not require the time to frequency domain mapping used in these approaches. The benefits of regarding systems as operators between function spaces will further emerge in the remainder of this text.

3.2 Laplace Transforms

Laplace Transform methodologies have been used widely in control systems analysis and design. The basic definition of the Laplace Transform of a scalar function $f(t)$ is most easily presented in the case of an infinite time interval $T = \infty$. More precisely, the Laplace Transform of a function f (defined on an infinite interval $0 \le t < \infty$) is a function of the complex variable s defined by

$$\mathscr{L}[f(t)] \ = \ \int_0^\infty e^{-st} f(t)dt, \tag{3.17}$$

for all complex number s such that the integral exists. It is noted that a sufficient condition for the existence of such values is that f is *exponentially bounded*. That is, there exists real numbers $M \ge 0$ and λ such that

$$|f(t)| \leq Me^{\lambda t} \quad \textit{for all } \ t \geq 0. \tag{3.18}$$

For such a function, the integral has a finite value for all s with real part strictly greater than λ. Although only valid for values of s such that the integral converges, the functions can be extended by analytic continuation to all points of the complex plane other than singularities. For this reason the issues of integral convergence play only an occasional role in practical use of the ideas.

Useful properties of the Laplace Transform used in the text and are summarized below for any function $f(t)$ defined on $0 \leq t < \infty$ with $f(0)$ written as f_0:

$$\begin{aligned} \mathscr{L}[\frac{df(t)}{dt}] &= s\mathscr{L}[f(t)] - f_0, & \mathscr{L}[t^k f(t)] &= (-1)^k \frac{d^k}{ds^k}\left(\mathscr{L}[f(t)]\right), \\ \mathscr{L}[e^{\alpha t}] &= \frac{1}{s-\alpha} \quad \textit{and} & \mathscr{L}[\frac{t^k}{k!}e^{\alpha t}] &= \frac{1}{(s-\alpha)^{k+1}}. \end{aligned} \tag{3.19}$$

As a simplification to the notation, the Laplace Transform $\mathscr{L}[f(t)]$ will often be denoted $f(s)$. This clearly could lead to some confusion. However, despite this possibility, the notation will be used as the differences between $f(t)$ and $f(s)$ will be clear to the reader from the context. With this convention, the *Inverse Laplace Transform* is simply the inverse process and is denoted by $\mathscr{L}^{-1}$, for example,

$$f(t) = \mathscr{L}^{-1}[f(s)] \tag{3.20}$$

The extension of the ideas to matrix valued functions of time is easily achieved. For a $p \times q$ time dependent matrix $H(t)$ the Laplace Transform $\mathscr{L}[H(t)]$ is denoted by $H(s)$ which has (i, j) element equal to the Laplace Transform of $H_{ij}(t)$.

3.3 Transfer Function Matrices, Poles, Zeros and Relative Degree

A linear, time-invariant, state space model has a frequency domain description obtained simply by taking Laplace Transforms of the state equations to obtain

$$sx(s) - x_0 = Ax(s) + Bu(s), \quad y(s) = Cx(s) + Du(s) \tag{3.21}$$

from which simple matrix algebra to eliminate $x(s)$ gives

$$\begin{aligned} x(s) &= (sI - A)^{-1}\left(Bu(s) + x_0\right) \qquad \textit{and hence} \\ y(s) &= G(s)u(s) + C(sI_n - A)^{-1}x_0 \end{aligned} \tag{3.22}$$

where the system $m \times \ell$ *Transfer Function Matrix* $G(s)$ is defined by

$$G(s) = C(sI_n - A)^{-1}B + D. \tag{3.23}$$

It has elements that are rational polynomials in the complex variable s. If $m = \ell = 1$, $G(s)$ is the familiar transfer function used to great effect in the analysis and design of SISO feedback controllers.

Poles and zeros play an important role in classifying systems dynamics in design. These two concepts are defined as follows

1. The denominator polynomial of $G(s)$ is the characteristic polynomial of A. As a consequence, the *poles* of $G(s)$ and their multiplicities are precisely the set of eigenvalues of the matrix A and their multiplicities.
2. The *zeros* of $G(s)$ are defined by rank conditions on a partitioned matrix dependent on (A, B, C, D). More precisely, the complex number s is a zero of $S(A, B, C, D)$ if, and only if,

$$rank \begin{bmatrix} sI_n - A & -B \\ C & D \end{bmatrix} < n + \min\{m, \ell\} \tag{3.24}$$

These zeros are the multivariable analogue of familiar transfer function zeros. If all zeros lie in the open left half complex plane then the system is said to be *minimum phase*. Otherwise it is said to be *non-minimum phase*. Non-minimum phase systems are known to lead to control problems and performance limitations in the design of feedback control systems. Similar problems will be seen to be the case for inverse-model-based, gradient and Norm Optimal Iterative Control methodologies described in this text.
Note that if $m = \ell$, the zero definition reduces to the computation of the roots of a determinant, namely, the solutions of

$$\begin{vmatrix} sI_n - A & -B \\ C & D \end{vmatrix} = 0 \tag{3.25}$$

Note also that, when $m \neq \ell$, the generic situation is that the system has no zeros. If $m = \ell$ then the use of *Schur's Formula* yields

$$\begin{vmatrix} sI_n - A & -B \\ C & D \end{vmatrix} \equiv |sI_n - A| \ \det[G(s)]. \tag{3.26}$$

When $m = \ell = 1$, the zeros are precisely those of the system transfer function.

Other useful properties and parameters associated with $G(s)$ can be constructed as follows

1. by using the power series expansion of $(sI_n - A)^{-1}$ to obtain the expression

$$G(s) = D + s^{-1}CB + s^{-2}CAB + \cdots = D + \sum_{k=1}^{\infty} s^{-k} CA^{k-1}B \tag{3.27}$$

which is convergent for $|s| > r(A)$. The matrices CB, CAB, CA^2B, ... are called the system *Markov Parameter Matrices* or, for simplicity, the system Markov parameters.

2. The *relative degree* k^* *of* $G(s)$ is defined as the unique integer such that $G_\infty = \lim_{s\to\infty} s^k G(s)$ exists and is non-zero. From the series expansion, it is clear that either $k^* = 0$ when $D \neq 0$ or, when $D = 0$, k^* is the smallest integer k for which $CA^{k-1}B$ is non-zero. Clearly the system is strictly proper if, and only if, $k^* > 0$. In the SISO case of $m = \ell = 1$, the relative degree is precisely equal to the number of poles of $G(s)$ minus the number of zeros of $G(s)$.
3. Clearly $G_\infty \neq 0$ from the definition but it does not necessarily have full rank as a matrix. This complication requires the development of useful practical refinements of the idea of relative degree. Such refinements will be discussed later in the text when the need for them arises.

3.4 The System Frequency Response

The frequency response function of the linear, time-invariant, system $S(A, B, C, D)$ is defined to be the (matrix-valued) function $G(i\omega)$ where i is the familiar "square root of -1" and ω is a real frequency with units radians/unit time. This function is a foundation of classical feedback control systems design.

It can also be used to describe the response of an asymptotically stable system from zero initial conditions $x_0 = 0$ to sinusoidal input signals. More generally, if the data (A, B, C, D) contains only real numbers and α is a constant, possibly complex $\ell \times 1$ vector, define inputs $u_c(t)$ and $u_s(t)$ to be the real and imaginary parts of the complex input $u(t) = \alpha e^{st}$ where s is any complex number. The resultant output responses from zero initial conditions are, respectively, equal to the real and imaginary parts of the response $y(t)$ from zero initial conditions to the complex input $u(t)$. It is left as an exercise for the reader to verify that

$$y(t) = G(s)u(t) - C(sI_n - A)^{-1}e^{At}B\alpha \tag{3.28}$$

which provides a direct link between the transfer function matrix $G(s)$ and well-defined time domain behaviours. In particular, if $s = i\omega$ (with ω a real frequency), the stability assumption indicates that, for all large enough values of time t, the steady state oscillation $y(t) \approx G(s)u(t) = G(i\omega)\alpha e^{i\omega t}$ is an accurate description of steady state oscillatory output behaviour. This is because e^{At} becomes infinitesimally small as t becomes large.

3.5 Discrete Time, Sampled Data State Space Models

In many applications of control systems engineering, the basis of measurement and control is sampling of continuous signals with simultaneous changing of control input values at sample times $0, h, 2h, 3h, \ldots$. This situation is commonly described as uniform synchronous sampling with *sample interval* h and *sampling rate* h^{-1}. In such circumstances, the sampled output signal is a *time series* $y(0), y(h), y(2h), \ldots$ and the input signal takes the form of a *piece-wise constant* input

$$u(t) = u(kh), \quad \textit{for} \quad kh \leq t < (k+1)h, \quad \textit{and} \quad 0 \leq k \leq N-1 \tag{3.29}$$

where $T = Nh$ is the time interval of interest to the control study. The corresponding input time series is hence $u(0), u(h), u(2h), \ldots$. For notational simplicity, the dependence on h will be suppressed and the input and output time series will be denoted $u(0), u(1), u(2), u(3), \ldots$ and $y(0), y(1), y(2), y(3), \ldots$. That is, they will be indexed by the sample number only.

3.5.1 State Space Models as Difference Equations

A general *nonlinear* discrete time, state space model can be written in the following form indexed by the integer sample time index $t = 0, 1, 2, 3, \ldots$

$$\begin{aligned} x(t+1) &= f_d(x(t), u(t), t), \quad 0 \leq t \leq N-1, \quad x(0) = x_0 \\ y(t) &= h_d(x(t), u(t), t), \quad 0 \leq t \leq N \end{aligned} \tag{3.30}$$

where

1. N represents the (finite or infinite) time interval of interest,
2. the column matrices $y(t), u(t), x(t)$ define output vectors, input vectors and state vectors as for the continuous time case.
3. The $n \times 1$ vector x_0 is the column matrix of *initial conditions* $x_1(0), x_2(0), \ldots, x_n(0)$ for the state variables $x_1(t), x_2(t), \ldots, x_n(t)$.
4. If both $f_d(x, u, t)$ and $h_d(x, u, t)$ have no explicit dependence on t, the system is said to be *time invariant*. Otherwise it is said to be *time varying*.

If both f_d and g_d are linear in x and u, the discrete state space model is said to be linear. The linear model used to describe this situation takes the general form

$$\begin{aligned} x(t+1) &= \Phi_d(t)x(t) + \Delta_d(t)u(t), \quad 0 \leq t \leq N-1, \quad x(0) = x_0 \\ y(t) &= C(t)x(t) + D(t)u(t), \quad 0 \leq t \leq N \end{aligned} \tag{3.31}$$

where the, possibly sample dependent, matrices Φ_d, Δ_d, C, D are of dimension $n \times n$, $n \times \ell$, $m \times n$ and $m \times \ell$ respectively. The following terminology is adopted from the continuous time case:

1. If $D(t)$ is identically zero (written $D(t) \equiv 0$), the system is said to be *strictly proper*. Otherwise it is said to be *proper*.
2. If any of the matrices Φ_d, Δ_d, C, D vary with sample number, then the system is said to be time varying. Otherwise it is said to be time-invariant.
3. As the linear state space model can be defined by the four matrices Φ_d, Δ_d, C, D, the system will be referred to as the discrete system $S(\Phi_d, \Delta_d, C, D)$. If it is strictly proper, then the notation is simplified to $S(\Phi_d, \Delta_d, C)$.

Finally, the discrete system is always linear and time invariant if it is is constructed from an underlying linear, time-invariant system $S(A, B, C, D)$. The constant matrices Φ_d and Δ_d can be computed, in this case, from the equations

$$\Phi_d = e^{Ah} \quad and \quad \Delta_d = \int_0^h e^{At} B dt'. \tag{3.32}$$

Important Note: in the remainder of the text the notation used for discrete systems will vary to suite the circumstance and to provide simplicity of notation in the presentation. For example, the notation $S(A, B, C, D)$ will be used, on occasion, to denote a continuous system but also a discrete system with Φ_d equal to some matrix A and Δ_d equal to a matrix B. This should not cause any confusion in cases where any underlying continuous systems dynamics plays no role in the discussions.

3.5.2 Solution of Linear, Discrete Time State Equations

For a given initial state x_0 and input time series $u(t)$, the solution can be obtained using computational simulation software. For the purposes of theory and algorithm development however, a theoretical description of the solution is extremely useful. This is only possible for linear state space models and takes the form

$$\begin{aligned}
x(t) &= \Phi(t-1,0)\Phi_d(0)x_0 + \sum_{t'=0}^{t-1} \Phi(t-1,t')\Delta_d(t')u(t'), \quad t \geq 1, \\
y(t) &= C(t)x(t) + D(t)u(t), \quad t \geq 0,
\end{aligned}$$

so that

$$\begin{aligned}
y(0) &= C(0)x_0 + D(0)u(0) \; and \\
y(t) &= \underbrace{D(t)u(t) + \sum_{t'=0}^{t-1} C(t)\Phi(t-1,t')\Delta_d(t')u(t')}_{\text{“}input \quad dependent \quad term\text{”}} \\
&\quad + C(t)\Phi(t-1,0)\Phi_d(0)x_0, \quad t \geq 1.
\end{aligned} \tag{3.33}$$

Here

1. The *discrete state transition matrix* $\Phi(t, t')$ is the solution of the family of difference equations

$$\Phi(t, t') = \Phi_d(t)\Phi(t-1, t'), \quad \textit{with} \;\; \Phi(t', t') = I_n, \;\; 0 \le t' < t. \tag{3.34}$$

2. If $\Phi_d(t)$ is a constant matrix Φ_d, then

$$\Phi(t, t') = \Phi_d^{t-t'}. \tag{3.35}$$

3. If $S(\Phi_d, \Delta_d, C, D)$ is linear and time-invariant, then

   a. the solution of the state equations is computed from the following *discrete convolution* equations

$$\begin{aligned} x(t) &= \Phi_d^t x_0 + \sum\nolimits_{t'=0}^{t-1} \Phi_d^{t-t'-1}\Delta_d u(t'), && t \ge 0 \\ y(t) &= Du(t) + \sum\nolimits_{t'=0}^{t-1} C\Phi_d^{t-t'}\Delta_d u(t') + C\Phi_d^t x_0, && t \ge 0. \end{aligned} \tag{3.36}$$

   b. A linear time-invariant, discrete system is asymptotically stable if, and only if, all eigenvalues of the matrix Φ_d have modulus strictly less than unity. Equivalent statements are that all eigenvalues have values in the open unit circle in the complex plane and, as a consequence, for some $M \ge 1$ and $0 \le \lambda < 1$,

$$\|\Phi_d^t\| \le M\lambda^t, \;\; t = 0, 1, 2, 3, \ldots \tag{3.37}$$

3.5.3 The Discrete Convolution Operator and the Discrete Impulse Response Sequence

In a similar manner to the continuous time case, the mapping between the input time series u and the output time series y on the time interval $0 \le t \le N$ has the operational form

$$y = Gu + d \tag{3.38}$$

where d represents the time series $C(0)x_0$ followed by $C(t)\Phi(t-1, 0)\Phi_d(0)x_0$, $1 \le t \le N$, and G represents the linear mapping that maps u into the times series $z(0) = D(0)u(0)$ followed by

$$z(t) = D(t)u(t) + \sum_{t'=0}^{t-1} C(t)\Phi(t-1, t')\Delta_d(t')u(t'), \;\; 1 \le t \le N. \tag{3.39}$$

In the linear time-invariant case, the sequence $\{D, C\Delta, C\Phi_d\Delta_d, C\Phi_d^2\Delta_d, \ldots\}$ is termed the *discrete systems impulse response sequence*.

To formalize this representation, regard the time series $u \in \mathscr{U}$ and $y \in \mathscr{Y}$ as signals in suitable sequence spaces $\mathscr{U}$ and $\mathscr{Y}$. Here, $G : \mathscr{U} \mapsto \mathscr{Y}$ is a linear operator mapping $u(t),\ 0 \leq t \leq N$, into $z(t),\ 0 \leq t \leq N$. The choice of sequence space is non-unique. For the purposes of the current discussion, consider the case of N finite and the choices of Cartesian product Hilbert spaces

$$\mathscr{U} = \underbrace{\mathscr{R}^\ell \times \mathscr{R}^\ell \times \cdots \times \mathscr{R}^\ell}_{(N+1)-times} \quad \mathscr{Y} = \underbrace{\mathscr{R}^m \times \mathscr{R}^m \times \cdots \times \mathscr{R}^m}_{(N+1)-times}. \tag{3.40}$$

Define inner products by symmetric positive definite matrices R and Q using

$$\langle u, v\rangle_{\mathscr{U}} = \sum_{t=0}^{N} u^T(t)Rv(t) \quad \langle y, w\rangle_{\mathscr{Y}} = \sum_{t=0}^{N} y^T(t)Qw(t). \tag{3.41}$$

Note: For notational simplicity, the inner products $\langle u, v\rangle_{\mathscr{U}}$ and $\langle y, w\rangle_{\mathscr{Y}}$ may be denoted by $\langle u, v\rangle_R$ and $\langle y, w\rangle_Q$ to identify the associated weighting matrices.

3.6 $\mathscr{Z}$-Transforms and the Discrete Transfer Function Matrix

The $\mathscr{Z}$-transform is a methodology of analysing time series in the form of a sequence of scalar or matrix valued data points $\{f(0), f(1), f(2), \ldots\}$ using functions of the complex variable. It has been used widely in control systems analysis and design. The basic definition of the $\mathscr{Z}$-Transform of a scalar time series $f(t),\ 0 \leq t \leq N$ is most easily presented in the case on an infinite interval $N = \infty$. More precisely, the $\mathscr{Z}$-Transform of the (real or complex) series $f = \{f(0), f(1), f(2), \ldots\}$ is defined as follows

$$\mathscr{Z}[f] = \sum_{j=0}^{\infty} z^{-j} f(t), \tag{3.42}$$

for all complex number z such that the series converges. It is noted that a sufficient condition for the existence of such values of z is that f is *exponentially (or geometrically) bounded*. That is, there exists real numbers $M \geq 0$ and λ such that

$$|f(t)| \leq M\lambda^t \quad for \quad all \ \ t \geq 0. \tag{3.43}$$

For such a time series, the infinite sum converges for all z satisfying $|z| > \lambda$. Although only valid for values of z such that the series converges, the functions can be extended by analytic continuation to all points of the complex plane other than singularities. For this reason the issue of convergence plays only an occasional role in practical use of the ideas.

For notational convenience, the $\mathscr{Z}$-transform of the sequence f will be denoted by $f(z)$. In the discrete time case this does not lead to any confusion as f denotes a times series whilst $f(z)$ is a function of the complex variable. With this convention, the *Inverse $\mathscr{Z}$-Transform* is simply the inverse process of constructing the time series f from a given transform $f(z)$ and is denoted by $\mathscr{Z}^{-1}$, for example,

$$f = \mathscr{Z}^{-1}[f(z)]. \tag{3.44}$$

The extension of the ideas to matrix valued time series is easily achieved. For a $p \times q$ time series $H(t),\ t \geq 0$, its $\mathscr{Z}$-Transform $\mathscr{Z}[H]$ is denoted by the $p \times q$ matrix function $H(z)$ which has (i, j) element equal to the $\mathscr{Z}$-Transform of the time series $H_{ij}(t),\ t \geq 0$.

Finally, a number of useful properties of the $\mathscr{Z}$-Transform are used in the text and are summarized below for any sequence $f(t)$ defined on $0 \leq t < \infty$ with $f(0)$ written as f_0,

$$\begin{aligned} \mathscr{Z}[\{0, f(0), f(1), \ldots\}] = z^{-1} f(z)\ , \quad & \mathscr{Z}[\{1, \alpha, \alpha^2, \ldots\}] \ = \ \tfrac{z}{z-\alpha}, \\ \mathscr{Z}[\{1, 2\alpha, 3\alpha^2, \ldots\}] \ = \ z \tfrac{\partial}{\partial \alpha} \left(\tfrac{z}{(z-\alpha)} \right)\ , \quad & \mathscr{Z}[\{f(1), f(2), \ldots\}] \ = \ zf(z) - zf_0. \end{aligned} \tag{3.45}$$

3.6.1 Discrete Transfer Function Matrices, Poles, Zeros and the Relative Degree

A linear, time-invariant, state space model has a frequency domain description obtained simply by taking $\mathscr{Z}$-Transforms of the state equations to obtain

$$zx(z) - zx_0 = \Phi_d x(z) + \Delta_d u(z), \quad y(z) = Cx(z) + Du(z) \tag{3.46}$$

from which simple matrix algebra to eliminate $x(z)$ gives

$$y(z) = G(z)u(z) + zC(zI_n - \Phi_d)^{-1} x_0. \tag{3.47}$$

where the system $m \times \ell$ *Discrete Transfer Function Matrix* $G(z)$ is defined by

$$G(z) \ = \ C(zI_n - \Phi_d)^{-1} \Delta_d + D \tag{3.48}$$

Poles and zeros are defined as follows

1. The *poles* of $G(z)$ and their multiplicities are precisely the set of eigenvalues of the matrix Φ_d and their multiplicities.
2. The *zeros* of $G(z)$ are defined by rank conditions on a partitioned matrix dependent on (Φ_d, Δ_d, C, D). More precisely, the complex number z is a zero of $S(\Phi_d, \Delta_d, C, D)$ if and only if

$$rank \begin{bmatrix} zI_n - \Phi_d & -\Delta_d \\ C & D \end{bmatrix} < n + \min\{m, \ell\} \tag{3.49}$$

   If all zeros lie in the open unit circle in the complex plane, the system is said to be *minimum-phase*. Otherwise, the system is said to be *non-minimum phase*.

A comparison with the equivalent formula for the transfer function matrix of a continuous time system shows the algebraic similarity, the discrete time case being obtained from the continuous time formula by the mapping $s \mapsto z$ and $(A, B, C, D) \mapsto (\Phi_d, \Delta_d, C, D)$. It is natural therefore that the discrete time case inherits the properties and nomenclature described previously for continuous time case. In particular, its relative degree is the uniquely defined integer k^* such that $\lim_{z\to\infty} z^k G(z) = G_\infty$ exists and is non-zero.

3.6.2 The Discrete System Frequency Response

The linear time-invariant discrete system $S(\Phi_d, \Delta_d, C, D)$ has frequency response function defined to be the (matrix-valued) function $G(z)$ where $|z| = 1$ or, equivalently, $z = e^{i\theta},\ 0 \le \theta \le 2\pi$. This function has found considerable application in multivariable feedback control systems design. It also describes the response of an asymptotically stable system from zero initial conditions $x_0 = 0$ to sinusoidal input time series u_c, u_s defined to be the real, (respectively) imaginary parts of the complex sequence $u = \{\alpha, z\alpha, z^2\alpha, \ldots\}$ (that is, $u(t) = \alpha z^t,\ t \ge 0$). More generally, if the data (Φ_d, Δ_d, C, D) contains only real numbers and z is an arbitrary complex number, then the resultant output responses from zero initial conditions are, respectively, equal to the real and imaginary parts of the response $y(t)$ from zero initial conditions to the complex input $u(t)$. It is left as an exercise for the reader to verify that

$$y(t) = G(z)u(t) - C(zI_n - \Phi_d)^{-1}\Phi_d^t \Delta_d \alpha, \quad t = 0, 1, 2, \ldots. \tag{3.50}$$

In particular, if $z = e^{i\theta}$, the stability assumption indicates that, for all large enough values of sample index t, the steady state oscillation $y(t) \approx G(z)\alpha z^t$ is an accurate description of steady state oscillatory output behaviour as Φ_d^t becomes infinitesimally small if t is large.

3.7 Multi-rate Discrete Time Systems

Multi-rate sampling is a common practical way of improving control for systems where the frequency at which output signals can be sampled is limited by measurement procedures or other factors. In such circumstances, the input signal could be continuous or discrete as discussed in the previous sections. Models of such systems are fairly easily parameterized using the previous approaches but with an additional computation that most easily is described as a multi-rate sampling operator. More precisely, suppose that input updates take place at times $0, h, 2h, 3h, \ldots$ but that output samples are taken at times $0, Mh, 2Mh, 3Mh \ldots$. Note that, sampling is synchronous to the extent that samples are only taken at a subset of times when the control signal is updated. A simple way of describing a mathematical model of such a system is as *two* operations. The first is a dynamical mapping of the input time series into an output time series $y = \{y(0), y(h), y(2h), \ldots\}$ and the second is a sampling operation that takes every Mth sample of y to produce the measured output y^e defined by $y^e = \{y^e(0), y^e(1), y^e(3), \ldots\} = \{y(0), y(Mh), y(2Mh), \ldots\}$. The intermediate output signals are not measured and play a role in the model description only.

The mapping of u into y can be described by a model typically of the form described in earlier sections and could be written, for linear systems, in operator form as $y = Gu + d$. The multi-rate operator, denoted $\mathscr{S}_M$ for simplicity, is a linear operator on times series with sampling frequency h^{-1} into a time series of sampling frequency $(Mh)^{-1}$. The dynamics is hence of the form of a series connection of two operators

$$y = Gu + d, \quad y^e = \mathscr{S}_M y \quad so \ that \quad y^e = \mathscr{S}_M G u + \mathscr{S}_M d. \tag{3.51}$$

In particular, the mathematical structure at the operator level is identical and any approach to design that uses the operator descriptions will have a similar form for continuous systems and for both uni-rate and multi-rate sampled data systems.

3.8 Controllability, Observability, Minimal Realizations and Pole Allocation

This section considers continuous systems only but, being algebraic in nature, the results apply with no change to uni-rate discrete systems simply by replacing the complex variable s by z and the quadruple of matrices (A, B, C, D) by Φ_d, Δ_d, C, D.

The analysis of linear, time-invariant, continuous state space models relies heavily on properties of (A, B, C, D). From time to time, the ideas of *complete state controllability* and *complete state observability* are crucial.

Definition 3.1 The linear system $S(A, B, C, D)$ is completely state controllable if, and only if, for any initial state $x(0)$ and any desired final state x_f, there exists a control input $u(t)$ and a time t_f such that $u(t)$ drives the state $x(t)$ from x_0 at $t = 0$ to x_f at time $t = t_f$.

Definition 3.2 The linear system $S(A, B, C, D)$ is completely state observable if, and only if, for any initial state $x(0)$ and any control input $u(t),\ t \geq 0$, there exists a time $t_f > 0$ such that the initial state x_0 can be computed uniquely from input and output records only on some time interval $0 \leq t \leq t_f$.

For linear time-invariant linear systems the relevant algebraic tests are as follows:

1. The system $S(A, B, C, D)$ is completely state controllable if, and only if, one of the following two (equivalent) conditions is true

$$\begin{gathered} rank[B, AB, A^2B, \ldots, A^{n-1}B] = n \\ rank[sI - A, -B] = n \quad \textit{for all complex numbers } s \end{gathered} \tag{3.52}$$

2. The system $S(A, B, C, D)$ is completely state observable if, and only if, one of the following two (equivalent) conditions is true

$$rank \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} = n, \quad rank \begin{bmatrix} sI - A \\ C \end{bmatrix} = n \ \textit{for all complex numbers } s \tag{3.53}$$

3. As a consequence,
   a. the system $S(A, B, C, D)$ is completely state controllable (respectively, observable) if, and only if, $S(A^T, C^T, B^T, D^T)$ is completely state observable (respectively, controllable).
   b. For all state feedback (respectively output injection) matrices F (respectively H), the completely state controllability (respectively observability) of the system $S(A, B, C, D)$ implies the complete state controllability (respectively observability) of the system $S(A - BF, B, C, D)$ (respectively $S(A - HC, B, C, D)$).

A *realization* of a proper or strictly proper transfer function matrix $G(s)$ is any linear, time-invariant model $S(A, B, C, D)$ whose transfer function matrix coincides with $G(s)$. A *minimal realization* is any realization where the state dimension n takes the lowest possible value. Any realization is minimal if, and only if, it is controllable and observable.

A useful property of controllable systems is the potential for *pole allocation*. More precisely, a *state feedback control law* takes the form

$$u(t) = Kv(t) - Fx(t) \tag{3.54}$$

where F is an $\ell \times n$ real matrix of state feedback gains, K is an $\ell \times \ell$ matrix of gains and $v(t)$ is the new input signal. The closed-loop state space model relating the input v to the output y is based on the same state vector model but with matrices $S(A - BF, BK, C - DF, DK)$. Note that

1. if $S(A, B, C, D)$ is completely state controllable and K is nonsingular, then the closed loop system $S(A-BF, BK, C-DF, DK)$ is completely state controllable.
2. The stability of the new feedback system is described by the characteristic polynomial $\rho_F(s) = \det[sI - A + BF]$. The important property of completely state controllable systems is the pole allocation property. That is, for any set of complex numbers $\{\mu_j\}_{1\leq j\leq n}$ (invariant under complex conjugation), there exists a real state feedback matrix F such that the eigenvalues of $A - BF$ are precisely the points $\{\mu_j\}_{1\leq j\leq n}$.

3.9 Inverse Systems

Dynamical systems G are normally regarded as mapping from inputs to outputs. The general concept of an *Inverse System* is a system that reconstructs the input from the observed output. As the output depends also on the initial state condition, it is usual to assume zero initial conditions. For square systems (where $m = \ell$) the operation parallels the simple observation that $y(s) = G(s)u(s)$ can be reconstructed using $u(s) = G^{-1}(s)y(s)$ provided that the inverse transfer function matrix exists.

Note: The analysis that follows applies equally to both continuous time and discrete time systems as the definitions and manipulations used are entirely algebraic.

*3.9.1 The Case of m = ℓ, Zeros and v**

If $m = \ell$ and $S(A, B, C, D)$ has transfer function matrix satisfying $\det[G(s)] \neq 0$ at all but a finite number of points s, then the system is said to be *invertible*. The *inverse transfer function matrix* takes the form

$$G^{-1}(s) \equiv P(s) + H(s) \tag{3.55}$$

where P and H are uniquely defined by the statements,

1. H(s) is strictly proper matrix of rational polynomials and
2. P(s) is either a constant matrix or has elements that are polynomials in s.

Theorem 3.1 *Let $S(A, B, C, D)$ have a transfer function matrix with inverse of the form of (3.55). Then, the matrix $P(s)$ is invertible for almost all values of s with*

$$\begin{aligned} \lim_{s\to\infty} G(s)P(s) = \lim_{s\to\infty} P(s)G(s) = I, \quad \textit{and} \\ \lim_{s\to\infty} P^{-1}(s)H(s) = \lim_{s\to\infty} H(s)P^{-1}(s) = 0. \end{aligned} \tag{3.56}$$

Moreover, $\lim_{s\to\infty} P^{-1}(s) = D$ and hence $P^{-1}(s)$ has a minimal realization in terms of a proper state space model. This model is strictly proper if $G(s)$ is strictly proper.

Proof Writing $GG^{-1} = I = GP + GH$ then $\lim_{|s|\to\infty} G(s)P(s) = I$, as required, as GH is strictly proper. Using a similar argument, $G^{-1}G = I = PG + HG$ so that $\lim_{|s|\to\infty} P(s)G(s) = I$. It follows that $P(s)$ is invertible. Also writing $(PG)^{-1} = I + HP^{-1}$ yields $\lim_{|s|\to\infty} H(s)P^{-1}(s) = 0$ and, similarly, $\lim_{|s|\to\infty} P^{-1}(s)H(s) = 0$. Finally, write $G = (I + P^{-1}H)^{-1}P^{-1}$ and note that, as a consequence $D = \lim_{|s|\to\infty} G(s) = \lim_{|s|\to\infty} P^{-1}(s)$ and hence $P^{-1}(s)$ has a minimal realization in terms of a proper state space model. Taking $D = 0$ shows that this model is strictly proper if $G(s)$ is strictly proper. □

Assuming zero initial conditions, the input output relations take the form $y(s) = G(s)u(s)$ and hence $u(s) = G^{-1}(s)y(s)$. That is, the inverse system reconstructs the input vector uniquely from the output vector. The transfer function matrix $G^{-1}(s)$, typically, is neither proper nor strictly proper. Its construction is computationally complex but has a simple state space form if D is nonsingular:

Theorem 3.2 *Suppose that the system $S(A, B, C, D)$ is such that D is nonsingular, then $G^{-1}(s)$ has a state space realization $S(A - BD^{-1}C, BD^{-1}, -D^{-1}C, D^{-1})$*

$$\dot{x}(t) = (A - BD^{-1}C)x(t) + BD^{-1}y(t), \quad u(t) = -D^{-1}Cx(t) + D^{-1}y(t) \tag{3.57}$$

Moreover, the poles of this system are the zeros of $S(A, B, C, D)$.

Proof The state space description follows easily from the output equation $y = Cx + Du$ written in the form $u = D^{-1}(y - Cx)$ and eliminating u from the state equations. The equivalence of its eigenvalues to the zeros follows from Schur's Formula

$$\begin{vmatrix} sI - A & -B \\ C & D \end{vmatrix} \equiv \det[D]\det[sI - A + BD^{-1}C]. \tag{3.58}$$

□

More generally, the poles of $H(s)$ are zeros of $S(A, B, C, D)$ and hence $H(s) \equiv 0$ if the system has no zeros. Conversely, any $m \times m$ matrix $P(s)$ of polynomials in s that is invertible and such that the limit $\lim_{s\to\infty} P^{-1}(s) = P_\infty$ exists, has a minimal (controllable and observable) state space realization that has no zeros.

The construction of P and H can be done using algebraic inversion of $G(s)$. An alternative approach valid for the case when $D = 0$ uses a direct sum decomposition of the state space

$$\mathscr{R}^n = \nu^* \oplus \tau^* \tag{3.59}$$

That is, every vector $x \in \mathscr{R}^n$ can be written uniquely as the sum of a vector $z_1 \in \tau^*$ and a vector $z_2 \in \nu^*$. These subspaces form the basis for the construction of the $\mathscr{C}^*$-canonical form due to Morse. The subspace ν^* has a particular connection with the zeros of the strictly proper system $S(A, B, C)$ stated as follows,

Theorem 3.3 *ν^* is the maximal $\{A, B\}$-invariant subspace in $ker[C]$. That is, it is the largest subspace $\nu \subset \mathscr{R}^n$ satisfying the inclusion conditions*

$$A\nu \subset \nu + \mathscr{R}[B], \quad \nu \subset ker[C]. \tag{3.60}$$

In addition, there exists a (non-unique) feedback matrix F such that $(A-BF)\nu^* \subset \nu^*$ and the eigenvalues of the restriction of $A - BF$ to ν^* are precisely the zeros of $S(A, B, C)$ (including multiplicities). Any minimal realization of $S(A - BF, B, C)$ has no zeros.

A simple description of system dynamics from $G(s), P^{-1}(s)$ and $H(s)$ starts with their minimal realizations $S(A, B, C)$, $S(A_P, B_P, C_P)$ and $S(A_H, B_H, C_H)$ and associated state vectors x, x_P, x_H. Writing $x^T = [x_P^T, x_H^T]^T$ appropriate matrices are computed from the feedback description $G = (I + P^{-1}H)^{-1}P^{-1}$ as follows

$$A = \begin{bmatrix} A_P & -B_P C_H \\ B_H C_P & A_H \end{bmatrix}, \quad B = \begin{bmatrix} B_P \\ 0 \end{bmatrix}, \quad C = [C_P \quad 0] \tag{3.61}$$

where the zeros of $S(A, B, C)$ are the eigenvalues of A_H.

3.9.2 Left and Right Inverses When $m \neq \ell$

When $m \neq \ell$, the transfer function matrix is non-square and matrix inversion techniques do not apply. The relevant ideas are those of left inverse and right inverse.

3.9.2.1 Left and Right Inverses

Let $S(A, B, C, D)$ have transfer function matrix $G(s)$ and suppose that $m > \ell$. Then a *Left Inverse* of $G(s)$ can be regarded as *any* $\ell \times m$ matrix $L(s)$ of rational polynomials in s such that

$$L(s)G(s) = I_\ell \tag{3.62}$$

In conceptual terms, assuming zero initial conditions, an input $u(s)$ produces an output $y(s) = G(s)u(s)$ from G. From the definition of L, the input is reconstructed using $u(s) = L(s)y(s)$. Note that $L(s)$ is non-unique and, in general, takes the form $L(s) = P(s) + H_0(s)$ where $H_0(s)$ is proper and $P(s)$ is a matrix of polynomials in s with $P(0) = 0$. The left inverse hence, in general, does not have a state space realization of an ℓ-output, m-input system $S(A_L, B_L, C_L, D_L)$. The case of interest when such a state space realization exists is defined as follows:

Theorem 3.4 *Suppose that $m > \ell$ and that $S(A, B, C, D)$ satisfies the condition $rank[D] = \ell$. Suppose also that K is any $\ell \times m$ matrix such that KD is nonsingular. Then the state space system*

$$S(A - B(KD)^{-1}KC, B(KD)^{-1}K, -(KD)^{-1}KC, (KD)^{-1}K) \tag{3.63}$$

maps $y \mapsto u$ and hence is a left inverse of $S(A, B, C, D)$.

Proof From the construction, write $y = Cx + Du$ and hence $u = -(KD)^{-1}KCx + (KD)^{-1}Ky$. The result follows by substitution of this expression into the equations for $x(t)$. □

The case of $m < \ell$ is analysed in a similar manner with a *Right Inverse* of $G(s)$ being regarded as any $m \times \ell$ matrix $R(s)$ of rational polynomials in s such that

$$G(s)R(s) = I_m \tag{3.64}$$

It is a simple matter to see that $R^T(s)$ is a left inverse of $G^T(s)$ and hence, applying the previous theorem to $S(A^T, C^T, B^T, D^T)$ gives,

Theorem 3.5 *Suppose that $m < \ell$ and that $S(A, B, C, D)$ satisfies the condition $rank[D] = m$. Suppose also that K is any $\ell \times m$ matrix such that DK is nonsingular. Then the state space system*

$$S(A - BK(DK)^{-1}C, -BK(DK)^{-1}, K(DK)^{-1}C, K(DK)^{-1}) \tag{3.65}$$

is a right inverse of $S(A, B, C, D)$.

Proof A left inverse of $S(A^T, C^T, B^T, D^T)$ is, using the previous theorem,

$$S(A^T - C^T(K^TD^T)^{-1}K^TB^T, C^T(K^TD^T)^{-1}K^T, -(K^TD^T)^{-1}K^TB^T, (K^TD^T)^{-1}K^T) \tag{3.66}$$

which has transfer function matrix $L^T(s)$. This is converted easily to the given right inverse by transposition. □

3.10 Quadratic Optimal Control of Linear Continuous Systems

The purpose of this section is to illustrate the application of the language described in this chapter to a basic optimal control problem. The problem is chosen to match the results typically described in specialist texts on optimal control and does not fully meet the needs of this text. The required generalizations will be described at appropriate points as, and when, needed and will refer back to this section for techniques and useful simplifications of the presentation.

A dynamic system is described by the ℓ-input, m-output, linear, time varying state space model

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t), \quad x(0) = x_0 \quad \textit{and} \\ y(t) &= C(t)x(t) \end{aligned} \tag{3.67}$$

with state vector $x(t) \in \mathscr{R}^n$ and piecewise continuous $A(t)$, $B(t)$, $C(t)$. Note that it is assumed that $D(t) \equiv 0$. Suppose that the control input $u(t)$ on a finite time interval $[0, T]$ is to be chosen to minimize the performance criterion $J(u)$ defined by

$$\begin{aligned} &J(u) = (r_T - y(T))^T F(r_T - y(T)) \\ &+ \int_0^T \left((r(t) - y(t))^T Q(t)(r(t) - y(t)) + (u(t) - u_0(t))^T R(t)(u(t) - u_0(t))\right) dt \end{aligned} \tag{3.68}$$

where both $R(t) = R^T(t) > 0$ and $Q(t) = Q^T(t) > 0$ are piecewise continuous, symmetric and positive definite for all $t \in [0, T]$, $F = F^T \geq 0$ is symmetric and positive semi-definite, $u_0 \in L_2^\ell[0, T]$, $r \in L_2^m[0, T]$ and $r_T \in \mathscr{R}^m$. Intuitively the control objective is to ensure that the output remains close to the *reference signal* $r(t)$, ensuring that $r(T)$ is close to r_T whilst keeping the input $u(t)$ close to $u_0(t)$. Both the phrase "close to" and the relative importance of the three objectives is specified by the choice of form and relative magnitudes of the weighting matrices F, $Q(t)$ and $R(t)$. In general, it might be expected that $r_T = r(T)$ but this is not necessary in what follows.

Representations of the solution to the problem are now constructed using operator methods, a two point boundary value problem and a Riccati solution.

3.10.1 The Relevant Operators and Spaces

The first step in the characterization of the optimal input is to write the problem as a minimum norm problem in a suitable real Hilbert space. To do this, suppose $\mathscr{U} = L_2^\ell[0, T]$ is the chosen input space, with inner product $\langle \cdot, \cdot \rangle_R$ and associated induced norm $\| \cdot \|_R$ identified with

$$\langle u, v\rangle_R = \int_0^T u^T(t)R(t)v(t)dt \quad \textit{and} \quad \|u\|_R = \sqrt{\int_0^T u^T(t)R(t)u(t)dt}. \tag{3.69}$$

The result of the application of this input to the system has *two* important components for the problem, the first being associated with the output response $y(t),\ t \in [0, T]$ and the second identified with the value $y(T)$ at the final time T. The weighting matrix F is not assumed to be positive definite and the problem needs a little more manipulation to make the Hilbert space formulation work effectively. For ease of presentation, it is assumed that $F \neq 0$ and that it is factorized into the product

$$F = F_f^T Q_f F_f \tag{3.70}$$

where F_f has full row rank m_f (that is, F_f is $m_f \times m$ with $ker[F_f^T] = \{0\}$) and $Q_f = Q_f^T > 0$. The results can be made to apply to the case of $F = 0$ by choosing $m_f = m$, $F_f = 0$ and $Q_f = Q_f^T > 0$ chosen arbitrarily. The details of this special case are left for the reader to complete.

Now regard the system as a mapping $y^e = Gu + d$ of u into the pair $y^e = (y_f, y) \in \mathscr{Y} = \mathscr{R}^{m_f} \times L_2^m[0, T]$. Here, $y_f = F_f y(T)$ and $(y_f, y) = Gu + d$ where the operator $G : \mathscr{U} \to \mathscr{Y}$ is defined by the notation $Gu = \big(G_f u, G_y u\big)$ where the operators $G_y : \mathscr{U} \to L_2^m[0, T]$ and $G_f : \mathscr{U} \to \mathscr{R}^{m_f}$ and the vector $d = (d_f, d_y) \in \mathscr{Y}$ are defined by the equations

$$\begin{aligned} G_f u &= F_f C(T) \int_0^T \Phi(T, t)B(t)u(t)dt, \\ (G_y u)(t) &= C(t) \int_0^t \Phi(t, t')B(t')u(t')dt' \quad \textit{and} \\ d(t) = (d_f, d_y(t)) &= \big(F_f C(T)\Phi(T, 0)x_0\ ,\ C(t)\Phi(t, 0)x_0\big) \quad \textit{for all}\ \ t \in [0, T]. \end{aligned} \tag{3.71}$$

Regarding $\mathscr{R}^{m_f}$ (respectively, $L_2^m[0, T]$) as real Hilbert spaces with inner products

$$\langle z_f, y_f\rangle = z_f^T Q_f y_f \quad (\textit{respectively} \quad \langle z, y\rangle = \int_0^T z^T(t)Q(t)y(t)dt), \tag{3.72}$$

then the product output space $\mathscr{Y} = \mathscr{R}^{m_f} \times L_2^m[0, T]$ is a real Hilbert space with respect to the inner product

$$\langle (z_f, z), (y_f, y)\rangle_{\mathscr{Y}} = z_f^T Q_f y_f + \int_0^T z^T(t)Q(t)y(t)dt \tag{3.73}$$

and the performance index $J(u)$ can be written in the form

$$J(u) = \|(r_f, r) - (y_f, y)\|_{\mathscr{Y}}^2 + \|u_0 - u\|_R^2 \quad \textit{where} \quad r_f = F_f r_T. \tag{3.74}$$

The optimal control problem hence fits into the framework of Theorem 2.18 and has a solution that can be written in the form

$$u = u_0 + G^* e \quad \textit{where} \quad e = (r_f, r) - (y_f, y) = (r_f - y_f, r - y). \tag{3.75}$$

It is therefore necessary to identify the form of the adjoint operator $G^* : \mathscr{R}^{m_f} \times L_2^m[0,T] \to L_2^\ell[0,T]$.

3.10.2 Computation of the Adjoint Operator

Using Theorem 2.5, the adjoint operator G^* of G can be computed from the definitions and the characterizations of its two components as follows

$$G^*(r_f - y_f, r - y) = G_f^*(r_f - y_f) + G_y^*(r - y) \tag{3.76}$$

The following results provide the appropriate characterization, firstly for the map $u \mapsto y$ and then for $u \mapsto y_f$. The notation used follows on from that used in the previous section.

Theorem 3.6 (The Adjoint Operator of a Linear State Space System) *Using the notation defined above, the linear map $u \mapsto y$ defined by the ℓ-input, m-output, linear, time varying state space model*

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t), \quad x(0) = x_0 \quad \textit{and} \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \tag{3.77}$$

is a bounded, linear operator from $L_2^\ell[0,T]$ into $L_2^m[0,T]$ with the defined Hilbert space inner products. Its adjoint operator is the map $z \mapsto u$ defined by

$$u(t) = R^{-1}(t)D^T(t)Q(t)z(t) + R^{-1}(t)B^T(t)\int_t^T \Phi^T(t',t)C^T(t')Q(t')z(t')dt' \tag{3.78}$$

which has the state space representation, with costate $p_y(t) \in \mathscr{R}^n$, of the form

$$\begin{aligned} \dot{p}_y(t) &= -A^T(t)p_y(t) - C^T(t)Q(t)z(t), \quad p_y(T) = 0, \\ \textit{and} \quad u(t) &= R^{-1}(t)D^T(t)Q(t)z(t) + R^{-1}(t)B^T(t)p_y(t). \end{aligned} \tag{3.79}$$

Proof Denote the map $u \mapsto y$ by G_y. The proof starts from the inner product

$$\langle z, G_y u\rangle = \int_0^T z^T(t)Q(t)\left(D(t)u(t) + C(t)\int_0^t \Phi(t,t')B(t')u(t')dt'\right)dt \tag{3.80}$$

The first term is easily manipulated to take the form

$$\int_0^T z^T(t)Q(t)D(t)u(t)dt = \int_0^T (R^{-1}(t)D^T(t)Q(t)z(t)(t))^T R(t)u(t)dt \quad (3.81)$$

whilst the second term, after interchanging the order of integration (and the symbols t and t'), has the form

$$\begin{aligned}
&\int_0^T \int_0^t z^T(t)Q(t)C(t)\Phi(t,t')B(t')u(t')dt'dt \\
&= \int_0^T \left(R^{-1}(t')B^T(t')\int_{t'}^T \Phi^T(t,t')C^T(t)Q(t)z(t)dt\right)^T R(t')u(t'dt)' \\
&= \int_0^T \left(R^{-1}(t)B^T(t)\int_t^T \Phi^T(t',t)C^T(t')Q(t')z(t')dt'\right)^T R(t)u(t)dt. \quad (3.82)
\end{aligned}$$

The first part of the theorem follows by summing the new representations of the two terms and comparing with $\langle G_y^* z, u\rangle_R$. It remains to show that this representation has the required state space model. First write,

$$\begin{aligned}
u(t) &= R^{-1}(t)D^T(t)Q(t)z(t) + R^{-1}(t)B^T(t)p_y(t), \\
\textit{where}\quad p_y(t) &= \int_t^T \Phi^T(t',t)C^T(t')Q(t')z(t')dt'. \quad (3.83)
\end{aligned}$$

Setting $t = T$ gives the terminal condition $p_y(T) = 0$ and differentiating $p_y(t)$ gives

$$\dot{p}_y(t) = -\Phi^T(t,t)C^T(t)z(t) + \int_t^T \frac{\partial \Phi^T(t',t)}{\partial t} C^T(t')Q(t')z(t')dt' \quad (3.84)$$

The result is now proved as $\Phi(t,t) = I$ and $\frac{\partial \Phi^T(t',t)}{\partial t} = -A^T(t)\Phi^T(t',t)$. □

A useful observation is obtained for the case of linear time invariant systems. It relates the adjoint operator to the *Time Reversal Operator* $\mathscr{T}_T$ defined by the map $f \mapsto \mathscr{T}_T f$ and the relation

$$(\mathscr{T}_T f)(t) = f(T-t) \quad (\textit{the Time Reversal of } f \textit{ on } [0,T]). \quad (3.85)$$

This notation forms a useful part of the following statement of the form and properties of the adjoint. In particular, it indicates that both the system and its adjoint can be related to simulations of state space models from zero initial conditions.

Theorem 3.7 (The Adjoint Operator of a Linear Time Invariant State Space System) *Assuming the notation of the previous theorem, assume that the state space model $S(A,B,C,D)$ is linear and time invariant with zero initial conditions and that the matrices $Q(t)$ and $R(t)$ are constant, symmetric and positive definite. Then the state*

space system is a bounded linear operator mapping the real Hilbert space $L_2^{\ell}[0,T]$ *into the real Hilbert space* $L_2^{m}[0,T]$ *with adjoint operator represented by the linear, time invariant system* $S(-A^T, -C^TQ, R^{-1}B^T, R^{-1}D^TQ)$ *with a terminal conditions* $p_y(T) = 0$.

In particular, the action of the adjoint operator on a signal z *can be written as the time reversal of the response of the system* $S(A^T, C^TQ, R^{-1}B^T, R^{-1}D^TQ)$ *from zero initial conditions to the time reversal of* z.

Proof The proof uses the substitution t to $t' = T - t$ and the relation $\frac{d}{dt'} = -\frac{d}{dt}$. It is left as an exercise for the reader. □

Now consider that operator G_f that maps $u \mapsto y_f$.

Theorem 3.8 (The Adjoint of the Operator Generating the Final Output) *Using the notation of Theorem 3.6, suppose that* $D(t) \equiv 0$. *The linear map* $u \mapsto y_f$ *defined by the* ℓ*-input, m-output, linear, time varying state space model*

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t), \quad x(0) = x_0 \quad \textit{and} \\ y_f &= F_f y(T), \quad \textit{where } y(T) = C(T)x(T) \end{aligned} \tag{3.86}$$

is a bounded, linear operator from $L_2^{\ell}[0,T]$ *into* $\mathscr{R}^{m_f}$ *with the defined Hilbert space inner products. Its adjoint operator is the map* $z_f \mapsto u$ *defined by*

$$u(t) \;=\; R^{-1}(t)B^T(t)\Phi^T(T,t)C^T(T)F_f^T Q_f z_f \tag{3.87}$$

which has the state space representation with costate $p_f(t) \in \mathscr{R}^n$ *satisfying*

$$\begin{aligned} \dot{p}_f(t) &= -A^T(t)p_f(t), \qquad p_f(T) = C^T(T)F_f^T Q_f z_f \\ \textit{and} \quad u(t) &= R^{-1}(t)B^T(t)p_f(t) \end{aligned} \tag{3.88}$$

Note: This equation is exactly that for $p_y(t)$ *with* $Q(t) \equiv 0$ *and with the addition of a possibly non-zero final condition at* $t = T$.

Proof Again denoting the map by G_f, the inner product $\langle z_f, G_f u\rangle$ has the form

$$\begin{aligned} & z_f^T Q_f F_f C(T) \textstyle\int_0^T \; \Phi(T,t)B(t)u(t)dt \\ = & \textstyle\int_0^T \; \left(R^{-1}(t)B^T(t)\Phi^T(T,t)C^T(T)F_f^T Q_f z_f\right)^T R(t)u(t)dt \end{aligned} \tag{3.89}$$

which proves the result by comparing this with $\langle G_f^* z_f, u\rangle_R$, defining the co-state $p_f(t) \;=\; \Phi^T(T,t)C^T(T)F_f^T Q_f z_f$ and using the properties of the state transition matrix. The details are left as an exercise for the reader. □

Unfortunately, a knowledge of the form of the adjoint operator is not, in general, sufficient for immediate real world application as the representation of the optimal input is not causal. The value of the required control input $u(t)$ at time $t < T$ depends on values of the output $y(t')$ for times $t' > t$, values that are not normally available at time t. This does not mean that the theory cannot be applied but it does mean that an alternative description of the input would be beneficial and probably essential for the purposes of most applications. The approach to solutions of this problem begin with the conversion of the representation into a two point boundary value problem.

3.10.3 The Two Point Boundary Value Problem

Returning now to the characterization $u = u_0 + G^*e$ using the results described above. Theorem 2.5 and the linearity of the costate equations indicates that the term G^*e can be described by setting

$$p(t) = p_y(t) + p_f(t) \quad for \quad all \quad t \in [0, T] \tag{3.90}$$

to obtain the equations (noting the assumption that $D(t) \equiv 0$)

$$\begin{aligned} \dot{p}(t) &= -A^T(t)p(t) - C^T(t)Q(t)(r(t) - y(t)), \\ p(T) &= C^T(T)F_f^T Q_f(r_f - y_f) \\ and \quad (G^*e)(t) &= R^{-1}(t)B^T(t)p(t) \end{aligned} \tag{3.91}$$

Using the relations $r_f = F_f r_T$ and $y_f = F_f y(T)$ then gives the result

Theorem 3.9 (Two Point Boundary Value Problem for the Optimal Control Input) *Using the notation and assumptions given above, the optimal control input to the linear, time varying system $S(A(t), B(t), C(t))$ that minimizes the objective function*

$$\begin{aligned} &J(u) = (r_T - y(T))^T F(r_T - y(T)) \\ &+ \int_0^T \left((r(t) - y(t))^T Q(t)(r(t) - y(t)) + (u(t) - u_0(t))^T R(t)(u(t) - u_0(t))\right) dt \end{aligned} \tag{3.92}$$

can be computed from the equation

$$u(t) = u_0(t) + R^{-1}(t)B^T(t)p(t), \quad t \in [0, T], \tag{3.93}$$

where the state vector $x(t)$ and the costate vector $p(t)$ are the solutions of the coupled differential equations

$$\begin{aligned} &\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad y(t) = C(t)x(t), \quad and \\ &\dot{p}(t) = -A^T(t)p(t) - C^T(t)Q(t)(r(t) - y(t)) \end{aligned} \tag{3.94}$$

subject to the initial and final conditions

$$x(0) = x_0 \quad \textit{and} \quad p(T) = C^T(T)F(r_T - y(T)) \tag{3.95}$$

The presence of boundary conditions at the two points $t = 0$ and $t = T$ have lead to the term *Two Point Boundary Value Problem* for this characterization. More generally, such problems are difficult to solve, particularly if a system is time varying and nonlinear. However, for the state space, linear systems of this text, solutions can be obtained using Riccati methods, an example of which is described next.

3.10.4 The Riccati Equation and a State Feedback Plus Feedforward Representation

The solution of any two point boundary value problem of the above general form can be approached using a simple parameterization of the solution. More precisely, suppose that $x(t)$ and $p(t)$ are related by the equation

$$p(t) = -K(t)x(t) + \xi(t), \quad \textit{for} \quad \textit{all} \quad t \in [0, T] \tag{3.96}$$

where $K(t)$ is an $n \times n$ time dependent matrix that is differentiable everywhere in $[0, T]$ and $\xi(t)$ is a differentiable vector function in $\mathscr{R}^n$ to be determined. It is useful at this stage to substitute this into the expression for $u(t)$ to obtain

$$\begin{aligned} u(t) &= u_0(t) + R^{-1}(t)B^T(t)p(t) \\ &= u_0(t) - R^{-1}(t)B^T(t)K(t)x(t) + R^{-1}(t)B^T(t)\xi(t), \quad \textit{for} \quad t \in [0, T]. \end{aligned} \tag{3.97}$$

Note: The expression indicates that the optimal controller is a modification of $u_0(t)$ to include a *time varying state feedback* term $-R^{-1}(t)B^T(t)K(t)x(t)$ and a term $R^{-1}(t)B^T(t)\xi(t)$. As will be shown below, $\xi(t)$ depends only on F, Q, R, K, r_T and the signals $u_0(t)$ and $r(t)$. As this term is independent of the dynamics of the state, it will be regarded as a *feedforward (or predictive) component* of the control law that transfers data from the control objectives and u_0 to the optimal solution.

Differentiating equation (3.96) gives

$$\dot{p}(t) = -\dot{K}(t)x(t) - K(t)\dot{x}(t) + \dot{\xi}(t). \tag{3.98}$$

Substituting from the state and costate equations and rearranging then requires that

$$\begin{aligned} &\left(\tfrac{dK(t)}{dt} + A^T(t)K(t) + K(t)A(t) - K(t)B(t)R^{-1}(t)B^T(t)K(t) + C^T(t)Q(t)C(t)\right)x(t) \\ &\equiv \tfrac{d\xi(t)}{dt} + \left(A^T(t) - K(t)B(t)R^{-1}(t)B^T(t)\right)\xi(t) + C^T(t)Q(t)r(t) - K(t)B(t)u_0(t). \end{aligned} \tag{3.99}$$

To remove any dependence of $\xi(t)$ on the state vector, set

$$\frac{d\xi(t)}{dt} = -\left(A^T(t) - K(t)B(t)R^{-1}(t)B^T(t)\right)\xi(t) - C^T(t)Q(t)r(t) + K(t)B(t)u_0(t) \quad (3.100)$$

and, to ensure that $K(t)$ is state independent, choose $K(t)$ as the unique solution of the *Differential Riccati Equation*

$$\frac{dK(t)}{dt} + A^T(t)K(t) + K(t)A(t) - K(t)B(t)R^{-1}(t)B^T(t)K(t) + C^T(t)Q(t)C(t) = 0. \quad (3.101)$$

The boundary conditions for these two equations are conditions defined at the terminal time $t = T$ and obtained from the condition defining $p(T)$, namely,

$$p(T) = C^T(T)F(r_T - y(T)) = C^T(T)F(r_T - C(T)x(T)) = -K(T)x(T) + \xi(T) \quad (3.102)$$

which is satisfied if $K(t)$ and $\xi(t)$ satisfy the terminal conditions,

$$K(T) = C^T(T)FC(T) \quad \textit{and} \quad \xi(T) = C^T(T)Fr_T. \quad (3.103)$$

Note that $K(T)$ is symmetric and positive definite. In computational terms, the Riccati matrix $K(t)$ is independent of the initial condition x_0 and the three tracking signals r_T, $r(t)$ and $u_0(t)$. Once computed, it can be used as part of the solution of problems where these signals are changed. In contrast, $\xi(t)$ is independent of the initial condition and is the direct consequence of a linear mapping $(r_T, r, u_0) \mapsto \xi$. It must therefore be recomputed if these signals are changed. The next subsection describes an alternative description of the solution in a special case of interest.

Finally, these solutions have a number of interesting properties when $F = 0$,

1. As $K(T)$ is symmetric, then $K(t)$ is symmetric for all $t \in [0, T]$ with $K(t') \geq K(t)$ whenever $t' \leq t$.
2. If $u_0(t) \equiv 0$ and $r(t) \equiv 0$, then the minimum value of the performance index takes the value $x_0^T K(0)x_0$. If, also, the system is time invariant, observable and controllable and both Q and R are constant, symmetric and positive definite, then the limit

$$\lim_{T\to\infty} K(0) = K_\infty \quad (3.104)$$

exists and is the uniquely defined positive definite solution of the *Algebraic Riccati Equation*

$$A^T K_\infty + K_\infty A - K_\infty B R^{-1} B^T K_\infty + Q = 0 \quad (3.105)$$

3.10.5 An Alternative Riccati Representation

The representation defined above provides a causal solution that, in principle, can be used in practice as a feedback/feedforward implementation. It is useful to note however that this representation is not unique. To illustrate this fact suppose that $u_0(t)$ generates an output $y_0(t)$ from the system from the same initial condition. That is, if the associated state vector is denoted by $x_0(t)$ solves the state equations

$$\begin{aligned} \dot{x}_0(t) &= A(t)x_0(t) + B(t)u_0(t), \quad x_0(0) = x_0, \\ \textit{and} \quad y_0(t) &= C(t)x_0(t), \end{aligned} \tag{3.106}$$

then, defining $\delta x(t) = x(t) - x_0(t)$, the system dynamics can be written in the form

$$\begin{aligned} \dot{\delta x}(t) &= A(t)\delta x(t) + B(t)(u(t) - u_0(t)), \quad \delta x_0(0) = 0, \\ \textit{and} \quad y(t) - y_0(t) &= C(t)\delta x(t). \end{aligned} \tag{3.107}$$

Examining the performance index

$$J(u) = \|(r_f, r) - (y_f, y)\|^2_{\mathscr{Y}} + \|u - u_0\|^2_R \tag{3.108}$$

indicates that the change in the model description does not change the solution of the original optimization problem if the following replacements are made

$$\begin{aligned} x(t) \;\mapsto\; x(t) - x_0(t)\,, \quad & u(t) \;\mapsto\; u(t) - u_0(t) \\ u_0(t) \;\mapsto\; 0\,, \quad & r(t) \;\mapsto\; r(t) - y_0(t) \\ \textit{and} \quad & r_T \;\mapsto\; r_T - y_0(T). \end{aligned} \tag{3.109}$$

The representation of $p(t)$ now takes the form

$$p(t) = -K(t)(x(t) - x_0(t)) + \xi(t). \tag{3.110}$$

Using the methodology of the last subsection then leads to the choice of $K(t)$ as the same solution of the Riccati equation but that the definition of $\xi(t)$ changes to the solution of the differential equation in $\mathscr{R}^n$,

$$\begin{aligned} \dot{\xi}(t) &= -\left(A^T(t) - K(t)B(t)R^{-1}(t)B^T(t)\right)\xi(t) - C^T(t)Q(t)(r(t) - y_0(t)), \\ \textit{with} \quad \xi(T) &= C^T(T)F(r_T - y_0(T)). \end{aligned} \tag{3.111}$$

In contrast to the previous identification of $\xi(t)$ as being generated by a linear map $(r_T, r, u_0) \mapsto \xi$, this choice is a linear map $(r_T - y_0(T), r - y_0) \mapsto \xi$. If $r_T = r(T)$, then the map is a map $r - y_0 \mapsto \xi$.

3.11 Further Reading and Bibliography

The introductory material in this chapter is covered, in part and in some form, by many texts from undergraduate texts introducing state space methods to more advanced texts where the details of the mathematics is covered more fully (see Sect. 1.5). These texts also tend to provide sufficient material on both Laplace and $\mathscr{Z}$-transforms to satisfy almost all engineering applications. However, this chapter has focussed on a presentation that associates dynamical models with linear operators between a real Hilbert spaces of input signals and a real Hilbert space of output signals. This point of view is conceptually similar to the use of transfer function matrices in defining input output relationships and, at the algebraic level, operators can be visualized as transfer functions operating in the time domain. The algebraic similarity is superficial however and does not change the need to use and understand many aspects of more classical systems descriptions including transfer function matrices, poles and zeros and frequency response relationships. These concepts are similar in the cases of continuous and discrete time systems. Although the differences are small, the reader should be careful to note the differences and their effects on mathematical relationships when they occur.

The analysis of invertibility is complex when examined in fuller detail [16, 103], particularly when combined with notions of geometric theory. Some knowledge of these ideas is useful in clarifying details of the theoretical ideas. In particular, the relationship of the maximal $\{A, B\}$ invariant subspace in the kernel of C [75] to system zeros and to state space geometry and, from there, to representations of the inverse system is very useful. In following chapters, the case of full rank matrix D provides a link between left and right system inverses and the existence of left and right inverses of the matrices appearing in the supervector descriptions of the next chapter. It is hence a formal link between transfer function matrix descriptions, state space models and dynamics on finite time intervals.

Many control theory texts (Sect. 1.5) that include material on state space design methods include material on the properties of controllability and observability and the link to the ideas of pole allocation. They often include introductory material on optimal control [9, 11, 20, 21, 24, 30], particularly linear quadratic optimal control using either the Calculus of Variations, Dynamic Programming or the Maximum Principle of Pontriagin. Some texts provide considerable information on the properties of the Riccati equation that generates the Riccati matrix $K(t)$. The properties of $K(t)$ and K_∞ have been extensively studied [62] and form an object of study in their own right [1]. The problem definition used in this chapter (Sect. 3.10) fits into these approaches and a knowledge of these methods could be useful to some readers. The approach preferred in this chapter (and the remainder of the text) uses the operator methodology to demonstrate the generality of the ideas and to exploit the geometry of the signal spaces to propose and analyse algorithms. In this sense the representation is in the spirit of texts such a Luenberger [69], Collatz [28] and Ortega and Rheinbolt [79]. It is interesting to note the central role of adjoint operators and their properties

in the construction of the solutions to the Linear Quadratic Optimal Control problem. The geometry of Hilbert spaces and the computation of adjoint operators are core tools to these descriptions and to what follows in later chapters.

Chapter 4
Matrix Models, Supervectors and Discrete Systems

Let $S(A, B, C, D)$ be an ℓ-input, m-output, state dimension n, linear, time-invariant, discrete, state space model of a dynamical system written in the form

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \quad x(0) = x_0, \quad t = 0, 1, 2, 3, \ldots, N-1 \\ y(t) &= Cx(t) + Du(t), \quad t = 0, 1, 2, 3, \ldots, N \end{aligned} \tag{4.1}$$

Note: For notational convenience, the matrices Φ_d and Δ_d used in Sect. 3.5 will now be denoted by A and B. No confusion should occur as the underlying continuous state space model is not used in what follows.

4.1 Supervectors and the Matrix Model

The state space recursion generates output time series $\{y(t)\}_{t\geq 0}$ from input time series $\{u(t)\}_{t\geq 0}$. The relationship between these times series can be represented by a single equation using the idea of "stacking" vectors into output (respectively, input) *supervectors* as follows

$$y = \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} \in \mathscr{R}^{m(N+1)}, \quad u = \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N) \end{bmatrix} \in \mathscr{R}^{\ell(N+1)} \tag{4.2}$$

of dimension $m(N+1)$ (respectively $\ell(N+1)$). Defining the block, lower triangular $m(N+1) \times \ell(N+1)$ matrix

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_4

$$G(A,B,C,D) = \begin{bmatrix} D & 0 & \cdots & 0 & 0 & 0 \\ CB & D & \cdots & 0 & 0 & 0 \\ CAB & CB & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ CA^{N-2}B & CA^{N-3}B & \cdots & CB & D & 0 \\ CA^{N-1}B & CA^{N-2}B & \cdots & CAB & CB & D \end{bmatrix}, \tag{4.3}$$

and the vector

$$d(C,A,x_0) = \begin{bmatrix} Cx_0 \\ CAx_0 \\ CA^2x_0 \\ \vdots \\ CA^Nx_0 \end{bmatrix} \in \mathscr{R}^{m(N+1)} \tag{4.4}$$

then $d(C, A, x_0)$ represents the effect of initial conditions whilst the $m(N+1) \times \ell(N+1)$ matrix $G(A, B, C, D)$ represents the state space model as a linear (matrix) mapping between the supervectors of the input times series and output time series. The equations of motion now have the simple, linear matrix form

$$y = Gu + d. \tag{4.5}$$

Note: For notational simplicity, the arguments in the matrix symbols will be suppressed if this leads to no confusion. Any possible confusion should be resolved by the context.

In principle, this matrix form releases the full power of matrix algebra for application to analysis and design of discrete dynamical systems on finite time intervals. It also places the model clearly into the framework of operator descriptions of system dynamics used later in the text. However, the matrices involved typically have very large dimension as the time series length $N + 1$ is almost always long. As a consequence, the direct use of matrix algebra in computation may be limited in practice. Note however that the equation is exactly equivalent to the original state space recursion equations so an output time series y described by $y = Gu + d$ is best computed using $S(A, B, C, D)$ and simulation. This matrix description does, however, have great theoretical value and possesses a number of other properties, some of which are summarized in the following sections.

4.2 The Algebra of Series and Parallel Connections

The reader will be familiar with the use of transfer functions and series and parallel connections of linear time-invariant systems. The ideas transfer quite naturally to the use of supervectors with one simple qualification. More precisely, for models on the interval $t = 0, 1, 2, \ldots, N$,

1. if $S(A_1, B_1, C_1, D_1)$ is $m \times \ell$, has input $u(t)$, output $\hat{y}(t)$ and transfer function matrix $G_1(z)$ and $S(A_2, B_2, C_2, D_2)$ is $m \times \ell$, has input $u(t)$ and output $\tilde{y}(t)$ and transfer function matrix $G_2(z)$, then the parallel connection relating $u(t)$ to the output $y(t) = \hat{y}(t) + \tilde{y}(t)$ has a supervector representation with matrix

$$G(A_1, B_1, C_1, D_1) + G(A_2, B_2, C_2, D_2). \tag{4.6}$$

   This formula is the analogue of the formula $G_1(z) + G_2(z)$ in the transfer function matrix description.
2. If $S(A_1, B_1, C_1, D_1)$ is $m \times r$, has input $v(t)$, output $y(t)$ and transfer function matrix $G_1(z)$ and $S(A_2, B_2, C_2, D_2)$ is $r \times \ell$, has input $u(t)$, output $v(t)$ and transfer function matrix $G_2(z)$, then the series connection relating $u(t)$ to the output $y(t)$ has a supervector representation with matrix computed from the matrix product

$$G(A_1, B_1, C_1, D_1)G(A_2, B_2, C_2, D_2). \tag{4.7}$$

   This formula is the analogue of the formula $G_1(z)G_2(z)$ in the transfer function matrix description.

The above shows that the algebraic properties of transfer function matrices are inherited by the matrix descriptions using supervectors. The qualification that needs to be stated is simply that care is needed to ensure the compatibility of the time series supervectors when related to the underlying time interval. This is not a problem if the original supervector model described in Sect. 4.1 is used for each subsystem as the underlying time series and time intervals are unmodified. Problems could occur however if, for example, $G(A_1, B_1, C_1, D_1)$ and $G(A_2, B_2, C_2, D_2)$ are constructed from different operations using relative degree concepts (see Sect. 4.4).

4.3 The Transpose System and Time Reversal

Time reversal for a time series $f = \{f(0), f(1), \ldots, f(N)\}$ with terms in $f(t) \in \mathscr{R}^p$ is the replacement of f by the time series $\{f(N), f(N-1), \ldots, f(0)\}$ where the order of the terms is reversed. Writing f in supervector form, the reverse time supervector can be written as $\mathscr{T}(p, N)f$ where $\mathscr{T}(p, N)$ is the square, $p(N+1) \times p(N+1)$ *matrix time reversal operator*

$$\mathscr{T}(p, N) = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & I_p \\ 0 & 0 & \cdots & 0 & I_p & 0 \\ 0 & 0 & \cdots & I_p & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & I_p & \cdots & 0 & 0 & 0 \\ I_p & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}. \tag{4.8}$$

The proof of the following theorem is left as an exercise in matrix manipulation for the reader

Theorem 4.1 (Properties of the Time Reversal Matrix) *The time reversal matrix has the properties:*

1. *Invariance under transposition and inversion,*

$$\mathscr{T}(p,N) = \mathscr{T}^T(p,N), \;\; (\mathscr{T}(p,N))^2 = I_{p(N+1)} \;\; and \;\; (\mathscr{T}(p,N))^{-1} = \mathscr{T}(p,N). \tag{4.9}$$

2. *The following identity relates the transpose of $G(A,B,C,D)$ to time reversal operators and the "transpose system" $S(A^T,C^T,B^T,D^T)$*

$$G^T(A,B,C,D) = \mathscr{T}(\ell,N)G(A^T,C^T,B^T,D^T)\mathscr{T}(m,N). \tag{4.10}$$

3. *If $S(A,B,C,D)$ has a transfer function matrix $G(z)$, then $S(A^T,C^T,B^T,D^T)$ has transfer function matrix $G^T(z)$.*
4. *If v is the supervector corresponding to a time series $\{v(0),v(1),\ldots,v(N)\}$ in $\mathscr{R}^m$, then the computation of the time series in $\mathscr{R}^\ell$ generated by the supervector $G^T(A,B,C,D)v$ can be expressed as*

$$G^T(A,B,C,D)v = \mathscr{T}(\ell,N)\left(G(A^T,C^T,B^T,D^T)\left(\mathscr{T}(m,N)v\right)\right). \tag{4.11}$$

That is, the time series in $\mathscr{R}^\ell$ represented by $G^T(A,B,C,D)v$ is the time reversal of the response of the system $S(A^T,C^T,B^T,D^T)$ from zero initial conditions to the time reversal of the time series v.

Finally, note that the time reversal operators will often be simply denoted $\mathscr{T}$ with the dimensions understood by the context of the discussion.

4.4 Invertibility, Range and Relative Degrees

It is left as an exercise for the reader to use the structure of $G(A,B,C,D)$ to verify that,

1. the kernels of $G(A,B,C,D)$ and D are closely related via a rank condition as follows,

$$ker[G] = \{0\} \;\; if, \; and \; only \; if, \;\; rank\,[D] \;=\; \ell \;\; (i.e. \;\; ker[D] = \{0\})\,. \tag{4.12}$$

This is never the case if $\ell > m$ and may not be the case if $\ell \leq m$. It excludes strictly proper systems immediately (as $D = 0$) with the physical interpretation that $u(N)$ has no effect whatsoever on any point $y(0), y(1), \ldots, y(N)$ of the output time series on the interval $0, 1, 2, 3, \ldots, N$.

In dynamical systems terms, $ker[G] \neq \{0\}$ if, and only if, there exists a non-zero input time series that produces zero output time series from the system with zero initial conditions. If $ker[G] = \{0\}$ (respectively $ker[G] \neq \{0\}$) for some value of N, then it is also true for all other values of N. That is, changing the length of the time series retains the property.

2. The range of $G(A, B, C, D)$ need not be the whole of $\mathscr{R}^{m(N+1)}$. More precisely,

$$\mathscr{R}[G] = \mathscr{R}^{m(N+1)} \quad if, \ and \ only \ if, \quad rank\,[D] = m \quad \left(i.e. \ \mathscr{R}[D] = \mathscr{R}^{m}\right). \tag{4.13}$$

This cannot be satisfied if $m > \ell$ but also may not be satisfied if $m \leq \ell$. Again, strictly proper systems do not have a full range as $y(0)$ cannot be influenced by the input time series.

In dynamical systems terms, if the range of $G(A, B, C, D)$ is a proper vector subspace of $\mathscr{R}^{m(N+1)}$, there are output time series that cannot be produced by any input time series from zero initial conditions. Finally, $\mathscr{R}[G] = \mathscr{R}^{m(N+1)}$ for some N if, and only if, it is also true for all values of N.

3. More generally, if $S(A, B, C, D)$ has relative degree k^*, then the outputs $y(t)$, $0 \leq t \leq k^* - 1$, are unaffected by the input sequence. Also the inputs $u(t)$ on the discrete interval $N + 1 - k^* \leq t \leq N$ have no effect on the output sequence on $0, 1, 2, \ldots, N$.
4. The system matrix $G(A, B, C, D)$ and its upper, block triangular "transpose system" matrix G^T have range and kernels related by the properties

$$\begin{aligned} &\mathscr{R}[G(A^T, C^T, B^T, D^T)]^{\perp} = \mathscr{T}(\ell, N)ker[G(A, B, C, D)] \\ and \ \ &\mathscr{T}(m, N)ker[G(A^T, C^T, B^T, D^T)] = \mathscr{R}[G(A, B, C, D)]^{\perp} \end{aligned} \tag{4.14}$$

where $X^{\perp}$ denotes the orthogonal complement of a vector subspace X with respect to the Euclidean inner product $\langle x, z\rangle = x^T z$.

5. The full rank condition on D links the supervector description to the ideas of inverse systems. More precisely

   a. If $m = \ell$, and D is nonsingular, then the inverse system $G^{-1}(z)$ is proper with a supervector description incorporating the matrix

$$G^{-1}(A, B, C, D) = G(A - BD^{-1}C, \ BD^{-1}, \ -D^{-1}C, \ D^{-1}). \tag{4.15}$$

   b. Similar observations apply to the cases of $m \neq \ell$ using Theorems 3.4 and 3.5 which provide examples of dynamical systems with left and right inverses in dynamical system form. The simple algebraic properties of series combinations of systems then indicate that their supervector descriptions satisfy the left and right (matrix) inverse properties illustrated by the fact that (from Theorem 3.4) the product

$$\underbrace{G(A-B(KD)^{-1}KC, B(KD)^{-1}K, -(KD)^{-1}KC, (KD)^{-1}K)}_{Left\ \ Inverse\ \ of\ \ G(A,B,C,D)}\, G(A,B,C,D) \tag{4.16}$$

is equal to the identity $I_{\mathscr{R}^{\ell(N+1)}}$ whenever $rank[D] = \ell$ and KD is nonsingular. It is left as an exercise for the reader to use Theorem 3.5 to deduce a right inverse

$$\underbrace{G(A-BK(DK)^{-1}C, BK(DK)^{-1}, -K(DK)^{-1}C, K(DK)^{-1})}_{Right\ \ Inverse\ \ of\ \ G(A,B,C,D)} \tag{4.17}$$

when $rank[D] = m$ and DK is nonsingular.

In general terms, if the supervector description of dynamics is used on a finite interval of time, a loss of rank for D implies that there is redundancy in the input supervector and/or the existence of output sequences that cannot be achieved by any input times series. A full rank D removes these problems and simultaneously ensures the existence of left and/or right inverses generated by dynamical systems. Range and kernel properties of the system can be related precisely to those of the transpose system using orthogonality conditions.

Under some circumstances, the kernel and range properties can be simplified by redefining the supervectors and the associated matrices G and d to eliminate some or all of the redundancy. The basic idea used in this text is to replace the system $S(A, B, C, D)$ by a system $S(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$ with essentially the same input output behaviours but improved conditioning of $G(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$. In the following sections, some techniques for achieving this are described using the relative degree, modifications suggested by decoupling theory and, for square, invertible systems, techniques using a decomposition of the inverse transfer function matrix and a technique based on the $\mathscr{C}^*$-canonical form.

4.4.1 The Relative Degree and the Kernel and Range of G

Let k^* be the relative degree of the discrete system $S(A, B, C, D)$. If, $k^* = 0$ (that is, $D \neq 0$) and $ker[D] = \{0\}$, then $ker[G(A, B, C, D)] = \{0\}$. Similarly if, $k^* = 0$ and $\mathscr{R}[D] = \mathscr{R}^m$, then $\mathscr{R}[G(A, B, C, D)] = \mathscr{R}^{m(N+1)}$.

More generally, if $k^* \geq 1$, replace the supervector y by y^e (of lower dimension) as follows

$$y^e = \begin{bmatrix} y(k^*) \\ y(k^*+1) \\ \vdots \\ y(N) \end{bmatrix} \in \mathscr{R}^{m(N+1-k^*)}, \quad u = \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N-k^*) \end{bmatrix} \in \mathscr{R}^{\ell(N+1-k^*)} \tag{4.18}$$

of dimension $m(N+1-k^*)$ (respectively $\ell(N+1-k^*)$. Note the replacement of the output sequence by a shifted sequence $y^e(t') = y(k^*+t')$ and the replacement of the time interval of length $N+1$ by one of length N^*+1 with $N^* = N-k^*$. The matrix G relating u to y^e is seen, using the lower block triangular structure of $G(A, B, C, D)$, to be

$$\begin{bmatrix} CA^{k^*-1}B & 0 & \cdots\cdots & 0 & 0 & 0 \\ CA^{k^*}B & CA^{k^*-1}B & \cdots\cdots & 0 & 0 & 0 \\ CA^{k^*+1}B & CA^{k^*}B & \cdots\cdots & 0 & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ CA^{N-2}B & CA^{N-3}B & \cdots\cdots & CA^{k^*}B & CA^{k^*-1}B & 0 \\ CA^{N-1}B & CA^{N-2}B & \cdots\cdots & CA^{k^*+1}B & CA^{k^*}B & CA^{k^*-1}B \end{bmatrix}, \tag{4.19}$$

which can be written in the form $G(A, B, CA^{k^*}, CA^{k^*-1}B)$ with initial condition term

$$d(CA^{k^*}, A, x_0) = \begin{bmatrix} CA^{k^*}x_0 \\ CA^{k^*+1}x_0 \\ CA^{k^*+2}x_0 \\ \vdots \\ CA^{N}x_0 \end{bmatrix} \in \mathscr{R}^{m(N+1-k^*)}. \tag{4.20}$$

The equivalent state space model is just $S(A, B, CA^{k^*}, CA^{k^*-1}B)$ with the same state vector as $S(A, B, C, D)$ and $m \times \ell$ transfer function matrix $z^{k^*}G(z)$ and initial condition $x(0) = x_0$. This model has a "D" term equal to $CA^{k^*-1}B$ and it is immediately concluded that the new representation has $ker[G(A, B, CA^{k^*}, CA^{k^*-1}B)] = \{0\}$ if, and only if, $ker[CA^{k^*-1}B] = \{0\}$. The relevant necessary and sufficient condition for $\mathscr{R}[G(A, B, CA^{k^*}, CA^{k^*-1}B)] = \mathscr{R}^{m(N+1-k^*)}$ is that $\mathscr{R}[CA^{k^*-1}B] = \mathscr{R}^m$.

4.4.2 The Range of G and Decoupling Theory

The modified supervector model introduced above covers many situations and indicates, for example, that, if $m = \ell, k^* \geq 1$ and the $m \times m$ matrix $CA^{k^*-1}B$ is nonsingular, then the $m(N+1-k^*) \times m(N+1-k^*)$ square matrix $G(A, B, CA^{k^*}, CA^{k^*-1}B)$ is nonsingular.

This construction covers the SISO case completely but, for MIMO systems, there remains the possibility that the first non-zero Markov parameter matrix will not have full rank. One situation where full rank conditions can be recovered is that when the strictly proper discrete system $S(A, B, C)$ can be *decoupled using state feedback*. More precisely, suppose that $m \leq \ell$. Then the system can be decoupled using state feedback if, and only if, there exists a state feedback control law

$$u(t) = Kv(t) - Fx(t), \quad t = 0, 1, 2, 3, \ldots, \tag{4.21}$$

with $\ell \times m$ matrix K of rank m, $\ell \times n$ state feedback matrix F and new input vector $v(t) \in \mathscr{R}^m$ such that the resultant m-input, m-output state space model

$$\begin{aligned} x(t+1) &= (A - BF)x(t) + BKv(t), \quad x(0) = x_0 \\ y(t) &= Cx(t) \end{aligned} \tag{4.22}$$

has an $m \times m$ transfer function matrix $G(zI - A + BF)^{-1}BK$ which is diagonal at all values of z. The physical interpretation of this construction is that each individual element of the new input $v(t)$ affects its corresponding output element only. *The system is said to have been decoupled*. MIMO systems that can be decoupled in this way can then be controlled by regarding each loop as a separate, non-interacting SISO control loop. Control design for such systems is well established and understood and is not pursued further here.

The relevance of the concept of decoupling to the supervector description follows from the known necessary and sufficient condition for decoupling. More precisely, for a system $S(A, B, C)$, identify the m separate rows $C_1, C_2, \ldots, C_m$ of C and the unique integers k_i^*, $1 \leq i \leq m$ (the *row relative degrees*) as follows

$$C = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{bmatrix}, \quad C_i A^{j-1} B = 0, \ 1 \leq j \leq k_i^* - 1, \ \ C_i A^{k_i^* - 1} B \neq 0\,, \ \ 1 \leq i \leq m. \tag{4.23}$$

With this parameterization, the system can be decoupled using state feedback if, and only if, the $m \times \ell$ matrix

$$G_\infty = C_\infty B, \quad where \quad C_\infty = \begin{bmatrix} C_1 A^{k_1^* - 1} \\ C_2 A^{k_2^* - 1} \\ \vdots \\ C_m A^{k_m^* - 1} \end{bmatrix}, \tag{4.24}$$

has rank equal to m . It is a simple matter (a) to use the transfer function matrix $G(z) = C(zI - A)^{-1}B$ to identify k_j^* as the relative degree of the jth row $C_j(zI - A)^{-1}B$ of $G(z)$ and (b) to verify that

$$G_\infty = \lim_{|z|\to\infty} \left(diag[z^{k_1^*}, z^{k_2^*}, \ldots, z^{k_m^*}] G(z) \right) \tag{4.25}$$

The test for decoupling can hence be done using the transfer function matrix alone.

It is now a matter of algebra, using the properties of the Markov parameter matrices of the rows of $G(z)$, to show that the system $S(A, B, C_\infty A, G_\infty)$ with state space model

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \quad x(0) = x_0 \\ y^e(t) &= C_\infty A x(t) + G_\infty u(t) \end{aligned} \tag{4.26}$$

has transfer function matrix

$$diag[z^{k_1^*}, z^{k_2^*}, \ldots, z^{k_m^*}] G(z) \; . \tag{4.27}$$

This model has a physical interpretation as being the original system with input u and a new output vector y^e constructed from y using shifted real outputs as follows

$$y^e(t') = \begin{bmatrix} y_1(t' + k_1^*) \\ y_2(t' + k_2^*) \\ \vdots \\ y_m(t' + k_m^*) \end{bmatrix}, \quad t' \geq 0. \tag{4.28}$$

The condition $rank[G_\infty] = m$ ensures that the matrix $G(A, B, C_\infty A, G_\infty)$ has range $\mathscr{R}^{m(N^*+1)}$ for any "time" interval $0 \leq t' \leq N^*$. A technical issue arises here, namely that any choice of interval $0 \leq t' \leq N^*$ for y^e corresponds to different underlying time intervals for each element of y. For example, the interval $0 \leq t' \leq N^*$ for y_j describes behaviour on the underlying time interval $k_j^* \leq t \leq N^* + k_j^*$, an interval that differs for different values of k_j^*.

There is hence a choice to be made if the modified model $G(A, B, C_\infty A, G_\infty)$ is to effectively describe dynamics of $S(A, B, C)$ on the whole interval $0 \leq t \leq N$. A simple choice that ensures that the resultant time interval covers output behaviour on the original interval $0 \leq t \leq N$ for y is to let $y^e(t')$ vary on $t' = 0, 1, 2, 3, \ldots, N^*$ with

$$N^* = N - k_{min}^* \qquad (where \quad k_{min}^* = \min\{k_1^*, k_2^*, \ldots, k_m^*\}). \tag{4.29}$$

If all k_j^* are identical in value (the so-called *uniform rank case*), then this causes no problem. In all other cases, the sequence $y^e(t')$, $0 \leq t' \leq N^*$ contains values of the original output outside the interval $0, 1, 2, \ldots, N$. This is easily demonstrated by examining the final value of $y^e(t')$ on $0, 1, 2, \ldots, N^*$, namely,

$$y^e(N^*) = \begin{bmatrix} y(N + k_1^* - k_{\min}^*) \\ y(N + k_2^* - k_{\min}^*) \\ \vdots \\ y(N + k_m^* - k_{\min}^*) \end{bmatrix} \tag{4.30}$$

which clearly contains values of $y(t)$ outside the original interval $0, 1, 2, \ldots, N$ unless all k_j^* are the same (and hence equal to $k_{\min}^*$). If this underlying modification is accepted, then the objective of ensuring that $G(A, B, C_\infty A, G_\infty)$ has range equal to the whole of $\mathscr{R}^{m(N^*+1)}$ has been achieved. As a by-product, the reader will note that, if $m = \ell$, then G_∞ is square and nonsingular and the objective of ensuring that the kernel is $\{0\}$ has also been achieved.

Note: In the case of $\ell \leq m$ the reader may wish to explore the parallel idea of column relative degree and transformation of the input to ensure that the kernel becomes $\{0\}$. *The simplest approach is to apply the theory described above to the transpose system* $S(A^T, C^T, B^T, D^T)$.

4.5 The Range and Kernel and the Use of the Inverse System

The use of decoupling concepts extends the idea of relative degree and provides useful range and kernel properties of G that reduce the redundancy present in the supervector model. Unfortunately, although this construction covers almost all cases in practice, there is still the mathematical possibility that G_∞ loses rank (that is, $S(A, B, C)$ cannot be decoupled by state feedback). The technique cannot then be used. A more general approach for the case of $m = \ell$ uses the idea of *inversion of a subsystem* generated by a partition of the inverse transfer function matrix.

4.5.1 A Partition of the Inverse

Let $m = \ell$ and suppose that $D = 0$. Suppose that $S(A, B, C)$ is controllable and observable and invertible in the sense that

$$\det[G(z)] \neq 0 \tag{4.31}$$

except at a finite number of points (which are zeros of the system). It follows that $G^{-1}(z)$ exists and has elements that are rational polynomials in the complex variable z. As the system is strictly proper, the inverse has the form

$$G^{-1}(z) = P(z) + H_0(z) \tag{4.32}$$

where $P(z)$ is assumed to have elements that are polynomials in the complex variable z. In this scenario, it is always possible to choose H_0 to be strictly proper, from which the resultant $P(z)$ has the properties that

1. its elements are polynomials in z,
2. $P^{-1}(z)$ exists and is strictly proper and
3. the product $H(z) = H_0(z)P^{-1}(z)$ is strictly proper.

If, however, the requirement that H_0 is strictly proper is removed, many more pairs (P, H_0) exist with these three properties. For example, consider the transfer function with inverse

$$G^{-1}(s) = s^2 + 2s + 1 + \frac{4}{s+3}, \tag{4.33}$$

then suitable $P(s)$ are written $P(s) = s^2 + as + b$ for any scalars a, b.

In all such cases, the system transfer function matrix hence has the form

$$G(z) = (P(z) + H_0(z))^{-1} = P^{-1}(z)(I + H_0(z)P^{-1}(z))^{-1} \tag{4.34}$$

which, for any nonsingular matrix P_0, is the product of a strictly proper system $P^{-1}(z)P_0$ and a proper system $H_1(z) = P_0^{-1}(I + H_0(z)P^{-1}(z))^{-1}$ of uniform rank $k^* = 0$. With this construction, the system model has two components

$$y^e = H_1 u, \quad \textit{and} \quad y = P^{-1}P_0 y^e \tag{4.35}$$

the first of which has a supervector representation with a corresponding nonsingular matrix H_1. Note that

1. The choice of P is non-unique in general as is the choice of nonsingular matrix P_0. A simple choice of P_0 could be based on ensuring a simple order of magnitude relationship between y and y^e. For example, choosing $P_0 = P(1)$ provides a simple equivalence between the two signals at low frequency.
2. The equation $y = P^{-1}P_0 y^e$ is a dynamic relationship between y and y^e that permits computation of the time series of y from that of y^e. Conversely, writing $y^e = P_0^{-1}Py$ inverts this relationship and identifies y^e as a vector of forward shifted values of y. The precise composition of this forward shifted version depends on the precise nature of the polynomial matrix $P(z)$ and, as in the use of the "relative degree" methodologies of the previous sections, leads to a similar need to reconsider the underlying time interval.

An interesting link with the uniform rank $k^* \geq 1$ case is obtained by noting that the nonsingularity of $CA^{k^*-1}B$ indicates that it is always possible to choose $P(z) = z^{k^*}(CA^{k^*-1}B)^{-1}$, a choice that satisfies the required conditions for both $P(z)$ and the consequent $H_0(z)$. Choosing $P_0 = P(1)$ yields the result that $P^{-1}(z)P_0 = z^{-k^*}I$ so that y is just y^e delayed by k^* samples. Also

$$H_1(z) = P_0^{-1}(I + H_0(z)P^{-1}(z))^{-1} = z^{k^*}G(z). \tag{4.36}$$

Note: Similar computations can be undertaken for the case of a system that can be decoupled where the choice is $P(z) = G_\infty^{-1} diag[z^{k_1^*}, \ldots, z^{k_m^*}]$ *and* $P_0 = G_\infty^{-1}$. *The details are left for the reader.*

4.5.2 Ensuring Stability of $P^{-1}(z)$

A potential problem implicit in the use of partitions of the inverse is that the minimal realization of $P^{-1}(z)$ may be unstable. It is natural to ask whether or not it is possible to choose P^{-1} to be stable. This requirement is strengthened here to ask whether or or not P^{-1} can be written as a finite polynomial in z^{-1}. This is equivalent to requiring that all poles take the value $z = 0$. In such cases, the relation between y^e and y takes the "moving average" form

$$y(t) = \sum_{j=q_1^*}^{q_2^*} P_j y^e(t-j) \tag{4.37}$$

where $\{P_j\}$ is a set of coefficient matrices and the integers $q_1^* \leq q_2^*$ are finite.

Theorem 4.2 *Let $m = \ell$ and suppose that $D = 0$. Suppose also that $S(A, B, C)$ is controllable and observable and that $G(z)$ is invertible. Then $G(z)$ can be factorized into the product of two $m \times m$ transfer function matrices $G(z) = P^{-1}(z)H_1(z)$ where $H_1(z)$ is proper with $\lim_{z\to\infty} H_1(z) = I_m$, $P^{-1}(z)$ has no zeros and is a finite polynomial in the complex variable z^{-1} and $P(z)$ is a finite polynomial in the complex variable z. The factorization is not necessarily unique.*

Proof At least one factorization exists if the requirement that $P^{-1}(z)$ is a polynomial in z^{-1} is ignored. Let $S(A_P, B_P, C_P)$ be a minimal realization of such a $P^{-1}(z)$ and let F be a feedback matrix that allocates all eigenvalues of $A_P - B_P F$ to the origin. Such a choice is normally non-unique if $m > 1$. Assume zero initial conditions and write the state equations for $S(A_P, B_P, C_P)$ in the form

$$x_P(t+1) = (A_P - B_P F)x_P(t) + B_P(u(t) + Fx_P(t)). \tag{4.38}$$

It follows that the transfer function matrix is the product of two factors

$$P^{-1}(s) = C_P(zI - A_P + B_P F)^{-1}B_P \left[I + F(zI - A_P)^{-1}B_P\right] \tag{4.39}$$

The first term $C_P(zI - A_P + B_P F)^{-1}B_P$ is strictly proper, has no zeros and only has poles at $z = 0$. It is hence a finite polynomial in z^{-1}. Invertibility follows from the invertibility of P and the invertibility of the second factor. It follows that P^{-1} can be replaced by $C_P(zI - A_P + B_P F)^{-1}B_P$ with the second factor H_1 replaced by a minimal realization of $(I + F(zI - A_P)^{-1}B_P)H_1(z)$. Finally the polynomial nature of $P(z)$ as a matrix with elements that are polynomials in z follows from the fact that $P^{-1}(z)$ has no zeros. □

Finally, the fact that such moving average representations exist does not imply that they are unique. In practice, it would be natural to seek a solution with additional minimality properties such as the "spread" $q_2^* - q_1^*$ should be minimal. For uniform

rank systems of relative degree k^*, the relevant minimal values are $q_1^* = q_2^* = k^*$ and, in this case, no smaller spread is possible.

4.6 The Range, Kernel and the $\mathscr{C}^*$ Canonical Form

When applicable, the use of relative degree concepts can be seen as an effective methodology for identifying "natural delays" $z^{-k_j^*}$ for each output in the system with the aim of producing the required properties for the kernel and range of the matrix in the supervector description relating u to a modified output y^e. The resultant dynamics are defined on a new interval length $0 \leq t' \leq N^*$. More generally, the use of inverse system representations replaces delays by matrices $P(z)$ whose elements are polynomials of finite degree in the complex variable z. The resultant new output signal y^e can then be taken to be a time series of weighted "moving average" values of the original output y over a finite horizon.

The technique explored in this section is to use the idea of the $\mathscr{C}^*$- canonical form of a strictly proper, invertible system $S(A, B, C)$ with $m = \ell$ to show that every such system has a natural set of simple delays $1 \leq k_1^* \leq \cdots \leq k_m^*$ embedded *within* its structure. These delays are revealed by a simple transformation to be the row relative degrees of a system mapping u into a new output $\tilde{y}$. This system satisfies the decoupling condition so $\tilde{y}$ leads naturally to a suitable y^e. The approach has connections to inverse system methods but is approached using state space descriptions.

4.6.1 Factorization Using State Feedback and Output Injection

For any *state feedback matrix* F, the state space model $S(A, B, C)$ can be written

$$x(t+1) = (A - BF)x(t) + B(u(t) + Fx(t)), \quad y(t) = Cx(t). \tag{4.40}$$

Assuming zero initial conditions, the application of the $\mathscr{Z}$-transform gives

$$y(z) = C(zI - A + BF)^{-1}B\,(u(z) + Fx(z))\,. \tag{4.41}$$

Noting that $x(z) = (zI - A)^{-1}Bu(z)$ then indicates that the transfer function matrix $G(z)$ of $S(A, B, C)$ can be written in the form of the product

$$G(z) = C(zI - A + BF)^{-1}B\left[I + F(zI - A)^{-1}B\right] \tag{4.42}$$

Applying the same idea to the transpose of the first term by introducing an *output injection matrix* H yields the algebraic description

$$\text{“}Factorization\ \ One\text{”}\quad \left(\begin{array}{r} G(z) = H_2(z)P^{-1}(z)H_1(z) \\ where P^{-1}(z) = C(zI - A + BF + HC)^{-1}B, \\ H_1(z) = \left[I + F(zI - A)^{-1}B\right], \\ and H_2(z) = \left[I + C(zI - A + BF)^{-1}H\right]. \end{array}\right) \tag{4.43}$$

H_1 has a, possibly non-minimal, realization $S(A, B, F, I)$ whilst H_2 has a, possibly non-minimal, state space realization $S(A - BF, H, C, I)$. The matrices in their supervector representations are nonsingular with unit determinant.

An alternative description starting from the transfer function matrix $G^T(z)$ for the transpose system $S(A^T, C^T, B^T, D^T)$ gives

$$\text{“}Factorization\ \ Two\text{”}\quad \left(\begin{array}{r} G(z) = H_2(z)P^{-1}(z)H_1(z) \\ where \quad P^{-1}(z) = C(zI - A + BF + HC)^{-1}B, \\ H_1(z) = \left[I + F(zI - A + HC)^{-1}B\right], \\ and \quad H_2(z) = \left[I + C(zI - A)^{-1}H\right]. \end{array}\right) \tag{4.44}$$

Here, H_1 has a, possibly non-minimal, realization $S(A-HC, B, F, I)$ whilst H_2 has a, possibly non-minimal, state space realization $S(A, H, C, I)$. Again the matrices in their supervector representations are nonsingular with unit determinant.

The important observation here is that, in both cases, the transfer function matrix $P^{-1}(z)$ has the same form and is strictly proper and invertible whilst both $H_1(z)$ and $H_2(z)$ are proper and have uniform rank zero. For the purposes of discussion, Factorization One is the one of interest and is assumed for the rest of this section.

4.6.2 The $\mathscr{C}^*$ Canonical Form

The $\mathscr{C}^*$ Canonical Form due to Morse is a theoretical description of properties of a matrix triple (A, B, C) under the group of transformations defined by five matrices (P_1, P_2, F, H, T) (with each P_1, P_2 and T nonsingular) and the group action

$$(P_1, P_2, F, H, T) \circ (A, B, C) \ \mapsto\ (T^{-1}(A - BF - HC)T, T^{-1}BP_1, P_2CT) \tag{4.45}$$

Associated with this group action is a canonical form for (A, B, C) describing the state space structure of the system and an associated transfer function matrix. The simplest expression of the relevant implications of the $\mathscr{C}^*$ Canonical Form is that there always exists uniquely defined integers $1 \le k_1^* \le k_2^* \le \cdots \le k_m^*$ (the *so-called structural invariants* of the $\mathscr{C}^*$-transformation group), nonsingular $m \times m$ matrices P_1, P_2, a state feedback matrix F and an output injection matrix H such that

$$P_2P^{-1}(z)P_1 = diag\left[z^{-k_1^*}, z^{-k_2^*}, \ldots, z^{-k_m^*}\right] \tag{4.46}$$

The two matrices F, H are non-unique but all have a number of general properties of interest, namely that,

$$\begin{array}{c}(A-BF)\mathscr{V}^* \subset \mathscr{V}^*, \quad (A-BF-HC)\mathscr{V}^* \subset \mathscr{V}^* \\ C\,(A-BF-HC)^k = 0\,, \quad for\ \ all\ \ k \geq k_m^*,\end{array} \tag{4.47}$$

where $\mathscr{V}^*$ is the maximal $\{A, B\}$-invariant subspace in $ker[C]$ with dimension

$$\dim(\mathscr{V}^*) = n_z = n - \sum_{j=1}^{m} k_j^*. \tag{4.48}$$

Here n_z is the number of zeros of $S(A, B, C)$. One consequence is that,

Theorem 4.3 (H_2^{-1} is a Moving Average Filter) *The inverse transfer function matrix, $H_2^{-1}(z)$ is a matrix with elements that are finite polynomials in the complex variable z^{-1} of the form*

$$\begin{array}{c}H_2^{-1}(z) = \sum_{j=0}^{k_m^*} H_2^{(j)} z^{-j}, \quad H_0^{(0)} = I_m, \\ and\ \ H_2^{(j)} = -C(A-BF-HC)^{j-1}H, \ \ 1 \leq j \leq k_m^*.\end{array} \tag{4.49}$$

Proof The proof is a simple consequence of the observation that $H_2^{-1}(z)$ has a state space model of the form $S(A-BF-HC, H, -C, I)$ with a minimal realization that has poles at the point $z = 0$ only. More precisely, the proof follows using the power series expansion,

$$\begin{aligned} H_2^{-1}(z) &= I - C(zI - A + BF + HC)^{-1}H \\ &= I - \textstyle\sum_{j=1}^{\infty} z^{-j} C(A+BF+HC)^{j-1}H \\ &= I - \textstyle\sum_{j=1}^{k_m^*} z^{-j} C(A+BF+HC)^{j-1}H \end{aligned} \tag{4.50}$$

□

This simple characterization suggests, in particular, that a change in the output definition provides some simplifications to the general case:

Theorem 4.4 (Embedded Delays and Decoupling) *Assuming zero initial state conditions,*

1. *The signal $P_2 H_2^{-1} y$ is a moving average filtered version of y,*
2. *the elements in the signal $P_2 H_2^{-1} y$ and the modified input $P_1^{-1} H_1 u$ are related in a simple way by the time shifts defined by the integers $\{k_j^*\}_{1 \leq j \leq m}$.*
3. *A link to decoupling characterizations is obtained through the fact that the state space model defining the mapping $u \mapsto P_2 H_2^{-1} y$ is a dynamical system with row relative degrees $\{k_j^*\}_{1 \leq j \leq m}$ which can be decoupled by state feedback.*

Proof The first statement follows from the preceding theorem. The second statement follows from the factorization of $G(z) = H_2(z) P_2^{-1} \left[P_2 P^{-1}(z) P_1 \right] P_1^{-1} H_1(z)$. The final statement follows as P_1 is nonsingular and

$$\lim_{|z|\to\infty} diag[z^{k^*}, \ldots z^{k_m^*}] P_2 H_2^{-1}(z) G(z) = P_1^{-1}. \tag{4.51}$$

□

The natural interpretation of this result is that a suitable moving average filtering of the output y yields a new output $\tilde{y}$ for which the decoupling based results described previously can be used. The filter $P_2 H_2^{-1}(z)$ used in the above theorems is a candidate for such a filter. This choice is non-unique as (a) the values of F and H are non-unique and (b), for a given pair (F, H), a simple calculation proves that scaling of the rows of P_2 retains the essential properties described provided that the columns of P_1 are scaled by the inverse scaling factors.

A (non-unique) characterization of the output y^e required is as the output response of a minimal realization of the state space model $S(A, B, P_1^{-1}F, P_1^{-1})$ of the system $P_1^{-1}H_1(z)$ to the input u. It is seen that y^e is related to the original output y by the $\mathscr{Z}$-transform relation

$$y^e(z) = M(z)y(z) \quad where \quad M(z) = diag[z^{k_1^*}, \ldots, z^{k_m^*}] P_2 H_2^{-1}(z). \tag{4.52}$$

The process can be regarded as a two step procedure $y \mapsto \tilde{y} \mapsto y^e$ using the time series representations

$$\tilde{y}(t) = P_2 \sum_{j=0}^{k_m^*} H_2^{(j)} y(t-j) \quad and \quad y^e(t) = \begin{bmatrix} \tilde{y}(t+k_1^*) \\ \vdots \\ \tilde{y}(t+k_m^*) \end{bmatrix}, \quad t \geq 0. \tag{4.53}$$

Computational Note: For practical purposes, it is natural to choose a scaling to avoid any possible numerical/computational problems arising from the wide range of gain from steady state to high frequency. An intuitive choice is a scaling that ensures that the value taken by $P_1^{-1}H_1(z)$ at $z = 1$, that is $P_1^{-1}[I + F(I - A)^{-1}B])$, has elements with similar magnitude to those in $G(1)$. This would, intuitively, ensure similar steady state behaviour from both y and y^e.

4.6.3 The Special Case of Uniform Rank Systems

For the special case of a uniform rank k^* system where $CA^{j-1}B = 0$ for $1 \leq j < k^*$ and $CA^{k^*-1}B$ is nonsingular, it is true that $k_j^* = k^*$ for all $j = 1, 2, \ldots, m$. The values of (F, H, P_1, P_2) are non-unique.

Example One: A candidate set can be computed from the following direct sum decomposition of the state space,

$$\mathscr{R}^n = \mathscr{V}^* \oplus \mathscr{R}[B] \oplus \mathscr{R}[AB] \oplus \cdots \oplus \mathscr{R}[A^{k^*-1}B] \tag{4.54}$$

with $n_z = n - k^*m$ and noting that

$$\mathscr{V}^* = ker[\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{k^*-1} \end{bmatrix}]. \tag{4.55}$$

In this case, if $\mathscr{V}_b^*$ is an $n \times n_z$ matrix whose columns span $\mathscr{V}^*$, the matrix

$$T = \left[B, AB, \ldots, A^{k^*-1}B, \mathscr{V}_b^*\right], \tag{4.56}$$

is nonsingular. Analysis of the form of the state space model in this basis leads to the following formulae defining candidates for the required F, H, P_1 and P_2

$$\begin{aligned} H = A^{k^*}B(CA^{k^*-1}B)^{-1}, \qquad & F = (CA^{k^*-1}B)^{-1}C(A - HC)^{k^*}, \\ P_2 = I \quad and \quad & P_1 = (CA^{k^*-1}B)^{-1}. \end{aligned} \tag{4.57}$$

from which a state space realization of $P_1^{-1}H_1$ and the relevant coefficient matrices $\{H_2^{(j)}\}$ in the inverse H_2^{-1} can be computed.

Example Two: Applying the procedure above to $S(A^T, C^T, B^T)$ suggests the choice of

$$\begin{aligned} F = (CA^{k^*-1}B)^{-1}CA^{k^*}, \qquad & H = (A - BF)^{k^*}B(CA^{k^*-1}B)^{-1}, \\ P_1 = I \quad and \quad & P_2 = (CA^{k^*-1}B)^{-1}. \end{aligned} \tag{4.58}$$

Example Three: The choices in Example Two can be simplified by replacing H by $H = 0$ and hence H_2 by the identity I_m. Although this no longer represents the $\mathscr{C}^*$-canonical form, this choice retains the required transfer function matrix property. More precisely,

$$G(z) = C(zI_n - A + BF)^{-1}B\left[I_m + F(zI - A)^{-1}B\right] \tag{4.59}$$

and

$$C(zI_n - A + BF)^{-1}B = z^{-k^*}CA^{k^*-1}B \tag{4.60}$$

as a simple calculation yields the observation that

$$\begin{gathered} C(A - BF)^{k^*} = 0, \\ C(A - BF)^{j-1}B = 0, \;\; 1 \le j < k^* \;\; and \;\; C(A - BF)^{k^*-1}B = CA^{k^*-1}B. \end{gathered} \tag{4.61}$$

This construction leads to the familiar choice of $y^e(t) = y(t + k^*)$ and the transfer function matrix $P_1^{-1}H_1(z) = CA^{k^*-1}B(I + F(zI - A)^{-1}B) = z^{k^*}G(z)$.

4.7 Quadratic Optimal Control of Linear Discrete Systems

Section 3.10 provided an introduction to operator based approaches to the solution of a continuous time linear quadratic optimal control problem. A similar problem for the ℓ-input, m-output, linear, time invariant, discrete time system

$$\begin{array}{l} x(t+1) = Ax(t) + Bu(t), \quad t = 0, 1, 2, \ldots, N-1, \quad x(0) = x_0 \\ and \quad y(t) = Cx(t) + Du(t) \quad for \quad t = 0, 1, 2, 3, \ldots, N, \end{array} \tag{4.62}$$

uses a quadratic objective function

$$J(u) = \sum_{j=0}^{N} \Big((r(t) - y(t))^T Q(t)(r(t) - y(t)) + (u(t) - u_0(t))^T R(t)(u(t) - u_0(t)) \Big) \tag{4.63}$$

where the time varying weight matrices $R(t) = R^T(t) > 0$ and $Q(t) = Q^T(t) > 0$ for $t = 0, 1, 2, \ldots, N$. Note that, in this description, the discrete time structure makes the separation of the term associated with the final "time" superfluous.

The supervector description provides a simple way to characterize the solution procedure. The relevant system model is simply

$$y = G(A, B, C, D)u + d(C, A, x_0), \tag{4.64}$$

where the supervectors $y \in \mathscr{Y} = \mathscr{R}^{m(N+1)}$ and $u \in \mathscr{U} = \mathscr{R}^{\ell(N+1)}$. Defining

$$R = blockdiag\,[R(0), R(1), \ldots, R(N)] \quad and \quad Q = blockdiag\,[Q(0), Q(1), \ldots, Q(N)], \tag{4.65}$$

the relevant inner products in $\mathscr{U}$ (respectively, $\mathscr{Y}$) are taken to be

$$\langle u, v \rangle_{\mathscr{U}} = u^T R v \quad (respectively \quad \langle y, w \rangle_{\mathscr{Y}} = y^T Q w) \tag{4.66}$$

Denoting the supervectors corresponding to the time series $\{r(0), r(1), \ldots, r(N)\}$ (respectively $\{u_0(0), u_0(1), \ldots, u_0(N)\}$) as r (respectively u_0), the objective function takes the form

$$J(u) = \|r - y\|^2_{\mathscr{Y}} + \|u - u_0\|^2_{\mathscr{U}} \tag{4.67}$$

The optimal solution is then obtained using Eq. (2.104) and is characterized by the supervector equation

$$u = u_0 + G^*(A, B, C, D)(r - y) \quad where \quad G^* = R^{-1} G^T(A, B, C, D) Q. \tag{4.68}$$

This matrix characterization of the solution is precise but may lead to computational problems if N is large. A characterization of solutions using two point boundary value problems and Riccati matrices are given in the followings sections.

4.7.1 The Adjoint and the Discrete Two Point Boundary Value Problem

The relationship defining the optimal solution is now converted into a two point boundary value problem. First note that, as $S(A, B, C, D)$ is assumed to be time invariant, the time variation in the problem is contained solely in the matrices R and Q. The transpose $G^T(A, B, C, D)$ is associated with the matrix representation of the linear, time invariant system $G(A^T, C^T, B^T, D^T)$. The precise form of this relationship is expressed using time reversal matrices (4.8) in the form (Theorem 4.1)

$$G^T(A, B, C, D) = \mathscr{T}(\ell, N)G(A^T, C^T, B^T, D^T)\mathscr{T}(m, N) \tag{4.69}$$

so that

$$u - u_0 = R^{-1}\mathscr{T}(\ell, N)G(A^T, C^T, B^T, D^T)\mathscr{T}(m, N)Q(r - y). \tag{4.70}$$

This calculation is now done in a number of simple steps. For simplicity, introduce the supervector $z = \mathscr{T}(m, N)Q(r - y)$ and note that the components of the supervector $w = G(A^T, C^T, B^T, D^T)z$ can be computed from the state space simulation

$$\begin{aligned} \psi(t' + 1) &= A^T\psi(t') + C^T z(t'), \quad t' = 0, 1, 2, \ldots, N - 1, \\ \psi(0) &= 0, \\ with \quad w(t') &= B^T\psi(t') + D^T z(t') \\ and \quad z(t') &= Q(N - t')\left(r(N - t') - y(N - t')\right) \ for \ t' = 0, 1, 2, \ldots, N. \end{aligned} \tag{4.71}$$

The equations defining the time reversed supervector $\eta = \mathscr{T}(\ell, N)w$ of w are obtained by the substitution $t = N - t'$ and the introduction of the *costate vector* $p(t) = \psi(N - t)$ and notation $\eta(t) = w(N - t)$ to give

$$\begin{aligned} p(t - 1) &= A^T p(t) + C^T Q(t)\left(r(t) - y(t)\right), \quad t = 1, 2, \ldots, N, \\ p(N) &= 0 \quad and \\ \eta(t) &= B^T p(t) + D^T Q(t)\left(r(t) - y(t)\right), \quad t = 0, 1, 2, \ldots, N. \end{aligned} \tag{4.72}$$

The final computation is that of $u - u_0 = R^{-1}\eta$ to give the optimal input as the solution of the two point boundary value problem

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \quad \textit{and} \\ p(t) &= A^T p(t+1) + C^T Q(t+1)\,(r(t+1) - y(t+1)) \\ &\qquad \textit{for} \quad t = 0, 1, 2, \ldots, N-1, \\ \textit{with} \quad u(t) &= u_0(t) + R^{-1}(t)B^T p(t) + R^{-1}(t)D^T Q(t)\,(r(t) - y(t)) \\ &\qquad \textit{for} \quad t = 0, 1, 2, \ldots, N. \\ &\textit{The boundary conditions are} \\ &\qquad x(0) = x_0 \quad \textit{and} \quad p(N) = 0. \end{aligned} \tag{4.73}$$

This representation is a discrete equivalent to the two point boundary value problem for continuous systems derived in Sect. 3.10.3. There are clear structural similarities between the two solutions and, in fact, the reader might like to demonstrate that they become equivalent if T is fixed and the sampling interval $h = T/N$ tends to zero as $N \to \infty$.

4.7.2 A State Feedback/Feedforward Solution

The two point boundary value representation can be converted into a state feedback plus feedforward solution in a similar manner to that described in Sect. 3.10.4. The first step in such a conversion is to propose a linear relationship between $p(t)$ and $x(t)$. The simplest such representation takes the form

$$p(t) = -K(t)x(t) + \xi(t), \quad t = 0, 1, 2, \ldots, N \tag{4.74}$$

where, as in the continuous time case, the $n \times n$ matrix $K(t)$ and the term $\xi(t) \in \mathscr{R}^n$ are, as yet, unspecified. The optimal input is then given by

$$\begin{aligned} u(t) = u_0(t) + R^{-1}D^T Q(t)\,(r(t) - y(t)) + R^{-1}(t)B^T\,(-K(t)x(t) + \xi(t))\,, \\ \textit{for} \quad t = 0, 1, 2, \ldots, N. \end{aligned} \tag{4.75}$$

The characterization of a suitable $K(t)$ and $\xi(t)$ is now described when $D(t) \equiv 0$.

The Case of $D(t) \equiv 0$: Substituting for the costate in the equations for the two point boundary problem gives

$$\begin{aligned} x(t+1) &= \left(A - BR^{-1}(t)B^T K(t)\right) x(t) + Bu_0(t) + BR^{-1}(t)B^T \xi(t) \\ &\quad \textit{and} \\ -K(t)x(t) + \xi(t) &= -\left(A^T K(t+1) + C^T Q(t+1)C\right) x(t+1) + A^T \xi(t+1) \\ &\qquad + C^T Q(t+1)r(t+1). \end{aligned} \tag{4.76}$$

Using the first equation to substitute for $x(t+1)$ into the second of these equations, collecting terms and setting the coefficient matrix of $x(t)$ to zero then implies that these equations are satisfied if the feedback matrix satisfies the nonlinear recursion relations

$$K(t) = \left(A^T K(t+1) + C^T Q(t+1)C\right)\left(A - BR^{-1}(t)B^T K(t)\right), \quad for \quad t = 0, 1, 2, \ldots, N-1 \tag{4.77}$$

and the sequence $\{\xi(t)\}$ satisfies the equations

$$\begin{aligned}\xi(t) = &-\left(A^T K(t+1) + C^T Q(t+1)C\right)\left(Bu_0(t) + BR^{-1}(t)B^T \xi(t)\right) \\ &+A^T \xi(t+1) + C^T Q(t+1)r(t+1), \\ &for \quad t = 0, 1, 2, \ldots, N-1.\end{aligned} \tag{4.78}$$

The boundary conditions are obtained by requiring that $p(N) = 0$ which is satisfied if

$$K(N) = 0 \quad and \quad \xi(N) = 0. \tag{4.79}$$

The equations must hence be solved by reverse time recursion.

More explicit representations for the solutions can be obtained by a simple rearrangement of the equations into the form, for $t = 0, 1, \ldots, N-1$,

$$\begin{aligned}\tilde{K}(t+1) &= A^T K(t+1) + C^T Q(t+1)C, \\ K(t) &= \left(I + \tilde{K}(t+1)BR^{-1}(t)B^T\right)^{-1} \tilde{K}(t+1)A\end{aligned} \tag{4.80}$$

and

$$\begin{aligned}\xi(t) = \left(I + \tilde{K}(t+1)BR^{-1}(t)B^T\right)^{-1} &(A^T \xi(t+1) \\ &-\tilde{K}(t+1)Bu_0(t) + C^T Q(t+1)r(t+1))\end{aligned} \tag{4.81}$$

This representation is a basis for the construction of the time varying state feedback gain matrix $-R^{-1}(t)B^T K(t)$ from the model data A, B, C and objective function weighting matrices $Q(t)$ and $R(t)$, $t = 0, 1, 2, \ldots, N$. The feedforward or predictive term $\xi(t)$, $t = 0, 1, 2, \ldots, N$ can then be constructed by backwards recursion from this data plus the signals $r(t)$ and $u_0(t)$, $t = 0, 1, 2, \ldots, N$.

4.8 Frequency Domain Relationships

The concept of a supervector (or "lifted") description is simply the combing of data sequences into a single vector and has been used by many authors in several forms (see, for example, [3, 93]). It has the benefit that it releases matrix methods for analysis on *finite* time intervals. This, at first sight, appears to be incompatible with the use of $\mathscr{Z}$-transform descriptions which normally look at *infinite* time series. There are however a number of useful links between the two when $S(A, B, C, D)$ is asymptotically stable. These links include the use of $\mathscr{Z}$-transforms and transfer function matrices to compute or bound quadratic forms and operator norms.

4.8.1 Bounding Norms on Finite Intervals

Consider, the supervector relation $y = Gu$ for $S(A, B, C, D)$ on an interval $0 \le t \le N$ for an asymptotically stable, ℓ-input, m-output system with $m \times \ell$ transfer function matrix $G(z)$. Extend the input to the infinite time interval by setting $u(t) = 0,\ t \ge N+1$ and denote the $\mathscr{Z}$-transform of the resultant signal by $u(z) = \sum_{j=0}^{\infty} z^{-j}u(j) = \sum_{j=0}^{N} z^{-j}u(j)$. Note that the first $N+1$ terms in the infinite time series computed from the relationship $y(z) = G(z)u(z)$ are precisely the subvectors of the supervector y but that the output response on the infinite interval $0 \le t < \infty$ is typically non-zero for $t \ge N+1$. It follows from contour integration in the complex plane and the easily proved property

$$\frac{1}{2\pi i}\oint_{|z|=1} z^{p-j}\frac{dz}{z} = \delta_{jp} \qquad (the\ Kronecker\ Delta) \tag{4.82}$$

(where p and j are integers and the contour integration on the unit circle in undertaken in the anti-clockwise sense (formally by writing $z = e^{i\theta}$ with $0 \le \theta \le 2\pi$) that the Euclidean norm in $\mathscr{R}^{m(N+1)}$ can be bounded as follows,

$$\begin{aligned} \|y\|^2 = y^T y &= \textstyle\sum_{t=0}^{N} y^T(t)y(t) \le \sum_{t=0}^{\infty} y^T(t)y(t) \\ &= \tfrac{1}{2\pi i}\oint_{|z|=1} y^T(z^{-1})y(z)\tfrac{dz}{z} \\ &= \tfrac{1}{2\pi i}\oint_{|z|=1} u^T(z^{-1})G^T(z^{-1})G(z)u(z)\tfrac{dz}{z} \end{aligned} \tag{4.83}$$

which is finite as the system is asymptotically stable. Note that, if the system matrices A, B, C, D are real, then, using the properties of complex quadratic forms and singular values yields the result that the supervector norm

$$\|y\|^2 \le \left(\sup_{|z|=1}\ \sigma_\ell^2(z)\right)\frac{1}{2\pi i}\oint_{|z|=1} u^T(z^{-1})u(z)\frac{dz}{z} = \left(\sup_{|z|=1}\ \sigma_\ell^2(z)\right)\|u\|^2 \tag{4.84}$$

where $0 \le \sigma_1(z) \le \sigma_2(z) \le \cdots \le \sigma_\ell(z)$ are the ordered singular values of the complex matrix $G(z)$ on the unit circle. Recalling that $\|G(A, B, C, D)\|$ is the smallest scalar M satisfying the expression $\|y\| \le M\|u\|$ for all $u \in \mathscr{C}^{\ell(N+1)}$, it follows that the norm of the matrix $G(A, B, C, D)$ and the frequency dependence of the singular values of $G(z)$ are closely related. More precisely, and more generally,

Theorem 4.5 (Matrix Norm and the Transfer Function Matrix Norm) *Suppose that $S(A, B, C, D)$ is asymptotically stable and has transfer function matrix $G(z)$. Let Q and R be symmetric positive definite $m \times m$ and $\ell \times \ell$ matrices respectively. Then, the norm of G induced by inner products in $\mathscr{U} = \mathscr{R}^{\ell(N+1)}$ and $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ defined by*

$$\langle u, v\rangle_{\mathscr{U}} = \sum_{t=0}^{N} u^T(t)Rv(t) \quad and \quad \langle y, w\rangle_{\mathscr{Y}} = \sum_{t=0}^{N} y^T(t)Qw(t) \tag{4.85}$$

is simply bounded as follows,

$$\begin{aligned}\|G(A,B,C,D)\| &\le \sup_{|z|=1}\ \sigma_\ell(z) \ = \ \|Q^{1/2}G(z)R^{-1/2}\|_\infty \\ &= \sqrt{\sup_{|z|=1}\ r(R^{-1}G^T(z^{-1})QG(z))},\end{aligned} \tag{4.86}$$

where $\sigma_1^2(z) \le \sigma_2^2(z) \le \cdots \le \sigma_\ell^2(z)$ *are the ordered eigenvalues of* $R^{-1}G^T(z^{-1})QG(z)$ *and* $\|M(z)\|_\infty$ *denotes the so-called* H_∞*-norm of a transfer function matrix* $M(z)$.

Note: The eigenvalues $\sigma_j^2(z)$ *are the singular values of* $Q^{1/2}G(z)R^{-1/2}$.

Proof The topology reduces to a Euclidean topology if y (respectively u) is replaced by $Q^{1/2}y$ (respectively $R^{1/2}u$) and $G(z)$ is replaced by $Q^{1/2}G(z)R^{-1/2}$. The discussion preceding this result then provides the required proof with the final link to the spectral radius following from the relation,

$$\begin{aligned}\|Q^{1/2}G(z)R^{-1/2}\|^2 &= \sup_{|z|=1}\ r(R^{-1/2}G^T(z^{-1})Q^{1/2}Q^{1/2}G(z)R^{-1/2}) \\ &= \sup_{|z|=1}\ r(R^{-1}G^T(z^{-1})QG(z))\end{aligned} \tag{4.87}$$

as the spectral radius is invariant under similarity transformations. □

In what follows, it is seen that H_∞-norms can be arbitrarily accurate estimates of the induced norm $\|G(A,B,C,D)\|$ if N is large.

4.8.2 Computing the Norm Using the Frequency Response

Frequency responses are normally computed using complex number representations of signals. In what follows, the inner products and norms defined in Theorem 4.5 are extended to the complexifications $\mathscr{U}_c = \mathscr{C}^{\ell(N+1)}$ and $\mathscr{Y}_c = \mathscr{C}^{m(N+1)}$ of $\mathscr{U} = \mathscr{R}^{\ell(N+1)}$ and $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ using

$$\langle u, v\rangle_{\mathscr{U}_c} = \sum_{t=0}^{N} \overline{u}^T(t)Qv(t) \quad \textit{and} \quad \langle y, w\rangle_{\mathscr{Y}_c} = \sum_{t=0}^{N} \overline{y}^T(t)Rw(t). \tag{4.88}$$

Consider the *complex valued* input sequence $\{z^j\alpha\}_{0\le j\le N} = \{\alpha, z\alpha, \ldots, z^N\alpha\}$ with $|z| = 1$ and $\alpha \in \mathscr{C}^\ell$ with inner product $\langle u, v\rangle = \overline{u}^T Rv$. The resultant complex output response from zero initial conditions is regarded as a time series in $\mathscr{C}^m$ with inner product $\langle y, w\rangle = \overline{y}^T Qw$. It has a supervector y with components

$$y(t) = G(z)u(t) + d(t), \quad d(t) = -C(zI - A)^{-1}A^t B\alpha, \quad 0 \le t \le N. \tag{4.89}$$

Note that $\|u\|^2_{\mathscr{U}_c} = (N+1)\|\alpha\|^2 = (N+1)\overline{\alpha}^T R\alpha$. If $S(A,B,C,D)$ is asymptotically stable, there exists a real number M such that, for all z of unit magnitude, $\|d\| \le$

$M\|\alpha\| = M(N+1)^{-1/2}\|u\|$. It follows that the fractional error in approximating y by setting $d = 0$ goes to zero as $N \to \infty$. For the complex input considered,

$$\lim_{N\to\infty} (N+1)^{-1/2} \sup\{\|y\|_{\mathscr{Y}_c} \ : \ \|\alpha\| = 1, \ |z| = 1\} = \|Q^{-1/2} G(z) R^{-1/2}\|_\infty \tag{4.90}$$

and hence the induced matrix norm is related to the transfer function matrix norm,

Theorem 4.6 (Matrix Norm as $N \to \infty$) *Using the notation above, suppose that the discrete state space system $S(A, B, C, D)$ is asymptotically stable. Then, as N increases, the induced norm of $G(A, B, C, D) : \mathscr{U}_c \to \mathscr{Y}_c$ has the limiting property*

$$\lim_{N\to\infty} \|G(A, B, C, D)\| = \|Q^{1/2} G(z) R^{-1/2}\|_\infty = \sqrt{\sup_{|z|=1} r(R^{-1} G^T(z^{-1}) Q G(z))}. \tag{4.91}$$

Note: As $G(A, B, C, D)$ is real, this expression also holds for $G(A, B, C, D)$ as a map $\mathscr{U} \to \mathscr{Y}$.

The computational benefit of this result is best seen in later chapters where the maximum gain ranges that guarantee algorithm convergence are related to norms of matrix operators. If a model is known, the norm can be accurately computed. Otherwise, system identification or experimental frequency response testing could be used to construct $G(z)$ at selected frequency points. For iterative control over time intervals much longer than the time constants of the plant, the results produced can provide good estimates of useful control parameters.

4.8.3 Quadratic Forms and Positive Real Transfer Function Matrices

Using the notation of the previous section, suppose that $m = \ell$ and that dynamics from zero initial conditions is considered on the interval $0, 1, 2, \ldots, N$. The two spaces $\mathscr{U}$ and $\mathscr{Y}$ are identical vector spaces with $R = Q$. Suppose also that $S(A, B, C, D)$ is asymptotically stable. Consider $\langle u, G(A, B, C, D)u\rangle_{\mathscr{Y}}$. Extending the analysis to infinite sequences by setting $u(t) = 0, \ t > N$ so that the $\mathscr{Z}$-transform of the resultant input signal is just $u(z) = \sum_{t=0}^{N} z^{-t} u(t)$, contour integration gives

$$\langle u, G(A, B, C, D)u\rangle_{\mathscr{Y}} = \frac{1}{2\pi i} \oint_{|z|=1} u^T(z^{-1}) Q y(z) \frac{dz}{z} = \frac{1}{2\pi i} \oint_{|z|=1} u^T(z^{-1}) Q G(z) u(z) \frac{dz}{z} \tag{4.92}$$

where $y(z) = G(z)u(z)$. It follows that,

Theorem 4.7 (Quadratic Forms and the Frequency Domain) *Suppose that $S(A, B, C, D)$ is asymptotically stable and that $m = \ell$. Suppose that $\underline{g}$ and $\overline{g}$ are real scalars with the property that*

$$\underline{g} \ Q \le Q G(z) + G^T(z^{-1}) Q \le \overline{g} \ Q \quad \textit{for all} \ \ |z| = 1. \tag{4.93}$$

where the inequalities are matrix inequalities in the familiar Euclidean sense in $\mathscr{C}^m$. *Then the matrix* $G(A, B, C, D)$ *satisfies the inequality*

$$\frac{1}{2}\underline{g}\,\langle u, u\rangle_{\mathscr{Y}} \leq \langle u, G(A, B, C, D)u\rangle_{\mathscr{Y}} \leq \frac{1}{2}\overline{g}\,\langle u, u\rangle_{\mathscr{Y}} \quad for\ all\ \ u \in \mathscr{R}^{m(N+1)}. \tag{4.94}$$

Proof Write $z = e^{i\theta}$ to obtain

$$\begin{aligned}\langle u, G(A, B, C, D)u\rangle_{\mathscr{Y}} &= \tfrac{1}{2\pi}\int_0^{2\pi}\ u^T(e^{-i\theta})QG(e^{i\theta})u(e^{i\theta})d\theta \\ &= \tfrac{1}{2\pi}\int_0^{2\pi}\ u^T(e^{-i\theta})\left(\tfrac{1}{2}\right)\left[QG(e^{i\theta}) + G^T(e^{-i\theta})Q\right]u(e^{i\theta})d\theta\end{aligned} \tag{4.95}$$

as $\langle u, G(A, B, C, D)u\rangle_{\mathscr{Y}}$ is real. The result is now completed by writing

$$\tfrac{1}{2}\underline{g}\,\langle u, u\rangle_{\mathscr{Y}} = \left(\tfrac{1}{2}\underline{g}\right)\tfrac{1}{2\pi}\int_0^{2\pi}\ u^T(e^{-i\theta})Qu(e^{i\theta})d\theta \leq \langle u, G(A, B, C, D)u\rangle_{\mathscr{Y}} \tag{4.96}$$

and

$$\langle u, G(A, B, C, D)u\rangle_{\mathscr{Y}} \leq \left(\tfrac{1}{2}\overline{g}\right)\tfrac{1}{2\pi}\int_0^{2\pi}\ u^T(e^{-i\theta})Qu(e^{i\theta})d\theta = \tfrac{1}{2}\overline{g}\,\langle u, u\rangle_{\mathscr{Y}}. \tag{4.97}$$

□

Cases of special interest are those when either $\underline{g} = 0$ or $\underline{g} > 0$. More precisely, it follows from the above result that, using the same topology as above,

Theorem 4.8 (Positive Matrices and Positive Real Conditions) *Suppose that* $S(A, B, C, D)$ *is asymptotically stable with* $m = \ell$ *and that* $Q = Q^T > 0$. *Then, the matrix* $G(A, B, C, D)$ *(and hence* $G+G^*$*) is positive (respectively, positive definite) if the associated* $m \times m$ *transfer function matrix* $G(z)$ *is positive real (respectively, strictly positive real) in the sense that*

$$QG(z) + G^T(z^{-1})Q \geq 0 \quad (respectively \quad QG(z) + G^T(z^{-1})Q > 0) \quad for\ all\ \ |z| = 1. \tag{4.98}$$

Note: for SISO systems these sufficient conditions require that the transfer function $G(z)$ *has positive (respectively, strictly positive) real part for all* z *on the unit circle of the complex plane.*

Proof The result is a consequence of the previous theorem when $G(z)$ is positive real as $\underline{g} = 0$ is the natural choice. If $G(z)$ is strictly positive real, then continuity of $G(z)$ on the (compact) unit circle implies that $\underline{g} > 0$ can be chosen. □

It is useful to note that the condition is only sufficient but, from the analysis of the frequency response, it can be expected to be quite accurate when N is large. For strictly proper systems, the positive real conditions are satisfied in only a minority of cases. The presence of a "D" term plays an important role in producing positive real properties. In this context, the effects of using the relative degree modifications described previously can produce positivity. For example, if the system

$S(A, B, C, D)$ is the stable first order system with transfer function $G(z) = 1/(z-\lambda)$ with $|\lambda| < 1$ then $G(z)$ is not positive real as the product $G(1)G(-1) < 0$. The supervector matrix representation in $\mathscr{R}^{N+1}$ in this case is $G(\lambda, 1, 1, 0)$. However, noting that the relative degree is $k^* = 1$ then the modified representation of $zG(z) = 1+\frac{\lambda}{z-\lambda}$ in $\mathscr{R}^N$ is $G(\lambda, \lambda, 1, 1)$. That this matrix is positive definite then follows from the fact that $zG(z)$ (written in the form $1/(1 - z^{-1}\lambda)$) is strictly positive real as its real part is $\geq 1/(1 + |\lambda|) > 0$.

4.8.4 Frequency Dependent Lower Bounds

The connection between the frequency domain properties and time domain inner products and norms can be extended by considering the impact of carefully chosen frequency domain inequalities between two transfer function matrices $G(z)$ and $K(z)$. The results take the form of defining frequency domain inequalities of the general form $F_z(G(z), K(z), \beta) \geq 0$ for $|z| = 1$ that are sufficient to ensure that associated supervector model operators satisfy inequalities $F_s(G, K, \beta) \geq 0$.

1. Suppose that $G(z)$ is the transfer function matrix of an asymptotically stable, ℓ-input, m-output, linear, time invariant system $S(A, B, C, D)$ with input and output spaces $\mathscr{U} = \mathscr{R}^{\ell(N+1)}$ and $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ and associated inner products defined by matrices Q and R as in Theorem 4.5. The matrix $G(A, B, C, D)$ in the supervector description is denoted, for simplicity, by the single symbol G.
2. Suppose also that $K(z)$ is the transfer function matrix of an asymptotically stable, ℓ-input, ℓ-output, linear, time invariant system $S(A_K, B_K, C_K, D_K)$ with input space $\mathscr{U}$ and output space $\mathscr{Y}_K = \mathscr{U}$ with its inner product. The matrix $K(A_K, B_K, C_K, D_K)$ in its supervector description is denoted by K. Then,

Theorem 4.9 (From Frequency Domain to Operator Inequalities) *The following two statements are true,*

1. *If $G(z)$ and $K(z)$ satisfy, for some real scalar $\beta > 0$, the frequency domain inequality*

$$RK(z) + K^T(z^{-1})R \geq \beta G^T(z^{-1})QG(z) \quad \textit{whenever} \quad |z| = 1, \tag{4.99}$$

then this inequality is also satisfied if β is replaced by any other value $\tilde{\beta} \in (0, \beta)$. In particular,

$$2\langle u, Ku\rangle_{\mathscr{U}} = \langle u, (K + K^*)u\rangle_{\mathscr{U}} \geq \beta\langle u, G^*Gu\rangle_{\mathscr{U}} = \beta\|Gu\|^2_{\mathscr{Y}} \quad \textit{for all} \quad u \in \mathscr{U} \tag{4.100}$$

and hence, in the inner product topology of $\mathscr{U}$,

$$(a) \quad K + K^* \geq \beta G^*G \quad \textit{and} \quad (b) \quad \beta\|G\|^2 \leq \|(K + K^*)\|. \tag{4.101}$$

If $\beta^* > 0$ is the largest value of β satisfying (4.99), then, for all $\beta \in (0, \beta^*)$,

$$2\langle u, Ku\rangle_{\mathscr{U}} = \langle u, (K + K^*)u\rangle_{\mathscr{U}} > \beta\langle u, G^*Gu\rangle_{\mathscr{U}} = \beta\|Gu\|^2_{\mathscr{Y}} \quad if \quad Gu \neq 0. \tag{4.102}$$

Note: The final inequality is an inequality relating quadratic forms. It can be expressed in a simple way by saying that $K + K^* > \beta G^*G$ on any subspace V that is the complement of $ker[G]$ in $\mathscr{U}$. That is, any subspace V such that $\mathscr{U} = V \oplus ker[G]$. If $ker[G] = \{0\}$ then $V = \mathscr{U}$.

2. *If (4.99) is replaced by the stronger assumption that*

$$RK(z) + K^T(z^{-1})R > \beta G^T(z^{-1})QG(z) \quad whenever \quad |z| = 1, \tag{4.103}$$

then, on the whole space $\mathscr{U}$,

$$K + K^* > \beta G^*G. \tag{4.104}$$

This relationship remains true for all $\beta \in (0, \beta^*)$.
Note: Although (4.99) is implied by (4.103), the additional detail adds value to the result and removes the need for $Gu \neq 0$ to ensure that $K + K^* > \beta G^*G$. The implications of this change are, in part, that, whereas (4.99) may allow $K(z)$ to be singular for some z on the unit circle, (4.103) demands that this is not possible.

Proof Consider Part 1. First β can be replaced by $\tilde{\beta}$ as the right hand side of (4.99) is positive. Next, in a similar manner to previous analysis,

$$\begin{aligned} 2\langle u, Ku\rangle_{\mathscr{U}} &= \tfrac{1}{2\pi i}\oint_{|z|=1} u^T(z^{-1})\left(RK(z) + K^T(z^{-1})R\right)u(z)\tfrac{dz}{z} \\ &\geq \beta\left(\tfrac{1}{2\pi i}\right)\oint_{|z|=1} u^T(z^{-1})G^T(z^{-1})QG(z)u(z)\tfrac{dz}{z} \\ &\geq \beta\langle Gu, Gu\rangle_{\mathscr{Y}} = \beta\|Gu\|^2_{\mathscr{Y}}. \end{aligned} \tag{4.105}$$

Inequality (a) follows as $\langle u, Ku\rangle_U = \langle u, K^*u\rangle_U$ so that $2\langle u, Ku\rangle_{\mathscr{U}} = \langle u, (K + K^*)u\rangle_{\mathscr{U}}$ together with $\langle Gu, Gu\rangle_{\mathscr{Y}} = \langle u, G^*Gu\rangle_{\mathscr{U}}$. Inequality (b) then follows as the characterization of positive self-adjoint operators then gives $\beta\|G\|^2 = \beta\|G^*G\| \leq \|(K + K^*)\|$. Finally, the strict inequality whenever $Gu \neq 0$ follows from the above as, in these circumstances, $\beta^*\|Gu\|^2 = \beta\|Gu\|^2 + (\beta^* - \beta)\|Gu\|^2 > \beta\|Gu\|^2$. Part 2 is proved in a similar way by making maximum use of the strict inequality expressed in (4.103). □

Theorem 4.9 is related to Theorem 4.5 when $K = G$. The proof of the following result is a direct consequence of Theorem 4.9 and is left as an exercise for the reader.

Theorem 4.10 (An Application to Iterative Learning Control) *Suppose that G represents a dynamical system with asymptotically stable, ℓ-input, m-output, linear, time invariant, discrete time, state space model $S(A, B, C, D)$ and transfer function matrix $G(z)$ operating on a finite time interval $0 \leq t \leq N$. Using the definitions*

and notation of the previous theorem, let $m = \ell$, $\mathscr{U} = \mathscr{Y}$, $Q = R$, $ker[G] = \{0\}$ *and* $\beta > 0$. *Then the condition* $\|(I - \beta G)\| < 1$ *is equivalent to the requirement that* $(I - \beta G)^*(I - \beta G) < I$ *in* $\mathscr{Y}$ *and hence that* $G + G^* > \beta G^* G$. *A sufficient condition for this to be true is that,*

$$QG(z) + G^T(z^{-1})Q > \beta G^T(z^{-1})QG(z) \quad whenever \quad |z| = 1. \tag{4.106}$$

This statement remains true for all $\beta \in (0, \beta^*)$ *where* β^* *is the largest value of* $\beta > 0$ *satisfying*

$$QG(z) + G^T(z^{-1})Q \geq \beta^* G^T(z^{-1})QG(z) \quad whenever \quad |z| = 1. \tag{4.107}$$

Note: In terms of Iterative Learning Control, this result indicates that the iterative error evolution $e_{k+1} = (I - \beta G)e_k$ *is convergent to zero with monotonically decreasing norms in* $\mathscr{Y}$ *for all "gains"* $\beta \in (0, \beta^*)$. *If* $m = \ell = 1$, *then the factor* Q *is a scalar and can be cancelled and the frequency domain condition reduces to the requirement that the* H_∞ *norm of* $1 - G(z)$ *is strictly less than unity.*

Theorem 4.9 can be rewritten if the term G^*G is replaced by GG^* and $K : \mathscr{Y} \to \mathscr{Y}$. The result is as follows. It uses the same notation and assumptions as the above.

Theorem 4.11 (Another Frequency Domain to Operator Inequality) *The following two statements are true*

1. *Suppose that the* $m \times \ell$ *transfer function matrix* $G(z)$ *and the* $m \times m$ *transfer function matrix* $K(z)$ *satisfy the frequency domain matrix inequality*

$$K^T(z)Q + QK(z^{-1}) \geq \beta QG(z^{-1})R^{-1}G^T(z)Q \quad whenever \ |z| = 1, \tag{4.108}$$

for some real scalar $\beta > 0$. *Then, this inequality is satisfied if* β *is replaced by any other value* $\tilde{\beta} \in (0, \beta)$. *In particular, in the supervector descriptions,*

$$2\langle y, Ky\rangle_{\mathscr{Y}} = \langle y, (K + K^*)y\rangle_{\mathscr{Y}} \geq \beta\langle y, GG^*y\rangle_{\mathscr{Y}} = \beta\|G^*y\|^2_{\mathscr{U}} \quad for\ all \ \ y \in \mathscr{Y} \tag{4.109}$$

and hence, in the topology of $\mathscr{Y}$,

$$(a) \quad K + K^* \geq \beta GG^* \quad so\ that \quad (b) \quad \beta\|G^*\|^2 \leq \|(K + K^*)\|. \tag{4.110}$$

If $\beta^* > 0$ *is the largest value of* β *satisfying* (4.108), *then, for all* $\beta \in (0, \beta^*)$,

$$2\langle y, Ky\rangle_{\mathscr{Y}} > \beta\langle y, GG^*y\rangle_{\mathscr{Y}} = \beta\|G^*y\|^2_{\mathscr{U}} \quad whenever \ \ G^*y \neq 0. \tag{4.111}$$

Note: The final inequality can be expressed by saying that $K + K^* > \beta GG^*$ *on any subspace* V *that is the complement of* $ker[G^*]$ *in* $\mathscr{Y}$. *That is, any subspace* V *such that* $\mathscr{Y} = V \oplus ker[G^*]$.

2. *If* (4.108) *is replaced by the stronger assumption that*

$$K^T(z)Q+QK(z^{-1}) > \beta QG(z^{-1})R^{-1}G^T(z)Q \quad \textit{whenever } |z|=1, \tag{4.112}$$

then, on the whole space $\mathscr{Y}$*,*

$$K + K^* > \beta GG^*. \tag{4.113}$$

Proof The proof has a similar structure to that of Theorem 4.11 with a number of changes. First note that $K : \mathscr{Y} \to \mathscr{Y}$. Next consider an arbitrary time series $\{y(t)\}_{0\le t\le N}$ and associated $\mathscr{Z}$-transform $y(z) = \sum_{t=0}^{N} z^{-t}y(t)$. Its time reversed time series will be denoted by w and a simple calculation indicates that it can be associated with a transform $w(z) = z^{-N}y(z^{-1})$. In a similar manner to previous analysis, using complex variable theory and the simple relation $R^{-1} = R^{-1}RR^{-1}$,

$$\frac{1}{2\pi i}\oint_{|z|=1} w^T(z^{-1})QG(z^{-1})R^{-1}G^T(z)Qw(z)\frac{dz}{z} \;\ge\; \langle \tilde{G}w, \tilde{G}w\rangle_{\mathscr{U}} = \|G^*y\|^2_{\mathscr{U}} \tag{4.114}$$

where $\tilde{G}$ is the matrix in the supervector description of $S(A^T, C^TQ, R^{-1}B^T, R^{-1}D^TQ)$. The final equality follows as the adjoint matrix of G in the given topologies is given by $G^* = \mathscr{T}(\ell, N)\tilde{G}\mathscr{T}(m, N)$ (using time reversal representations of transpose matrices as in Sect. 4.3), and $w = \mathscr{T}(m, N)y$. In addition, the time reversal operators are self adjoint with $\mathscr{T}^2 = I$. These facts imply that

$$\begin{gathered}\langle \tilde{G}w, \tilde{G}w\rangle_{\mathscr{U}} = \langle \tilde{G}\mathscr{T}(m, N)y, \tilde{G}\mathscr{T}(m, N)y\rangle_{\mathscr{U}} \\ = \langle \mathscr{T}(\ell, N)\tilde{G}\mathscr{T}(m, N)y, \mathscr{T}(\ell, N)\tilde{G}\mathscr{T}(m, N)y\rangle_{\mathscr{U}} = \langle G^*y, G^*y\rangle_{\mathscr{U}} = \|G^*y\|^2_{\mathscr{U}}\end{gathered} \tag{4.115}$$

as required. Turning now to the terms containing K, the methodology of Theorem 4.11 gives,

$$\begin{gathered}\tfrac{1}{2\pi i}\oint_{|z|=1} w^T(z^{-1})\left(K^T(z)Q + QK(z^{-1})\right)w(z)\tfrac{dz}{z} \\ = 2\left(\tfrac{1}{2\pi i}\right)\oint_{|z|=1} w^T(z^{-1})K^T(z)Qw(z)\tfrac{dz}{z} \\ = 2\left(\tfrac{1}{2\pi i}\right)\oint_{|z|=1} w^T(z^{-1})Q\left[Q^{-1}K^T(z)Q\right]w(z)\tfrac{dz}{z} = 2\langle w, \tilde{K}w\rangle_{\mathscr{Y}}\end{gathered} \tag{4.116}$$

where $\tilde{K}$ is the supervector matrix for the system $S(A_K^T, C_K^TQ, Q^{-1}B_K^T, Q^{-1}D_K^TQ)$. A similar calculation using time reversal representations of the adjoint operator/matrix $K^* = \mathscr{T}(m, N)\tilde{K}\mathscr{T}(m, N)$ gives $2\langle w, \tilde{K}w\rangle_{\mathscr{Y}} = 2\langle y, K^*y\rangle_{\mathscr{Y}} = 2\langle y, Ky\rangle_{\mathscr{Y}}$. This proves Part 1 of the result. Part 2 is proved in a similar way making maximum use of the strict inequality in (4.112). □

4.9 Discussion and Further Reading

The use of the idea and notation of supervectors (often called a "lifted" description) has been used by many authors (see, for example, [3, 93]) and has the advantage of reducing many calculations for discrete state space systems to more familiar matrix operations. The main advantage is probably theoretical as matrices such as $G(A, B, C, D)$ are typically of large dimension. The theoretical advantage is valuable for many reasons including the fact that it opens up the possibility of the use of eigenvector and singular value analysis and, as has been seen, a simple derivation and presentation of the solution of linear quadratic optimal control problems. Causality is represented by lower triangular structures and time reversal by a simple matrix operation. The ideas are used extensively in this text although it is important to note that, for applications purposes, problems associated with the high dimensionality are avoided, in the main, by reducing algorithms to computations and simulations using the underlying state space models or transfer function matrices.

The range and kernel of $G(A, B, C, D)$ defining dynamics on an interval $0 \leq t \leq N$ have clear physical meaning in terms of the input and output time series. For theoretical purposes, a range equal to $\mathscr{R}^{m(N+1)}$ and kernel equal to $\{0\}$ has the advantage that all reference output trajectories demanded for control purposes will be achievable and the input achieving this desired result will be unique. Methodologies for achieving one or both of these objectives by redefining the output vector hence have control design value as well as theoretical value. There is no obvious best approach to creating situations where these properties are present but the methods described suggest that the solution is certainly not unique. The methods presented here relate closely to systems theoretical concepts already available in the classical control literature. The ideas of relative degree are a natural generalization of the ideas used in transfer function analysis and corresponds to a shift that removes signals and inputs that cause problems. The simplest solution is for uniform rank systems [81] and sets the scene for the rest of the discussion. Decoupling concepts and the notion of relative degrees for each loop extend these ideas to a wider class of systems in a simple way. The use of decoupling theory and the ideas of pole allocation using state feedback has a long history and is described in classic texts referenced in Sect. 1.5. The basic ideas are easily stated but computational tools are more complex for high dimensional problems.

The ideas suggest that the notion of relative degree for MIMO systems is naturally associated with m "delays". The importance of the $\mathscr{C}^*$ canonical form introduced by Morse [75] is that it indicates that these delays are present in all square systems but are not necessarily associated with individual outputs. The ideas underpinning the $\mathscr{C}^*$ canonical form has been included as a logical next step in the identification of "natural", state feedback independent delays $k_1^*, k_2^*, \ldots, k_m^*$ that exist in the internal structure of all linear, time invariant systems. The main consequence of the canonical form is that, for square systems (when $m = \ell$), it is always possible (non-uniquely) to achieve the desired range and kernel properties using state space computations to redefine the output vector. Details of derivation and background to the canonical

form can be found in the original paper by Morse. The technical problem here is that its construction is not only based on a search for invariants under state feedback but also simultaneous invariants under output injection. Output injection has control theoretical interest and in used in the design of state observers [35, 70, 109], which, as control systems design elements, could also play a role in implementations of ILC laws/algorithms.

The application of Iterative Learning Control is firmly embedded in finite time experiments. The derivation of the link between finite time dynamics and frequency domain properties relies on well known contour integration methodologies for functions of a complex variable with norm bounds that provide a link to H_∞ control theory [21, 23]. The ideas of positive real transfer function matrices and transfer functions also has a long history in control theory, particularly in adaptive control [76]. Despite its apparently restrictive properties, the reader should note that positivity is a natural property in Iterative Control as it is linked to the need for the spectrum of an operator to be in the open unit circle. This property inevitably means that retention of the convergence property requires directional limitations on perturbations. Positive real properties will be used extensively in this text as a condition that supports algorithm convergence and robustness. The basic relationships in Theorems 4.9 and 4.11 provide frequency domain bounding of the effects of modelling error on monotonicity and convergence of the error sequence are also central to what follows.

Finally, their are many derivations of the solution of discrete linear quadratic optimal control problems. The one chosen here has the advantage of simplicity and consistency with the structure of the matrix methods used in the text. The text does not discuss the issues that arise when the methods are used in practice but the interested reader will be able to find useful information elsewhere (for example, [20]).

Chapter 5
Iterative Learning Control: A Formulation

Classical feedback control theory has a problem formulation that starts with the issue of stability and follows this requirement closely with issues to do with dynamic performance, robustness to modelling errors and disturbance rejection. Iterative Learning Control has a similar hierarchy of design requirements. In this chapter, the major design requirements are defined in a general context to form the language needed for the remainder of the text.

5.1 Abstract Formulation of a Design Problem

The careful formulation of Iterative Control design criteria provides a bedrock of ideas to underpin the algorithm developments that follow. Many of the criteria map over from those familiar in classical feedback control but differ in detail. The main issues are those of

1. the use of a mathematical model of plant input/output behaviour,
2. a definition of the control objective,
3. an assumption defining the structure of the control system to be designed (that is, defining the design paradigm),
4. a recognition that the causality structure of Iterative Control has a special form,
5. a recognition that design computations can be a mixture of off-line calculations between each iteration and on-line (feedback) control decisions,
6. a precise definition of success in terms of convergence conditions and
7. a consideration of other issues including

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_5

a. a characterization of what is meant by good and bad dynamic performance as the iteration sequence evolves,
b. insights into, and tests for, the robustness of the convergence in the presence of plant/model behavioural differences and
c. a characterization of the effects of imprecise initialization of the plant before each iteration begins.

5.1.1 The Design Problem

The design problem considered has several components:

The Plant Model: *Suppose that a system is driven by an input signal $u \in \mathscr{U}$ which produces the output signal $y \in \mathscr{Y}$ where $\mathscr{U}$ and $\mathscr{Y}$ are normed linear vector spaces of plant input and output signals respectively. Suppose also that, the relationship between the input and output signals can be represented by a mathematical model of the form*

$$y = \mathscr{G}(u, d) \ (A\ General\ Nonlinear\ Input/output\ Model) \tag{5.1}$$

where $\mathscr{G} : \mathscr{U} \times \mathscr{Y} \to \mathscr{Y}$ is a linear or nonlinear mapping and $d \in \mathscr{Y}$ represents plant initial conditions or other known phenomena of interest.

Note: For linear systems the model will be written in the form

$$y = Gu + d \ (A\ General\ Linear\ Input/output\ Model) \tag{5.2}$$

where $G : \mathscr{U} \mapsto \mathscr{Y}$ is a bounded linear operator. Examples include the continuous or discrete state space models $S(A, B, C, D)$ discussed in previous chapters.

The Tracking Objective: *Let $r \in \mathscr{Y}$ be a reference or demand signal chosen by the user and consider the problem of finding an input $u_\infty \in \mathscr{U}$ such that*

$$\begin{aligned} r &= y_\infty \qquad (The\ Perfect\ Tracking\ Requirement) \\ where\ y_\infty &= \mathscr{G}(u_\infty, d) \ \ (The\ Input/output\ Relation). \end{aligned} \tag{5.3}$$

Note: this idea can be regarded as a theoretical idealization of a practical desire to achieve excellent accuracy. A formal statement of this is to represent the objective as that of achieving an accuracy target expressed in the form

$$\|r - y\|_{\mathscr{Y}} < \varepsilon \tag{5.4}$$

where $\|\cdot\|_{\mathscr{Y}}$ represents the norm in $\mathscr{Y}$ and $\varepsilon > 0$ is a "small" parameter defined by the user. It represents the tracking accuracy sought for. It is worth noting at this

stage that the convergence conditions required for Iterative Control ensure that this condition can be satisfied *for any value of* ε.

An Iterative Control Structure: *Consider the solution of the tracking problem using an iterative strategy where plant operation is repetitive. An iteration (also called a repetition, trial or pass) consists of application of an input signal to the plant. Control signals can vary from iteration to iteration. More precisely,*

1. *let* $u_0 \in \mathscr{U}$ *be an initial choice of input signal and*
2. *suppose that* $d \in \mathscr{Y}$ *is the same for each and every iteration. That is, the initial condition component and the environmental conditions within which the plant operates are the same for each and every iteration.*

Let the integer $k = 0, 1, 2, \ldots$ *be a counter for the number of repetitions completed. Denote the input signal used on the kth iteration by* u_k *and let the corresponding output generated by the plant be* $y_k = \mathscr{G}(u_k, d)$ *for indices* $k \geq 0$.
The signal representing tracking accuracy on iteration k is the error signal

$$e_k = r - y_k, \ k \geq 0. \tag{5.5}$$

Now suppose that the iterative control strategy is defined by an Iterative Learning Control law that constructs the input signal $u_{k+1} \in \mathscr{U}$ *on the* $(k+1)^{th}$ *iteration from known measurement data. This typically takes the form of input and error data*

$$e_0, e_1, \ldots, e_k, e_{k+1} \text{ and } u_0, u_1, \ldots, u_k \tag{5.6}$$

but other measurements should not be excluded from the design process.
In functional terms, the input update procedure is illustrated by a relationship

$$u_{k+1} = f_{k+1}(e_0, e_1, \ldots, e_k, e_{k+1}, u_0, u_1, \ldots, u_k) \tag{5.7}$$

where $f_{k+1}(\cdot) : \mathscr{Y} \times \mathscr{Y} \times \cdots \mathscr{Y} \times \mathscr{U} \times \cdots \times \mathscr{U} \to \mathscr{U}$ *represents a computational procedure or algorithm processing the data in its argument list to produce* u_{k+1}.

The choice of f_{k+1} is the main design objective as, together with plant dynamics $\mathscr{G}$, it dictates the form of the sequences of input iterates $u_0, u_1, u_2, \ldots$, the resultant output iterates $y_0, y_1, y_2, \ldots$ and hence the error iterates $e_0, e_1, e_2, \ldots$. It can be assumed to change from iteration to iteration but the most common form taken is the "*Finite Memory Law*" represented by

$$u_{k+1} = f(e_{k-M_e+1}, e_{k-M_e+2}, \ldots, e_k, e_{k+1}, u_{k-M_u+1}, u_{k-M_u+2}, \ldots, u_k) \tag{5.8}$$

where f is independent of k. The positive integers M_u and M_e represent the *"memory"* of the algorithm by defining the range of iterations used in input signal updates. The most common choice is $M_u = M_e = 1$ when

$$u_{k+1} = f(e_k, e_{k+1}, u_k) \ (\textit{A Unit Memory Algorithm}). \tag{5.9}$$

Causality Requirements: *Input update laws need to satisfy the causality requirement that states that only available data is used at and during each iteration. More precisely, the data needed to construct the input signal at any time t on iteration $k+1$ is only the actual data recorded at any time on previous iterations (with index $k' \leq k$) and the data recorded up to and including time t on iteration $k+1$.*

This causality definition allows for the use of data at all times $t' > t$ from iterations with index $k' \leq k$. In this sense the concept differs greatly from classical mathematical views of causality.

The Convergence Requirement: *The successful completion of the tracking task using the "closed loop" combination of plant input output dynamics and input update rules leads to a tracking error sequence $\{e_k\}_{k\geq 0}$ satisfying the convergence condition*

$$\lim_{k\to\infty} e_k = 0 \tag{5.10}$$

in the norm topology in $\mathscr{Y}$. That is

$$\lim_{k\to\infty} \|e_k\|_{\mathscr{Y}} = 0 \tag{5.11}$$

If these conditions are satisfied then arbitrarily accurate tracking (as measured by the norm $\|\cdot\|_{\mathscr{Y}}$) can be achieved in a finite number of iterations in the sense that, for all $\varepsilon > 0$, there will exist an integer N_ε (dependent on ε) such that

$$\|e_k\|_{\mathscr{Y}} < \varepsilon \quad forall \quad k \geq N_\varepsilon. \tag{5.12}$$

A number of important refinements are as follows:

1. As stated above, convergence from the specified initial error e_0, only, is required. The more general requirement is that the convergence condition is satisfied either for all $e_0 \in \mathscr{Y}$ or for all $e_0 \in \mathscr{Y}_0$ where $\mathscr{Y}_0$ is a linear vector subspace of $\mathscr{Y}$.
2. If $\mathscr{Y}$ is a real Hilbert space, then convergence in norm could, if necessary, be replaced by weak convergence. This would not normally be the ideal situation but may be a necessary consequence of the issues faced in mathematical analysis.
3. No explicit requirement for convergence of the input is assumed in general. For practical applications, this is not necessarily a problem as the algorithm will be terminated after a finite number of iterations which, if large enough, can yield a sufficiently accurate outcome. Theoretical conditions to guarantee convergence of the input sequence to a signal $u_\infty \in \mathscr{U}$ will normally need assumptions to be made about the initial error e_0 and hence the reference signal r.

Performance Requirements: *As in classical feedback control, there may be a wide range of requirements put in place to meet the specific needs of the application. The major requirements assumed in this text are those of convergence plus additional criteria including*

1. *an acceptable pattern of changes in the errors from iteration to iteration,*
2. *ensuring that errors e_k are sufficiently small after a small number of iterations (rapid convergence) and*
3. *embedding a degree of robustness into the design predictions.*

Monotonic Convergence: *Of particular interest is the pattern of error changes. This may take many forms but is illustrated here by a requirement that tracking accuracy improves on each and every iteration. A simple way of characterizing this idea is to require that the error norm reduces monotonically from iteration to iteration in the sense that*

$$\|e_{k+1}\|_{\mathscr{Y}} < \|e_k\|_{\mathscr{Y}} \quad \textit{for all} \quad k \geq 0 \ (\textit{Error Norm Monotonicity}). \tag{5.13}$$

The choice of norm in $\mathscr{Y}$ then influences the details of the observed, physical form of the convergence. For example, in situations where the norm is the mean square value, a simple, useful and familiar interpretation is available.

5.1.2 Input and Error Update Equations: The Linear Case

The nature of the error evolution from iteration to iteration depends on the control law assumed. A useful class of linear algorithms for linear systems has the unit memory form, for all $k \geq 0$,

$$\begin{array}{rl} u_{k+1} = u_k + K_0 e_k + K_1 e_{k+1} & (\textit{The Input Update Rule}) \\ \textit{with } y_k = Gu_k + d & (\textit{Plant Dynamics}) \end{array} \tag{5.14}$$

Here, both K_0 and K_1 are linear and bounded operators mapping $\mathscr{Y}$ into $\mathscr{U}$. Choosing them is the core of the resultant design problem and the aim of achieving desired convergence and performance objectives.

Error evolution is obtained using $Gu = y - d = r - e - d$ which gives

$$e_{k+1} = e_k - GK_0 e_k - GK_1 e_{k+1}. \tag{5.15}$$

Note that the input update relationship is implicit as e_{k+1} depends on u_{k+1} and yet e_{k+1} is the result of application of u_{k+1}. In this sense the unit memory update relationship is a direct parallel of the use of feedback (represented by $K_1 e_{k+1}$) and feedforward (represented by $K_0 e_k$) control. Denoting the identity operator by I and assuming that $I + GK_1$ has a bounded inverse, gives the *error evolution*

$$\begin{array}{rcl} e_{k+1} & = & Le_k \ (\textit{The Error Evolution Equation}) \\ & & \textit{where the operator } L = (I + GK_1)^{-1}(I - GK_0) \end{array} \tag{5.16}$$

is linear and bounded. It follows that the iteration update formula can be written in the equivalent "feedforward form"

$$u_{k+1} = u_k + (K_0 + K_1 L)e_k \tag{5.17}$$

Note: In particular, this suggests that the use of feedforward control laws is quite general although feedback terms could, intuitively, have a useful influence on issues such as robustness and disturbance rejection in practice.

Using (5.16), the iteration error evolution has the property $e_1 = Le_0,\ e_2 = Le_1 = L^2 e_0$ or, more generally, the "power law" formula

$$e_k = L^k e_0,\ \textit{for all}\ k \geq 0. \tag{5.18}$$

That is, the errors observed in the iterative process evolve in a dynamic way dependent on several factors including

1. the initial error e_0, G and both K_0 and K_1 plus
2. the properties of powers L^k of the operator L.

The first factor is a design issue with many solutions, some of which are explored in this text. The second links the notion of convergence to properties of $L : \mathscr{Y} \to \mathscr{Y}$ and, more specifically, to its spectrum. Using Theorem 2.1, convergence to zero tracking error is guaranteed (for all initial errors e_0) if at least one of the (sufficient) conditions

$$r(L) \ < \ 1 \ \textit{or}\ \|L\| < 1 \tag{5.19}$$

is satisfied. If L is a matrix (or, equivalently, $\mathscr{Y}$ is finite dimensional), then the spectral radius condition becomes a condition on eigenvalues. For infinite dimensional $\mathscr{Y}$, the analysis is more complex but convergence turns out also to be possible if $r(L) = \|L\| = 1$ under certain conditions that will arise later.

5.1.3 Robustness and Uncertainty Models

Uncertainty Representations and Robustness: The primary robustness requirement is a formalization of the need to retain convergence despite differences between plant behaviour and the behaviour predicted by the model. Such differences can take many forms and analysis is, to date, confined to certain special representations of modelling error. Assuming linear plant dynamics, two forms of uncertainty are easily stated, namely,

1. uncertainty in the value of parameters present in the plant model and
2. uncertainty in the structural form of the plant.

Parametric uncertainty is often expressed in the form of interval bounds for an imprecisely known parameter p and could take the form $p \in [\underline{p}, \overline{p}]$ where both $\underline{p}$ and $\overline{p}$ are

known upper and lower parameter bounds. Structural uncertainty models could take many forms in practice but, for theoretical analysis purposes, are often additive or multiplicative. More precisely, if, $G : \mathscr{U} \to \mathscr{Y}$ represents the plant *model*, then the real plant dynamics could be assumed to be characterized by, for example, either the operator

$$\begin{array}{c} G + \Delta G \ \ (\textit{the Additive Uncertainty Model}), \\ \textit{or } UG \ (\textit{a Left Multiplicative Uncertainty Model}) \end{array} \tag{5.20}$$

where $\Delta G : \mathscr{U} \to \mathscr{Y}$ and $U : \mathscr{Y} \to \mathscr{Y}$ are linear bounded operators. They can be regarded as alternative representations of the difference between plant and model behaviours. Other representations are also possible including plant dynamics

$$GU \quad (\textit{a Right Multiplicative Uncertainty Model}) \tag{5.21}$$

where $U : \mathscr{U} \to \mathscr{U}$ is linear and bounded. Alternatively, if both G and the plant have inverses, the perturbation could be regarded as a perturbation of the inverse with plant inverse dynamics described by

$$G^{-1} + U \quad (\textit{an Inverse Additive Uncertainty Model}) \tag{5.22}$$

where $U : \mathscr{Y} \to \mathscr{U}$ is linear and bounded.

The choice of perturbation description, at least in part, is dictated by the feasibility of deriving useful theoretical results and robustness tests. Few general comments can be made about their relative merits.

Robustness of the Convergence of an Algorithm requires that convergence to zero error is retained in the presence of the modelling error U. Convergence can have many forms and may include unacceptable temporary growth characteristics. To avoid this problem, the idea of "*Robust Monotonic Convergence*" adds structure to the convergence property.

Definition 5.1 (*Robust Monotonic Convergence*) Suppose that a linear model $y = Gu + d$ is used for a plant with linear dynamics. Suppose also that an input update algorithm is chosen and that the use of the model predicts that error norm evolution will have the properties of converging to zero with monotonically decreasing error norm. That is, *for all* e_0 and $k \geq 0$, the monotonic behaviour

$$\|e_{k+1}\|_{\mathscr{Y}} < \|e_k\|_{\mathscr{Y}} \ \textit{and}, \ \textit{also}, \ \lim_{k\to\infty} e_k = 0 \tag{5.23}$$

is predicted. Then, if the control law is applied to the plant and the plant is represented by G perturbed by a modelling error U, the resultant closed loop Iterative Learning Control system is said to be robust monotonically convergent in the presence of the modelling error U if, and only if, the monotonic convergence properties (5.23) are retained.

The reader should note that, from time to time in the text, convergence to zero is not possible as the reference is unattainable. In such cases the ideas of monotonic convergence and robust monotonic convergence have natural extensions.

Demonstrating Robustness is Easy? The answer is a definitive "no" in general but the problem may not be intractable. The simplest approach is to manipulate operator norms to bound the magnitude of perturbations. For example, taking $K_1 = 0$ so that $L = I - GK_0$ and supposing that the norm $\|L\| < 1$, Theorem 2.1 predicts convergent behaviour when using the control law $u_{k+1} = u_k + K_0 e_k$. However, when applied to the real plant $G + \Delta G$, the operator L becomes $I - (G + \Delta G)K_0 = L + \Delta L$ with $\Delta L = -\Delta G K_0$. Again using Theorem 2.1, an inequality that guarantees convergence of the algorithm is $\|L + \Delta L\| < 1$. As $\|L + \Delta L\| \leq \|L\| + \|\Delta L\|$, it follows that,

Theorem 5.1 *Using the notation of the preceding discussion, a sufficient condition for robust convergence of the algorithm $u_{k+1} = u_k + K_0 e_k$ in the presence of additive uncertainty is that*

$$\|\Delta L\| = \|\Delta G K_0\| < 1 - \|L\| \tag{5.24}$$

This simple norm inequality demonstrates a degree of robustness whenever $\|L\| < 1$ and also that this robustness is described by an interaction between the modelling error and the choice of control element K_0. It also suggests that modelling errors might need to be small if $\|L\|$ takes a value close to unity. If $\|L\| = 1$, it provides no information whatsoever. This approach can provide insight into many finite dimensional problems but, for the infinite dimensional case, the use of more refined techniques than the simple use of norm bounds and inequalities is required. Details of this will appear later in the text.

Robustness to Initial Condition Variation. Suppose that the plant model is exact but that the initialization of each iteration is subject to variation. A simple way of representing this is to replace d by an iteration dependent term d_k. Again, using $u_{k+1} = u_k + K_0 e_k + K_1 e_{k+1}$, the model of dynamics on any iteration k now takes the form $y_k = G u_k + d_k$ and the error evolution becomes

$$e_{k+1} = L e_k + (I + GK_1)^{-1}(d_k - d_{k+1}),\ k \geq 0. \tag{5.25}$$

Clearly the variation in initialization conditions has an effect on convergence. If the variation of d_k has a degree of randomness but is bounded in the sense that the sequence $\|d_k\|_{\mathscr{Y}}$ is bounded, then the best that can be hoped for is that the iteration does not diverge. This does not mean that the error converges to any value but it would mean that the error sequence is bounded in norm—a desirable but not necessarily acceptable performance. The following theorem illustrates potential boundedness behaviours.

Theorem 5.2 (Bounded Response with Varying Initialization) *For the linear repetitive system $y_k = G u_k + d_k,\ k \geq 0,$ and control update law $u_{k+1} = u_k + K_0 e_k + K_1 e_{k+1}$, suppose that the terms $d_k,\ k \geq 0$, can be described by the relation $d_k = H\eta_k,\ k \geq 0$, where $\eta_k \in \mathscr{X}_0$, $\mathscr{X}_0$ is a normed linear vector space and*

$H : \mathscr{X}_0 \to \mathscr{Y}$ is linear and bounded. Suppose also that

$$\|L\| < 1 \text{ and that } M_d = \sup_{k\geq 0} \|\eta_k - \eta_{nom}\|_{\mathscr{X}_0} < \infty. \tag{5.26}$$

where η_{nom} is the nominal value that generates the ideal initialization condition. Then the resultant error sequence is bounded in the sense that, for all $p \geq 1$,

$$\sup_{k\geq p} \|e_k\|_{\mathscr{Y}} \leq \|L\|^p \|e_0\|_{\mathscr{Y}} + \frac{2M_d}{(1-\|L\|)} \|(I + GK_1)^{-1}H\| < \infty \tag{5.27}$$

and hence

$$\limsup_{k\geq 0} \|e_k\|_{\mathscr{Y}} \leq \frac{2M_d}{(1-\|L\|)} \|(I + GK_1)^{-1}H\| < \infty. \tag{5.28}$$

Proof Using arguments identical to those used for discrete state space systems

$$e_k = L^k e_0 + \sum_{j=0}^{k-1} L^{k-j-1}(I + GK_1)^{-1}(d_j - d_{j+1}), \qquad k \geq 1 \tag{5.29}$$

Using $d = H\eta$ and taking norms then gives the required result via

$$\begin{aligned}
\|e_k\| &\leq \|L\|^k \|e_0\| + \sum\nolimits_{j=0}^{k-1} \|L\|^{k-j-1} \|(I + GK_1)^{-1}H\| \|(\eta_j - \eta_{j+1})\|, \\
&\leq \|L\|^k \|e_0\| \\
&\quad + \sum\nolimits_{j=0}^{\infty} \|L\|^j \|(I + GK_1)^{-1}H\| \sup_{j\geq 0} \|((\eta_j - \eta_{nom}) - (\eta_{j+1} - \eta_{nom}))\| \\
&\leq \|L\|^k \|e_0\| + 2\sum\nolimits_{j=0}^{\infty} \|L\|^j \|(I + GK_1)^{-1}H\| \sup_{j\geq 0} \|(\eta_j - \eta_{nom})\| \\
&\leq \|L\|^k \|e_0\| + 2\sum\nolimits_{j=0}^{\infty} \|L\|^j \|(I + GK_1)^{-1}H\| M_d \\
&= \|L\|^k \|e_0\| + \frac{2M_d}{(1-\|L\|)} \|(I + GK_1)^{-1}H\|, \text{ for } k \geq 1.
\end{aligned} \tag{5.30}$$

□

The error bound computed above is highly conservative and is likely to have little practical value as an numerical indicator of performance. Its primary contributions are that it demonstrates conditions for boundedness of the error sequence response and that the bound depends on the way in which the terms d_k are generated via H. Clearly, the bound is influenced by the nature and magnitude of the operator $(I + GK_1)^{-1}H$ and it may be desirable to design a feedback compensator K_1 to reduce this magnitude. For example, consider a continuous time, time invariant, linear state space system $S(A, B, C, D)$ operating on a time interval $[0, T]$ from an assumed, desirable initial condition x_0 and in the absence of external disturbances. In this case, $d(t)$ can be identified with the function on $[0, T]$

$$d(t) = Ce^{At}x_0,\ t \in [0, T] \tag{5.31}$$

which can be written as $d = Hx_0$ with $x_0 = \eta_{nom} \in \mathscr{X}_0 = \mathscr{R}^n$. The operator H is multiplication by the matrix valued function Ce^{At}. Suppose that initialization variation is identified with imprecise setting of the initial condition $x(0)$. That is, the initial condition on iteration k is, in reality, x_{0k}. $d_k(t)$ can be assumed to take the form $Hx_{0k},\ k \geq 0$. The bound on variation is the bound on initial condition variation,

$$\|x_{0k} - x_0\|_{\mathscr{R}^n} \leq M_d \text{ and } \eta_k = x_{0k},\ k \geq 0. \tag{5.32}$$

Under these conditions, let $\mathscr{Y} = L_2^m[0, T]$ and use intuitive classical frequency domain ideas to identify H with a low pass signal suggesting that the norm of $\|(I + GK_1)^{-1}H\|$ in $L_2^m[0, T]$ could be small if the frequency response of $(I + GK_1)^{-1}$ is small over the bandwidth of H. This is a classical design concept as $(I + GK_1)^{-1}$ is the sensitivity function of a unity feedback control system for G with forward path controller K_1.

5.2 General Conditions for Convergence of Linear Iterations

Many of the iterations described in this text have the general form an iterative evolution in some normed linear vector space $\mathscr{Y}$ of the form

$$e_{k+1} = Le_k + d,\ \textit{with } e_k \in \mathscr{Y},\ k \geq 0,\ \textit{and } d \in \mathscr{Y}, \tag{5.33}$$

with the linear operator $L : \mathscr{Y} \to \mathscr{Y}$ linear and bounded and d is an iteration independent bias term. For a given initial condition $e_0 \in \mathscr{Y}$, the solution has the form

$$e_k = L^k e_0 + \left(I + L + L^2 + \cdots + L^{k-1}\right) d,\ k \geq 1. \tag{5.34}$$

This equation leads to the following theorem that provides properties of both L and d that guarantee convergence of the signals e_k to a limit e_∞ in $\mathscr{Y}$.

Theorem 5.3 (General Conditions for Iterative Convergence) *Using the notation above, suppose that, for all* $e_0 \in \mathscr{Y}$,

$$\lim_{k\to\infty} \|L^k e_0\|_{\mathscr{Y}} = 0 \textit{ and also that } d \in \mathscr{R}[I - L]. \tag{5.35}$$

Then,

1. *it is necessary that* $ker(I - L) = \{0\}$.
2. *The sequence* $\{e_k\}_{k\geq 0}$ *converges in the norm topology of* $\mathscr{Y}$ *to a uniquely defined limit* $e_\infty \in \mathscr{R}[I - L]$ *satisfying*

$$d = (I - L)e_\infty. \tag{5.36}$$

3. *If $\mathscr{Y}$ is a real Hilbert space, then convergence in the norm topology can be replaced by convergence in the weak topology if $d \in \mathscr{R}[I-L]$ and $\lim_{k\to\infty} L^k e_0 = 0$ weakly for all $e_0 \in \mathscr{Y}$.*

Proof First note that $ker[I - L] = \{0\}$ as otherwise there exists a non-zero vector $z \in \mathscr{Y}$ such that $Lz = z$ which, choosing $e_0 = z$ indicates that $L^k z = z$ for all $k \geq 0$ which contradicts the assumption that $L^k e_0 \to 0$ for all e_0. Next, write $d = (I-L)\psi$ and note that such a $\psi \in \mathscr{Y}$ is unique. Substituting into the expression for e_k gives

$$\begin{aligned} e_k &= L^k e_0 + \left(I + L + L^2 + \cdots + L^{k-1}\right)(I - L)\psi \\ &= L^k e_0 + \left(I - L^k\right)\psi \end{aligned} \tag{5.37}$$

and hence $\lim_{k\to\infty} e_k = \psi$ which is the required value of e_∞. The theorem is proved as the argument applies equally using firstly norm and then weak convergence assumptions. □

The result relates both convergence in the norm topology and weak convergence to general properties of L and d that, in principle, have to be checked for each application of the theory. Little can be said about the condition $d \in \mathscr{R}[I-L]$ without further information of the nature of $\mathscr{Y}$ and L. The convergence of $L^k e_0$ for all e_0 is a property that has useful consequences even at the level of operator theory. In the following paragraphs, a number of general results are discussed and proven. They form the basic conditions guiding much of the algorithm analysis in this and following chapters.

5.2.1 Spectral Radius and Norm Conditions

The first general result is a slightly extended version of Theorem 2.1 expressed using the notation defined above,

Theorem 5.4 (Convergence of Iterations in Norm and the Spectral Radius) *Let $\mathscr{Y}$ be a Banach space. Then, given an arbitrary starting vector $e_0 \in \mathscr{Y}$ and a bounded linear operator $L : \mathscr{Y} \to \mathscr{Y}$, the sequence $\{e_k\}_{k\geq 0}$ generated by the iteration $e_{k+1} = Le_k,\ k \geq 0,$ converges (in norm) to zero if (a sufficient condition)*

$$r(L) < 1. \tag{5.38}$$

(a) A sufficient condition for $r(L) < 1$ is that $\|L\| < 1$.
(b) If $\mathscr{Y}$ is finite dimensional, then the spectral radius condition is both necessary and sufficient.

Proof The proof is simply that of Theorem 2.1 and using familiar matrix methods to a matrix representation of L. □

In general, dynamical systems have large and possibly infinite dimensional input and/or output spaces and the computation of the spectral radius could be a challenging problem. For a class of linear, time invariant discrete (respectively, continuous) state space systems evolving in $\mathscr{R}^{m(N+1)}$ (respectively $L_2^m[0, T]$), the spectrum can be computed precisely as follows

Theorem 5.5 (The Spectral Radius of State Space Systems) *Suppose that L is the operator associated with an m-input, m-output, discrete or continuous-time state space system $S(A, B, C, D)$ on a finite time interval. Then, the spectrum of L is precisely the set of eigenvalues of D and hence $r(L) = r(D)$.*

Proof The proof examines the solution of the relation $(\eta I - L)w = v$ for arbitrary $v \in \mathscr{Y}$. A state space realization of $\eta I - L$ has the form $S(A, B, -C, \eta I - D)$ which, using Theorem 3.3 (which apples to both continuous and discrete time cases), has a bounded inverse on the interval if, and only if, $\eta I - D$ is invertible. The spectral values of L are hence the eigenvalues of D and the theorem is proved. □

The theorem provides sufficient conditions for convergence of the iteration. It is intriguing to note that the condition depends only on the D matrix in the model $S(A, B, C, D)$.

More generally, the use of norm methodologies can provide good insight into behaviours. For example, if $\|L\| < 1$, then Theorem 5.2 has already demonstrated that variable (non-zero) initialization of the iterations (that is, replacing d by a sequence of bounded d_k) will lead to a sequence $\{e_k\}_{k\geq 0}$ bounded in norm. Even greater detail can obtained if the requirement of monotonicity is added to the list of desired behaviours of the iterations. Monotonicity depends on the choice of norm in $\mathscr{Y}$ and the iterative process may be monotonically convergent with respect to a norm $\|\cdot\|_0$ in $\mathscr{Y}$ but not monotonic in the preferred norm $\|\cdot\|_{\mathscr{Y}}$. Using this notation, the following result is easily proved,

Theorem 5.6 *Suppose that the two norms $\|\cdot\|_{\mathscr{Y}}$ and $\|\cdot\|_0$ on $\mathscr{Y}$ are topologically equivalent in the sense that, for some real scalars $0 < a \leq b$,*

$$a\|e\|_0 \leq \|e\|_{\mathscr{Y}} \leq b\|e\|_0 \text{ for all } e \in \mathscr{Y}. \tag{5.39}$$

Suppose that the iterative process $e_{k+1} = Le_k$ generates a monotonically decreasing sequence satisfying, for some $\lambda < 1$, $\|e_{k+1}\|_0 \leq \lambda\|e_k\|_0$ for all $k \geq 0$. Then, for all $k \geq 0$, convergence is achieved in the $\|\cdot\|_{\mathscr{Y}}$ norm topology with the, possibly non-monotonic, behaviour

$$\|e_k\|_{\mathscr{Y}} \leq b\lambda^k\|e_0\|_0 \leq \frac{b}{a}\,\lambda^k\,\|e_0\|_{\mathscr{Y}}. \tag{5.40}$$

There is a strong connection between operator norms and monotonicity.

Theorem 5.7 (Monotonic Convergence and the Operator Norm) *Consider the iteration $e_{k+1} = Le_k$ in the normed vector space $\mathscr{Y}$. Then, a necessary condition for this iteration to be monotonic is that*

$$\|L\| \leq 1. \tag{5.41}$$

Moreover, for the cases where $\mathscr{Y}$ is finite dimensional, it is both necessary and sufficient that $\|L\| < 1$ when the iterations are also monotonically convergent. This condition is also sufficient for monotonic convergence if $\mathscr{Y}$ is infinite dimensional.

Proof Note that $\|e_{k+1}\|_{\mathscr{Y}} \leq \|L\|\|e_k\|_{\mathscr{Y}}$ and that, if $\|L\| > 1$, the definition of the operator norm implies that there exists an initial error e_0 such that $\|e_1\|_{\mathscr{Y}} > \|e_0\|_{\mathscr{Y}}$. It is therefore necessary that $\|L\| \leq 1$. In finite dimensions, L has a matrix representation and hence, if $\|L\| = 1$, there must exist a non-zero vector e_0 such that $\|e_1\|_{\mathscr{Y}} = \|L\|\|e_0\|_{\mathscr{Y}} = \|e_0\|_{\mathscr{Y}}$ so that $\|L\| < 1$ is necessary for monotonic convergence. That it is also sufficient follows as then $\|e_{k+1}\|_{\mathscr{Y}} \leq \|L\|\|e_k\| < \|e_k\|_{\mathscr{Y}}$ for all $k \geq 0$ and $\|e_k\|_{\mathscr{Y}} \leq \|L\|^k\|e_0\| \to 0$ as $k \to \infty$. □

The implications of monotonicity are even greater if additional structure is added to either $\mathscr{Y}$ or the operator L. First, consider the situation when the structure of Hilbert spaces is added to the mix. More precisely, if $\mathscr{Y}$ is a real Hilbert space, then the following result leads to a useful insight into the nature of L in the form of positivity conditions on $I - L$ that are essential if monotonic iteration is to be achieved.

Theorem 5.8 (Positivity and Monotonic Convergence) *Suppose that $\mathscr{Y}$ is a finite or infinite dimensional, real Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathscr{Y}}$ and associated induced norm. Consider the iteration $e_{k+1} = Le_k$, $k \geq 0$. Write $L = I - H$ where $H : \mathscr{Y} \to \mathscr{Y}$ is linear and bounded. Then, a necessary condition for monotonic convergence is that*

$$H + H^* > 0 \text{ (A Strict Positivity Condition for Monotonicity)} \tag{5.42}$$

where H^* is the adjoint of H. Moreover, if $\mathscr{Y}$ is finite dimensional, it is necessary that there exists a real number $\varepsilon_0 > 0$ such that

$$H + H^* \geq \varepsilon_0^2 I \text{ (A Stronger Strict Positivity Condition)}. \tag{5.43}$$

Suppose that the iteration is modified to replace L by $L_\beta = I - \beta H$ where $\beta > 0$ is a real scalar. Then, in both the finite and infinite dimensional cases, the validity of condition (5.43) implies that there exists a real number $\beta^* > 0$ such that, for any choice of $\beta \in (0, \beta^*)$, it is possible to compute a real number $\lambda(\beta) < 1$ (dependendent on β) such that monotonic convergence of the modified iteration $e_{k+1} = L_\beta e_k$ is guaranteed as $\|L_\beta\| \leq \lambda(\beta) < 1$. Finally, the largest possible choice of β^* lies in the range

$$\frac{\varepsilon_0^2}{\|H\|^2} \leq \beta^* \leq \frac{2}{\|H\|}. \tag{5.44}$$

Proof Consider the general case of $L_\beta = I - \beta H$ noting that the original operator L is regained if $\beta = 1$. For monotonicity of the modified iteration independent of

e_0, it is both necessary and sufficient that $\langle e_1, e_1\rangle_{\mathscr{Y}} < \langle e_0, e_0\rangle_{\mathscr{Y}}$ for all e_0. This is equivalent to the condition

$$L_\beta^* L_\beta = (I - \beta H)^*(I - \beta H) < I \tag{5.45}$$

which is just

$$H + H^* > \beta H^* H \tag{5.46}$$

which is positive. The need for $H + H^*$ to be strictly positive follows. If $\mathscr{Y}$ is finite dimensional, then H can be identified as a matrix for which strict positivity implies the existence of $\varepsilon_0 > 0$ in (5.43).
Next, if (5.43) holds,

$$L_\beta^* L_\beta = I - \beta(H + H^*) + \beta^2 H^* H \leq (1 - \beta\varepsilon_0^2 + \beta^2\|H\|^2)I < I \tag{5.47}$$

for all $\beta \in (0, \beta^{**})$ with $\beta^{**} = \varepsilon_0^2/\|H\|^2$ and the result follows by defining $\lambda^2(\beta) = 1 - \beta\varepsilon_0^2 + \beta^2\|H\|^2 < 1$ and noting that $\|L_\beta\| \leq \lambda(\beta) < 1$ for all $\beta \in (0, \beta^{**})$. Finally, it is clear that the largest possible choice of β^* is $\geq \varepsilon_0^2/\|H\|^2$. The required upper bound follows as $H + H^* > \beta H^* H$ requires that $H + H^* \geq \beta^* H^* H$ and hence, using the Cauchy Schwarz inequality, that $2\|H\| \geq \beta^*\|H\|^2$. □

5.2.2 *Infinite Dimensions with $r(L) = \|L\| = 1$ and $L = L^*$*

A simple summary of the above is that, if $r(L) < 1$, convergence of the iterations to zero is guaranteed but the resultant sequence $\{e_k\}_{k\geq 0}$ may not have monotonically decreasing norms. If, in addition, $\|L\| < 1$, then monotonic convergence is guaranteed. Although these conditions cover many potential applications, some, including the important case of optimal Iterative Control of strictly proper continuous time state space systems $S(A, B, C)$, will be seen in later chapters to satisfy the conditions

$$r(L) = \|L\| = 1 \tag{5.48}$$

and none of the results described above apply. This condition sits, intuitively, on the stability boundary of the iterative process but, as it has primary relevance to the case when $\mathscr{Y}$ is infinite dimensional, it has significant but less obvious consequences. More precisely, with the addition of an assumption that L (and hence $H = I - L$) is self adjoint, the following result adds considerable value,

Theorem 5.9 (Error Monotonicity Properties when $r(L) = \|L\| = 1$) *Suppose that $\mathscr{Y}$ is a real Hilbert space with inner product $\langle\cdot,\cdot\rangle_{\mathscr{Y}}$ and associated induced norm. Consider the iteration $e_{k+1} = Le_k,\ k \geq 0$ and assume that L is self adjoint, that $r(L) = 1$ and that L satisfies the inequality*

$$(-1+\mu_0^2)I \le L \le I \tag{5.49}$$

for some $\mu_0 > 0$. *Then* $r(L) = \|L\| = 1$ *and, for all* e_0 *in the closure* $\overline{\mathscr{R}[I-L]}$ *of the range of* $I - L$, *the iteration satisfies*

$$\|e_{k+1}\|_{\mathscr{Y}} < \|e_k\|_{\mathscr{Y}},\ k \ge 0 \text{ and } \lim_{k\to\infty} |\|L^k e_0\|_{\mathscr{Y}} = 0. \tag{5.50}$$

In addition,

1. *if* $e_0 \in \mathscr{R}(I-L)$, *then*

$$\sum_{k=0}^{\infty} \|e_k\|_{\mathscr{Y}}^2 < \infty \quad (Norm - squared\ Summability). \tag{5.51}$$

2. *For an arbitrary* $e_0 \in \mathscr{Y}$, *the iteration is monotonic with a limit*

$$\lim_{k\to\infty} e_k = e_\infty = P_{ker[I-L]} e_0 \tag{5.52}$$

where $P_{ker[I-L]}$ *is the self adjoint, positive, orthogonal projection operator onto* $ker[I-L]$.

Proof The conditions $(-1+\mu_0^2)I \le L \le I$ implies that $\|L\| \le 1$ and hence, as $r(L) \le \|L\|$, that $\|L\| = r(L) = 1$. Consider now $e_0 \in \mathscr{R}[I-L]$ and note that the condition $\mu_0 > 0$ ensures that the operator $(I+L)$ has a bounded inverse on $\mathscr{Y}$. It is hence possible to write $e_0 = (I-L)w_0$ and $w_0 = (I+L)w_1$ for some w_0 and w_1 in $\mathscr{Y}$. That is $e_0 = (I-L)(I+L)w_1 = (I-L^2)w_1$ and hence, for any $p \ge 0$, algebraic methods plus the use of the Cauchy-Schwarz inequality and $\|L\| = 1$ gives

$$\begin{aligned} &\textstyle\sum_{k=0}^{p} \|e_k\|^2 = \sum_{k=0}^{p} \langle L^k e_0, L^k e_0\rangle = \sum_{k=0}^{p} \langle e_0, L^{2k} e_0\rangle \\ &= \textstyle\sum_{k=0}^{p} \langle e_0, L^{2k}(I-L^2)w_1\rangle = \langle e_0, \left[\left(\sum_{k=0}^{p} L^{2k}\right)(I-L^2)\right] w_1\rangle \\ &= \langle e_0, (I-L^{2(p+1)})w_1\rangle \le \|e_0\|\ \|w_1\|\ (1+\|L\|^{2(p+1)}) = 2\|e_0\|\ \|w_1\|. \end{aligned} \tag{5.53}$$

Letting $p \to \infty$ proves that $\sum_{k=0}^{\infty} \|e_k\|^2$ converges and hence that $e_k \to 0$ in norm. Now suppose that $e_0 \in \overline{\mathscr{R}[I-L]}$ and note that the invariance condition $L\overline{\mathscr{R}[I-L]} \subset \overline{\mathscr{R}[I-L]}$ implies that all iterates lie in $\overline{\mathscr{R}[I-L]}$. Choose $\varepsilon > 0$ arbitrarily. It is always possible to write $e_0 = \tilde{e}_0 + \delta_0$ with $\tilde{e}_0 \in \mathscr{R}[I-L]$ and $\|\delta_0\| < \varepsilon$ and hence to note that

$$\limsup_{k\to\infty} \|L^k e_0\| \le \limsup_{k\to\infty} \|L^k \tilde{e}_0\| + \limsup_{k\to\infty} \|L^k \delta_0\| \le \varepsilon. \tag{5.54}$$

which proves that $e_k \to 0$ as ε was arbitrary. Each norm $\|e_{k+1}\| \le \|L\|\|e_k\| = \|e_k\|$ but, in fact, $\|e_{k+1}\| < \|e_k\|$ as, if equality holds for some $e_k \ne 0$, it follows that $\langle e_k, (I-L^2)e_k\rangle = 0$. As $I-L^2$ is self adjoint and positive, a square root argument then

indicates that $(I - L^2)e_k = 0$ from which $(I - L)e_k = 0$. That is $e_k \in ker[I - L] = \mathscr{R}[I-L]^{\perp}$ and hence $e_k = 0$ as $ker[I-L] \cap \overline{\mathscr{R}[I - L]} = \{0\}$ which is a contradiction.

Finally, if $e_0 \in \mathscr{Y}$ is chosen arbitrarily, the sum $\mathscr{Y} = ker[I - L] + \overline{\mathscr{R}[I - L]}$ is a direct sum of orthogonal subspaces. Write $e_0 = \tilde{e}_0 + \delta_0$ with $\tilde{e}_0 \in \overline{\mathscr{R}[I - L]}$ and $\delta_0 \in ker[I - L]$ uniquely defined. A simple calculation gives, for all $k \geq 0$,

$$e_k = L^k \tilde{e}_0 + \delta_0, \text{ and } \|e_k\|^2 = \|L^k \tilde{e}_0\|^2 + \|\delta_0\|^2 \tag{5.55}$$

As $L^k \tilde{e}_0 \to 0$ monotonically, the monotonic nature of $\|L^k e_0\|_{\mathscr{Y}}$ and the formula for the limit follow easily. □

When $e_0 \in \overline{\mathscr{R}[I - L]}$, the result demonstrates convergence properties of $\{e_k\}_{k\geq 0}$ in $\overline{\mathscr{R}[I - L]}$ which may be a proper subspace of $\mathscr{Y}$. As it is closed, it is a Hilbert space with the same inner product as that used in $\mathscr{Y}$. Also $L\overline{\mathscr{R}[I - L]} \subset \overline{\mathscr{R}[I - L]}$ so that the iteration is an iteration in $\overline{\mathscr{R}[I - L]}$ and the result can be interpreted as proving the monotonic iteration and monotonic convergence properties in that space.

The above result can be refined if $H = I - L > 0$ when $ker[I - L] = \{0\}$) and, as $ker[I - L] = \mathscr{R}[I - L]^{\perp}$,

Theorem 5.10 (Monotonic Convergence when $r(L) = \|L\| = 1$ *and* $H = I - L > 0$) *Using the same constructions and assumptions of Theorem 5.9, suppose also that*

$$(-1 + \mu_0^2)I \leq L < I. \tag{5.56}$$

Then the conclusion of Theorem 5.9 remain valid but, in addition,

$$\overline{\mathscr{R}[I - L]} = \mathscr{Y} \tag{5.57}$$

and hence monotonic convergence on $\mathscr{Y}$ is achieved.

Note: Monotonicity in $\mathscr{Y}$ hence needs the range of $H = I - L$ to be dense.

5.2.3 Relaxation, Convergence and Robustness

The spectrum and properties of L are important properties in the behaviour of Iterative Algorithms in terms of convergence and robustness. In this section, a simple modification to iteration update formulae will provide a means of ensuring convergence properties at the price of convergence to non-zero limit errors. More precisely, using the notation Sect. 5.2.2, suppose that $L = I - H$, that L is not necessarily self adjoint but that it has a spectrum in the closed ball

$$spec[L] \subset \mathscr{B}_c(\frac{1}{2}\mu_0^2; 1 - \frac{1}{2}\mu_0^2)) \text{ , for some } \mu_0^2 > 0. \tag{5.58}$$

A simple calculation indicates that the spectral radius $r(L) \leq 1$ and that the only possible point on the unit circle that could be in the spectrum is the point $z = 1$. Note that none of the previously described results applies to this case unless $r(L) < 1$. The loss of self-adjoint properties will occur depending on the choice of K_0 and the effects of any modelling errors in plant dynamics.

If $r(L) = 1$, then a simple mechanism for modifying the iteration to ensure the convergence of the algorithm is to use a *relaxation* technique. More precisely, replace the feedforward input update rule by the formula

$$u_{k+1} = \alpha u_k + K_0 e_k \ , \ k \geq 0 \ , \ (\textit{A Relaxed Input Update Rule}). \tag{5.59}$$

The parameter $\alpha > 0$ is a *relaxation parameter* and has a default value of $\alpha = 1$ that is used almost exclusively in this text. To examine the effect of α on algorithm performance, use the dynamics $y = Gu + d$ to obtain the *relaxed error evolution*,

$$e_{k+1} = Le_k + (1-\alpha)(r-d), \qquad \textit{where, now,} \quad L = \alpha I - H \textit{ and } H = GK_0. \tag{5.60}$$

The spectral mapping theorem now places the spectrum of $L = (\alpha - 1)I + (I - H)$ in $(\alpha - 1) + \mathscr{B}_c(\frac{1}{2}\mu_0^2; 1 - \frac{1}{2}\mu_0^2))$ and hence a sufficient condition for the spectral radius $r(L) < 1$ is that α lies in the range

$$1 - \mu_0^2 < \alpha < 1 \ , \ (\textit{the "Permissible Range" of } \alpha). \tag{5.61}$$

That is, the effect of α is to shift the spectrum of L into the open unit circle. The price paid is that the error evolution now has the additional term $(1 - \alpha)(r - d)$ so that any converged error must be non-zero unless $r = d$ when the solution $u = 0$ is obtained easily and iteration is not needed.

Theorem 5.11 (Convergence and Relaxation Techniques) *With the assumptions of the discussion preceding this result, the error evolution with the relaxed input update rule and a relaxation parameter α in its "permissible range" will converge to a limit error e_∞^α for all $r \in \mathscr{Y}$. The value of e_∞^α satisfies*

$$(1 - \alpha)(r - d) = [(1 - \alpha)I + H]e_\infty^\alpha. \tag{5.62}$$

If, in addition, $\mathscr{Y}$ is a real Hilbert space and H is such that $H + H^* \geq \varepsilon_0^2 I$ for some $\varepsilon_0 > 0$, then

$$\|e_\infty^\alpha\|_{\mathscr{Y}} \leq \|r - d\|_{\mathscr{Y}} \textit{ and } \lim_{\alpha \to 1-} e_\infty^\alpha = 0 \textit{ in the norm topology in } \mathscr{Y}. \tag{5.63}$$

If no such value of ε_0 exists and the output space $\mathscr{Y}$ has the direct sum decomposition

$$\mathscr{Y} = \overline{\mathscr{R}[H]} \oplus ker[H], \tag{5.64}$$

then $\alpha I - H$ maps $\overline{\mathscr{R}[H]}$ into itself. If $H + H^* > 0$ on $\overline{\mathscr{R}[H]}$, then, writing $r - d = r^{(1)} + r^{(2)}$ with $r^{(1)} \in \overline{\mathscr{R}[H]}$ and $r^{(2)} \in ker[H]$, the limit error converges to $r^{(2)}$ as $\alpha \to 1-$ in the weak topology in $\mathscr{Y}$.

Proof Convergence follows, with a change in notation, from Theorem 5.3 and 5.4 noting that, as the point $(1, 0)$ is not in the spectrum of $L = \alpha I - H$, it follows that $\mathscr{R}[I - L] = \mathscr{Y}$. Next, using the Cauchy-Schwarz inequality,

$$\begin{aligned} &(1-\alpha)\|r-d\|\|e_\infty^\alpha\|_{\mathscr{Y}} \geq \langle e_\infty^\alpha, (1-\alpha)r\rangle_{\mathscr{Y}} \\ &= \langle e_\infty^\alpha, [(1-\alpha)I + H]e_\infty^\alpha\rangle_{\mathscr{Y}} \geq (1-\alpha)\|e_\infty^\alpha\|_{\mathscr{Y}}^2 + \tfrac{1}{2}\varepsilon_0^2\|e_\infty^\alpha\|_{\mathscr{Y}}^2 \geq 0 \end{aligned} \tag{5.65}$$

which proves that $\|e_\infty^\alpha\| \leq \|r - d\|$ for all permissible α and the limit formula when $\varepsilon_0 > 0$. Finally, if $H + H^* > 0$, write $e_k = e_k^{(1)} + e_k^{(2)}$ with $e_k^{(1)} \in \overline{\mathscr{R}[H]}$ and $e_k^{(2)} \in ker[H]$ and deduce that, for $k \geq 0$,

$$e_{k+1}^{(1)} = (\alpha I - H)e_k^{(1)} + (1-\alpha)r^{(1)} \text{ and } e_k^{(2)} = \alpha e_k^{(2)} + (1-\alpha)r^{(2)}. \tag{5.66}$$

Convergence of the component $e_k^{(2)}$ to $r^{(2)}$ in norm follows as $|\alpha| < 1$. For the iteration in the Hilbert (sub-)space $\overline{\mathscr{R}[H]}$, let $\{\alpha_j\}_{j\geq 1}$ be a sequence of relaxation parameters converging to unity from below. Without loss of generality, weak compactness of any closed ball indicates that the sequence $\{e_\infty^{\alpha_j}\}_{j\geq 0}$ has a weak limit $e_\infty^{(weak)}$ and, for all $f \in \overline{\mathscr{R}[H]}$, that $\langle f, He_\infty^{(weak)}\rangle_{\mathscr{Y}} = 0$. Choosing $f = e_\infty^{(weak)}$, the condition $H + H^* > 0$ leads to the conclusion that $e_\infty^{(weak)} = 0$ and hence that the limit error sequence converges weakly to zero as $\alpha \to 1-$. □

Conclusion: Benefits of Relaxation Methods: Relaxation concepts provide a stabilizing modification for applications where the non-relaxed algorithm leads to the condition $r(L) = 1$.

1. Perhaps the most important potential application of the technique is in the infinite dimensional case when, in the absence of relaxation, the condition $r(L) = 1$ may hold but L is not self-adjoint. Introducing relaxation brings the spectrum into the open unit circle and guarantees convergence to some limit albeit a non-zero one. The form of the limit error will depend on the choice of α and the nature of the dynamics although the general rule that α should be close to unity is necessary for the limit error to be small.
2. In convergence and robustness studies, the results provide a way forward to useful convergence as a two step process with the initial objective of ensuring that $r(L) \leq 1$ followed by a choice of α to pull the spectrum into the open unit circle.
3. The value of μ_0^2 may not be known in a given situation but a bound can be deduced from a norm on H in some circumstances. More precisely, using the fact that, for any real scalar λ, the spectrum of H lies in any closed ball $\mathscr{B}_c(\lambda; \|H - \lambda I\|)$, the reader will be able to verify that, if $|1 - \lambda| + \|H - \lambda I\| \leq 1$, then $r(L) \leq 1$ and any value of μ_0^2 in the range

$$0 < \mu_0^2 < 2 - \lambda - \|H - \lambda I\| \tag{5.67}$$

can be used. The norm of $H - \lambda I$ can be computed in a number of cases. For example, using the frequency domain bounds available for linear, discrete (and continuous), time-invariant, state space systems when identification of H with $S(A, B, C, D)$ leads to the identification of $H - \lambda I$ with $S(A, B, C, D - \lambda I_m)$.

4. The form of the limit error e_∞^α may need careful consideration. Theorem 5.11 indicates that the strict positivity of H is sufficient to provide some control over the limit by the choice of α close to unity. Some insight into the issues can be seen if H is a linear state space system $S(A, B, C, D)$.

   a. For discrete systems the strict positivity of the operator is guaranteed if the associated transfer function matrix $H(z)$ is strictly positive real. Using $\mathscr{Z}$-transform notation by extending signals to $0 \leq t < \infty$,

   $$e_\infty^\alpha(z) = (I + (1 - \alpha)^{-1} H(z))^{-1} (r(z) - d(z)). \tag{5.68}$$

   That is, the limit error on $0 \leq t \leq N$ can be computed from the negative feedback system where $H(z)$ is subjected to a "gain" $(1 - \alpha)^{-1}$. This is a high gain system if α is close to unity. The requirement that the error is stable underlines the importance of the relative degree of $H(z)$ (and hence the shift techniques discussed in the modelling chapter), the problems associated with non-minimum-phase zeros and the benefit arising from making $H(z)$ strictly positive real. In addition, the reference signal r also clearly has an influence.

   b. For continuous state space systems, positivity of the associated operator is guaranteed if the transfer function matrix $H(s)$ is strictly positive real. The underlying output space is now infinite dimensional and convergence is achieved but behaviour of the limit error as $\alpha \to 1-$ is now governed by weak convergence rules. For functions on $[0, T]$, this is equivalent to convergence (to zero) of all coefficients in the Fourier series expansion, a fact that suggest that any anomalous limit behaviours will be associated with high frequency components.

5. Finally, the behaviour of the limit error e_∞^α as α varies is seen to be very different in the two cases of $\alpha = 1$ and $\alpha \to 1-$. In the second case, the effect of the initial input choice u_0 is seen to disappear. In effect, the algorithm *"forgets"* its starting condition and for this reason, α could be called a *"forgetting factor"*. In practice, these subtle differences may play little role as α will be close to unity and only a finite number of iterations will be undertaken until a desired tracking accuracy has been achieved.

5.2.4 Eigenstructure Interpretation

Using the notation of the previous sections, consider the iterations $e_{k+1} = Le_k$ in a real Hilbert space $\mathscr{Y}$ and write $L = I - \beta H$ with $\beta > 0$. Again assume that L is self adjoint and, in particular, that H is self adjoint and positive so that $\beta^{-1}(I - L) = H = H^* \geq 0$. More detail of iteration performance can, in principle, be obtained by analysing the situation where, either

1. $\mathscr{Y}$ is finite dimensional (when H can be regarded as a matrix) or,
2. H is a compact (completely continuous) operator on an infinite dimensional space $\mathscr{Y}$.

The common property of these cases is that H then has a complete set of linearly independent, orthonormal eigenvectors $v_1, v_2, \ldots$ with positive eigenvalues $r(H) = \|H\| = \lambda_1 \geq \lambda_2 \geq \ldots$. The eigenvalues are the spectral values of H and, in the infinite dimensional case, all non-zero eigenvalues are isolated points and $\lambda = 0$ is either an eigenvalue and/or a cluster point of the spectrum. That is

$$Hv_j = \lambda v_j, \ \textit{and} \ \langle v_i, v_j \rangle = \delta_{ij}, \tag{5.69}$$

where δ_{ij} is the Kronecker delta. It follows that

$$(I - \beta H)v_j = (I - \beta\lambda_j)v_j, \ j \geq 0 \tag{5.70}$$

and hence that L has eigenvalues $\{1 - \beta\lambda_j\}_{j\geq 1}$ and eigenvectors $\{v_j\}_{j\geq 0}$. Note that L is positive definite if, and only if, all eigenvalues are strictly positive.

Using the completeness assumption, any initial error e_0 can be written in the form $e_0 = \sum_{j\geq 1} \gamma_j v_j$ where $\gamma_j = \langle v_j, e_0 \rangle, \ j \geq 1$, and

$$\|e_0\|^2_{\mathscr{Y}} = \sum_{j\geq 1} \gamma_j^2 < \infty. \tag{5.71}$$

The value of γ_j is a measure of the contribution made by the signal v_j to the initial error. By analogy with frequency analysis of signals, it is expected that, often, the contribution will vary from being substantial (the "low frequency" components) and very small (the "high frequency" components).

A simple calculation gives, for any $k \geq 0$,

$$e_k = L^k e_0 = \sum_{j\geq 1} \gamma_j (1 - \beta\lambda_j)^k v_j \ \textit{with} \ \|e_k\|^2 = \sum_{j\geq 1} \gamma_j^2 (1 - \beta\lambda_j)^{2k} \tag{5.72}$$

from which the evolution of the different eigenvectors can be seen. More precisely, the contribution of v_j evolves with magnitude $\gamma_j(1 - \beta\lambda_j)^k, \ j \geq 1$, and it follows that

1. convergence to zero error for all initial error e_0 is achieved if, and only if, all $\lambda_j > 0$ (that is, H is positive definite) and $\beta\|H\| < 2$.
2. More generally, the self adjoint nature of H indicates that

$$\mathscr{Y} = kerH \oplus \overline{\mathscr{R}[H]} \tag{5.73}$$

the second component of which is the closure of the subspace generated by arbitrary linear combinations of the eigenvectors with non-zero eigenvalues. As a consequence, the limit of the iterative process is

$$\lim_{k\to\infty} e_k = e_\infty = \sum_{\lambda_j=0} \gamma_j v_j. \tag{5.74}$$

3. Speed of convergence depends on both the spread and magnitude of the eigenvalues and the values of γ_j.
   a. Convergence is rapid for components v_j where $(1 - \beta\lambda_j)$ is close to zero.
   b. Convergence is slow for components where $(1-\beta\lambda_j)$ is either close to unity or the value -1. Slow convergence of these components can only generate small tracking errors quickly if they start from already small magnitudes γ_j. That is, rapid practical convergence (in a small number of iterations) to small error norm values will only be achieved if the initialization produces an error e_0 dominated by contributions made by eigenvectors with the largest eigenvalues.

The above discussion can be read as being simple statements of matrix theory. For application to, for example, discrete supervector models, this is the case but it is important to note that infinite dimensional examples do exist where the eigenstructure assumption is satisfied, although the calculation of the eigenvalues and eigenvectors may be impossible or, at best, very difficult.

5.2.5 *Formal Computation of the Eigenvalues and Eigenfunctions*

The purpose of this section is to illustrate the complexities and nature of eigenstructure computation and provide, by example, some evidence that eigenstructure assumptions can be valid for infinite dimensional systems in state space form. The general nature of the eigenvalue equation $Hv = \lambda v$ (where solutions λ such that $v \neq 0$ are sought) can be envisaged when $H = GG^*$ and G is described by linear, time-invariant, state space model with zero initial conditions. Firstly note that the adjoint operator is then described by linear, time-invariant, state space models with zero terminal condition at an end-time T. To illustrate the issues that arise in the eigenvalue analysis of the problem, a simple example is considered below, namely computing the eigenvalues and eigenfunctions of GG^* where G is the operator form of

the linear, time-invariant, continuous time system $S(A, B, C)$. The eigenvalue equation is simply the equation $GG^*v = \lambda v$ for some non-zero vector v. That is, writing $y = \lambda v,\ y = Gu,\ u = G^*v$,

$$\begin{array}{c} y(t) = \lambda v(t) \quad \textit{where} \quad , \\ \dot{x}(t) = Ax(t) + Bu(t)\ ,\ x(0) = 0\ ,\ \ y(t) = Cx(t), \\ \dot{p}(t) = -A^T p(t) - C^T Q v(t)\ ,\ p(T) = 0\ ,\ \ u(t) = R^{-1}B^T p(t). \end{array} \tag{5.75}$$

First note that the case of $\lambda = 0$ corresponds to solutions $v \in ker[G^*]$. The remaining cases of $\lambda > 0$ are considered by writing this expression in the form

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ p(t) \end{bmatrix} = \begin{bmatrix} A & BR^{-1}B^T \\ -\lambda^{-1}C^T QC & -A^T \end{bmatrix}\begin{bmatrix} x(t) \\ p(t) \end{bmatrix} = \mathscr{H}(\lambda)\begin{bmatrix} x(t) \\ p(t) \end{bmatrix} \tag{5.76}$$

where $\mathscr{H}(\lambda)$ is defined naturally. The initial conditions are $x(0) = 0$ with $p(0)$ to to be found to ensure that $p(T) = 0$. This equation has the formal solution in terms of the matrix exponential $e^{\mathscr{H}(\lambda)t}$, written as

$$e^{\mathscr{H}(\lambda)t} = \begin{bmatrix} E_{11}(t, \lambda)\ E_{12}(t, \lambda) \\ E_{21}(t, \lambda)\ E_{22}(t, \lambda) \end{bmatrix}. \tag{5.77}$$

Using the initial conditions, it follows that the equation defining the non-zero eigenvalues and the consequent (un-normalized) eigenvectors are solutions of the equation defining initial conditions $p(0)$ that produce the desired terminal condition $p(T) = 0$,

$$p(T) = E_{22}(T, \lambda)p(0) = 0,\ p(0) \neq 0,\ v_\lambda(t) = CE_{12}(t, \lambda)\ p(0). \tag{5.78}$$

In particular, the eigenvalues are precisely the solutions of the nonlinear equation defined by the determinant

$$|E_{22}(T, \lambda)| = 0. \tag{5.79}$$

Formulae for the solutions and the forms of the eigenfunctions are not available. It is known that all solutions λ are real and positive and, from the matrix exponential form for linear time-invariant systems, the eigenfunctions consist of sums of products of exponential, polynomial and trigonometric functions associated with the eigenvalues of $\mathscr{H}(\lambda)$. To illustrate the issues, a simple example is now described:

Example: Consider the simple case of $m = \ell = 1$ with $A = 0, B = C = Q = R = 1$ when, writing $\lambda = \omega^{-2}$,

$$\mathscr{H}(\lambda) = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \textit{ and hence } e^{\mathscr{H}(\lambda)t} = \begin{bmatrix} \cos(\omega t) & \omega^{-1}\sin(\omega t) \\ -\omega\sin(\omega t)) & \cos(\omega t) \end{bmatrix}. \tag{5.80}$$

The relevant values of ω are obtained from the solutions of $\cos(\omega T) = 0$. That is, there are an infinite number of solutions parameterized by the relation $\omega_k = \frac{\pi(2k+1)}{2T},\ k =$

0, 1, 2, 3, ... This gives the eigenvalues and associated (un-normalized) eigenfunctions as

$$\lambda_k = \left(\frac{2T}{\pi(2k+1)}\right)^2,\ v_k(t) = \sin\left(\frac{\pi(2k+1)t}{2T}\right) \text{ for } k = 0, 1, 2, 3, \ldots \\ \text{with } \lambda_1 > \lambda_2 > \lambda_3 > \ldots \text{ and } \lim_{k\to\infty} \lambda_k = 0. \tag{5.81}$$

In particular, the eigenfunctions are sinusoidal functions with frequencies spanning low to high frequencies in a similar manner to the components in Fourier series. The important point to note is that $\{v_k\}_{k\geq 0}$ form a basis for $\mathscr{Y} = L_2[0, T]$. To prove this, note that each point in $f \in L_2[0, T]$ can be associated uniquely with a function f_e in $L_2[-2T, 2T]$ via the construction $f_e(2T - t) = f(t),\ \forall\, t \in [0, T]$ and $f_e(-t) = -f_e(t),\ \forall\, t \in [0, 2T]$. Being odd functions they can be uniquely expressed as a Fourier series with only sinusoidal terms $\{\sin\left(\frac{2p\pi\ t}{4T}\right)\}_{p\geq 1}$. The $\{v_k\}_{k\geq 0}$ are simply the terms in this set with $p = 2k + 1$ (i.e. the odd index values). The property $f(2T - t) = f(t),\ \forall\, t \in [0, T]$ ensures that these are the only non-zero terms in the representation of f_e. It is concluded that any function $f \in L_2[0, T]$ can be expressed as a uniquely defined linear combination of the $\{v_k\}_{k\geq 0}$ as stated at the beginning of the discussion. Note that, as a by-product of the analysis, it follows that $ker[G^*] = \{0\}$.

5.3 Robustness, Positivity and Inverse Systems

This short section anticipates some of the developments of following chapters. It aims to help the reader more fully understand the structure of robustness analyses by pointing out a particular robustness condition that is common to many important cases. Some of the details for each topic are left for the relevant section but it is useful to identify the pattern before starting on that journey. For this reason, some of the statements made below will be proved in later chapters. Let $\mathscr{Y}$ and $\mathscr{U}$ be real Hilbert spaces and $P : \mathscr{U} \to \mathscr{Y}$ a linear, bounded operator with adjoint P^*. Suppose that an iterative process in $\mathscr{Y}$ is described by a recursion, for $k \geq 0$,

$$e_{k+1} = L_e e_k,\ \text{where } L_e = I - \varepsilon^{-2} PP^*(I + \lambda PP^*)^{-1},\ \varepsilon^2 > 0 \text{ and } \lambda \geq 0. \tag{5.82}$$

Then $L_e = L_e^*$ and

$$\left(1 - \frac{\varepsilon^{-2}\|P^*\|^2}{1 + \lambda\|P^*\|^2}\right) I \ \leq\ L_e \ \leq\ I. \tag{5.83}$$

In particular, L is invertible if the left-hand-side is strictly positive. One consequence of this assumption is that the norm $\|e\|_{\mathscr{Y}}$ in $\mathscr{Y}$ is topologically equivalent to the norm $\|e\|_0$ induced by the inner product $\langle e, w\rangle_0 = \langle e, (I + \lambda PP^*)^{-1} w\rangle_{\mathscr{Y}}$. Note, in particular that $\|e_k\|_0 \to 0$ as $k \to \infty$ if, and only if, $\|e_k\|_{\mathscr{Y}} \to 0$.

Let $U : \mathscr{U} \to \mathscr{U}$ be a linear bounded operator and consider the perturbed iteration

$$e_{k+1} = L_U e_k, \ \textit{where } L_U = I - \varepsilon^{-2} PUP^*(I + \lambda PP^*)^{-1}. \tag{5.84}$$

Note that, in general, $L_U \neq L_U^*$. In future material, regarding U as a right multiplicative modelling error of the plant model G, the reader should note that, amongst other examples,

1. the case of $P = I$, $\lambda = 0$ and $\beta = \varepsilon^{-2}$ forms the basis of robustness of the Left Inverse Model Algorithm 6.2 for left-invertible systems (Chap. 6).
2. Using the data $P = G$, $\lambda = 0$ and $\beta = \varepsilon^{-2}$ forms the basis of robustness analysis of the Steepest Descent Algorithm 7.2 described in Chap. 7.
3. The data $P = G$ and $\lambda = \varepsilon^{-2}$ forms the basis of robustness analysis of the Norm Optimal Iterative Learning Control (NOILC) Algorithm 9.1 in Chap. 9.

The robustness question is formulated here as the requirement that, for any choice of e_0, the resultant sequence $\{e_k\}_{k\geq 0}$ remains bounded and the monotonicity condition

$$\|e_{k+1}\|_0 \leq \|e_k\|_0 \ , \ \textit{for all } k \geq 0, \tag{5.85}$$

is satisfied. This can be written in the form $L_U^*(I + \lambda PP^*)^{-1} L_U \leq (I + \lambda PP^*)^{-1}$ which is just

$$(I + \lambda PP^*)^{-1} P \underbrace{\left(U + U^* - \varepsilon^{-2} U^* P^* (I + \lambda PP^*)^{-1} PU\right)} P^* (I + \lambda PP^*)^{-1} \geq 0. \tag{5.86}$$

A sufficient condition for this to hold is that U and P satisfy the condition

$$U + U^* \geq \varepsilon^{-2} U^* P^* (I + \lambda PP^*)^{-1} PU \tag{5.87}$$

with respect to the original topology in $\mathscr{Y}$. Robust monotonicity is hence typically connected to positivity conditions on U although the nature of P and parameters such as ε^2 and λ play a role.

The inequality can be further relaxed to produce other sufficient conditions. For example, it is satisfied if, either

$$\begin{array}{lllll} (A) & U + U^* & \geq & \beta U^* U \ , & \beta = \frac{\varepsilon^{-2}\|P\|^2}{1+\lambda\|P\|^2}, \ \textit{or} \\ (B) & U + U^* & \geq & \beta U^* P^* P U \ , & \beta = \varepsilon^{-2}. \end{array} \tag{5.88}$$

The first of these is closely related to the robustness conditions for Inverse Model Algorithms in Chap. 6 whilst the second is closely related to those for Steepest Descent Algorithms in Chap. 7. Both can be applied to the NOILC Algorithm described in Chap. 9 as well as some of its extensions in Chap. 10. These observations demonstrate that the robustness of these algorithms can be underpinned by an analysis of Inverse and Descent Algorithms and suggests that all such algorithms, in some sense, are approximate inversion processes.

The above material has been included here for presentational reasons. The reader will see it in various guises in many places in the rest of the text but with more detail

and refinement, particularly related to the replacement of $\leq$ in the monotonicity requirement by strict inequality $<$ and, for discrete, linear, state space systems, the conversion of the criteria into frequency domain inequalities that, in principle, can be tested by the user. Here, the material in Sect. 4.8 will play a role.

5.4 Discussion and Further Reading

The general formulation of the Iterative Learning Control problem has been expressed in a way that meets the needs of the following chapters and the focus on quadratic optimization as a tool in algorithm development. There are several ways of formulating the problem but the approach chosen here covers many cases of interest and, in particular, the approaches that form the foundations of Chap. 9 and its extensions (Chap. 10). Other approaches can be found in [2, 3, 17, 19, 25, 68, 78, 88, 93, 100, 111, 113 and 114]. It is important to note that the formulation inevitably requires modification from time to time to fit applications needs or the nature of the control philosophy used. Some of these changes will appear later in this text and include the introduction of auxiliary variables in Chap. 11, additional control criteria, constraints (Chap. 12) and iteration dependent parameters as in the development of Parameter Optimal Iterative Learning Control (Chap. 14).

The use of a relatively abstract presentation is in the spirit of the mathematical texts such as Collatz [28] and Ortega and Rheinboldt [79] which discuss numerical iterations for solving sets of algebraic equations. The advantage of the approach is a clear indication of the generality of the algorithms and hence the wide range of applications that could be considered. This is precisely why this approach is taken here for Iterative Control as systems can be regarded as mappings on input data and tracking requirements defined by equations of the form

$$r = \mathscr{G}(u, d) \tag{5.89}$$

Given r and d, the iterative solution of this (normally dynamic) relationship for the "solution" u has a structural similarity to equation solving problems in general. The important point is that it allows many specific interpretations from discrete and continuous state space relationships, to multi-rate sampled system models and other more complex dynamical structures including delays, integral operators and differential-algebraic systems. Operator theoretical tools have already been used with some success on the problems of Multipass Processes in Edwards and Owens [34] and, in part, for the related area of repetitive systems [99] although the results provided in this text go well beyond work in these areas.

Attention has focussed on linear iterations as this is consistent with classical control theory and design and the engineering philosophy of undertaking analysis and some modelling based on simplified linear models of plant behaviour. In the author's view, linear analysis provides practical solutions with the bonus of providing some familiar links to feedback theory where the mathematical description has strong con-

nections to performance of the algorithms when applied to real world processes. The core requirement is convergence of iterations which, for linear systems, is described by a combination of circumstances depending on the values and properties of the spectral radius and induced norm of an operator L characterizing the evolution of the tracking error from iteration to iteration. For the purposes of this text and the optimization related algorithms described in later chapters, the concept of monotonic iteration is central to the idea of improving behaviours from iteration to iteration and integrates well into the objective of achieving convergence. This practical benefit has theoretical consequences in the three cases of interest

$$r(L) < 1 \; or \; \|L\| < 1 \; or \; r(L) = \|L\| = 1. \tag{5.90}$$

For applications, these general criteria are ideally converted to more easily understood relationships between familiar system properties or parameters. This is achieved in this text for linear, time invariant, state space models using matrix methodologies and/or frequency domain inequalities.

The eigenstructure interpretation of the situation when relevant operators are self adjoint and positive provides interesting insights into iteration dynamics and links the ideas to more familiar effects of eigenvalue positions for discrete state space systems. Most of the mathematics can be viewed, in finite dimensions, as being consequences of familiar matrix theory but, for infinite dimensional spaces, properties such as operator compactness [31] have considerable technical value and interest in providing a conceptual interpretation that parallels the use of Fourier series representations of functions on finite time intervals [105, 106].

Finally, the use of relaxation techniques has parallels in the area of numerical analysis as a means of accelerating or controlling convergence rates. For Iterative Control, the presentation focusses on their use as aids to ensure convergence in the case when, in the absence of relaxation, $r(L) = 1$. The technique therefore has a long history in numerical mathematics and, given the observed benefits, a place in Iterative Control algorithm development. Readers might note that it is a special case of, for example, more general *feedforward* algorithms of the form

$$u_{k+1} = F_U u_k + K_0 e_k, \tag{5.91}$$

where $F_U : \mathscr{U} \to \mathscr{U}$ is a filter designed to influence convergence rates, the frequency content of input iterates and/or to reduce the effect of noise in the input signal. Note that, whenever $F_U \neq I$ (or $\alpha \neq 1$), convergence to a limit error may be achieved but it is unlikely to be zero and may not have a form that is desirable for the application.

Chapter 6
Control Using Inverse Model Algorithms

Iterative Learning Control has a close conceptual relationship to the notion of an inverse model of the plant. The usual notion of a system is that of an input signal u producing a uniquely defined output signal y via a relationship $y = \mathscr{G}(u, d)$. In contrast, the idea of an *inverse model* regards the plant input as the response to the plant output by assuming the existence of a relationship $u = \mathscr{H}(y, d)$ obtained "by solving for u as a function of y and d". The link to the idea of inverse is strengthened by noting that, for example,

$$u = \mathscr{H}(\mathscr{G}(u, d), d). \tag{6.1}$$

Given such a relationship, the input $u_\infty = \mathscr{H}(r, d)$ is a solution of the Iterative Learning Control problem. For a linear system $y = Gu + d$ where G has an inverse G^{-1}, the inverse system is simply $u = G^{-1}(y - d)$ and $u_\infty = G^{-1}(r - d)$.

6.1 Inverse Model Control: A Benchmark Algorithm

Consider now a linear system and an Iterative Control algorithm defined by the feedforward input update rule $u_{k+1} = u_k + K_0 e_k$. Suppose also that there is no modelling error. Two choices of K_0 are of interest in this section. They are both closely related to ideas of inverse systems and are constructed as follows.

6.1.1 Use of a Right Inverse of the Plant

The operator G may not have an inverse but the idea of inversion is still relevant by using the idea of *Right Inversion*. More precisely, a *right inverse* of G is a linear, bounded operator $G_R : \mathscr{Y} \to \mathscr{U}$ that satisfies $GG_R y = y$ for all $y \in \mathscr{Y}$ i.e.

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_6

$$GG_R = I. \tag{6.2}$$

For the feedforward input update rule $u_{k+1} = u_k + K_0 e_k$, the error update relationship is $e_{k+1} = (I - GK_0)e_k$. This expression forms the basis of the following algorithm stated with proven properties in the form.

Algorithm 6.1 (*A Right Inverse Model Algorithm*) Suppose that G has a right inverse G_R that is both linear and bounded and set $K_0 = \beta G_R$ where β is a real scalar "learning gain". Consider the "Right Inverse Model" input update law

$$u_{k+1} = u_k + \beta G_R e_k. \tag{6.3}$$

Then, the error evolution in the resultant Iterative Control algorithm satisfies

$$e_{k+1} = (1 - \beta)e_k \;\; \textit{and hence} \;\; e_k = (1 - \beta)^k e_0 \tag{6.4}$$

for all $k \geq 0$. In particular, the error e_k converges to zero for all initial errors e_0, if and only if,

$$0 < \beta < 2. \tag{6.5}$$

Under these conditions, the error norm is monotonically decreasing with $\|e_{k+1}\| = |1 - \beta|\|e_k\| < \|e_k\|$ for all $k \geq 0$.

Proof The error evolution equations follow easily as $e_{k+1} = (I - \beta GG_R)e_k = (1 - \beta)e_k$ using the properties of G_R. This indicates that $\|e_{k+1}\| = |1 - \beta|\|e_k\|$. The range of β needed for stability then follows from the requirement that $|1 - \beta| < 1$ as does the monotonicity property of the norms. □

Note that, in particular, all iterates are proportional to the initial error e_0. Also the rate of convergence is controlled by choosing the value of β. For example, choosing $\beta = 1/2$ will lead to halving of the error each iteration. More generally, values of β close to 2 produce slow convergence and an oscillatory error signal sequence whilst values of β close to zero produce slow convergence without oscillation. Values of β close to unity produce fast convergence with convergence in one iteration if $\beta = 1$. The choice of β is a design choice based upon the desired convergence rate and, as will be seen in following sections, to the degree of robustness to modelling error that is required.

Existence of a Bounded Right Inverse: A *necessary* condition for the existence of a right inverse G_R is that G has range equal to $\mathscr{Y}$, written as

$$\mathscr{R}[G] = \mathscr{Y}. \tag{6.6}$$

The reader can construct a proof by supposing that this relationship is not satisfied and yet G has a right inverse G_R. Choosing y that does not lie in $\mathscr{R}[G]$ then leads to a contradiction as $GG_R = I$ implies that $y = GG_R y \in \mathscr{R}[G]$. That this condition

is also sufficient for cases when both $\mathscr{Y}$ and $\mathscr{U}$ are finite dimensional is proved by assuming that G is a matrix with full row rank and noting that $QG^T(GQG^T)^{-1}$ is a right inverse for any nonsingular matrix Q. If G is non-square, then the right inverse is non-unique.

6.1.2 Use of a Left Inverse of the Plant

A similar algorithm can be constructed for linear plants with bounded left inverses although, in this case, the form of the initial error e_0 must be constrained if convergence to zero is to be guaranteed. The relevant assumption is that $G : \mathscr{U} \to \mathscr{Y}$ has a *left inverse operator* $G_L : \mathscr{Y} \to \mathscr{U}$ that is both linear and bounded and satisfies

$$G_L G = I. \tag{6.7}$$

It is easily verified that $(GG_L)^2 = GG_LGG_L = G(G_LG)G_L = GG_L$ from which it is easily seen that $(GG_L)^k = GG_L$ for all $k \geq 1$.

Existence of a Bounded Left Inverse: A *necessary* condition for such a left inverse to exist is that

$$ker[G] = \{0\}, \tag{6.8}$$

a condition that is proved by assuming that $ker[G] \neq \{0\}$, choosing a non-zero $u \in ker[G]$ and noting that a contradiction is achieved from the equation $u = G_LGu = 0$. That this condition is also sufficient for cases when both $\mathscr{Y}$ and $\mathscr{U}$ are finite dimensional is proved by assuming that G is a matrix with full column rank and noting that $(G^TQG)^{-1}G^TQ$ is a left inverse for any nonsingular matrix Q. If G is non-square, then the left inverse is non-unique.

The following statement defines a left inverse based algorithm and its properties.

Algorithm 6.2 (*A Left Inverse Model Algorithm*) Suppose that G has a left inverse G_L that is both linear and bounded. Consider the "Left Inverse Model" input update law

$$u_{k+1} = u_k + \beta G_L e_k. \tag{6.9}$$

Then, the error evolution in the resultant Iterative Control algorithm satisfies

$$\begin{aligned} e_{k+1} &= (1 - \beta GG_L)e_k \qquad \textit{and hence} \\ e_k &= (1 + \Big[(1-\beta)^k - 1\Big] GG_L)e_0 \end{aligned} \tag{6.10}$$

for all $k \geq 0$. In particular, if $e_0 \in \mathscr{R}[G]$, the error evolution is described by $e_{k+1} = (1 - \beta)e_k$ for all $k \geq 0$ and the error converges to zero, if and only if, $0 < \beta < 2$. The error norm sequence is then monotonically decreasing to zero.

Proof The first error evolution equations follow easily by operating on the input update law with G whilst the second follows by an inductive argument assuming that $e_k = (I + \gamma_k GG_L)e_0$. This is trivially true for $k = 1$ with $\gamma_1 = -\beta = (1 - \beta) - 1$ and, if true for k, the property $G_L G = I$ leads to the calculation

$$e_{k+1} = (I - \beta GG_L)e_k = (I - \beta GG_L)(I + \gamma_k GG_L)e_0 = (I + \gamma_{k+1} GG_L)e_0 \quad (6.11)$$

where $\gamma_{k+1} = (1 - \beta)\gamma_k - \beta$. The equation $\gamma_k = (1 - \beta)^k - 1$ for all $k \geq 1$ is then easily deduced by an inductive argument. Finally, if $e_0 \in \mathscr{R}[G]$, set $e_0 = Gw_0$ for some uniquely defined $w_0 \in \mathscr{U}$, then, using $G_L G = I$, it is easily seen that $e_k = G(1 + \left[(1 - \beta)^k - 1\right])w_0 = (1 - \beta)^k e_0$ which shows that $e_{k+1} = (1 - \beta)e_k$ for $k \geq 0$. The proofs that the error converges to zero if, and only if, β lies in the defined range and the monotonicity of the error norms follows easily. □

The general conclusions on algorithm performance and the choice of β are identical to those deduced for the right inverse algorithm but conditions on e_0 are required for the left inverse algorithm. As $e_0 = r - Gu_0 - d$, the condition $e_0 \in \mathscr{R}[G]$ becomes the condition $r - d \in \mathscr{R}[G]$.

The result indicates that the choice of reference r can be an issue in Iterative Control and should be regarded as part of the design problem. The condition $r - d \in \mathscr{R}[G]$ simply means that there exists a control input in $\mathscr{U}$ that generates the output $y = r$ exactly. This is reassuring but, if there are signals in $\mathscr{Y}$ that cannot be generated by any input in $\mathscr{U}$, convergence to zero error is impossible. This behaviour of the algorithm when $r - d$ does not lie in $\mathscr{R}[G]$ depends both on properties of G and the choice of G_L. A simple description of possible behaviours can be given using the fact that the output space $\mathscr{Y}$ has a direct sum decomposition of the form

$$\mathscr{Y} = \mathscr{R}[G] \oplus ker[G_L]. \quad (6.12)$$

To prove this, let $y \in \mathscr{Y}$ be arbitrary and write $y = (I - GG_L)y + GG_L y$. It is easily verified that $(I - GG_L)y \in ker[GG_L] = ker[G_L]$ and that $GG_L y \in \mathscr{R}[G]$. This proves that $\mathscr{Y} = \mathscr{R}[G] + ker[G_L]$. The fact that this is a direct sum follows as, if $y \in \mathscr{R}[G] \cap ker[G_L]$ is non-zero, then $y = Gw_0$ for some non-zero $w_0 \in \mathscr{U}$ and $0 = G_L y = G_L G w_0 = w_0$ which is a contradiction. Hence

$$\mathscr{R}[G] \cap ker[G_L] = \{0\}. \quad (6.13)$$

Given the discussion above, write $e_0 = Gw_0 + v_0$ where $v_0 = (I - GG_L)e_0 \in ker[G_L]$ is uniquely defined and $w_0 = G_L e_0 \in \mathscr{U}$. A simple calculation then gives

$$e_k = (1 + \left[(1 - \beta)^k - 1\right] GG_L)e_0 = (1 - \beta)^k GG_L e_0 + v_0. \quad (6.14)$$

It is clearly seen that the errors converge to the signal v_0 as $k \to \infty$. The component v_0 present in e_0 remains unchanged as the iterations progress with the changes in e_k being associated, solely, with the initial component w_0. This analysis proves that,

Theorem 6.1 *With the above construction, the algorithm* $u_{k+1} = u_k + \beta G_L e_k$ *where* $0 < \beta < 2$ *has the following convergence properties for all initial errors* e_0

$$\lim_{k\to\infty} e_k = e_\infty = (I - GG_L)e_0. \tag{6.15}$$

Little can be said about the behaviour of the sequence of norms $\|e_k\|_{\mathscr{Y}}$, $k \geq 0$, unless $GG_L = I$ or $e_0 \in \mathscr{R}[G]$ when $e_\infty = 0$. In particular, monotonic decreases in norms from iteration to iteration will only occur if $v_0 = 0$. More generally, however, it is easy to see that the norms $\|e_k - v_0\|$ reduce monotonically as $e_k - v_0 = (1-\beta)^k GG_L e_0$. This observation typically has little practical value as the signal v_0 is either unknown or remains uncomputed.

6.1.3 Why the Inverse Model Is Important

The use of inverse model algorithms demonstrates the feasibility of constructing algorithms that converge but that this convergence depends on properties of G, the chosen algorithm and e_0. If G itself has a bounded inverse, denoted G^{-1}, then $G_R = G_L = G^{-1}$ and the two algorithms are identical. In all cases, the analysis

1. proves that a simple, single parameter algorithm can provide a detailed prediction of the behaviour of the sequence of iterates,
2. highlights the need to consider the nature of the initial error e_0 and
3. indicates that control of the rate of convergence is a feasible design objective. In fact, choosing $\beta = 1$ in a right inverse algorithm leads to convergence to zero error in one iteration.

This immediately suggests that iteration may be unnecessary but this is incorrect!

Why Iterate? The reasons why iteration is an essential component of high accuracy control in repetitive environments can be found, for example, in three observations, namely that,

1. either *the complexity of an inverse operator may make implementation of inverse algorithms unrealistically complex*,
2. or the fact that *errors between the model G and the plant dynamics will mean that the first iterate is highly unlikely to be zero in practice even if* $\beta = 1$ (an observation compounded by the probable presence of noise on measured signals),
3. or that *the inverse system may have dynamic characteristics that make its use in feedforward control difficult*. For example, if G is associated with a linear, time invariant, state space system, the inverse system may be unstable depending upon the position of the system zeros. The use of unstable update laws is, it is assumed, a computational and implementation problem in practice.

The case when the plant has a transfer function description provides an intuitive underpinning of these observations.

1. **Effect of Transfer Function Properties**: Suppose that G is a SISO, linear time invariant system with transfer function $G(s)$ of relative degree $k^* \geq 0$. Assuming zero initial conditions and letting the duration of each trial $T \to \infty$ to allow the use of Laplace transform descriptions, the input/output relationship takes the form $y(s) = G(s)u(s)$ and the input update equation is $u_{k+1}(s) = u_k(s) + \beta G^{-1}(s)e_k(s)$. Perfect tracking is achieved for the input signal with Laplace Transform $u_\infty(s) = G^{-1}(s)r(s)$. The inverse transfer function could have a complex form or be unstable making the input update equation unnecessarily complicated with the added complication of generating possibly unstable signals.
2. The transfer function will also contain modelling errors. For strictly proper systems (that is, $k^* \geq 1$),

   a. these errors could be substantial, particularly at high frequencies where the inverse contains derivative operations that could increase sensitivity to noise.
   b. Poor modelling of the system zeros that are very close to the imaginary axis can lead to substantial errors in the inverse and potentially destabilizing amplification of frequencies close to that zero.

   A combination of noise and inadequate modelling could lead to poor input replication and, potentially, algorithm divergence.
3. **The Influence of the Choice of Demand Signal**: The effect of r can be illustrated by assuming that $r(s)$ is a rational polynomial of relative degree k_r. If $k_r \leq k^*$, it is a simple matter to show that the formula for time signal $u_\infty(t) = \mathscr{L}^{-1}[u_\infty(s)]$ contains impulsive (delta function) type terms at $t = 0$. Such inputs cannot be generated in practice and it is concluded that the Iterative Control problem has no practically useful solution unless $k_r > k^*$.

Iteration as Approximate Inversion: The simplicity and effectiveness of the inverse model algorithms provide an excellent conceptual basis for understanding the issue of iteration but the complexity of the control law and robustness issues can be a real problem in practice.

1. The issue of robustness will be discussed in a later section where it becomes clear that modelling errors can be tolerated but the consequences of such uncertainty are that the modelling error has to satisfy certain technical positivity conditions and, also, the usable range of learning gain β is likely to be reduced.
2. Reductions in the complexity of the control law can only be addressed by modifications to the input update rule and yet any such modification must aim to retain much of the performance benefits of inverse model control. For this reason, many Iterative Learning Control algorithms can be regarded as being based on structured approximations to a plant inverse. Conceptually,

$$u_{k+1} = u_k + \beta K_0 e_k \quad \textit{where} \quad GK_0 \approx I. \tag{6.16}$$

The approximation may not be explicit and may not even be good in the normal mathematical sense. It should, however, ideally be capable of creating observed behaviours that are similar to inverse model control. The dynamic structure of the input update rule can have many forms and include many more parameters than the single learning gain β. This text describes a number of such structures based on optimization ideas.

6.1.4 Inverse Model Algorithms for State Space Models

Application of inverse model algorithms starts with the selection of a computational representation of an appropriate left or right inverse of the operator G used in the model of the plant. This representation, together with a choice of β, can then be used in the chosen input update law to compute the new input signal off-line between iterations. Because of the generality of the presentation, the operator G can take many structures. This leads to the inevitable fact that inverse systems can take many forms. The ideas of left and right inverse systems for m-output, ℓ-input state space systems $S(A, B, C, D)$ was discussed in Sect. 3.9 in the form of Theorems 2.2, 2.4 and 2.5. These theorems apply to both continuous time and discrete time cases. For completeness, the essential conclusion was that

Left Inverse Representation: *Suppose that $m \geq \ell$, $rank[D] = \ell$ and K is any $\ell \times m$ matrix such that KD is nonsingular. Then the state space system*

$$S(A - B(KD)^{-1}KC, B(KD)^{-1}K, -(KD)^{-1}KC, (KD)^{-1}K) \tag{6.17}$$

is a left inverse of $S(A, B, C, D)$ on any finite interval.

Right Inverse Representation: If *$m \leq \ell$, $rank[D] = m$ and K is any $\ell \times m$ matrix such that DK is nonsingular, then the state space system*

$$S(A - BK(DK)^{-1}C, -BK(DK)^{-1}, K(DK)^{-1}C, K(DK)^{-1}) \tag{6.18}$$

is a right inverse of $S(A, B, C, D)$ on any finite interval.

Given the familiar form of these representations, the core computation for inverse model algorithms is easily stated as follows,

Computation of the Input Update: *Inverse model algorithms using these representations construct the input change $u_{k+1} - u_k$ in the (off-line) period between the completion of the kth iteration and initiation of the $(k + 1)$th iteration. The change is precisely the signal obtained as the output from a simulation of the state space model of the relevant inverse model from zero initial conditions to the input e_k.*

Other observations relevant to design include

1. When $m = \ell$ the two representation given above are identical and independent of K. Without loss of generality, the value $K = I$ can be taken.
2. If $m \neq \ell$, K is non-unique and there is a degree of choice available for design. One particularly important effect of the choice of K is its impact on the poles/eigenvalues of the left or right inverse system. Ideally, the choice ensures the stability of the inverse could be but it is possible that no such K exists.
3. The representations given above assume not only that $D \neq 0$ but also that it has full row or column rank. The analysis of supervector descriptions of discrete time models has already indicated, using matrix methods, that such properties are essential to ensuring that either $\mathscr{Y} = \mathscr{R}[G]$ or that $ker[G] = \{0\}$. Chapter 4 presented a number of techniques (based on the relative degree, inverse system partition, decoupling theory parameterizations and the $\mathscr{C}^*$ canonical form) that indicate that a change in the definition of the output signal can transform a strictly proper system $S(A, B, C)$ into a proper system $S(A_0, B_0, C_0, D_0)$ where D_0 has full row or column rank. Despite their theoretical benefits, the acceptability of such a process may depend on the nature of the design application being considered.

6.1.5 Robustness Tests and Multiplicative Error Models

The robustness of inverse model algorithms can be approached in a general way. The methodology is based on the tools provided in Sect. 5.2 and illustrates the challenges seen in developing useful robustness tests. In particular, it provides a general insight into the role of positivity conditions in the analysis.

6.1.5.1 Left Multiplicative Perturbations: A General Analysis of Right Inverse Algorithms

Using the notation of the previous sections, consider the right inverse model algorithm for a linear plant with model $G : \mathscr{U} \to \mathscr{Y}$ possessing a bounded right inverse G_R. Suppose that the input update rule is $u_{k+1} = u_k + \beta G_R e_k$ but that the plant can be described by the left multiplicative uncertainty model UG where $U : \mathscr{Y} \to \mathscr{Y}$ is a bounded linear operator. The evolution of the observed plant error is obtained by operating on the update rule with UG to obtain

$$UGu_{k+1} = UGu_k + \beta UGG_R e_k \quad which \ \ gives \quad e_{k+1} = (I - \beta U)e_k \tag{6.19}$$

as the plant dynamics takes the form of the relationship $y = UGu + d_U$ where d_U is the modified (iteration independent) initial condition term. The predictions of the model G will be correct when $U = I$ and $d_U = d$ (which can be thought of as the "*nominal values*" of U and d_U).

Consider the characterization of the property of Robust Monotonic Convergence of the resultant dynamics in the presence of the modeling error. It is a simple matter to use the error update relations and Theorem 5.7 (with $L = I - \beta U$) to prove the following result,

Theorem 6.2 *Consider the Right Inverse Model algorithm with β in the range $0 < \beta < 2$. Then, a necessary condition for robust monotonic convergence in the presence of the left multiplicative modelling error U is that*

$$\|(I - \beta U)\| \leq 1. \tag{6.20}$$

Moreover, for the cases where $\mathscr{Y}$ is finite dimensional, it is both necessary and sufficient that $\|(I - \beta U)\| < 1$. This condition is also sufficient if $\mathscr{Y}$ is infinite dimensional.

More specifically, if $\mathscr{Y}$ is a real Hilbert space, then the following result follows from Theorem 5.8 with $L = I - \beta U$ and $H = U$. It provides a useful insight into the nature of U in the form of positivity conditions that are essential if robust monotonic convergence is to be achieved.

Theorem 6.3 (Positivity and Robust Monotonic Convergence) *Suppose that $\mathscr{Y}$ is a finite or infinite dimensional, real Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathscr{Y}}$. Consider the Right Inverse Model algorithm with $0 < \beta < 2$. Then, a necessary condition for robust monotonic convergence in the presence of the left multiplicative modelling error $U : \mathscr{Y} \to \mathscr{Y}$ is that*

$$U + U^* > 0 \quad (A\ \textit{Strictly Positive Modelling Error Condition}) \tag{6.21}$$

where U^* is the adjoint of U.
Moreover, if $\mathscr{Y}$ is finite dimensional, it is necessary that there exists a real number $\varepsilon_0 > 0$ such that

$$U + U^* \geq \varepsilon_0^2 I \quad (A\ \textit{Stronger Strict Positivity Condition}). \tag{6.22}$$

In both the finite and infinite dimensional cases, the validity of condition (6.22) implies that there exists a real number $\beta^* > 0$ such that, for any choice of $\beta \in (0, \beta^*)$, it is possible to compute a real number $\lambda(\beta) < 1$ (dependent on β) such that robust monotonic convergence is guaranteed as $\|(I - \beta U)\| \leq \lambda(\beta) < 1$. Finally, the largest possible choice of β^* lies in the range

$$\frac{\varepsilon_0^2}{\|U\|^2} \leq \beta^* \leq \frac{2}{\|U\|}. \tag{6.23}$$

The theorem throws light on many aspects of robust monotonic convergence including the role of positivity conditions on U and the inevitable restrictions on the range of learning gains β that can be used. In the absence of modelling error, U

takes the nominal value $U = I$ when, noting that $\|U\| = 1$ and that it it possible to choose $\varepsilon_0^2 = 2$, the value $\beta^* = 2$ is regained. A simple continuity argument then suggests that the conditions will be satisfied by all U "close enough" to I. This observation provides the basis for robustness tests based on modelling errors that are not known but assumed to lie in a set defined by suitably constructed norm bounds. More precisely, if U is not known, the ideas can still be applied to provide a link between the choice of β and a definition of a set of possible modelling errors as follows.

Theorem 6.4 (Robustness and Norm Bounded Modelling Errors) *Suppose that $\mathscr{Y}$ is a real Hilbert space. Consider the Right Inverse Model algorithm with $0 < \beta < 2$ and suppose that all that is known about the modelling error between model and plant is that U satisfies the condition, for some scalars $\gamma > 0$ and $0 \leq \delta < 1$,*

$$\|I - \gamma U\| \leq \delta \quad (A\ Defined\ Set\ of\ Modelling\ Errors). \tag{6.24}$$

Under these conditions, there exists a real number $\beta^* > 0$ such that the Iterative Control process is robust monotonic convergent for all $\beta \in (0, \beta^*)$ for all U satisfying (6.24). In particular a lower bound for the largest such β^* is $\beta^{**} = \gamma\left(\frac{1-\delta}{1+\delta}\right) > 0$.

Proof The norm condition (6.24) defining plant perturbations is equivalent to

$$\gamma(U + U^*) \geq (1 - \delta^2)I + \gamma^2 U^* U \geq (1 - \delta^2)I \tag{6.25}$$

so it is possible to choose $\varepsilon_0^2 = \gamma^{-1}(1 - \delta^2) = \gamma^{-1}(1 - \delta)(1 + \delta) > 0$. In addition, $\|\gamma U\| \leq \|I\| + \|(I - \gamma U)\|$, so that $\|U\| \leq \gamma^{-1}(1 + \delta)$. The results of the preceding theorem now prove robust monotonic stability for gains $0 < \beta < \beta^{**} = \varepsilon_0^2/\|U\|^2 = \gamma\left(\frac{1-\delta}{1+\delta}\right)$ as required. □

The reader should note that:

(a) The range of β permitted depends only on the defined parameters γ and δ.
(b) The norm condition defining the set of modelling errors is equivalent to a statement that the operator U lies in the closed ball $B_c(\gamma^{-1}I, \delta\gamma^{-1})$ in the normed space $\mathscr{L}(\mathscr{Y}; \mathscr{Y})$ of bounded linear operators from $\mathscr{Y}$ into itself.
(c) The nominal value $U = I$ lies in the modelling error set if $|1 - \gamma| \leq \delta$.

6.1.5.2 Right Multiplicative Perturbations: A General Analysis of Left Inverse Algorithms

Suppose that the plant has a model G with bounded left inverse G_L and the relationship between plant and model is described by a right multiplicative perturbation U. It follows that $ker[G] = \{0\}$ and the error update equation for the plant tracking error is

$$e_{k+1} = (I - \beta GUG_L)e_k \tag{6.26}$$

For the nominal case of $U = I$, monotonic convergence to zero error requires that $0 < \beta < 2$ and that the initial error $e_0 \in \mathscr{R}[G]$. For robustness analysis, it is necessary to use a changed perspective by *associating the monotonicity with some subsidiary signal*. More precisely, assume that $e_0 \in \mathscr{R}[G]$ and construct uniquely defined subsidiary signals $\{w_k\}_{k\geq 0}$ in $\mathscr{U}$ by writing $e_k = Gw_k, \;\; k \geq 0$.

Note: If u_∞ is an input signal with the property that the resultant tracking error $e = 0$, then a simple calculation gives,

$$w_k = U(u_\infty - u_k), \quad \textit{for all } k \geq 0, \tag{6.27}$$

which links the monotonicity of the norms of w_k to the monotonicity of the norm of $U(u_\infty - u_k)$. If $U = I$, then the link is precisely that the norm of $u_\infty - u_k$ is monotonic.

Using this construction and the condition $ker[G] = \{0\}$ in the analysis of the plant dynamics, the update equation becomes an update equation in $\mathscr{U}$

$$w_{k+1} = (I - \beta UG_LG)w_k = (1 - \beta U)w_k. \tag{6.28}$$

This equation is identical in structure to that seen in the right inverse algorithm with left multiplicative model error. If the robust monotonic convergence property is now redefined to require that

$$\textit{for all } k \geq 0 \textit{ and } w_0 \in \mathscr{U}, \quad \|w_{k+1}\|_{\mathscr{U}} < \|w_k\|_{\mathscr{U}} \quad \textit{and, also,} \quad \lim_{k\to\infty} w_k = 0, \tag{6.29}$$

then monotonic error convergence is not achieved but $e_k \to 0$ as required. With this understanding, all of the results of the previous section can be applied to the case of right multiplicative perturbations. It is left as an exercise for the reader to pursue this observation in more detail.

Finally, using $e_k = GU(u_\infty - u_k)$ and $w_k = U(u_\infty - u_k), \;\; k \geq 0$, assume that the perturbations lie in the set defined by (6.24) so that $\|\gamma U\| \leq 1 + \delta$. Also, using $(I - \gamma U)^*(I - \gamma U) \leq \delta^2 I$, the scalar $v_0 = \inf\{\|Uu\| \;:\; \|u\| = 1\} > 0$ as $\delta < 1$ so that

$$v_0^2 I \leq U^*U \leq \gamma^{-2}(1 + \delta)^2 I. \tag{6.30}$$

This implies that the norm $\|\cdot\|_{\mathscr{U}}$ in $\mathscr{U}$ is topologically equivalent to the norm $\|\cdot\|_0$ defined by $\|u\|_0 = \|Uu\|_{\mathscr{U}}$. More precisely,

$$v_0\|u\|_{\mathscr{U}} \leq \|u\|_0 \leq \gamma^{-1}(1 + \delta)\|u\|_{\mathscr{U}} \quad \textit{for all } u \in \mathscr{U} \tag{6.31}$$

and the monotonicity of $\|w_k\|_{\mathscr{U}}$ ensures the monotonicity of $\|u_\infty - u_k\|_0$ which, in turn, "pulls" $\|u_\infty - u_k\|_{\mathscr{U}}$ to zero. If δ is small then the sequence $\{\|u_\infty - u_k\|_{\mathscr{U}}\}_{k\geq 0}$ is "almost" monotonic.

6.2 Frequency Domain Robustness Criteria

The norm-based conditions for robustness provide some evidence for robustness of both left and right inverse model algorithms but, for application, will need methods for computation of the norms and, ideally, provide a link between these norms and physical properties of the system. The ease with which these issues can be progressed will depend on the nature of G and the input and output spaces used. For linear, time-invariant state space systems $S(A, B, C, D)$, the natural link to classical control concepts lies in the use of frequency domain concepts applied to the error or subsidiary signal evolution equation with multiplicative modelling error U,

$$\begin{aligned} e_{k+1} &= (I - \beta U)e_k \quad (\textit{for the Right Inverse Algorithm}) \\ w_{k+1} &= (I - \beta U)w_k \quad (\textit{for the Left Inverse Algorithm}). \end{aligned} \tag{6.32}$$

The great similarity of the error evolution in these two cases means that the analysis can proceed assuming the use of a right inverse algorithm with left multiplicative modelling error.

6.2.1 Discrete System Robust Monotonicity Tests

Consider a right inverse algorithm algorithm applied to the discrete time system $S(A, B, C, D)$ (with $m \leq \ell$) on an interval $0 \leq t \leq N$ on which the matrix $G(A, B, C, D)$ in the supervector description has a right inverse (that is, $rank(D) = m$). If the model transfer function matrix is $G(z)$, suppose that the plant has transfer function matrix $U(z)G(z)$. For the left inverse algorithm, it is necessary that $\ell \leq m$, $rank(D) = \ell$ and that the representation of the plant is $G(z)U(z)$.

With this construction, let the matrix U be the matrix in the supervector description of $U(z)$. The block diagonal structure in the supervector description of the error evolution provides convergence conditions as follows.

Theorem 6.5 *If $U(z)$ has a state space realization $S(A_U, B_U, C_U, D_U)$, then the Iterative Control process is convergent for all initial errors e_0 if, and only if, the spectral radius $r(I - \beta D_U) < 1$.*

Proof The characteristic polynomial of $I - \beta U$ is $(det\,(\lambda I_m - (I - \beta D_U))^{N+1}$. □

The result indicates the ultimate dependence of the convergence on the accuracy of high frequency modelling and shows that convergence is not necessarily monotonic in any chosen norm. Monotonicity and the use of the Euclidean signal norm does however allow much more detail to be derived. A more general approach using bi-directional filters is provided later in Sect. 8.1.

Theorem 6.6 (Frequency Domain Monotonic Robustness for Discrete Inverse Model Algorithms) *With the conditions described in the preceding discussion, suppose that* $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ *is given the Hilbert space topology defined by the inner product*

$$\langle y, w\rangle_{\mathscr{Y}} = \sum_{t=0}^{N} y^T(t)Qw(t), \quad \text{with} \quad Q = Q^T > 0, \tag{6.33}$$

with induced norm $\|y\| = \sqrt{\langle y, y\rangle_{\mathscr{Y}}}$. *Suppose also that* $U(z)$ *is asymptotically stable and that it is strictly positive real in the sense that*

$$QU(z) + U^T(z^{-1})Q > 0 \quad \text{for all complex } z \text{ satisfying } \quad |z| = 1. \tag{6.34}$$

Then, in some non-empty range of gains $(0, \beta^*)$, *the inverse model algorithm is robust monotonically convergent (in the given topology) in the presence of the multiplicative modelling error* $U(z)$ *and, for all* $\beta \in (0, \beta^*)$, *the induced norm* $\|(I - \beta U)\| < 1$. *This norm requirement is satisfied if*

$$\sup_{|z|=1} r\left(Q^{-1}(I - \beta U(z^{-1})^T Q(I - \beta U(z)))\right) = \sup_{|z|=1} \sigma^2_{max}(z) < 1 \tag{6.35}$$

where $\sigma^2_{max}(z)$ *is the largest eigenvalue of* $Q^{-1}(I - \beta U(z^{-1})^T Q(I - \beta U(z))$. *That is,* $\sigma_{max}(z)$ *is the largest singular value of the complex matrix* $Q^{1/2}(I - \beta U(z))Q^{-1/2}$.

Proof The strict positive real condition on $U(z)$ implies (using Theorem 4.8) that the matrix $U + U^T > 0$ and hence, noting that the relevant spaces are finite dimensional, that $U + U^T \geq \varepsilon_0^2 I$ for some $\varepsilon_0 > 0$. In particular, $det(D_U) \neq 0$. The result now follows from Theorem 4.10. The permissible range of β are those where

$$QU(z) + U^T(z^{-1})Q > \beta U^T(z^{-1})QU(z) \quad \text{for all } |z| = 1. \tag{6.36}$$

This statement is equivalent to the stated spectral radius condition. □

If $U(z)$ is a known, the theorem leads to a computational frequency domain test for robustness for a given $U(z)$ and choice of β by first checking, frequency by frequency $z = e^{i\theta}$, $0 \leq \theta < 2\pi$, that $U(z)$ is strictly positive real. If successful, then the spectral radius condition can be checked in the same frequency range to verify that $\sigma_{max}(z) < 1$. The relationship between $U(z)$ and β^* is highly complex in this situation so the choice of a suitable β value is, in reality, a trial and error process.

If $U(z)$ is not known, then the result simply provides a useful statement that robust monotonic convergence can be attained provided that it is strictly positive real and β is small enough. It says nothing about the value of β^*. Rather more can be deduced if U satisfies some known defined conditions such as (6.24) in Theorem 6.4 but it can be envisaged that even this data may not be available in some applications.

These problems disappear to some extent for the case of SISO systems (when $m = \ell = 1$). In this case, $Q = 1$ without loss of generality, and

1. the robustness condition is simply

$$|\beta^{-1} - U(z)| < \beta^{-1} \ \textit{for all} \ z = e^{i\theta}, 0 \leq \theta < 2\pi \tag{6.37}$$

   which, simply stated, requires that—*The plot of the frequency response function of* $U(z)$ *lies in the interior of the circle* S_β *in the complex plant of centre* $(\beta^{-1}, 0)$ *and radius* β^{-1}. Simple graphical analysis of this construct also indicates that $U(z)$ must be strictly positive real in the sense that $Re\,[U(z)] > 0$, for all z satisfying $|z| = 1$.
2. Alternatively, the robustness condition could be used to define the permitted modelling errors that retain monotonic convergence. More precisely, for given β, *monotonic convergence will be retained for any error* $U(z)$ *with frequency response plot lying in the interior of* S_β.
3. If it is known that $U(z)$ satisfies, for some $\gamma > 0$ and $0 \leq \delta < 1$,

$$|\gamma^{-1} - U(z)| < \delta\gamma^{-1} \ \textit{for all} \ z = e^{i\theta}, 0 \leq \theta < 2\pi, \tag{6.38}$$

   then the algorithm is robust monotonically stable for all such perturbations provided that the circle of centre $(\gamma^{-1}, 0)$ and radius $\gamma^{-1}\delta$ lies in the interior of S_β.

6.2.2 Improving Robustness Using Relaxation

The development above applies to inverse algorithms without relaxation. Motivated by its role in reducing the spectral radius of the operator in the error evolution, Sect. 5.2.3 suggested, without proof, that relaxation will improve the robustness of Iterative Control. For inverse algorithms, the relaxed input update rule has the form, $u_{k+1} = \alpha u_k + \beta\hat{G}e_k, \ k \geq 0$, where G is a model of the plant and $\hat{G}$ is a left or right inverse. The effect of the relaxation parameter $0 < \alpha < 1$ can be assessed using similar techniques to those used in Sect. 6.2.

Right Inverse Algorithms: Suppose that the plant can be described by UG, where U is a left multiplicative perturbation, and that $\mathscr{R}[UG] = \mathscr{R}[G] = \mathscr{Y}$ so that a right inverse $\hat{G} = G_R$ exists. The input and error evolutions are described by

$$e_{k+1} = (\alpha I - \beta U)e_k + (1 - \alpha)(r - d), \quad k \geq 0. \tag{6.39}$$

A sufficient condition for convergence to a limit error is that $\beta > 0$ and $(\alpha I - \beta U)^*(\alpha I - \beta U) < I$ in $\mathscr{Y}$ which is achieved if

$$U + U^* - \alpha^{-1}\beta U^* U > -(\alpha\beta)^{-1}(1 - \alpha^2)I. \tag{6.40}$$

Comparing this expression with the case when $\alpha = 1$ indicates that, setting $\beta' = \alpha^{-1}\beta$ the range of errors that can be tolerated is greater than the range of errors that can be tolerated by the unrelaxed algorithm with gain β'. Note that, in principle, the need for U to satisfy a positivity condition is removed. This is underpinned by considering the case of discrete state space systems when, in a similar manner to Theorem 6.6, convergence is achieved if

$$\sup_{|z|=1} r\left(Q^{-1}(\alpha I - \beta U(z^{-1})^T Q(\alpha I - \beta U(z)))\right) < 1. \tag{6.41}$$

For SISO state space systems the condition reduces to the frequency domain requirement that the modelling error $U(z)$ with $|z| = 1$ lies inside the circle of radius β^{-1} and centre $(\beta^{-1}\alpha, 0)$ in the complex plane. When $\alpha < 1$, part of this circle lies in the open left half complex plane.

Left Inverse Algorithms: A similar argument leads to a very similar outcome in terms of increased robustness. More precisely, using the plant description GU where U is a right multiplicative modelling error with $ker[G] = ker[GU] = \{0\}$, then a left inverse $\hat{G} = G_L$ exists and the error evolution is described by

$$e_{k+1} = (\alpha I - \beta GUG_L)e_k + (1 - \alpha)(r - d), \quad k \geq 0, \tag{6.42}$$

A sufficient condition for convergence to a limit error is that $\beta > 0$ and $(\alpha I - \beta U)^*(\alpha I - \beta U) < I$ in $\mathscr{U}$. This reduces the algebraic analysis and conclusions reached for this case to those of $m \leq \ell$. It is left as an exercise for the reader to provide the details who should note that the $m \times m$ matrix Q will be replaced by an $\ell \times \ell$ matrix R defining an inner product in $\mathscr{U}$.

6.2.3 *Discrete Systems: Robustness and Non-monotonic Convergence*

The flexibility in the choice of spaces and inner products can be used to approach issues of robust convergence that is not necessarily monotonic in the preferred choice of norm. To illustrate this fact, consider the ℓ-input, m-output, discrete system $S(A, B, C, D)$ with input u, output y and state x on the interval $0 \leq t \leq N$. Suppose also that D has full rank so that the system has a right (or left) inverse to which a right (respectively, left) inverse model algorithm is applied. Let $\varepsilon > 0$ and replace the input, output, reference, error and state vector time series by weighted sequences

$$\begin{aligned} u^\varepsilon &= \{u(0), \varepsilon u(1), \varepsilon^2 u(2), \ldots, \varepsilon^N u(N)\} \\ y^\varepsilon &= \{y(0), \varepsilon y(1), \varepsilon^2 y(2), \ldots, \varepsilon^N y(N)\} \\ r^\varepsilon &= \{r(0), \varepsilon r(1), \varepsilon^2 r(2), \ldots, \varepsilon^N r(N)\} \end{aligned}$$

$$e^{\varepsilon} = \{e(0), \varepsilon e(1), \varepsilon^2 e(2), \ldots, \varepsilon^N e(N)\}$$
$$x^{\varepsilon} = \{x(0), \varepsilon x(1), \varepsilon^2 x(2), \ldots, \varepsilon^N x(N)\} \tag{6.43}$$

A simple calculation easily reveals that the state space model describing the evolution of the weighted signals is obtained by the map

$$S(A, B, C, D) \quad \mapsto \quad S(\varepsilon A, \varepsilon B, C, D) \tag{6.44}$$

with unchanged initial condition $x(0)$. The transfer function matrix becomes

$$G_{\varepsilon}(z) = C(zI - \varepsilon A)^{-1}\varepsilon B + D = G(\varepsilon^{-1}z). \tag{6.45}$$

Asymptotic stability of $S(A, B, C, D)$ implies that of $S(\varepsilon A, \varepsilon B, C, D)$ if $\varepsilon \leq 1$.

Define the inner product in $\mathscr{Y}$ as in Theorem 6.6 and introduce a new inner product and norm on ε-weighted time series in $\mathscr{Y}$ using

$$\langle f, g\rangle_{\varepsilon} = \langle f^{\varepsilon}, g^{\varepsilon}\rangle_{\mathscr{Y}} \quad \textit{and} \quad \|f\|_{\varepsilon} = \|f^{\varepsilon}\|_{\mathscr{Y}}. \tag{6.46}$$

Note that, the norm is unchanged if $\varepsilon = 1$ and that, more generally, the two norms are topologically equivalent with

$$\|f\|_{\varepsilon} \leq \|f\|_{\mathscr{Y}} \leq \varepsilon^{-N}\|f\|_{\varepsilon} \quad \textit{if} \quad \varepsilon \leq 1. \tag{6.47}$$

Monotonicity of the sequence of un-weighted norms $\{\|e_0\|_{\mathscr{Y}}, \|e_1\|_{\mathscr{Y}}, \|e_2\|_{\mathscr{Y}}, \ldots\}$ is not equivalent to monotonicity of the weighted norms $\{\|e_0^{\varepsilon}\|_{\mathscr{Y}}, \|e_1^{\varepsilon}\|_{\mathscr{Y}}, \|e_2^{\varepsilon}\|_{\mathscr{Y}}, \ldots\} = \{\|e_0\|_{\varepsilon}, \|e_1\|_{\varepsilon}, \|e_2\|_{\varepsilon}, \ldots\}$.

For simplicity, consider the case of a right inverse algorithm and the asymptotically stable, left multiplicative modelling error $U(z)$. The error evolution still takes the form $e_{k+1} = (I - \beta U)e_k$. However, robust monotonic convergence with respect to the norm $\|\cdot\|_{\varepsilon}$ is described by the following generalization of Theorem 6.6.

Theorem 6.7 (Frequency Domain Robustness Test—A Non-monotonic Case) *With the conditions described in the preceding discussion and $\varepsilon \leq 1$, suppose that $U(z)$ is asymptotically stable and that it is strictly positive real in the sense that*

$$QU(z) + U^T(z^{-1})Q > 0 \quad \textit{for all complex } z \textit{ satisfying} \quad |z| = \varepsilon^{-1}. \tag{6.48}$$

Then, in some non-empty range of gains $(0, \beta^*)$,

$$\sup_{|z|=\varepsilon^{-1}} r\left(Q^{-1}(I - \beta U^T(z^{-1}))Q(I - \beta U(z))\right) \;=\; \sup_{|z|=\varepsilon^{-1}} \sigma_{max}^2(z) \;<\; 1. \tag{6.49}$$

The inverse model algorithm with any such choice of β is robust monotonically convergent with respect to the $\|\cdot\|_{\varepsilon}$-norm in the presence of the multiplicative modelling error $U(z)$.

Proof The proof is identical to that of Theorem 6.6 noting that the weighted time series satisfied $e^{\varepsilon}_{k+1} = (I - U_{\varepsilon})e^{\varepsilon}_{k}$ where U_{ε} is the matrix appearing in the supervector description of $U_{\varepsilon}(z) = U(\varepsilon^{-1}z)$. The frequency domain conditions for $U_{\varepsilon}(z)$ with $|z| = 1$ then become conditions on $U(z)$ with $|z| = \varepsilon^{-1}$. □

Using the principle of the maximum from complex variable theory, the quantity $\sup_{|z|=\varepsilon^{-1}} \sigma_{max}(z)$ is monotonically decreasing with ε which suggests that reducing ε allows a greater range of perturbations U to be included. The ultimate robustness characterization is hence obtained by letting $\varepsilon \to 0+$. Suppose that $U(z)$ is a given modelling error and that the "D" term in the state space model of U is denoted by D_U. It is easily seen that $\lim_{|z|\to\infty} U(z) = D_U$ so that

Theorem 6.8 *A sufficient condition for the existence of a value of* $\varepsilon \leq 1$ *such that the conditions of Theorem* 6.7 *are valid is that* $r\left(Q^{-1}(I - \beta D_U^T)Q(I - \beta D_U)\right) < 1$.

Note that, in practice, this implies that convergence with respect to some ε-weighted norm may depends ultimately on the nature of the high frequency properties of the modelling error U.

The theoretical implications of the above are that robust monotonic convergence can, in principle, be relaxed to prove that convergence can be achieved without monotonicity with respect to the preferred norm. A frequency domain tool is used to develop conditions for the monotonicity using an ε-weighted norm. The reader will note that Theorem 6.7 provides computational and graphical tools for checking the robust convergence conditions for any $\varepsilon \leq 1$ but that, for $\varepsilon < 1$, the theory provides little physical insight into the detailed nature of the resultant convergence expressed in terms of the original un-weighted norm sequence.

6.3 Discussion and Further Reading

The Iterative Control problem can be seen as an inverse problem as it attempts to solve the equation $r = Gu + d$. If a solution exists then, intuitively speaking, the converged solution $u_{\infty} = \hat{G}(r - d)$ where $\hat{G}$ is some inverse mapping of G. The chapter has provided the mathematical background to the formal use of inverse models to create convergent algorithms with monotonically decreasing tracking error norms. The algorithm is remarkably simple in structure with clear properties dependent only on a single gain parameter β. This simplicity hides a multitude of issues that influence practical applications. Although the ideas are believed to be part of commercial Iterative Control software solutions, formal, rigorous research in the area is more recent and has concentrated on the single-input, single-output case (see, for example, [55]) where the simple monotonic convergence rules were first proved and the need for modelling errors that satisfy positivity conditions became clear. Together with nonlinearities and other related effects, the violation of positivity conditions is probably one of the main reasons that poor convergence and divergence can be observed in practical applications.

The ideas have great generality when expressed in terms of operator theory, the main assumptions being the boundedness of the operators and the existence of inverses. Although algebraically simple, it is necessary to check these assumptions before use. The focus on multi-input, multi-output, discrete time, state space systems is natural in this context as it is possible to create (state space or supervector) models that map $0 \leq t \leq N$ onto itself and do, indeed, have a left or right inverse which is also described by a proper state space model. In contrast, the derivatives that exist in proper or strictly proper, continuous inverse models for continuous time systems when D does not have full rank will cause problems of boundedness in $L_2[0, T]$ spaces and introduce sensitivity properties, particularly in noisy environments. The mathematical solution to these problems is to apply exactly the same transformations to the continuous system matrices (A, B, C, D) as would be applied to them as if they described a discrete model. The technical problem with this is, for example, that the shift operation z^{k^*} is replaced by the s^{k^*} operation (mapping signals into their $(k^*)^{th}$ derivatives). For uniform rank k^* systems, the transformation regards the plant as mapping input functions into a new output equal to the $(k^*)^{th}$ derivative of y. A bounded inverse now exists and the Iterative Control problem would then be that of ensuring that the $(k^*)^{th}$ derivative of the output y tracks the $(k^*)^{th}$ derivative of the original reference signal r.

The robustness of the algorithms to multiplicative perturbations has a natural frequency domain structure that provides the first indication of the need for positivity conditions if the monotonicity of the error norm sequence is to be retained. If violated, convergence may still occur but may be characterized by iterations where, albeit temporarily, the norm can increase. This may not be a practical problem if the increases are small in both magnitude and number but this may not be the case. For discrete time systems, the analysis of such poor convergence properties can be approached in a theoretical way by focussing on the use of monotonicity but expressing this in terms of weighted norms parameterized by a single parameter $\varepsilon \in (0, 1]$. This has no effect on the analysis if there is no modelling error but, in the presence of such errors, reducing the value from unity makes it possible to analyze robust convergence but at the expense of allowing transient increases in the preferred error norm. The approach extends the range of modelling errors that can be tolerated to those where $U(\varepsilon^{-1}z)$ is strictly positive real. An intuitive description of the weighting methodology is that, by providing increased weighting on the initial part of the interval, the algorithm concentrates efforts in this area but allows larger deviations at later times. Having achieved substantial reductions early in the interval, the algorithm then reduces the error magnitudes at later times.

Finally, the text will return, from time to time, to the ideas of inversion as an underpinning theme for algorithm development and demonstrate that

the robustness of the inverse model algorithms, in general terms, implies the robustness of many other optimization-based ILC algorithms.

The main focus will be on Norm Optimal Iterative Learning Control beginning in Chap. 9 but, perhaps the simplest link to other approaches comes in Chap. 8 where the inversion methodologies are combined with the gradient approaches introduced in

Chap. 7 to provide filtering methodologies, design approaches and other techniques for improving robustness and shaping convergence behaviours to meet practical needs (including the specific needs for tracking of a reference r). Of particular interest in this context, design possibilities include

1. the use of bidirectional filters to condition algorithm behaviours (Sect. 8.1) and
2. the consequences of using a stable (approximate) inverse model when the plant is non-minimum-phase (Sect. 8.2.3).

Chapter 7
Monotonicity and Gradient Algorithms

The inverse model algorithms have the benefits of being applicable, of ensuring monotonic convergence and of being amenable to analysis. Another example with these properties is that of *Gradient Algorithms*. The concept behind the approach is, again, to construct control algorithms that guarantee a reduction of some measure of error magnitude from iteration to iteration. In practice, such measures could include its mean square value. This chapter takes a general viewpoint and extends this simple idea by taking an error measure that is the error norm in some chosen normed space $\mathscr{Y}$ and expresses error improvement from iteration to iteration by requiring that, for all $k \geq 0$,

$$\|e_{k+1}\|_{\mathscr{Y}} < \|e_k\|_{\mathscr{Y}} \qquad (\textit{Error Norm Monotonicity}). \tag{7.1}$$

Clearly the ease with which this can be achieved will depend on the choice of norm in $\mathscr{Y}$. It also depends on the chosen change in the input signal $u_{k+1} - u_k$. In what follows it is assumed that both the input space $\mathscr{U}$ and the output space $\mathscr{Y}$ are real Hilbert spaces. The plant dynamics is assumed to be described by the relationship $y = Gu + d$ with $G : \mathscr{U} \to \mathscr{Y}$ linear and bounded. Writing

$$\begin{aligned} \|e_{k+1}\|^2_{\mathscr{Y}} &= \|e_k + (e_{k+1} - e_k)\|^2_{\mathscr{Y}} \\ &= \|e_k\|^2_{\mathscr{Y}} + 2\langle e_k, e_{k+1} - e_k\rangle_{\mathscr{Y}} + \|e_{k+1} - e_k\|^2_{\mathscr{Y}}, \end{aligned} \tag{7.2}$$

and using the identity $e_{k+1} - e_k = -G(u_{k+1} - u_k)$ and the properties of adjoint operators gives

$$\begin{aligned} \|e_{k+1}\|^2_{\mathscr{Y}} &= \|e_k\|^2_{\mathscr{Y}} - 2\langle e_k, G(u_{k+1} - u_k)\rangle_{\mathscr{Y}} + \|G(u_{k+1} - u_k)\|^2_{\mathscr{Y}} \\ &= \|e_k\|^2_{\mathscr{Y}} - 2\langle G^* e_k, (u_{k+1} - u_k)\rangle_{\mathscr{U}} + \|G(u_{k+1} - u_k)\|_{\mathscr{Y}}. \end{aligned} \tag{7.3}$$

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_7

The second term is linear in the input change whilst the third is quadratic and positive. In general terms, a necessary condition for the norm to reduce in magnitude is to ensure that the second term $\langle G^* e_k, u_{k+1} - u_k \rangle > 0$ whilst making $u_{k+1} - u_k$ small enough to ensure that the positive contribution of the third term does not negate this effect. This leads to the following conceptual algorithm,

Algorithm 7.1 (*Conceptual Gradient Algorithm*) Let $K_0 : \mathscr{Y} \to \mathscr{U}$ be linear and bounded. Consider the feedforward Iterative Control input update law

$$u_{k+1} = u_k + \beta K_0 e_k, \quad k \geq 0. \tag{7.4}$$

Then there exists a value of β on iteration $k+1$ that reduces the error norm as is easily seen by writing

$$\|e_{k+1}\|_{\mathscr{Y}}^2 = \|e_k\|_{\mathscr{Y}}^2 - 2\beta \langle e_k, GK_0 e_k \rangle_{\mathscr{Y}} + \beta^2 \|GK_0 e_k\|_{\mathscr{Y}}^2 \tag{7.5}$$

This is quadratic in β and, assuming that $\|GK_0 e_k\|_{\mathscr{Y}} \neq 0$, is minimized by the choice

$$\beta_{k+1} = \frac{\langle e_k, GK_0 e_k \rangle_{\mathscr{Y}}}{\|GK_0 e_k\|_{\mathscr{Y}}^2} \tag{7.6}$$

The value of the norm that results from this choice is then

$$\|e_{k+1}\|_{\mathscr{Y}}^2 = \|e_k\|_{\mathscr{Y}}^2 - \left(\frac{\langle e_k, GK_0 e_k \rangle_{\mathscr{Y}}}{\|GK_0 e_k\|_{\mathscr{Y}}} \right)^2 < \|e_k\|_{\mathscr{Y}}^2. \tag{7.7}$$

A real reduction in norm is achieved if both $GK_0 e_k$ and $\langle e_k, GK_0 e_k \rangle_{\mathscr{Y}}$ are non-zero. No change in norm is possible if $GK_0 e_k = 0$.

In this chapter, the central issues considered are the choice of K_0, and the choice of gain β that is *iteration independent*, to provide and control descent properties.

7.1 Steepest Descent: Achieving Minimum Energy Solutions

The *Steepest Descent* algorithm is defined by the choice of $K_0 = G^*$ where $G^* : \mathscr{Y} \to \mathscr{U}$ is the adjoint operator of G in the defined topologies of $\mathscr{Y}$ and $\mathscr{U}$. The formal algorithm is then defined by the statement

Algorithm 7.2 (*The Steepest Descent Algorithm for Iterative Control*) The Steepest Descent algorithm is defined by the choice of a scalar gain β and the use of the input update rule

$$u_{k+1} = u_k + \beta G^* e_k, \quad \textit{for all } k \geq 0. \tag{7.8}$$

The resultant error evolution satisfies the recursive relation

$$e_{k+1} = Le_k, \quad \textit{for all } k \geq 0 \quad \textit{where } L = I - \beta GG^*. \tag{7.9}$$

The operator L is self adjoint.

The algorithm requires the construction of a computational procedure to generate the signal G^*e_k but, once achieved, has quite general convergence properties. The following theorem applies in both finite and infinite dimensions and provides an insight into optimality properties and the choice of initializing control input u_0,

Theorem 7.1 (Convergence of Steepest Descent in Finite and Infinite Dimensions) *Consider the steepest descent control law $u_{k+1} = u_k + \beta G^* e_k$ and suppose that $0 < \beta \|G^*\|^2 < 2$. Then, for all e_0 in the closure $\overline{\mathscr{R}[GG^*]}$ of the range of GG^*, the iteration satisfies*

$$\|e_{k+1}\|_{\mathscr{Y}} < \|e_k\|_{\mathscr{Y}}, \; k \geq 0, \quad \textit{and} \quad \lim_{k\to\infty} \|e_k\|_{\mathscr{Y}} = 0. \tag{7.10}$$

In particular, $\beta > 0$ and

1. *for an arbitrary $e_0 \in \mathscr{Y}$, the iteration is again monotonic with a limit*

$$\lim_{k\to\infty} e_k = e_\infty = P_{ker[G^*]} e_0 \tag{7.11}$$

   where $P_{ker[G^*]}$ is the self adjoint, positive, orthogonal projection operator onto $ker[G^*]$.
2. *If $ker[G^*] = \{0\}$ then monotonic convergence is achieved on the whole of $\mathscr{Y}$.*
3. *If only an upper bound $B(G^*)$ for $\|G^*\|^2$ is available, then the gain range can be replaced by $0 < \beta B(G^*) < 2$.*

Proof The result follows from Theorems 5.9 and 5.10 using the identification $L = I - \beta GG^*$, the fact that $GG^* \leq \|G^*\|^2 I$ and the inequality $(-1 + (2 - \beta\|G^*\|^2))I = (1 - \beta\|G^*\|^2)I \leq L \leq I$ so that $\mu_0^2 = (2 - \beta\|G^*\|^2) > 0$. The result follows as $ker[I - L] = ker[\beta GG^*] = ker[G^*]$ and $ker[G^*] = \{0\}$ implies that $GG^* > 0$ and hence that $L < I$. □

The convergence of the input sequence $\{u_k\}_{k\geq 0}$ is similarly proved by the following result which also provides a useful insight into the nature of the limit u_∞ and its connections to the choice of u_0.

Theorem 7.2 (Input Convergence and Minimum Norm Solutions) *Using the notation and assumptions of Theorem 7.1, suppose that $e_0 \in \mathscr{R}[GG^*]$. Then the input sequence $\{u_k\}_{k\geq 0}$ in $\mathscr{U}$ converges in the norm topology to a limit $u_\infty \in \mathscr{U}$ described by the relations*

$$r = Gu_\infty + d, \quad u_\infty = u_0 + G^* v_0 \quad \textit{where} \quad e_0 = \beta GG^* v_0 \textit{ and } v_0 \in \overline{\mathscr{R}[GG^*]}. \tag{7.12}$$

In particular, u_∞ is the unique solution of the (minimum norm) optimization problem

$$u_\infty = \arg\min_{u\in\mathscr{U}}\{\|u-u_0\|^2_{\mathscr{U}}: \textit{subject to } r = Gu + d\}. \tag{7.13}$$

Proof Convergence of the error follows from Theorem 7.1. Apply induction to the input update equation to show that

$$u_{k+1} = u_0 + G^* \sum_{j=0}^{k} L^k e_0 \quad \textit{where} \quad L = I - \beta GG^*. \tag{7.14}$$

Write $e_0 = \beta GG^* v_0 = (I-L)v_0$ for some $v_0 \in \mathscr{Y}$ and, using $\mathscr{Y} = ker[G^*] \oplus \overline{\mathscr{R}[GG^*]}$, take, without loss of generality $v_0 \in \overline{\mathscr{R}[GG^*]}$. It follows that $L^k v_0 \to 0$ as $k \to \infty$ and

$$u_{k+1} = u_0 + G^* \left(\sum_{j=0}^{k} L^k \right) (I-L)v_0 = u_0 + G^*(I - L^{k+1})v_0 \to u_\infty \tag{7.15}$$

as required. Finally, the equations define the minimum norm solution. This can be verified by the calculation, for any u in the closed linear variety $S = \{u: r = Gu + d\} \subset \mathscr{U}$, of the inner product $\langle u - u_\infty, u_\infty - u_0\rangle_{\mathscr{U}} = \langle u - u_\infty, G^* v_0\rangle_{\mathscr{U}} = \langle G(u-u_\infty), v_0\rangle_{\mathscr{Y}} = 0$ as $G(u-u_\infty) = 0$. This is precisely the condition defining the orthogonal projection of u_0 onto S which, from the Projection Theorem 2.17, defines the unique solution of the minimum norm problem. □

The results provide a general characterization of convergence properties, guaranteed monotonicity and a natural description of the limits. One consequence is that the choice of u_0 has more significance than the simple intuition that a good choice will influence convergence rates beneficially. More generally, the limit u_∞ is the closest input to u_0 in the $\mathscr{U}$ topology. That is, the "drift" from u_0 during the iterative process is minimized and choosing $u_0 = 0$ leads to the input u_∞ of minimum norm which, in optimal control terminology, is the *minimum energy* solution.

7.2 Application to Discrete Time State Space Systems

Application of the Steepest Descent Algorithm 7.2 and Theorem 7.1 to a linear, time invariant, discrete/sampled-data system with state space model $S(A, B, C, D)$ on a time interval $0 \le t \le N$ requires the characterization of the adjoint of the operator in the supervector description and an estimate of relevant operator norms.

7.2.1 Algorithm Construction

Choice of Spaces: Take the general case where the output space $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ (respectively, input space $\mathscr{U} = \mathscr{R}^{\ell(N+1)}$) has inner product

$$\langle y, z\rangle_{\mathscr{Y}} = \sum_{t=0}^{N} y^T(t)Q(t)z(t) \quad \left(\textit{respectively} \;\; \langle u, v\rangle_{\mathscr{U}} = \sum_{t=0}^{N} u^T(t)R(t)v(t)\right), \tag{7.16}$$

where the $m \times m$ (respectively, $\ell \times \ell$) matrices $Q(t) = Q^T(t) > 0$ (respectively, $R(t) = R^T(t) > 0$) for $0 \le t \le N$. Using the associated characterization of G^* then gives,

Algorithm 7.3 (*Steepest Descent for Linear, Discrete, State Space Systems*) Using the methodology of Sect. 4.7.1 transforms the input supervector update rule $u_{k+1} = u_k + \beta G^* e_k$ into feedforward state space computations as follows

$$\begin{aligned} u_{k+1}(t) &= u_k(t) + \beta R^{-1}(t)v_{k+1}(t) \quad (\textit{The Input Update Rule}) \\ \textit{where } v_{k+1}(t) &= B^T p_{k+1}(t) + D^T Q(t)e_k(t) \\ &\quad \textit{for} \quad t = 0, 1, 2, \ldots, N \\ \textit{and} \quad p_{k+1}(t) &= A^T p_{k+1}(t+1) + C^T Q(t+1)e_k(t+1) \\ &\quad \textit{for} \quad t = 0, 1, 2, \ldots, N-1, \quad \textit{with } p_{k+1}(N) = 0. \end{aligned} \tag{7.17}$$

These computations are done between iterations using data $e_k(t)$ and reverse time simulation of the state equations for $p_{k+1}(t)$.

Choice of Weighting Matrices: The choice of weighting matrices $\{Q(t)\}_{0\le t\le N}$ and $\{R(t)\}_{0\le t\le N}$ is a design choice reflecting

1. the physical units used for inputs and outputs and their relative importance in measuring accuracies and convergence rates and/or
2. the relative importance of different times/time intervals in measuring tracking accuracy and the required convergence rates. For example,

   a. if it is more important to achieve convergence rapidly in the initial parts of the time interval, the choice of "ε-weighting" and $Q(t) = \varepsilon^{2t}Q$ (with Q time independent) and $0 < \varepsilon < 1$ will slowly reduce the weighting as t increases. The range of weights is from unity (at $t = 0$) to ε^{2N} at $t = N$ which provides an insight into the choice of ε to meet the needs of the application.
   b. If the priority is to meet accuracy requirements, very quickly, at specified intermediate times $0 < t_1 < t_2 < \cdots < t_M$ with other times being of much less importance, then an intuitive choice is obtained using a "small" parameter $\varepsilon > 0$ and setting

$$Q(t) = Q \quad \textit{if } t = t_j \quad \textit{and} \quad Q(t) = \varepsilon^2 Q \quad \textit{otherwise}. \tag{7.18}$$

Note: The reader may like to consider how this idea can be generalized to allocate greater emphasis to several *subintervals* $t_1 \leq t \leq \hat{t}_1$, $t_2 \leq t \leq \hat{t}_2$, $t_3 \leq t \leq \hat{t}_3$ etc. Applications for this generalization can be illustrated if there is only one such interval and convergence on the intervals $0 \leq t < t_1$ and $\hat{t}_1 < t \leq N$ is relatively unimportant to the practical outcome. Intuitively, the small weighting on these intervals will lead to iteration evolution where emphasis is initially placed on the tracking on $t_1 \leq t \leq \hat{t}_1$. The very small weighting on the other intervals will allow control actions on those intervals to operate in a manner that will accelerate convergence on $t_1 \leq t \leq \hat{t}_1$.

3. The choice of $\{R(t)\}_{0 \leq t \leq N}$ may arise out of a real need to converge to a minimum input energy solution. More generally, there will be a need to reflect the physical units used and the priorities or preferences describing time intervals or loops where control action/input change activity should be limited. They could also be constructed for algorithmic reasons such as the use of ε-weighting by choosing $R(t) = \varepsilon^{2t} R$.

Together with the choice of β, the choices made will have an impact on the observed rate and nature of the observed convergence.

The Role of β and R: β and R appear in the update formula as a product $\beta R^{-1}(t)$. This product controls the relative magnitude of the changes in loop inputs and is crucial to algorithm performance and the issue of allowing the user to tailor the evolution to meet the needs of the application. It suggests that β can be absorbed into R without any loss of generality. In this sense, β is irrelevant to the mathematics of the problem. However, for the purposes of this text and applications, β is retained as a design parameter and

1. the choice of the values of $R(t)$, $0 \leq t \leq N$ is assumed to be fixed before iterations begin, the choice being based on issues of energy measurement, unit choice and the degree of activity required in individual control loops.
2. β is then regarded as a "tuning parameter" available to the user as a simple mechanism for influencing/changing convergence rates and robustness.

Computation of β^*: Computing a range of learning gains $\beta \in (0, \beta^*)$ that produce convergence can be approached in the cases when the weights are iteration independent. That is, when $Q(t) = Q$ and $R(t) = R$ for all $0 \leq t \leq N$. Here, Q and R are symmetric and positive definite matrices. First use Theorem 2.13 to show that

$$r(GG^*) = r(G^*G) \quad \textit{and hence that} \quad \|G^*\|^2 = \|GG^*\| = \|G^*G\| = \|G\|^2. \tag{7.19}$$

Then compute the required range $0 < \beta < \beta^* = \frac{2}{\|G\|^2}$. Applying, Theorem 4.5 then provides a slightly smaller, but useful, range of gains in the form

$$0 < \beta < \frac{2}{\|Q^{\frac{1}{2}} G(z) R^{-\frac{1}{2}}\|_\infty^2} = 2\left(\sup_{|z|=1} r(R^{-1} G^T(z^{-1}) Q G(z) \right)^{-1} \tag{7.20}$$

computed using eigenvalue evaluation for $z = e^{i\theta}$ with $0 \leq \theta < 2\pi$.

Choice of β: The final choice of β for an application could be guided by intuitive ideas suggested by eigenvalue structure and robustness characterizations or, given an estimate of β^*, simple trial and error. The user may wish to change the gain β on each iteration. The analysis of the consequences can be approached using eigenanalysis to each such iteration but the nature of the convergence will depend upon the chosen pattern of gain variations. One such approach could be to start iterations in a cautious manner with "small" values of β such as $\beta = 0.1\beta^*$. This value can then be increased each iteration until adequate convergence rates and perceived robustness are regarded as being satisfactory. Following this initial gain selection, the reminder of the iterations could proceed with some confidence with the final chosen value β_{final} on, say, iteration k_{final}. Note that variations in β require that the minimum norm characterization of u_∞ needs to be reinterpreted as the input (solving the tracking problem) minimizing $\|u - u_{final}\|^2_{\mathscr{U}}$.

7.2.2 *Eigenstructure Interpretation: Convergence in Finite Iterations*

Suppose that G is the matrix in a supervector description of an ℓ-input, m-output, discrete time, state space system $S(A, B, C, D)$. In this circumstance, the condition $ker[G] = ker[G(A, B, C, D)] = \{0\}$ holds if, and only if, $ker[D] = \{0\}$. Similarly, the condition $ker[G^*] = \{0\}$ holds if, and only if, $ker[D^T] = \{0\}$. That is, $rank[D] = \ell$ (respectively $rank[D] = m$). As the matrix GG^* is self adjoint, the eigenstructure interpretation of steepest descent iteration is possible and is identical to that of Sect. 5.2.4 with orthogonality of eigenvectors expressed in terms of the inner product defined by the $\{Q(t)\}_{0\le t\le N}$ in Eq. (7.16).

Suppose initially that β **is constant from iteration to iteration**. Using the notation of Sect. 5.2.4, note that algorithm convergence, combined with very rapid convergence of the component corresponding to a chosen eigenvector v_j, is only possible if, within the range $0 < \beta r(GG^*) < 2$, there exists a β such that $1 - \beta\lambda_j$ is close to zero. In fact, if $1 - \beta\lambda_j = 0$, then the contribution of v_j is eliminated from the error in one iteration. Once eliminated, it will never reappear unless the plant is subject to external disturbances and/or the plant model is inaccurate.

However, it is clearly open to the user to change β **on each iteration** opening up the possibility of eliminating each eigenvector component sequentially. The application of this idea motivates a consideration of the following conceptual algorithm, based on the sequential elimination of the eigenvector components corresponding to an ordering $\lambda_{j_1}, \lambda_{j_2}, \ldots$ of the non-zero eigenvalues of GG^*.

Algorithm 7.4 (*Theoretical Convergence in a Finite Number of Iterations*) Suppose that $\mathscr{Y}$ is finite dimensional of dimension $dim[\mathscr{Y}]$. Let $\lambda_{j_1}, \lambda_{j_2}, \lambda_{j_3}, \ldots$ be a chosen ordering of the non-zero eigenvalues of GG^* and suppose that GG^* has q zero eigenvalues (that is $ker[G^*]$ has dimension q). Then the parameter varying Iterative

Control algorithm,

$$u_{k+1} = u_k + \beta_{k+1} G^* e_k,\ k \geq 0, \quad \textit{with the choice} \quad 1 - \beta_{k+1}\lambda_{j_{k+1}} = 0, \tag{7.21}$$

converges to the limit e_∞ defined by Eq. (5.74) in a finite number of iterations.

Proof of Algorithm Properties: Write $e_0 = \sum_{p\geq 1} \gamma_p v_p$. Note that the proposed iteration generates the error sequence

$$e_k = \left(\prod_{j=1}^{k} \left(I - \beta_j GG^*\right)\right) e_0 = \sum_{p\geq 1} \gamma_p \left(\prod_{j=1}^{k} \left(I - \beta_j \lambda_p\right)\right) v_p,\ k \geq 1. \tag{7.22}$$

The discussion of eigenstructure indicates that the choice of $\beta = \beta_1$ on the first iteration eliminates the component of v_{j_1} from the error. By induction, the use of $\beta = \beta_k$ on iteration k eliminates the component of v_{j_k} but does not reintroduce the previously eliminated components. The iteration terminates after, at most, $dim[\mathscr{Y}]-q$ iterations as all non-zero eigenvalues have been covered and hence all corresponding eigenvectors v_j eliminated. □

Discussion and Extension: Although conceptually interesting, the approach is not presented as a practical design policy as small values of λ_j will lead to very high values of β_j and hence to extremely large transient variations in error norm. This problem will become intolerable if model errors are present when elimination of eigenvector components will not be achieved in any iteration and/or may be re-introduced in later iterations.

If the algorithm is to stay within its range of convergence, it is essential only to use gains in the range $0 < \beta_k \|G\|^2 < 2$. In this case, the variable gain algorithm can, in principle, be used to eliminate all components with eigenvalues satisfying $\lambda_j > \frac{1}{2}\|G\|^2$ in a finite number of iterations after which convergent iterations can continue with a constant value of β. Although this looks, at first sight, like a practical approach, it is possible that $\|G\|^2$ can be computed or estimated accurately (using frequency domain analysis) whereas it is highly unlikely that the eigenvalues will be known or computed. However, substantial reductions to all components with eigenvalues satisfying $\lambda_j > \frac{1}{2}\|G\|^2$ can be achieved by using the following algorithm

Algorithm 7.5 (*Suppression of Eigenvalues* $\lambda \in (\frac{1}{2}\|G\|^2, \|G\|^2]$) Choose a finite number N_p of points $p_1, p_2, \ldots$ spread over the half-open interval $(\frac{1}{2}\|G\|^2, \|G\|^2]$ and set the gain $\beta_k = p_k^{-1}$ on any iteration $k \leq N_p$. If, on subsequent iterations, a fixed value in the range $0 < \beta\|G\|^2 < 2$ is used, it follows that the resultant error sequence has the form

$$\begin{aligned} e_k &= \left(\prod\nolimits_{j=1}^{k} (I - \beta_j GG^*)\right) e_0, \quad \textit{for } 1 \leq k \leq N_p, \textit{ and} \\ e_k &= (I - \beta GG^*)^{(k-N_p)} e_{N_p}, \quad \textit{for } k \geq N_p. \end{aligned} \tag{7.23}$$

Here, e_{N_p} is, in effect, the initial error for the remainder of the iterative process and the properties of fixed gain iteration are known, namely convergence to $e_\infty = P_{ker[G^*]}e_{N_p} = P_{ker[G^*]}e_0$. The error e_{N_p} will have had the contributions of eigenvectors with eigenvalues in the interval $(\frac{1}{2}\|G\|^2, \|G\|^2]$ greatly reduced provided that, for each such eigenvalue, at least one value of p_j is very close to that eigenvalue. This is easily verified by computing the coefficient of v_j in e_{N_p} to have magnitude

$$\left|\gamma_j \prod_{i=1}^{N_p} (1 - p_i^{-1}\lambda_j)\right| \leq |\gamma_j| \min_{1\leq i\leq N_p}\{|1 - p_i^{-1}\lambda_j|\} \ll |\gamma_j|. \tag{7.24}$$

Two observations can be made:

1. The choice of N_p is open to the user. Intuitively, N_p should be large enough to provide "adequate cover" of the interval $(\frac{1}{2}\|G\|^2, \|G\|^2]$. In addition the set $\{p_j\}_{1\leq j\leq N_p}$ could contain repeated entries, a property that is equivalent to repeated efforts to reduce the eigenvectors in the selected range.
2. Intuitively, the approach will increase convergence speed in the first N_p iterations as compared with the case of a constant β taken from the same interval.

A Link to the Frequency Domain: An empirical relationship between eigenvalues and frequency response characteristics is provided in the next subsection. This link suggests that careful choice of the p_j could target specific frequency ranges and problematic issues such as resonance. For the SISO case with $Q = R = 1$, this link suggests the choice of

$$p_j = |G(z_j)|^2, \quad 1 \leq j \leq N_p, \tag{7.25}$$

where $\{z_j\}_{1\leq j\leq N_p}$ are chosen frequency points that provide "cover" for the frequency range where the transfer function $G(z)$ has gain in the range

$$\sqrt{\frac{1}{2}}\,\|G(z)||_\infty < |G(z)| \leq \|G(z)||_\infty \quad \textit{where} \quad \|G(z)\|_\infty = \sup_{|z|=1} |G(z)|. \tag{7.26}$$

For the MIMO case, the natural extension is to cover frequencies in the range where

$$\frac{1}{2}\sup_{|z|=1} r(R^{-1}G^T(z^{-1})QG(z)) < r(R^{-1}G^T(z^{-1})QG(z)) \leq \sup_{|z|=1} r(R^{-1}G^T(z^{-1})QG(z)). \tag{7.27}$$

It is interesting to note that:

1. **The Effect of Resonance**: If there is a substantial resonance in $G(z)$, this phenomenon will mean that the value of $\|G\|$ is possibly very large with the majority of frequencies having gains that are outside of the interval $(\frac{1}{2}\|G\|^2, \|G\|^2]$. This will lead to slow convergence as the corresponding values of β will be small. With this interpretation, the elimination or reduction of resonances in $G(z)$ is seen to be essential for good algorithm performance.

2. **A Frequency Range Interpretation**: Interpreting the number of eigenvalues in the range $(\frac{1}{2}\|G\|^2, \|G\|^2]$ as a measure of the width of the frequency band covered by the algorithm, it is seen that the choice of Q and R will influence this aspect. It might be expected that Q is selected on the basis of the required measure of error convergence. In contrast R, unless required for other design purposes, is available to increase the frequency band covered. At this time there is no additional theoretical information about this issue and there are no algorithms that consider this option.

7.2.3 Frequency Domain Attenuation

Practicing engineers tend to be familiar with frequency domain characterizations of control design processes and dynamics. In Iterative Control, this is possible in some cases using a frequency domain interpretation of the error evolution equation. However, as Iterative Control is based on finite time intervals, the interpretation is approximate and improves in accuracy only as $N \to \infty$. For steepest descent iteration, the following analysis relates frequency response characteristics of the operator GG^* to the transfer function matrix $G(z)R^{-1}G^T(z^{-1})Q$. For SISO systems this quantity is just $QR^{-1}|G(z)|^2$ which provides a link to familiar frequency domain gain characteristics such as resonance, bandwidth, "notch frequencies" and the significance of pole and zero positions.

For an asymptotically stable, discrete, time invariant system $S(A, B, C, D)$ with transfer function matrix $G(z) = C(zI - A)^{-1}B + D$, the matrix $G(A, B, C, D)$ in the supervector description on $0 \le t \le N$ is the core object for eigenvalue analysis of GG^*. There is no formula for these eigenvalues but a link to frequency responses is provided using the ideas in Sect. 4.8.2. More precisely, the inner products and norms in $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ and $\mathscr{U} = \mathscr{R}^{\ell(N+1)}$ are taken to be defined by Eq. (7.16) with constant $Q(t) = Q$ and $R(t) = R$, $1 \le t \le N$. Using Theorem 2.7, the adjoint is then $G^* = \mathscr{T}(\ell, N)G(A^T, C^TQ, R^{-1}B^T, R^{-1}D^TQ)\mathscr{T}(m, N)$ where $\mathscr{T}(\cdot, \cdot)$ represents the time reversal operator of Sect. 4.3. Consider a complex valued supervector $u(z, \alpha)$ associated with a time series $u(z, \alpha) = \{u(t, z, \alpha)\}_{0 \le t \le N} = \{\alpha, \alpha z, \alpha z^2, \ldots, \alpha z^N\}$ with $\alpha \in \mathscr{C}^m$. Its time reversal is just $z^N u(z^{-1}, \alpha)$. Operating with $G(A^T, C^TQ, R^{-1}B^T, R^{-1}D^TQ)$ produces the time series $\psi_1(t)$, $0 \le t \le N$,

$$\begin{aligned} \psi_1(t) &= R^{-1}G^T(z^{-1})Qz^{N-t}\alpha + \eta_1(t) \\ \textit{where} \quad \eta_1(t) &= -R^{-1}B^T(z^{-1}I - A^T)^{-1}(A^T)^tC^TQz^N\alpha. \end{aligned} \tag{7.28}$$

A number of properties of this expression are important. Note that

1. the stability assumption implies that the norm of the supervector η_1 of the time series $\eta_1(t)$ is uniformly bounded over both N and the unit circle $|z| = 1$,

$$\sup_{N\geq 0 \ \&\ |z|=1} \|\eta_1\| < +\infty. \tag{7.29}$$

2. In particular, $\|\eta_1(t)\| \leq M\|\alpha\|\lambda^t$ for some $M > 0$ and $\lambda \in (0, 1)$ (both independent of N and $|z| = 1$). It takes small values for all $0 \ll t \leq N$ and values of $\eta_1(t)$ that are significant are clustered towards the beginning of the interval $0 \leq t \leq N$.

The time reversal of ψ_1 is just

$$\begin{aligned}\psi_1(N-t) &= R^{-1}G^T(z^{-1})Qz^t\alpha + \eta_1(N-t)\\ &= R^{-1}G^T(z^{-1})Qu(t,z,\alpha) + \eta_1(N-t).\end{aligned} \tag{7.30}$$

Operating on this time series with $G(A, B, C, D)$ then leads to the supervector $y(z, \alpha)$ associated with the output time series $\{y(t, z, \alpha)\}_{1\leq t\leq N}$ defined as follows,

Theorem 7.3 (Frequency Attenuation and GG^*) *Using the construction and notation of the discussion above,*

$$y(t,z,\alpha) = G(z)R^{-1}G^T(z^{-1})Qu(t,z,\alpha) + \eta_2(t), \quad 0 \leq t \leq N\,, \tag{7.31}$$

where $\eta_2(t)$ is uniformly bounded over all N and $|z| = 1$. It also has the property that its significant values are clustered at the beginning and end of the time interval by an expression of the form

$$\|\eta_2(t)\| \leq M\|\alpha\|(\lambda^t + \lambda^{N-t}), \quad \textit{for } 0 \leq t \leq N, \ \textit{and} \tag{7.32}$$

for some $M > 0$ and $\lambda \in (0, 1)$ (independent of N and $|z| = 1$). In particular, it follows that the limit $\limsup_{N\to\infty} \|\eta_2\|$ *is finite.*

Proof In supervector notation,

$$\begin{aligned}y(z,\alpha) &= G(A,B,C,D)\psi(z,\alpha)\\ &= G(A,B,C,D)\left(u(z,R^{-1}G^T(z^{-1})Q\alpha) + \mathscr{T}(\ell,N)\eta_1\right)\\ &= G(A,B,C,D)u(z,R^{-1}G^T(z^{-1})Q\alpha) + \mathscr{T}(m,N)G(A^T,C^T,B^T,D^T)\eta_1.\end{aligned} \tag{7.33}$$

The proof is completed using the techniques used in the preceding discussion. □

Note that $\eta_2(t)$ is small in the "mid-range" $0 \ll t \ll N$, a range which dominates the nature of the time series if N is large. The important conclusion is that, in this range, a good approximation to output behaviour is

$$y(t,z,\alpha) \approx G(z)R^{-1}G^T(z^{-1})Qu(t,z,\alpha)\,, \tag{7.34}$$

so the properties of the $m \times m$ transfer function matrix $G(z)R^{-1}G^T(z^{-1})Q$ are central to behaviour of the algorithm.

Now note that $G(z)R^{-1}G^T(z^{-1})Q = Q^{-1/2}\left(Q^{1/2}G(z)R^{-1}G^T(z^{-1})Q^{1/2}\right)Q^{1/2}$ is obtained, using a similarity transformation, from a positive Hermitian matrix when $|z| = 1$ and hence has real eigenvalues $\sigma_j^2(z) \geq 0$ with associated eigenvectors $w_j(z)$ that are orthonormal with respect to the inner product $\langle y, w\rangle = \bar{y}^T Qw$ in $\mathscr{C}^m$. Let $z_k = e^{2\pi ik/(N+1)}$, $0 \leq k \leq N$, be the distinct $(N+1)^{th}$ roots of unity. Then an orthogonal basis for the complexification $\mathscr{Y}_c = \mathscr{C}^{m(N+1)} = \mathscr{Y} \oplus i\mathscr{Y}$ of $\mathscr{Y}$ is,

$$W_j(z_k) = \left[w_j^T(z_k), w_j^T(z_k)z_k, w_j^T(z_k)z_k^2, \ldots, w_j^T(z_k)z_k^N\right]^T, \quad 1 \leq j \leq m,\ 0 \leq k \leq N, \tag{7.35}$$

where each basis vector has norm $\sqrt{N+1}$. Normalizing this basis by replacing $W_j(z_k)$ by $(N+1)^{-\frac{1}{2}}W_j(z_k)$ then gives, using Theorem 7.3,

$$\begin{aligned} GG^*W_j(z_k) &= \sigma_j^2(z_k)W_j(z_k) + \eta_{j,k} \\ \textit{where} \quad \|W_j(z_k)\| &= 1 \quad \textit{and} \quad \lim_{N\to\infty}\|\eta_{j,k}\| = 0. \end{aligned} \tag{7.36}$$

as $W_j(z_k)$ can be associated with the input sequence $u(t, z_k, (N+1)^{-\frac{1}{2}}w_j(z_k))$. Clearly the $W_j(z_k)$ are normally not eigenvectors of GG^* but they can be regarded as *approximate eigenvectors* when N is large.

The results above are summarized in the form of the following theorem. The result assumes the notation used in the preceding discussion.

Theorem 7.4 (Frequency Domain Approximate Eigenvector Properties of GG^*) *Suppose that the discrete system $S(A, B, C, D)$ is asymptotically stable and that the output space $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ (respectively, input space $\mathscr{U} = \mathscr{R}^{\ell(N+1)}$) has inner product*

$$\langle y, z\rangle_{\mathscr{Y}} = \sum_{t=0}^{N} y^T(t)Qz(t) \quad (\textit{respectively}\ \ \langle u, v\rangle_{\mathscr{U}} = \sum_{t=0}^{N} u^T(t)Rv(t)), \tag{7.37}$$

where the $m\times m$ (respectively, $\ell\times\ell$) matrix $Q = Q^T > 0$ (respectively, $R = R^T > 0$). Then, the response $y = GG^*u$ to the complex input $u(t) = u(t, z, \alpha)$ is the time series

$$y(t, z, \alpha) = G(z)R^{-1}G^T(z^{-1})Qu(t, z, \alpha) + \eta_2(t), \quad 0 \leq t \leq N. \tag{7.38}$$

Moreover, $\eta_2(t)$ is uniformly bounded over all N and the unit circle $|z| = 1$. It also has the property that its significant values are clustered at the beginning and end of the time interval as

$$\|\eta_2(t)\| \leq M\|\alpha\|(\lambda^t + \lambda^{N-t}), \quad \textit{for}\ \ 0 \leq t \leq N, \tag{7.39}$$

for some $M > 0$ *and* $\lambda \in (0,1)$ *(independent of* N *and* $|z| = 1$*). Finally, let* $z_k = e^{2\pi ik/(N+1)}$, $0 \le k \le N$, *be the* $(N+1)$th *roots of unity, and set* $\{w_j(z_k)\}_{1\le j\le m}$ *to be the orthonormal eigenvectors of* $G(z_k)R^{-1}G^T(z_k^{-1})Q$ *(in* $\mathscr{C}^m$*) associated with eigenvalues* $\{\sigma_j^2(z_k)\}_{1\le j\le m}$*, then the basis vectors* $\{W_j(z_k)\}$ *defined by Eq.* (7.35) *all have the approximate eigenstructure property*

$$\begin{aligned} GG^*W_j(z_k) &= \sigma_j^2(z_k)W_j(z_k) + \eta_{j,k} \\ \text{where} \quad \|W_j(z_k)\|_{\mathscr{Y}_c} &= 1 \quad \text{and} \quad \lim_{N\to\infty} \|\eta_{j,k}\|_{\mathscr{Y}_c} = 0. \end{aligned} \tag{7.40}$$

Consequences for the steepest descent algorithm? Assume that the spaces $\mathscr{Y}$ and $\mathscr{U}$ and the inner products of Theorem 7.4 are used. Remembering that (Theorem 2.13) $\|G\| = \|G^*\|$, it is only necessary to consider the gain range $0 < \beta\|Q^{1/2}G(z)R^{-1/2}\|_\infty^2 < 2$ where $\|Q^{1/2}G(z)R^{-1/2}\|^2$ can be computed as the spectral radius $\sup_{|z|=1} r(G(z)R^{-1}G^T(z^{-1})Q)$. The Iterative algorithm is then monotonically convergent. Also $0 < \beta\sigma_j^2(z) < 2$, $1 \le j \le m$, and $|1 - \beta\sigma_j^2(z_k)| < 1$. A simple calculation shows that

$$(I - \beta GG^*)W_j(z_k) = (1 - \beta\sigma_j^2(z_k))W_j(z_k) - \beta\eta_{j,k}\,. \tag{7.41}$$

That is, assuming zero initial conditions and a starting vector consisting of a single frequency component $e_0 = W_j(z_k)$, the approximate effect of each iteration is to produce a response with magnitude reduced by $|1 - \beta\sigma_j^2(z_k)|$ and dominated by the same frequency as the input. This approximation will be good, if N is large, at sample times in the mid-range $0 \ll t \ll N$ of the interval $0 \le t \le N$.

Attenuation of Discrete Frequencies: Normally e_0 will contain contributions from all frequency components. If the initial error is written in the form $e_0 = \sum_{j,k} \gamma_{jk}W_j(z_k)$ then, using the steepest descent algorithm, assuming that N is large, a simple conceptual model of iteration dynamics is to write the initial amplitudes

$$\gamma_{jk} = \langle W_j^T(z_k), e_0\rangle_{\mathscr{Y}_c} = \frac{1}{(N+1)^{1/2}} \sum_{t=0}^{N} \left(\overline{w_j^T(z_k)}Qe_0(t)\right) z_k^{-t}\,. \tag{7.42}$$

The amplitude of each discrete frequency $z_0, z_1, \ldots, z_N$ is then attenuated approximately by a factor of $1 - \beta\sigma_j^2(z_k)$ from iteration to iteration. This statement is approximate as the contribution of the term $\eta_{j,k}$, although small, does have an effect and is not included in the argument. Despite this approximation, the analysis does provide guidance on the effects in the frequency domain. For example, intuitively,

1. convergence rates for high frequency inputs will be slow if the plant $G(z)$ is low pass. This is because $\sigma_j^2(z_k)$ can be interpreted as as a system "gain" at the frequency z_k. First note that $\|G^*W_j(z_k)\| \approx \sigma_j^2(z_k)\|W_j(z_k)\|$ if N is large. That is, using G^*, the eigenvector component $W_j(z_k)$ is attenuated by $\sigma_j^2(z_k)$. As the eigenvalues of GG^* are identical to those of G^*G, $\|G\tilde{W}_j(z_k)\|^2 = \sigma_j^2(z_k)\|\tilde{W}_j(z_k)\|^2$

for some orthonormal eigenvectors $\{\tilde{W}_j(z_k)\} \subset \mathscr{U}$ which shows that the smallest values of the eigenvalues will be very small if G is low pass.

2. Resonances will cause problems as they will reduce the range of β that can be used. More generally, when $m = \ell$, the effect of plant zeros close to the unit circle will cause slow convergence at frequencies z_k close to that zero as the smallest singular value $\underline{\sigma}$ of $G(z)$ will be small and hence at least one value of $1 - \beta\sigma_j(z_k)$ will be close to unity.

A Useful Generalization: Finally, The frequency domain interpretation extends naturally to more general iterative operations. For example, noting that, for all $\varepsilon > 0$,

$$(I + \varepsilon^{-2}GG^*)W_j(z_k) = \left(1 + \varepsilon^{-2}\sigma_j^2(z_k)\right) W_j(z_k) + \varepsilon^{-2}\eta_{j,k}, \tag{7.43}$$

Theorem 7.5 (The Frequency Domain, $(I + \varepsilon^{-2}GG^*)^{-1}$ and NOILC) *With the notation and construction defined above*

$$\begin{aligned}(I + \varepsilon^{-2}GG^*)^{-1}W_j(z_k) = \; & (I + \varepsilon^{-2}\sigma_j^2(z_k))^{-1}W_j(z_k) + \eta_{j,k}^{NOILC} \\ \textit{where} \quad \|W_j(z_k)\|_{\mathscr{Y}_c} = 1 \quad & \textit{and} \quad \lim_{N\to\infty}\|\eta_{j,k}^{NOILC}\|_{\mathscr{Y}_c} = 0.\end{aligned} \tag{7.44}$$

The result provides insight into the frequency domain effects of the operator $(I + \varepsilon^{-2}GG^*)^{-1}$. This operator plays a central role in error evolution in Norm Optimal Iterative Learning Control in later chapters.

7.3 Steepest Descent for Continuous Time State Space Systems

Algorithm 7.6 (*Continuous Time Steepest Descent Algorithm*) Let G be generated by a linear, continuous time, time invariant, state space system $S(A, B, C, D)$ with input space $\mathscr{U} = L_2^\ell[0, T]$ and output space $\mathscr{Y} = L_2^m[0, T]$ with inner products, respectively,

$$\langle u, v\rangle_{\mathscr{U}} = \int_0^T u^T(t)R(t)v(t)dt \quad \textit{and} \quad \langle y, w\rangle_{\mathscr{Y}} = \int_0^T y^T(t)Q(t)w(t)dt \tag{7.45}$$

where $Q(t)$ and $R(t)$ are piece-wise continuous, symmetric , positive definite matrices. The adjoint system is constructed using the data $\{Q(t)\}_{0\le t\le T}$, $\{R(t)\}_{0\le t\le T}$ and dynamics $S(-A^T, -C^T, B^T, D^T)$ with a zero terminal condition.

The steepest descent algorithm constructs the input $u_{k+1} = u_k + \beta G^* e_k$ using the feedforward calculations

$$\begin{aligned} u_{k+1}(t) &= u_k(t) + \beta R^{-1}(t) v_{k+1}(t), \quad \textit{for } t \in [0, T], \\ \textit{where } v_{k+1}(t) &= B^T \psi_{k+1}(t) + D^T Q(t) e_k(t) \\ \textit{and } \dot{\psi}_{k+1}(t) &= -A^T \psi_{k+1}(t) - C^T Q(t) e_k(t), \quad \textit{with } \psi_{k+1}(T) = 0. \end{aligned} \tag{7.46}$$

Convergence properties are described by Theorems 7.1 and 7.2. The computation of $\psi_{k+1}(t)$ is approached using reverse time simulation from the terminal condition $t = T$. Alternatively, setting $\eta_{k+1}(t) = \psi_{k+1}(T - t)$, then $\eta_{k+1}(t)$ can be computed from the initial value problem

$$\dot{\eta}_{k+1}(t) = A^T \eta_{k+1}(t) + C^T Q(T - t) e_k(T - t), \quad \textit{with} \quad \eta_{k+1}(0) = 0\,. \tag{7.47}$$

The role of the weighting matrices in this case is a direct parallel to that seen in the case of discrete time systems and is not discussed further here. The calculation of $\|G^*\|$ is more complex however, even in the case of time independent $Q(t)$ and $R(t)$. This case can be analysed if G is asymptotically stable by identifying G^* as the product of a time reversal operator on $[0, T]$, the system $S(A^T, C^T Q, R^{-1} B^T, R^{-1} D^T Q)$ and a time reversal operator. As time reversal operators have unit norm, the use of the Laplace Transform description and Parseval's Theorem leads to the norm bound

$$\|G^*\| \leq \sqrt{\sup_{\omega \geq 0} r(G(i\omega) R^{-1} G^T(-i\omega) Q)} \tag{7.48}$$

which can be identified as the H_∞ norm of $R^{-1/2} G^T(s) Q^{1/2}$.

Finally, the introduction of relaxation parameters (see Sect. 5.2.3) yields,

Algorithm 7.7 (*Continuous Time Steepest Descent with Relaxation*) With the notation of Algorithm 7.6, replace the input update rule by

$$u_{k+1}(t) = \alpha u_k(t) + \beta R^{-1}(t) v_{k+1}(t), \quad \textit{for } t \in [0, T]\,, \quad \textit{and} \quad k \geq 0\,, \tag{7.49}$$

Convergence to a limit error e_∞^α follows for all choices of α in the range

$$-1 + \beta \|G^*\|^2 < \alpha \leq 1 \quad (\textit{as} \quad \mu_0^2 = 2 - \beta \|G^*\|^2)\,. \tag{7.50}$$

1. If $\alpha = 1$, then e_∞^α is equal to the orthogonal projection of e_0 onto the subspace $ker[G^*]$.
2. If $\alpha < 1$, then the limit error e_∞^α is the unique solution of the equation

$$(1 - \alpha)(r - d) = ((1 - \alpha) I + \beta G G^*) e_\infty^\alpha\,. \tag{7.51}$$

3. If, $\alpha < 1$ and, in addition, $GG^* \geq \varepsilon_0^2 I$ for some $\varepsilon_0 > 0$, then

$$lim_{\alpha \to 1-} e_\infty^\alpha = 0 \,. \tag{7.52}$$

   in the norm topology in $\mathscr{Y}$.
4. More generally, if no such ε_0 exists, the limit errors e_∞^α converge (as $\alpha \to 1-$) in the weak topology in $\mathscr{Y}$ to the orthogonal projection of $r - d$ onto $ker[G^*]$.

Proof of Algorithm Properties: The case of $\alpha = 1$ is the unrelaxed algorithm. It is only necessary, therefore, to consider the case of $\alpha < 1$. The error evolution is $e_{k+1} = Le_k + (1-\alpha)(r-d)$. $L = (\alpha I - \beta GG^*) = L^*$ is self adjoint with spectrum on the real interval $\alpha - \beta\|G^*\|^2 \leq \lambda \leq \alpha$. For the specified range of α, $r(L) < 1$. Noting that $I - L = (1-\alpha)I + \beta GG^*$) has a bounded inverse, Theorems 5.3 and 5.4 (with $H = \beta GG^*$) prove convergence to some limit e_∞^α for all $r \in \mathscr{Y}$. If $\varepsilon_0^2 > 0$, then the behaviour of e_∞^α as α varies follows from Theorem 5.11. Finally, $\mathscr{Y}$ has the orthogonal subspace decomposition $\overline{\mathscr{R}[GG^*]} \oplus ker[GG^*]$ and $ker[GG^*] = ker[G^*]$. Theorem 5.11 and the orthogonal decomposition of $\mathscr{Y}$ proves the weak convergence property. The details are left for the reader as an exercise. □

Finally, the behaviour of the limit error e_∞^α as α varies is seen to be very different in the two cases of $\alpha = 1$ and $\alpha \to 1-$. In the second case, the effect of the initial input choice u_0 is seen to disappear leaving the limit equivalent to the use of the unrelaxed algorithm with $u_0 = 0$. In effect, the relaxed algorithm *"forgets"* its starting condition and α could be called a *"forgetting factor"*.

7.4 Monotonic Evolution Using General Gradients

Return now to the general feedforward iteration $u_{k+1} = u_k + \beta K_0 e_k$. In this section, conditions for monotonic convergence are identified when K_0 has no presumed internal structure but, with carefully chosen β, produces gradient properties. More precisely, using Theorem 6.3 with U replaced by GK_0,

Algorithm 7.8 (*Positivity of GK_0 and Monotonic Convergence*) Suppose that $\mathscr{Y}$ is a finite or infinite dimensional, real Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathscr{Y}}$ and associated induced norm. Consider the feedforward Iterative Control algorithm $u_{k+1} = u_k + K_0 e_k$, $k \geq 0$, with $K_0 : \mathscr{Y} \to \mathscr{U}$ linear and bounded. Then, a necessary condition for monotonic convergence is that

$$GK_0 + (GK_0)^* > 0 \quad (\textit{A Strict Positivity Condition}). \tag{7.53}$$

In particular, it is necessary that $ker[GK_0] = \{0\}$.

If $\mathscr{Y}$ is finite dimensional, it is necessary that there exists a real number $\varepsilon_0 > 0$ such that

$$GK_0 + (GK_0)^* \geq \varepsilon_0^2 I \quad (\textit{A Stronger Strict Positivity Condition}). \tag{7.54}$$

In both the finite and infinite dimensional cases, the validity of condition (7.54) implies that there exists a real number $\beta^* > 0$ such that, for any choice of $\beta \in (0, \beta^*)$, it is possible to compute a real number $\lambda(\beta) < 1$ (dependent on β) such that monotonic error convergence is guaranteed as $\|(I - \beta GK_0)\| \leq \lambda(\beta) < 1$. Finally, the largest possible choice of β^* lies in the range

$$\beta^{**} = \frac{\varepsilon_0^2}{\|GK_0\|^2} \leq \beta^* \leq \frac{2}{\|GK_0\|}. \tag{7.55}$$

The algorithm can be interpreted in computational terms as providing conditions for $K_0 e_k$ to be a *Descent Direction* for $\|e\|_{\mathscr{Y}}^2$ at e_k. The existence of $\varepsilon_0^2 > 0$ then guarantees the existence of a range of gains β that produce norm reduction independent of e_0 and hence independent of e_k at each iteration. Insight into the general meaning of the conditions can be obtained from the special case where both G and K_0 are the matrices in the supervector descriptions of discrete time, state space models of transfer function matrices $G(z)$ and $K(z)$ on some interval $0 \leq t \leq N$.

1. Theorems 4.5 and 4.6 indicate that a sufficient condition for (7.54) to hold is that

$$H(z) = QG(z)K_0(z) + K_0^T(z^{-1})G^T(z^{-1})Q \geq \varepsilon_0^2 Q \quad \textit{whenever } |z| = 1\,. \tag{7.56}$$

   In the simplest case when $Q = I_m$, the matrix $H(z)$ is Hermitian with real eigenvalues $h_{min}(z) = h_1(z) \leq h_2(z) \leq \cdots \leq h_m(z)$ and the condition becomes $\min_{|z|=1} h_{min}(z) > 0$. It is then possible to choose $\varepsilon_0^2 = \min_{|z|=1} h_{min}(z)$.

   Note: this condition is particularly simple in the case of SISO systems ($m = \ell = 1$) when it is equivalent to the requirement that the Nyquist plot of $GK_0(z)$ lies in the region to the right of the vertical line $\{\, z \,:\, 2Re[z] = \varepsilon_0^2 > 0 \,\}$ in the complex plane. It indicates the role of K_0 as a phase compensation component as, if G has substantial phase lag, it will be essential that K_0 has compensating phase lead. The use of $K_0 = G^{-1}$ provided this property for inverse model algorithms but at the price of introducing complexity and possible controller instabilities.
2. Given a value of $\varepsilon_0^2 > 0$, Theorems 4.5 and 4.6 then indicate that convergence to zero is guaranteed if

$$\begin{aligned} &\|Q^{1/2}(I - \beta G(z)K_0(z))Q^{-1/2}\|_\infty \\ &\quad = \sqrt{\sup_{|z|=1} r\left(Q^{-1}(I_m - G(z^{-1})K_0(z^{-1}))^T Q(I_m - G(z)K_0(z))\right)} \;<\; 1\,, \end{aligned} \tag{7.57}$$

   a condition that can be checked computationally. The existence of suitable values of β is guaranteed up to a value of β^* that can be taken to be in the range obtained by replacing $\|GK_0\|$ by the upper bound

$$\|Q^{1/2}G(z)K_0(z)Q^{-1/2}\|_\infty = \sqrt{\sup_{|z|=1} r\left(Q^{-1}K_0^T(z^{-1})G^T(z^{-1})QG(z)K_0(z)\right)}\,. \tag{7.58}$$

Note: Again the SISO case simplifies the picture with Eq. (7.57) being interpreted as the need for the Nyquist plot of $G(z)K_0(z)$ to be contained in the open circle of centre $(\beta^{-1}, 0)$ and radius β^{-1} in the complex plane.

It is essential to examine the case when the strict positivity condition (7.54) is not satisfied. An alternative approach can be used if $ker[G] = \{0\}$ using subsidiary signals in a similar manner to the left inverse model algorithm. It is left as an exercise for the reader to confirm the following statements.

Algorithm 7.9 (*Positivity of K_0G and Monotonicity of Subsidiary Signals*) Suppose that $\mathscr{U}$ is a finite or infinite dimensional, real Hilbert space with inner product $\langle\cdot,\cdot\rangle_{\mathscr{U}}$ and associated induced norm. Suppose that $ker[G] = \{0\}$, that $e_0 \in \mathscr{R}[G]$ and that u_∞ is the unique input that provides exact tracking of r i.e. $r = Gu_\infty$. Consider the feedforward Iterative Control algorithm $u_{k+1} = u_k + K_0e_k,\ k \geq 0$ with $K_0 : \mathscr{Y} \to \mathscr{U}$ linear and bounded. Then $e_k \in \mathscr{R}[G]$ for all $k \geq 0$ and, writing $e_k = Gw_k$ for all $k \geq 0$, the subsidiary signals $\{w_k\}_{k\geq 0}$ in $\mathscr{U}$ are simply $w_k = u_\infty - u_k,\ k \geq 0$, and satisfy the evolution equation

$$w_{k+1} = (I - \beta K_0G)w_k, \quad k \geq 0\,. \tag{7.59}$$

Under these conditions, a necessary condition for monotonic convergence of the subsidiary signals is that

$$K_0G + (K_0G)^* > 0 \quad (\textit{A Strict Positivity Condition}). \tag{7.60}$$

Moreover, if $\mathscr{U}$ is finite dimensional, it is necessary that there exists a real number $\varepsilon_0 > 0$ such that

$$K_0G + (K_0G)^* \geq \varepsilon_0^2 I \quad (\textit{A Stronger Strict Positivity Condition}). \tag{7.61}$$

In both the finite and infinite dimensional cases, the validity of condition (7.61) implies that there exists a real number $\beta^* > 0$ such that, for any choice of $\beta \in (0, \beta^*)$, it is possible to compute a real number $\lambda(\beta) < 1$ (dependent on β) such that monotonic convergence of the subsidiary signals is guaranteed as $\|(I - \beta K_0G)\| \leq \lambda(\beta) < 1$. Finally, $\{e_k\}_{k\geq 0}$ is then convergent to zero and the largest possible choice of β^* lies in the range

$$\beta^{**} = \frac{\varepsilon_0^2}{\|K_0G\|^2} \leq \beta^* \leq \frac{2}{\|K_0G\|}. \tag{7.62}$$

If G and K_0 are the matrices in the supervector descriptions of discrete time state space models of transfer function matrices $G(z)$ and $K(z)$ on some interval $0 \leq t \leq N$

then Theorem 4.9 indicates that a sufficient condition for (7.61) to hold is that

$$RK_0(z)G(z) + G^T(z^{-1})K_0^T(z^{-1})R \geq \varepsilon_0^2 R \quad \textit{whenever} \quad |z| = 1\,. \tag{7.63}$$

Convergence is then guaranteed for $\beta \in (0, \beta^*)$ if

$$\begin{aligned}&\|R^{1/2}(I - \beta K_0(z)G(z))R^{-1/2}\|_\infty \\ &\quad = \sqrt{\sup_{|z|=1} r\left(R^{-1}(I_\ell - K_0(z^{-1})G(z^{-1}))^T R(I_\ell - K_0(z)G(z))\right)} \;<\; 1\end{aligned} \tag{7.64}$$

whilst Theorems 4.5 and 4.6 indicate that a suitable value of β^* can be taken to be in the range obtained by replacing $\|K_0 G\|$ by

$$\|R^{1/2}K_0(z)G(z)R^{-1/2}\|_\infty = \sqrt{\sup_{|z|=1} r\left(R^{-1}G^T(z^{-1})K_0^T(z^{-1})RK_0(z)G(z)\right)}\,. \tag{7.65}$$

Note: In the case of SISO systems the two analyses are identical as both Q and R cancel from the relationships and, trivially, $G(z)K_0(z) = K_0(z)G(z)$.

7.5 Discrete State Space Models Revisited

7.5.1 Gradients Using the Adjoint of a State Space System

Steepest descent algorithms use the adjoint G^* of the operator G associated with the plant as the choice of feedforward element K_0. The benefits and simplicity seen in this case suggests that the choice of K_0 as the adjoint K^* of an operator K that is simpler than, but not necessarily the same as, the plant G might provide useful control laws and design tools. Typically, it could be envisaged that, by choosing K as a simplified model of G with reduced state dimension, the computational load when computing the input signal u_{k+1} from u_k and e_k will be reduced whilst simultaneously satisfying other conditions (such as those discussed in what follows) that ensure monotonic convergence.

Suppose that G is generated by an m-output, ℓ-input, asymptotically stable, discrete, state space system $S(A, B, C, D)$ and that the compensator $K_0 = K^*$ is generated as the adjoint K^* of the m-output, ℓ-input, asymptotically stable system K described by the state space model $S(A_K, B_K, C_K, D_K)$. The topologies in $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ and $\mathscr{U} = \mathscr{R}^{\ell(N+1)}$ are assumed to be those defined in Theorem 7.4.

Algorithm 7.10 (*Gradient Algorithms Using an Adjoint System K^**) Using the notation defined above, an Iterative Algorithm using the adjoint K^* of $S(A_K, B_K, C_K, D_K)$, takes the form of the input supervector update computation

$$u_{k+1} = u_k + \beta K^* e_k, \quad k \geq 0\,, \tag{7.66}$$

with scalar gain $\beta > 0$. In state space form, this update takes the form

$$\begin{aligned} u_{k+1}(t) &= u_k(t) + \beta R^{-1} v_{k+1}(t) \quad (\textit{The Input Update Rule}) \\ \textit{where}\ \ v_{k+1}(t) &= B_K^T p_{k+1}(t) + D_K^T Q e_k(t) \quad \textit{for} \quad t = 0, 1, 2, \ldots, N \\ \textit{and}\ \ p_{k+1}(t) &= A_K^T p_{k+1}(t+1) + C_K^T Q e_k(t+1), \quad \textit{with}\ \ p_{k+1}(N) = 0. \end{aligned} \tag{7.67}$$

It is assumed throughout the following analysis that both G and K are asymptotically stable and that

$$rank[D] = rank[D_K] = \min\{m, \ell\}. \tag{7.68}$$

In particular, this guarantees the existence of left or right inverses.

Computational Note: *The condition* $rank[D_K] = \min\{m, \ell\}$ *is a design choice. In contrast, achieving the condition* $rank[D] = \min\{m, \ell\}$ *will typically require a redefinition of the output using techniques such as those described in Chap. 4.*

In a similar manner to Sect. 7.2.3, the response of the non-causal system GK^* to a single frequency input time series $u(t, z, \alpha) = z^t\alpha,\ 0 \leq t \leq N$ and $|z| = 1$, has the form of a time series

$$y(t, z, \alpha) = G(z)R^{-1}K^T(z^{-1})Qu(t, z, \alpha) + \eta_2(t)\,, \quad 0 \leq t \leq N\,, \tag{7.69}$$

where $\|\eta_2(t)\| \leq M\|\alpha\|\left(\lambda^t + \lambda^{N-t}\right)$ for some $M > 0$ and $\lambda \in (0, 1)$. This suggests that the transfer function matrices $G(z)$, $K(z)$ and $G(z)R^{-1}K^T(z^{-1})Q$ are strongly related to both performance and convergence properties. In what follows, frequency domain conditions describing monotonic convergence of the algorithm are derived.

The Case of $m \leq \ell$: Noting that the spaces are finite dimensional, monotonic convergence using the algorithm $u_{k+1} = u_k + \beta K^* e_k$ with $\beta > 0$ is guaranteed if, in the topology of $\mathscr{Y}$, the inequality $(I - \beta GK^*)^*(I - \beta GK^*) < I$ is satisfied. That is,

$$GK^* + KG^* > \beta KG^*GK^*\,, \tag{7.70}$$

which requires that $ker[GK^*] = ker[KG^*] = \{0\}$, and hence that both matrices GK^* and KG^* are nonsingular. In addition, this implies the necessary conditions

$$\begin{aligned} ker[G^*] = ker[K^*] = \{0\}\,, \quad \mathscr{R}[G] = \mathscr{R}[K] = \mathscr{Y}\,, \\ \textit{and} \quad \mathscr{R}[K^*] \cap ker[G] = \mathscr{R}[G^*] \cap ker[K] = \{0\}\,, \\ \textit{which implies that} \quad \mathscr{U} = \mathscr{R}[K^*] \oplus ker[G] = \mathscr{R}[G^*] \oplus ker[K]\,. \end{aligned} \tag{7.71}$$

The basic assumption made in the analysis is that the control element K and plant G are related by an expression

$$G = KU. \tag{7.72}$$

As both G and K are matrices in the supervector description of the plant and controller, it follows that U is a (possibly non-unique) matrix operator $U : \mathscr{U} \to \mathscr{U}$. This description is sufficient for a matrix analysis of algorithm properties but, if a frequency domain interpretation is sought, it seems to be natural to seek a realization of U as the matrix representation of an ℓ-input, ℓ-output, linear, state space system $S(A_U, B_U, C_U, D_U)$ with $\ell \times \ell$ transfer function matrix $U(z)$. From this perspective, the following relationship must hold,

$$G(z) = K(z)U(z), \tag{7.73}$$

Note that a simple algebraic analysis indicates that such a transfer function matrix always exists. For current purposes, $U(z)$ itself requires some structure and it is assumed that it is asymptotically stable.

The simplest example of this construction is when $m = \ell$ and $K(z)$ is invertible. It then follows that $U(z) = K^{-1}(z)G(z)$ which is stable if G is stable and K is minimum-phase. More generally, if $m \leq \ell$, suppose that K has an asymptotically stable right inverse K_R generated from a discrete state space model $S(A_{KR}, B_{KR}, C_{KR}, D_{KR})$ with $\ell \times m$ transfer function matrix $K_R(z)$. In particular, note that $K(z)K_R(z) = I_m$ and, in their supervector form, $KK_R = I_{m(N+1)}$. In this case, U can be expressed in the form

$$U = K_R G + U_0 \quad \textit{with transfer function matrix} \quad U(z) = K_R(z)G(z) + U_0(z), \tag{7.74}$$

where U_0 is arbitrary provided that it is asymptotically stable and satisfies $KU_0 = 0$. Note that, as it is a right inverse, it is necessary that $ker[K_R] = \{0\}$ from which it follows that

$$\begin{gathered} D_K D_{KR} = I_m, \quad ker[D_{KR}] = \{0\} \quad \textit{and also that} \\ ker[U] = ker[G] \quad \textit{if} \quad U_0 = 0. \end{gathered} \tag{7.75}$$

Using the matrix description $G = KU$, monotonic convergence requires that the operators satisfy

$$\begin{gathered} K\left(U + U^* - \beta G^* G\right) K^* > 0 \quad \textit{or, equivalently,} \\ \Psi(U, G, \beta) = U + U^* - \beta G^* G > 0 \quad \textit{on the subspace } \mathscr{R}[K^*] \subset \mathscr{U}. \end{gathered} \tag{7.76}$$

As operator properties, these define monotonic convergence although, if U is non-unique, it is necessary to find an appropriate choice. In particular, it is sufficient to find a U that satisfies the conditions and is also positive semi-definite on the full space $\mathscr{U}$. Assuming this possibility, the next step uses the approach described in Theorem 4.9

with K replaced by U. It reduces the analysis to an examination of the behaviour of the quadratic form $\langle u, \Psi(U, G, \beta)u\rangle_{\mathscr{U}}$ on the full space $\mathscr{U}$. In particular, assume now that there exists at least one $\beta_1 > 0$ such that $\Psi(U, G, \beta_1) \geq 0$ on $\mathscr{U}$. Then, as $\langle u, \Psi(U, G, \beta)u\rangle_{\mathscr{U}} = \langle u, \Psi(U, G, \beta_1)u\rangle_{\mathscr{U}} + (\beta_1 - \beta)\|Gu\|^2_{\mathscr{Y}}$, it follows that, for all $\beta \in (0, \beta_1)$, the operator $\Psi(U, G, \beta) \geq 0$ on $\mathscr{U}$ but $\Psi(U, G, \beta) > 0$ on $\mathscr{R}[K^*]$ if $\mathscr{R}[K^*] \cap ker[G] = \{0\}$. Let $\beta^* > 0$ be the largest value of β_1 with this property. That is

$$\beta^* = \sup\{\, \beta \;:\; \Psi(U, G, \beta) \geq 0 \quad \textit{on the full space } \mathscr{U} \,\}. \tag{7.77}$$

It follows that, for any $\beta \in (0, \beta^*)$, the condition $\mathscr{R}[K^*] \cap ker[G] = \{0\}$ ensures that $\Psi(U, G, \beta) > 0$ on $\mathscr{R}[K^*]$ and hence that the operator norm $\|(I - GK^*)\| < 1$. Expressed in frequency domain terms,

Theorem 7.6 (Frequency Domain Convergence Conditions for Non-causal Compensators) *Using the notation of the preceding discussion, suppose that $m \leq \ell$, that G is asymptotically stable and that G and K are related by an expression $G(z) = K(z)U(z)$ where $U(z)$ is proper and asymptotically stable. Suppose also that there exists a real scalar $\beta_1 > 0$ and a choice of $U(z)$ such that, for all z satisfying $|z| = 1$, the matrix equation*

$$RU(z) + U^T(z^{-1})R \geq \beta_1 G^T(z^{-1})QG(z) \tag{7.78}$$

is valid in the (complex) Euclidean Topology in $\mathscr{C}^\ell$. Let $\beta^* > 0$ be the largest such value of β_1. Then, for all choices of $\beta \in (0, \beta^*)$, the norm $\|(I - \beta GK^*)\| \leq 1$ and the feedforward algorithm represented by the supervector update formula $u_{k+1} = u_k + \beta K^* e_k$ is monotonically bounded with $\|e_{k+1}\|_{\mathscr{Y}} \leq \|e_k\|_{\mathscr{Y}} \leq \|e_0\|_{\mathscr{Y}}$ for all $k \geq 0$. If, in addition, $\mathscr{R}[K^*] \cap ker[G] = \{0\}$, then $\|(I - \beta GK^*)\| < 1$ and the sequence is monotonically convergent to the unique limit $e_\infty = 0$.

Proof In supervector form, using Theorem 4.9 with K replaced by U, the condition (7.78) is a sufficient condition to ensure that $U + U^* \geq \beta G^* G$ on $\mathscr{U}$ for all $\beta \in (0, \beta^*)$. This proves that $\|(I - \beta GK^*)\| \leq 1$ and hence the monotonic boundedness property. Also $U + U^* > \beta G^* G$ on $\mathscr{R}[K^*]$ (and hence $\|(I - \beta GK^*)\| < 1$) if $\mathscr{R}[K^*] \cap ker[G] = \{0\}$. This proves monotonic convergence to zero and completes the proof of the result. □

If the frequency domain inequality is satisfied, the boundedness of the error sequence is guaranteed by the result. This is reassuring for practical applications as it indicates that, in the absence of modelling errors and/or worst case disturbances on each iteration, error norm magnitudes will not increase. However, convergence to zero error then depends on the truth of the condition $\mathscr{R}[K^*] \cap ker[G] = \{0\}$. In matrix terms, this is achieved if GK^* is nonsingular, a condition satisfied by many choices of K including, whenever $\mathscr{R}[G] = \mathscr{Y}$, the choice of $K = G$ as then $\mathscr{R}[G^*]^\perp = ker[G]$. There are an infinity of other choices, including those "close to G" as exemplified by the following result,

Theorem 7.7 ($\mathscr{R}[K^*] \cap ker[G] = \{0\}$, $(K-G)^*$ and Algorithm Convergence) *With the notation of the discussion above, suppose that* $ker[G^*] = ker[K^*] = \{0\}$ *(and hence that* $\mathscr{R}[G] = \mathscr{R}[K] = \mathscr{Y}$*).*

1. *Then, a sufficient condition for the relation* $\mathscr{R}[K^*] \cap ker[G] = \{0\}$ *to be true is that, in the assumed topologies of* $\mathscr{U}$ *and* $\mathscr{Y}$*,*

$$\begin{array}{c} \|(K-G)^*y\|_{\mathscr{U}} < \|G^*y\|_{\mathscr{U}} \quad \textit{for all non-zero } y \in \mathscr{Y} \\ (\textit{so that } \; (K-G)(K-G)^* < GG^* \; \textit{ and } \; KK^* < GK^* + KG^* \;) \; . \end{array} \tag{7.79}$$

2. *Under these conditions, the Iterative Algorithm* $u_{k+1} = u_k + \beta K^* e_k$ *is monotonically convergent whenever* $0 < \beta \|G\|^2 \leq 1$.
3. *Suppose that K is one such operator, then, as the condition* $\mathscr{R}[K^*] \cap ker[G] = \{0\}$ *will also be valid if any such K is replaced by* γK *with* $\gamma > 0$*, it follows that the set of compensators can be extended to elements of the "positive cone" defined by the set*

$$\{ \, K = \gamma \tilde{K} \; : \; \gamma > 0 \;\; \textit{and} \;\; \|(\tilde{K}-G)^*y\|_{\mathscr{U}} < \|G^*y\|_{\mathscr{U}} \;\; \textit{for all} \;\; y \in \mathscr{Y} \, \} \; . \tag{7.80}$$

Proof If $\mathscr{R}[K^*] \cap ker[G] \neq \{0\}$, then there exists a non-zero y such that $GK^*y = 0$ and hence $G(K-G)^*y + GG^*y = 0$. That is, $\langle y, G(K-G)^*y \rangle_{\mathscr{Y}} + \|G^*y\|^2_{\mathscr{U}} = 0$. Using the Cauchy Schwarz inequality, $|\langle y, G(K-G)^*y \rangle| \leq \|G^*y\| \|(K-G)^*y\|$ which leads to a violation of the assumed inequality if $y \neq 0$. Hence $y = 0$ which proves the first part of the result. Next note that $\beta G^*G \leq \beta \|G\|^2 I \leq I$, so that

$$GK^* + KG^* > KK^* \geq \beta KG^*GK^* \tag{7.81}$$

which proves monotonic convergence. The cone characterization is a simple consequence of invariance of subspaces under scalar multiplication. □

In frequency domain terms,

Theorem 7.8 *A sufficient condition in the frequency domain for* $\mathscr{R}[K^*] \cap ker[G] = \{0\}$ *to be true is that there exists a proper, asymptotically stable, dynamical system* $S(A_U, B_U, C_U, D_U)$ *with* $\ell \times \ell$ *transfer function matrix* $U(z)$ *that satisfies the two conditions*

$$\begin{array}{c} G(z) = K(z)U(z) \; , \quad \textit{and also that} \\ RU(z) + U^T(z^{-1})R \; > \; 0 \; , \quad \textit{whenever} \;\; |z| = 1 \; . \end{array} \tag{7.82}$$

Proof Continuing with the notation of the proof of the previous result, write G=KU. The condition $RU(z) + U^T(z^{-1})R > 0$, whenever $|z| = 1$ implies the existence of a real scalar $\gamma > 0$ such that

$$RU(z) + U^T(z^{-1})R \; > \; \gamma R \; , \quad \textit{whenever} \;\; |z| = 1 \; . \tag{7.83}$$

Using Theorem 4.9 with K replaced by U, setting $\mathscr{Y} = \mathscr{U}$ and replacing G by the identity verifies the operator inequality $U + U^* > \gamma I$ and hence that $GK^* + KG^* > \gamma KK^*$. A simple calculation then indicates that γK satisfies the conditions derived in the first part of this result and hence $\mathscr{R}[K^*] \cap ker[G] = \{0\}$. □

The case of $m = \ell$ is simpler to state and is left as an exercise for the reader.

Theorem 7.9 ($\mathscr{R}[K^*] \cap ker[G] = \{0\}$ whenever $m = \ell$) *Using the notation given above, suppose that $m = \ell \geq 1$, that both D and D_K are nonsingular. Then $ker[G] = \{0\}$ and hence $\mathscr{R}[K^*] \cap ker[G] = \{0\}$ and $\mathscr{U} = \mathscr{R}[K^*] \oplus ker[G]$ trivially.*

The results given above require the identification of a suitable, asymptotically stable U to satisfy the desired inequality. If $m < \ell$, it is expected that the choices available for U will be non-unique. The consequences are simplified when $m = \ell$ and K is invertible as then there is only one possible choice for U, namely $U(z) = K^{-1}(z)G(z)$. The following result defines convergence conditions very precisely in this case. The proof is left as an exercise for the reader but consists, essentially, of substituting the expression $U(z) = K^{-1}(z)G(z)$ to simplify Eq. (7.78).

Theorem 7.10 (Spectral Radius Condition for Convergence when $m = \ell$) *Suppose that $m = \ell$, that $G(z)$ is asymptotically stable, that $K(z)$ is minimum-phase and asymptotically stable and that the determinant* $\det[G(z)R^{-1}K^T(z^{-1})Q] \neq 0$ *for all z satisfying $|z| = 1$. Under these conditions, $K_R(z) = K^{-1}(z)$ exists, is asymptotically stable and Eq. (7.78) is true if, whenever $|z| = 1$,*

$$G(z)R^{-1}K^T(z^{-1}) + K(z)R^{-1}G^T(z^{-1}) > \beta_1 K(z)R^{-1}G^T(z^{-1})QG(z)R^{-1}K^T(z^{-1}) \quad (7.84)$$

Moreover, this condition is equivalent to the spectral radius condition

$$\sup_{|z|=1} r\left((I_m - \beta K(z)R^{-1}G^T(z^{-1})Q)(I_m - \beta G(z)R^{-1}K^T(z^{-1})Q)\right) < 1 \quad (7.85)$$

Proof Let $|z| = 1$. Multiplying Eq. (7.78) from the left by the nonsingular matrix $K(z)R^{-1}$ and from the right by its complex conjugate transpose $R^{-1}K^T(z^{-1})$ yields the required inequality which, by construction, is equivalent to (7.78). The spectral radius condition then follows by multiplying from the left and right by $Q^{1/2}$ and writing the derived frequency domain inequality, in the Euclidean topology in $\mathscr{C}^m$, as, for all $|z| = 1$,

$$(I_m - \beta Q^{1/2}K(z)R^{-1}G^T(z^{-1})Q^{1/2})(I_m - \beta Q^{1/2}G(z)R^{-1}K^T(z^{-1})Q^{1/2}) < I\,, \quad (7.86)$$

which, as the left-hand-side is Hermitian, is just the spectral radius condition

$$\sup_{|z|=1} r\left(Q^{1/2}(I_m - \beta K(z)R^{-1}G^T(z^{-1})Q)(I_m - \beta G(z)R^{-1}K^T(z^{-1})Q)Q^{-1/2}\right) < 1\,. \quad (7.87)$$

This is the required result as the spectral radius is unchanged under the similarity transformation $H \mapsto Q^{-1/2}HQ^{1/2}$. □

The importance of the spectral radius condition to convergence in this case suggests the possibility of using these ideas as the basis of systematic frequency domain design. The concept is supported by considering the case of SISO systems. When $m = \ell = 1$, the analysis reduces to a graphical condition relating convergence to gain/phase characteristics of the transfer function $G(z)K(z^{-1})$. More precisely, bringing previous results together gives the following result.

Theorem 7.11 (The Single-input Single-output Case, $m = \ell = 1$) *Using the notation and assumptions given above, suppose that $m = \ell = 1$ and that both $D \neq 0$ and $D_K \neq 0$. Then $\mathscr{R}[K^*] \cap ker[G] = \{0\}$. In these circumstances, a sufficient condition for monotonic convergence of the iterative algorithm $u_{k+1} = u_k + \beta K^* e_k$, with $\beta > 0$, is that*

$$|\frac{R}{\beta Q} - G(z)K(z^{-1})| < \frac{R}{\beta Q} \quad \textit{for all } |z| = 1. \tag{7.88}$$

An equivalent graphical condition is that the Nyquist plot of $G(z)K(z^{-1})$ on the unit circle $z = e^{i\theta}, 0 \leq \theta \leq \pi$, lies entirely within the interior of the circle of radius $\frac{R}{\beta Q}$ and centre $(\frac{R}{\beta Q}, 0)$ in the complex plane. Finally,

1. *if β^* is the least upper bound of all such values of β, then monotonic convergence is assured for all choices $\beta \in (0, \beta^*)$.*
2. *A necessary condition for the graphical frequency domain condition to be valid (for some value of $\beta > 0$) is that the Nyquist plot of $G(z)K(z^{-1})$ lies entirely in the open right-half complex plane defined by $\{s : Re[s] > 0\}$ (another form of Positive Real Property!).*

Note: the result indicates that the effects of Q and R are simply to scale the gain β. It is expected in practice that the choice of $Q = R = 1$ will simplify the analysis. Alternatively, setting $\beta = 1$ will leave the ratio $R^{-1}Q$ as an effective "gain" to be chosen to achieve the required convergence condition.

The conditions are easily checked for a given choice of $K(z)$ but are expressed in terms of $K(z^{-1})$. The phase $\phi_{GK^*}(z)$ of $G(z)K(z^{-1})$ is the phase $\phi_G(z)$ of $G(z)$ minus the phase $\phi_K(z)$ of $K(z)$,which, together with the required positive real property indicates a design objective of ensuring that

$$-\frac{\pi}{2} < \phi_G(z) - \phi_K(z) < \frac{\pi}{2} \quad \textit{whenever} \quad |z| = 1. \tag{7.89}$$

If achieved, the Nyquist plot of $G(z)K(z^{-1})$ lies in the interior of the right half complex plane and the computation of the permissible range $(0, \beta^*)$ of β then follows from the graphical condition. Additional frequency shaping could be approached

using a suitable transfer function $F_B(z)$ and choosing $K(z)$ to satisfy an inequality of the form (assuming the choice of $Q = R = 1$)

$$|1 - \beta G(z)K(z^{-1})|^2 \leq 1 - |F_B(z)|^2 \quad \textit{on the unit circle} \quad |z| = 1. \tag{7.90}$$

The magnitude and bandwidth of F_B will provide a method of influencing the nature and convergence rate of the algorithm in situations where the reference signal has a frequency content dominated by frequencies in the chosen bandwidth.
The case of $m < \ell$ has more structure but has strong relationship with the spectral radius conditions. More precisely, an identical algebraic approach yields the statement,

Theorem 7.12 (Convergence and the Spectral Radius when $m < \ell$) *Suppose that $m < \ell$ and that the conditions of Theorem 7.6 are satisfied and that $U(z)$ satisfies Eq. (7.78). Then, whenever $|z| = 1$,*

$$G(z)R^{-1}K^T(z^{-1}) + K(z)R^{-1}G^T(z^{-1}) \geq \beta_1 K(z)R^{-1}G^T(z^{-1})QG(z)R^{-1}K^T(z^{-1}) \tag{7.91}$$

Moreover, this condition is equivalent to the spectral radius condition

$$\sup_{|z|=1} r\left((I_m - \beta K(z)R^{-1}G^T(z^{-1})Q)(I_m - \beta G(z)R^{-1}K^T(z^{-1})Q)\right) < 1\,. \tag{7.92}$$

The Final Case of $m \geq \ell$: This case uses subsidiary signals as the basis of analysis. Supposing that $ker[G] = \{0\}$, then any error $e \in \mathscr{R}[G]$ can be written uniquely in the form $e = Gw$ where $w \in \mathscr{U}$. Note that, if u_∞ is an input that generates a zero tracking error $e = 0$, then $w = u_\infty - u$. The evolution of e_k can then be replaced by the evolution of w_k using the expression

$$e_k = Gw_k \quad \textit{and} \quad w_{k+1} = (I - \beta K^* G)w_k, \quad \textit{for} \quad k \geq 0. \tag{7.93}$$

Convergence analysis for the case of $e_0 \in \mathscr{R}[G]$ can then proceed by examining conditions for the monotonic convergence of the associated sequence $\{w_k\}_{k\geq 0}$ to a limit $w_\infty = 0$. As $\|e_k\| \leq \|G\|\|w_k\|$, this will then ensure the convergence of the error to a limit $e_\infty = 0$. More generally, if e_0 is an arbitrary starting condition, then, assuming that the conditions

$$\mathscr{Y} = \mathscr{R}[G] \oplus \ker[K^*], \quad (\textit{and hence that} \quad \mathscr{R}[G] \cap ker[K^*] = \{0\})\,, \tag{7.94}$$

are satisfied, the reader can verify that, by writing $e_0 = Gw_0 + \xi$ with $\xi \in ker[K^*]$, the error evolution can be described by the equation $e_k = Gw_k + \xi$ for all $k \geq 0$ where $w_{k+1} = (I - K^*G)w_k$. That is, even if convergence of w_k to zero is achieved, the component ξ of e_0 in $ker[K^*]$ remains unchanged. In fact, under these conditions, $e_k \to e_\infty = \xi$ as $k \to \infty$.

Returning to the case of $e_0 \in \mathscr{R}[G]$, a simple calculation, using the assumption that $\beta > 0$, indicates that monotonic convergence of $\{w_k\}_{k\geq 0}$ to $w_\infty = 0$ is guaranteed if, in the topology of $\mathscr{U}$,

$$K^*G + G^*K > \beta G^*KK^*G \tag{7.95}$$

A simple comparison with the case of $m \leq \ell$ indicates that the mathematics is very similar with the substitution of GK^* by K^*G which is just the substitutions $G \mapsto K^*$ and $K^* \mapsto G$. The consequences of these facts can be used, together with Theorem 4.11, to give frequency domain characterizations of convergence properties. The approach and conclusions are summarized below.

Note that both K^*G and G^*K must have kernels $\{0\}$. As a consequence $ker[G] = ker[K] = \{0\}$ and $\mathscr{R}[K] \cap ker[G^*] = \mathscr{R}[G] \cap ker[K^*] = \{0\}$). As a consequence, $\mathscr{Y} = \mathscr{R}[G] \oplus \ker[K^*]$ as, if $y \in \mathscr{Y}$ is non-zero and orthogonal to both $\mathscr{R}[G]$ and $\ker[K^*]$, then $y \in \mathscr{R}[K] \cap ker[G^*]$. That is, $y = 0$ contrary to assumption.

Following the approach used for the case of $m \leq \ell$, assume that $K = UG$ where $U : \mathscr{Y} \to \mathscr{Y}$ is bounded. For example, suppose that G represents a discrete, state space system and has a proper left inverse $S(A_{GL}, B_{GL}, C_{GL}, D_{GL})$ where $rank[D_{GL}] = \ell$. Define the operator $U = KG_L + U_0$ where U_0 satisfies $U_0G = 0$ but is otherwise arbitrary. A simple calculation gives $UG = KG_LG + U_0G = K$. The condition for monotonic convergence then becomes

$$\begin{aligned} G^*\left(U + U^* - \beta KK^*\right)G > 0 \quad & on\ \mathscr{U} \\ which\ is\ just \quad U + U^* - \beta KK^* > 0 \quad & on\ \mathscr{R}[G]. \end{aligned} \tag{7.96}$$

Using Theorem 4.11, the natural parallel to Theorem 7.6 can be stated as follows

Theorem 7.13 (Frequency Domain Convergence Conditions when $m \geq \ell$) *Using the notation of the preceding discussion, suppose that $m \geq \ell$ and that $ker[G] = \{0\}$. Suppose also that there exists a real scalar $\beta_1 > 0$ and an $m \times m$, asymptotically stable system $U(z)$ such that $K(z) = U(z)G(z)$ and, for all z satisfying $|z| = 1$, the matrix equation*

$$U^T(z)Q + QU(z^{-1}) \geq \beta_1 QK(z^{-1})R^{-1}K^T(z)Q \tag{7.97}$$

is valid in the (complex) Euclidean Topology in $\mathscr{C}^m$. Let $\beta^* > 0$ be the largest such value. Then, for all choices of $\beta \in (0, \beta^*)$, the induced operator norm $\|(I - \beta K^*G)\| \leq 1$ and the feedforward algorithm represented by the supervector update formula $u_{k+1} = u_k + \beta K^*e_k$ generates bounded sequences $\{e_k\}_{k\geq 0}$ and $\{w_k\}_{k\geq 0}$ with $\|w_{k+1}\|_{\mathscr{U}} \leq \|w_k\|_{\mathscr{U}} \leq \|w_0\|_{\mathscr{U}}$ and $\|e_k\|_{\mathscr{Y}} \leq \|G\|\|w_k\|_{\mathscr{U}}$ for all $k \geq 0$. If, in addition, $\mathscr{R}[G] \cap ker[K^*] = \{0\}$, then $\|(I - \beta K^*G)\| < 1$ and the sequence $\{w_k\}_{k\geq 0}$ is monotonically convergent to the limit $w_\infty = 0$. In particular, $\lim_{k\to\infty} e_k$ is equal to the component of e_0 in $ker[K^*]$.

Note that the condition $\mathscr{R}[G] \cap ker[K^*] = \{0\}$ is essential to the proof and parallels the need for $\mathscr{R}[K^*] \cap ker[G] = \{0\}$ when $m \leq \ell$. The condition is easily seen to be

valid if $m = \ell$ and both D and D_K are nonsingular as then $ker[K^*] = \{0\}$ trivially. A parallel to Theorem 7.7 is as follows,

Theorem 7.14 (A Sufficient Condition for $\mathscr{R}[G] \cap ker[K^*] = \{0\}$ to be Valid) *Suppose that* $ker[K] = \{0\}$. *Then, a sufficient condition for the relation* $\mathscr{R}[G] \cap ker[K^*] = \{0\}$ *to be true is that, in the topology of* $\mathscr{U}$,

$$\begin{array}{c} \|(K-G)u\|_{\mathscr{Y}} < \|Ku\|_{\mathscr{Y}} \quad \textit{for all} \quad u \in \mathscr{U} \\ (\textit{so that} \quad (K-G)^*(K-G) < K^*K \quad \textit{and} \quad G^*G < G^*K + K^*G\ ) \ . \end{array} \tag{7.98}$$

Under these conditions, the Iterative algorithm is convergent for all β *satisfying* $0 < \beta \|K^*\|^2 \leq 1$.

Also a direct parallel to Theorems 7.10 and 7.12 can be stated in the form,

Theorem 7.15 (Spectral Radius Condition for Convergence when $m \geq \ell$) *Suppose that the determinant* $\det[R^{-1}K^T(z)QG(z^{-1})] \neq 0$ *for all* z *satisfying* $|z| = 1$. *Then,*

1. *If* $m = \ell$, *then Eq.* (7.97) *is equivalent to the existence of* $\beta_1 > 0$ *such that*

$$K^T(z^{-1})QG(z) + G^T(z^{-1})QK(z) \geq \beta_1 G^T(z^{-1})QK(z)R^{-1}K^T(z^{-1})QG(z) \tag{7.99}$$

for all z *satisfying* $|z| = 1$. *Moreover, if* β^* *is the largest such value of such* β_1 *and* $\beta \in (0, \beta^*)$, *convergence to zero error is guaranteed by the spectral radius condition*

$$\sup_{|z|=1} r\left((I_\ell - \beta R^{-1}G^T(z^{-1})QK(z))(I_\ell - \beta R^{-1}K^T(z^{-1})QG(z))\right) < 1 \tag{7.100}$$

2. *If* $m > \ell$ *and, also,* $U(z)$ *satisfies the requirements of Theorem* 7.13, *then the spectral radius condition is satisfied.*

7.5.2 Why the Case of $m = \ell$ May Be Important in Design

The previous section has shown that the case of $m = \ell$ is, on paper, simpler than the cases when $m \neq \ell$. Spectral radius conditions can be associated with all cases but, whenever $m \neq \ell$, the non-uniqueness of $U(z)$ adds a degree of complexity to the analysis. The purpose of this section is to show that the choice of compensator can reduce or eliminate these problems in practice. The compensation is a series connection of causal and anti-causal elements. More precisely,

1. If $m < \ell$, let $K_1 : \mathscr{V} \to \mathscr{Y}$ and $K_2 : \mathscr{V} \to \mathscr{U}$ be state space, linear systems with $m \times m$, respectively $\ell \times m$, transfer function matrices $K_1(z)$ and $K_2(z)$. The space $\mathscr{V}$ is a space of intermediate signals and is identical to $\mathscr{Y}$ but with Q (in

the associated inner product) replaced by the symmetric positive definite matrix Q_V. Now consider the control update law,

$$u_{k+1} = u_k + \beta K_2 K_1^* e_k \quad so\ that \quad e_{k+1} = (I - \beta G K_2 K_1^*) e_k, \quad k \geq 0. \tag{7.101}$$

This expression has the structure of a control law for the m-input, m-output, composite linear system GK_2 using the non-causal implementation of the $m \times m$ compensator $K = K_1$. Under these conditions, Theorem 7.10 requires that GK_2 is asymptotically stable and that K_1 is asymptotically stable and minimum-phase. It also indicates that monotonic error convergence of the algorithm can then be described precisely by the condition

$$\begin{gathered} r\left(Q^{-1}(I_m - \beta \Gamma^T(z^{-1}))Q(I_m - \beta \Gamma(z))\right) < 1 \\ (where\ \Gamma(z) = G(z)K_2(z)Q_V^{-1}K_1^T(z^{-1})Q) \end{gathered} \tag{7.102}$$

for all z satisfying $|z| = 1$.

2. If $m > \ell$, let $K_1 : \mathscr{U} \to \mathscr{V}$ and $K_2 : \mathscr{Y} \to \mathscr{V}$ be $\ell \times \ell$, respectively $\ell \times m$, linear, state space systems. The space $\mathscr{V}$ is taken to be as the same as $\mathscr{U}$ but with R replaced by R_V.
Consider the control update law,

$$u_{k+1} = u_k + \beta K_1^* K_2 e_k \ so\ that\ e_{k+1} = (I - \beta G K_1^* K_2) e_k, \quad k \geq 0. \tag{7.103}$$

Assuming that

$$ker[G] = \{0\}, \quad ker[K_1] = \{0\} \quad and \quad \mathscr{Y} = \mathscr{R}[G] \oplus ker[K_2], \tag{7.104}$$

define subsidiary variables w_k using the representation $e_k = GK_1^* w_k + \xi_k$, $\xi_k \in ker[K_2]$, $k \geq 0$. This gives the evolution

$$w_{k+1} = (I - K_2 G K_1^*) w_k, \quad k \geq 0 \quad and \quad \xi_k = \xi_0, k \geq 0. \tag{7.105}$$

Convergence of w_k to zero then guarantees convergence of e_k to $e_\infty = \xi_0$. Replacing convergence of the $\{w_k\}_{k\geq 0}$ by monotonic convergence makes it possible to apply Theorem 7.10 with appropriate changes of notation to produce the convergence requirement

$$\begin{gathered} r\left(R_V^{-1}(I_m - \beta \Gamma^T(z^{-1}))R_V(I_m - \beta \Gamma(z))\right) < 1, \ whenever \quad |z| = 1, \\ where\ now\ \Gamma(z) = K_2(z)G(z)R^{-1}K_1^T(z^{-1})R_V. \end{gathered} \tag{7.106}$$

Note: In this case, the information available about the nature of the convergence of the error is limited as the interpretation of w_k is unclear. Also the nature of the limit $e_\infty = \xi_0$ depends on K_2 and the initial input signal u_0.

7.5.3 Robustness Tests in the Frequency Domain

This section is devoted to developing the form of frequency domain robustness tests for the steepest descent algorithm. It will be assumed that the plant has an asymptotically stable, discrete state space model $S(A, B, C, D)$ and that the matrix D and the left or right multiplicative perturbation U satisfy the conditions

$$rank[D] = \min\{m, \ell\} \quad and \quad ker[U] = \{0\} \ \ (that\ is\ \det[U] \neq 0). \tag{7.107}$$

The supervector representation G then has a left and/or a right inverse. Also,

$$\begin{array}{l} (a)\ ker[GU] = ker[G] = \{0\}\ and\ \mathscr{R}[GU] = \mathscr{R}[G]\ if\ m \geq \ell \\ (b)\ \mathscr{R}[UG] = \mathscr{R}[G] = \mathscr{Y}\ and\ ker[UG] = ker[G] \quad if\ m \leq \ell. \end{array} \tag{7.108}$$

A Preliminary Discussion of the SISO Case: One approach to analyzing robustness of the steepest descent algorithms can be based on the results in the previous section. For example, when $m = \ell = 1$, suppose that K is replaced by G and G by UG. Theorem 7.10 then provides a computational approach to assessing the effects of a known perturbation $U(z)$ on convergence of the steepest descent algorithm. For the SISO case, the conditions reduce to the need for (a) $U(z)$ to be asymptotically stable, (b) for $G(z)$ to be asymptotically stable and minimum-phase and (c) for the following frequency domain inequality to be satisfied

$$\left| \frac{R}{\beta Q} - U(z)|G(z)|^2 \right| < \frac{R}{\beta Q} \quad whenever \quad |z| = 1. \tag{7.109}$$

Taking $Q = R = 1$ for simplicity, an equivalent statement is that the plot, in the complex plane, of the frequency response $U(z)|G(z)|^2$ with $|z| = 1$ lies entirely within the interior of the circle of radius β^{-1} and centre $(\beta^{-1}, 0)$.

This condition again underlines the need for $U(z)$ to be strictly positive real if monotonic convergence is to be retained. It also suggests that the model characteristics influence robustness by modulating the gain characteristics of the modelling error U. This is better seen by comparison with the equivalent condition for the inverse model algorithm, namely,

$$\left| \frac{1}{\beta} - U(z) \right| < \frac{1}{\beta} \quad whenever \quad |z| = 1. \tag{7.110}$$

In the inverse model case, if U contains serious modelling errors such as any parasitic resonances that exist outside of the bandwidth of the system, then, even if the positive real property is present, the magnitude of $U(z)$ may significantly reduce the range of gains β that can be tolerated to small values. This will inevitably limit achievable performance and produce slow convergence rates. In contrast, in the case of the steepest descent algorithm and assuming that $G(z)$ has low-pass characteristics, the presence

of the product $U(z)|G(z)|^2$ in the robustness condition implies that the effects of the high frequency modelling errors will be substantially reduced in magnitude. This, in turn, implies that higher gains can be used and, potentially, improved convergence rates achieved.

The analysis provided above gives considerable insight into the robustness of the steepest descent iteration. The need for a strictly positive real modelling error $U(z)$ is, again, inevitable but the frequency domain conditions provide for substantial variation in the detailed frequency domain characteristics. In the following paragraphs, direct proofs of robustness conditions with fewer assumptions are presented.

The Case of $m \leq \ell$: In this case, the compensator $K = G$, G represents a plant model whilst the actual plant is UG where U is a left multiplicative perturbation. The error evolution for the algorithm is $e_{k+1} = (I - \beta UGG^*)e_k$. Suppose that $\mathscr{R}[UG] = \mathscr{R}[G] = \mathscr{Y}$ and hence that $ker[G^*] = \{0\}$ and $\det[GG^*] \neq 0$. Suppose also that $u_\infty \in \mathscr{U}$ is an input signal that produces the required reference signal exactly. That is, $r = UGu_\infty + d$ and the corresponding tracking error is zero. As $ker[G] = \mathscr{R}[G^*]^\perp$ and hence $\mathscr{U} = ker[G] \oplus \mathscr{R}[G^*]$, it is possible to assume, without loss of generality, that both $u_\infty \in \mathscr{R}[G^*]$ and $u_0 \in \mathscr{R}[G^*]$ and hence that $u_k \in \mathscr{R}[G^*]$ for all $k \geq 0$. Being finite dimensional, $\mathscr{R}[G^*]$ is closed and hence is a Hilbert space in its own right. Let $w_k = u_\infty - u_k \in \mathscr{R}[G^*], k \geq 0$, and note that $e_k = UGw_k$. The update formula $u_{k+1} = u_k + \beta G^* e_k$ then indicates that the evolution of the sequence $\{w_k\}_{k\geq 0}$ in $\mathscr{R}[G^*]$ is governed by the recursion $w_{k+1} = (I - \beta G^* UG)w_k$. Robustness of the convergence of the algorithm, in a general context, would simply require that the sequence $w_k \to w_\infty$ with $w_\infty = 0$. A sufficient condition for this is that the convergence to zero is monotonic. That is, if $\beta > 0$ and $(I - \beta G^* UG)^*(I - \beta G^* UG) < I$ in $\mathscr{R}[G^*]$ which is achieved if

$$\begin{aligned} G^*\left(U + U^* - \beta U^* GG^* U\right) G > 0 \quad on \quad \mathscr{R}[G^*], \\ or, equivalently, \quad U + U^* - \beta U^* GG^* U > 0 \quad on \quad \mathscr{Y} \end{aligned} \tag{7.111}$$

as GG^* is invertible. As U is assumed to be invertible, this is equivalent to

$$\hat{U} + \hat{U}^* - \beta GG^* > 0 \quad on \quad \mathscr{Y}, \tag{7.112}$$

where $\hat{U}$ denotes the inverse of U. Application of Theorem 4.11 then yields

Theorem 7.16 (Robustness Tests when $m \leq \ell$) *Using the notation in the preceding discussion, suppose that $m \leq \ell$, that $G(z)$ is a model of a discrete system which has the transfer function matrix $U(z)G(z)$ where the left multiplicative modelling error $U(z)$ has a minimum-phase state space model $S(A_U, B_U, C_U, D_U)$ with $\det[D_U] \neq 0$. Suppose also that $\mathscr{R}[G] = \mathscr{R}[UG] = \mathscr{Y}$ and that the inner products defined in Theorem 7.4 are used in the input and output Hilbert spaces $\mathscr{U}$ and $\mathscr{Y}$.*
Under these conditions, there exists a unique input $u_\infty \in \mathscr{R}[G^*]$ such that $r = UGu_\infty + d$. Suppose that $w_k = u_\infty - u_k, k \geq 0$. Then a sufficient condition for the monotonic convergence of the subsidiary sequence $\{w_k\}_{k\geq 0}$ to a limit $w_\infty = 0$ (and

hence the convergence of the error sequence $\{e_k\}_{k\geq 0}$ *to* $e_\infty = 0$*) is that* $u_0 \in \mathscr{R}[G^*]$ *and that the following inequality holds for all* z *satisfying* $|z| = 1$*,*

$$\hat{U}^T(z)Q + Q\hat{U}(z^{-1}) > \beta QG(z^{-1})R^{-1}G^T(z)Q. \tag{7.113}$$

More generally, suppose that $u_0 = u_0^{(1)} + u_0^{(2)}$ *with* $u_0^{(1)} \in \mathscr{R}[G^*]$ *and* $u_0^{(2)} \in ker[G]$*. The result still stands in this case but* $w_k \to w_\infty = -u_0^{(2)}$*.*

Proof The proof when $u_0 \in \mathscr{R}[G^*]$ follows from the discussion that precedes the statement and the invertibility of GG^* ensures the existence of a unique value of $u_\infty \in \mathscr{R}[G^*]$. More generally, write $u_k = u_k^{(1)} + u_k^{(2)}$ with $u_k^{(1)} \in \mathscr{R}[G^*]$ and $u_k^{(2)} \in ker[G]$. An examination of the control update rule shows that $u_k^{(2)} = u_0^{(2)}$ and $u_{k+1}^{(1)} = u_k^{(1)} + \beta G^* e_k$ for $k \geq 0$. As a consequence $\lim_{k\to\infty} u_\infty - u_k^{(1)} = 0$ and the result is proved as the subsidiary signals $w_k = u_\infty - u_k \to -u_k^{(2)} = w_\infty$, $k \geq 0$. □

The Case of $m \geq \ell$: In this case, the compensator $K = G$ and the plant is taken to be GU where U is a right multiplicative perturbation. The error evolution for the algorithm is simply $e_{k+1} = (I - \beta GUG^*)e_k$. Note that $ker[G^*] = \mathscr{R}[G]^\perp$, $\mathscr{Y} = ker[G^*] \oplus \mathscr{R}[G]$ and $\det[G^*G] \neq 0$. This indicates that the error evolution can be described by writing e_0 as a component in $ker[G^*]$ and another in $\mathscr{R}[G]$. The component in $ker[G^*]$ is unchanged from iteration to iteration and represents unachievable reference signals. It is therefore necessary only to consider the case of $e_0 \in \mathscr{R}[G]$. Being finite dimensional, $\mathscr{R}[G]$ is closed and hence is a Hilbert space in its own right. In addition, $I - \beta GUG^*$ maps $\mathscr{R}[G]$ into itself. Robustness of the convergence of the algorithm, in a general context would simply require that the sequence $e_k \to e_\infty$ with $e_\infty = 0$. A sufficient condition for this is that the convergence to zero is monotonic. This will be achieved if $(I - \beta GUG^*)^*(I - \beta GUG^*) < I$ in $\mathscr{R}[G]$ which is achieved if

$$\begin{aligned} G\left(U + U^* - \beta U^* G^* GU\right) G^* &> 0 \quad on \quad \mathscr{R}[G], \\ or, equivalently, \quad U + U^* - \beta U^* G^* GU &> 0 \quad on \quad \mathscr{U}. \end{aligned} \tag{7.114}$$

as G^*G is invertible. Application of Theorem 4.9 then yields the robustness test

Theorem 7.17 (Robustness Tests when $m \geq \ell$) *Using the notation in the preceding discussion, suppose that* $m \geq \ell$*, that* $G(z)$ *is a model of a discrete system with transfer function matrix* $G(z)U(z)$ *where the right multiplicative modelling error* $U(z)$ *is asymptotically stable. Suppose also that* $ker[G] = ker[UG] = \{0\}$ *and that the topologies defined in Theorem 7.4 are used in the input and output spaces* $\mathscr{U}$ *and* $\mathscr{Y}$*.*

Under these conditions, a sufficient condition for the monotonic convergence of the error sequence $\{e_k\}_{k\geq 0}$ *to a limit* $e_\infty = 0$ *is that* $e_0 \in \mathscr{R}[G]$ *and that the following inequality holds for all* z *satisfying* $|z| = 1$*,*

$$RU(z) + U^T(z^{-1})R > \beta U^T(z^{-1})G^T(z^{-1})QG(z)U(z). \tag{7.115}$$

More generally, if $e_0 = e_0^{(1)} + e_0^{(2)}$ *with* $e_0^{(1)} \in \mathscr{R}[G]$ *and* $e_0^{(2)} \in ker[G^*]$*, then the inequality ensures that* $\lim_{k\to\infty} e_k = e_\infty = e_0^{(2)}$.

A Comment on the Structural Form and Assumptions: It is of interest to note that the robustness results require either that $U(z)$ is minimum-phase if $m \geq \ell$ but that it is asymptotically stable if $m \geq \ell$. Pre- and post multiplication will give a number of alternative but equivalent descriptions. This can be illustrated by the case of $m \geq \ell$ and right multiplicative perturbations when the relevant conditions in Theorem 7.17 are replaced by requiring that $U(z)$ is minimum-phase and that, for all z satisfying $|z| = 1$,

$$\hat{U}^T(z^{-1})R + R\hat{U}(z) > \beta G^T(z^{-1})QG(z). \tag{7.116}$$

This illustrates the need for the uncertainty to satisfy some form of positivity condition. This essentially constrains the "phase" of $\hat{U}(z)$ (and hence $U(z)$) but the positivity also limits the magnitude of $U(z)$. This is most easily seen in the case of SISO systems when the condition reduces to

$$Re\left[\hat{U}(z)\right] > \frac{\beta}{2}\left(\frac{Q}{R}\right)|G(z)|^2, \quad \textit{whenever} \quad |z| = 1. \tag{7.117}$$

This formula provides a precise, numerical expression of the phase and gain constraints and indicates the practical point that both Q and R can be set equal to unity as their effects are equivalent to a change in the learning gain β.

7.5.4 Robustness and Relaxation

The discussion in the previous section applies to the steepest descent algorithm without relaxation. Motivated by its role in reducing the spectral radius of the operator in the error evolution, Sect. 5.2.3 suggested, without proof, that relaxation will improve the robustness of Iterative Control. For steepest descent algorithms, the relaxed input update rule has the form,

$$u_{k+1} = \alpha u_k + \beta G^* e_k, \quad k \geq 0, \tag{7.118}$$

where G is a model of the plant. The effect of the relaxation parameter $0 < \alpha < 1$ can be assessed using similar techniques to those used in Sect. 7.5.3.

The Case of $m \leq \ell$: Suppose that the plant can be described by UG, where U is a left multiplicative perturbation, and that $\mathscr{R}[UG] = \mathscr{R}[G] = \mathscr{Y}$ so that $ker[G^*] = \{0\}$. The error evolution is described by

$$e_{k+1} = (\alpha I - \beta UGG^*)e_k + (1-\alpha)(r-d), \quad k \geq 0. \tag{7.119}$$

Define $w_k = G^* e_k, k \geq 0$ noting that convergence of $\{w_k\}_{k\geq 0}$ implies the convergence of $\{e_k\}_{k\geq 0}$. A sufficient condition for convergence of $\{w_k\}_{k\geq 0}$ to a limit error is that $\beta > 0$ and $(\alpha I - \beta G^* UG)^*(\alpha I - \beta G^* UG) < I$ in $\mathscr{R}[G^*]$ which is achieved if

$$\begin{aligned} &G^* \left(\alpha\beta(U + U^*) - \beta^2 U^* GG^* U\right) G > -(1-\alpha^2) I \text{ on } \mathscr{R}[G^*], \\ &\text{or, on } \mathscr{Y},\ U + U^* - \alpha^{-1}\beta U^* GG^* U > -(\alpha\beta)^{-1}(1-\alpha^2)(GG^*)^{-1}, \end{aligned} \tag{7.120}$$

as GG^* is invertible. The fact that the right-hand-side is strictly negative definite when $\alpha < 1$ removes the need for U to satisfy a positivity condition. The range of model errors that can be tolerated has therefore increased.

The Case of $m \geq \ell$: A similar argument leads to a very similar outcome in terms of increased robustness. More precisely, using the plant description GU where U is a right multiplicative modelling error, the error evolution is described by

$$e_{k+1} = (\alpha I - \beta GUG^*) e_k + (1-\alpha)(r-d), \quad k \geq 0, \tag{7.121}$$

A sufficient condition for convergence to a limit error is that $\beta > 0$ and $(\alpha I - \beta GUG^*)^*(\alpha I - \beta GUG^*) < I$ in $\mathscr{R}[G]$ which is achieved if

$$\begin{aligned} &G \left(\alpha\beta(U + U^*) - \beta^2 U^* G^* GU\right) G^* > -(1-\alpha^2) I \quad \text{on} \quad \mathscr{R}[G], \\ &\text{or,} \quad U + U^* - \alpha^{-1}\beta U^* G^* GU > -(\alpha\beta)^{-1}(1-\alpha^2)(G^*G)^{-1} \quad \text{on} \quad \mathscr{Y}, \end{aligned} \tag{7.122}$$

as G^*G is invertible. The fact that the right-hand-side is strictly negative definite when $\alpha < 1$ removes the need for U to satisfy a positivity condition. The range of model errors that can be tolerated has therefore increased.

7.5.5 Non-monotonic Gradient-Based Control and ε-Weighted Norms

The ideas described in previous sections are now extended to include convergence and robustness by a structured change to the idea of monotonicity. Care has to be taken however if the very useful link to the frequency domain is to be retained. A way forward is to note that the inner product induced topology used in the Theorems of the previous sections is a special case (by choosing $\varepsilon = 1$) of ε-weighted inner products used for inverse algorithms in Sect. 6.2.3. More precisely, let $\varepsilon \in (0, 1]$ and define $\mathscr{Y}_\varepsilon = \mathscr{R}^{m(N+1)}$ and $\mathscr{U}_\varepsilon = \mathscr{R}^{\ell(N+1)}$ to be Hilbert spaces (of supervectors) with inner products

$$\langle y, z\rangle_{\mathscr{Y}_\varepsilon} = \sum_{t=0}^{N} \varepsilon^{2t} y^T(t) Q z(t) \quad \textit{and} \quad \langle u, v\rangle_{\mathscr{U}_\varepsilon} = \sum_{t=0}^{N} \varepsilon^{2t} u^T(t) R v(t). \tag{7.123}$$

The cases of $\varepsilon = 1$ and any $\varepsilon \in (0, 1]$ can be connected for any supervector f associated with the time series $\{f(t)\}_{0\le t\le N}$ by defining a weighted time series $\{\varepsilon^t f(t)\}_{0\le t\le N}$ and constructing the associated supervector

$$f^\varepsilon = \left[f^T(0), \varepsilon f^T(1), \varepsilon^2 f^T(2), \ldots, \varepsilon^N f^T(N)\right]^T. \tag{7.124}$$

Working with weighted signals $u^\varepsilon, y^\varepsilon, r^\varepsilon, e^\varepsilon, x^\varepsilon$ rather than the original vectors u, y, r, e, x provides a simple approach to introducing and analyzing different convergence behaviours. One important aspect of this is that the inner products (and hence the norms) are related by

$$\begin{array}{ll} (a) & \langle y, z\rangle_{\mathscr{Y}_\varepsilon} = \langle y^\varepsilon, z^\varepsilon\rangle_{\mathscr{Y}} \ \textit{and} \ \ \langle u, v\rangle_{\mathscr{U}_\varepsilon} = \langle u^\varepsilon, v^\varepsilon\rangle_{\mathscr{U}}, \quad \textit{whilst} \\ (b) & \|y\|_{\mathscr{Y}_\varepsilon} = \|y^\varepsilon\|_{\mathscr{Y}} \quad \textit{and} \quad \|u\|_{\mathscr{U}_\varepsilon} = \|u^\varepsilon\|_{\mathscr{U}} \end{array} \tag{7.125}$$

Convergence properties with respect to these norms are related also as, for example, if $0 < \varepsilon_2 \le \varepsilon_1 \le 1$, then it is easily shown that, for all $e \in \mathscr{Y}$,

$$\left(\frac{\varepsilon_2}{\varepsilon_1}\right)^N \|e\|_{\mathscr{Y}_{\varepsilon_1}} \le \|e\|_{\mathscr{Y}_{\varepsilon_2}} \le \|e\|_{\mathscr{Y}_{\varepsilon_1}} \le \|e\|_{\mathscr{Y}} \le \varepsilon_1^{-N} \|e\|_{\mathscr{Y}_{\varepsilon_1}}, \tag{7.126}$$

and hence convergence in one topology guarantees convergence in the other. Specifically in terms of monotonic convergence,

1. the monotonic convergence of the sequence $\{e_k\}_{k\ge 0}$ in $\mathscr{Y}_{\varepsilon_1}$ implies the convergence of that sequence in $\mathscr{Y}_{\varepsilon_2}$.
2. The converse statement is that the monotonic convergence of the sequence $\{e_k\}_{k\ge 0}$ in $\mathscr{Y}_{\varepsilon_2}$ implies the convergence of the sequence in $\mathscr{Y}_{\varepsilon_1}$.

In particular, if $\varepsilon_1 = 1$, then convergence is guaranteed if monotonic convergence is achieved for the weighted signal with any choice of $0 < \varepsilon < 1$. The convergence in $\mathscr{Y} = \mathscr{Y}_1$ need not, however, be monotonic.

A simple calculation easily reveals that the state space model describing the evolution of the weighted signals $u^\varepsilon, x^\varepsilon, y^\varepsilon$ is obtained by the data map

$$\begin{aligned} S(A, B, C, D) &\mapsto S(\varepsilon A, \varepsilon B, C, D) \qquad \textit{and} \\ G(z) &\mapsto G_\varepsilon(z) = C(zI - \varepsilon A)^{-1}\varepsilon B + D = G(\varepsilon^{-1} z). \end{aligned} \tag{7.127}$$

The initial condition $x^\varepsilon(0) = x(0)$ remains the same and the values of $G_\varepsilon(z)$ for $|z| = 1$ are generated from the values of $G(z)$ when $|z| = \varepsilon^{-1}$.

The matrices in the supervector descriptions of $S(A, B, C, D)$ and $S(\varepsilon A, \varepsilon B, C, D)$ will be denoted by G and G_ε respectively. A simple calculation indicates

that the Markov parameter matrices CA^jB are replaced by $\varepsilon^{j+1}CA^jB$ for $j \geq 0$ and hence that

$$G_\varepsilon = \Theta_\varepsilon(m, N)G\Theta_\varepsilon^{-1}(\ell, N) \tag{7.128}$$

where, for example,

$$\Theta_\varepsilon(m, N) = \begin{bmatrix} I_m & 0 & 0 \cdots & 0 \\ 0 & \varepsilon I_m & 0 \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & 0 \cdots & \varepsilon^N I_m \end{bmatrix} \tag{7.129}$$

In what follows, the simplified symbol Θ_ε will be used with the required dimensions being understood from the context.

The poles and zeros of $G_\varepsilon(z)$ are those of $G(z)$ scaled by a factor of ε. As a consequence, the asymptotic stability of $S(A, B, C, D)$ then implies that of $S(\varepsilon A, \varepsilon B, C, D)$ if $\varepsilon \leq 1$. More generally,

1. The modification leaves the system's relative degree(s) unchanged.
2. If $S(A, B, C, D)$ is unstable, then $S(\varepsilon A, \varepsilon B, C, D)$ is stable in some interval $\varepsilon \in (0, \varepsilon_{as})$.
3. If $S(A, B, C, D)$ is non-minimum phase, then $S(\varepsilon A, \varepsilon B, C, D)$ is minimum-phase in some interval $\varepsilon \in (0, \varepsilon_{mp})$.
4. If $S(A, B, C, D)$ is both unstable and non-minimum-phase then $S(\varepsilon A, \varepsilon B, C, D)$ is asymptotically stable and minimum-phase whenever $\varepsilon \in (0, \min\{\varepsilon_{as}, \varepsilon_{mp}\})$.
5. If $\hat{G}(z)$ is an inverse of $G(z)$, then $\hat{G}_\varepsilon(z) = \hat{G}(\varepsilon^{-1}z)$ is an inverse of $G_\varepsilon(z)$.
6. As a consequence, assumptions of asymptotic stability, minimum-phase characteristics or the existence of stable inverses can be achieved by using the ε-weighted inner products and norms and examining the iterative control problem for $S(\varepsilon A, \varepsilon B, C, D)$ with ε chosen to ensure that the desired conditions are satisfied. Remember however that monotonicity with respect to an ε-weighted norm may not guarantee acceptable convergence properties with respect to the unweighted norm.
7. If $G_\varepsilon : \mathscr{U} \to \mathscr{Y}$ is the matrix in the supervector description of $S(\varepsilon A, \varepsilon B, C, D)$, then its adjoint is the familiar product $\mathscr{T}(\ell, N)G_E\mathscr{T}(m, N)$. Here $\mathscr{T}$ denotes the appropriate discrete time reversal operator (Chap. 4) and G_E is the matrix in the supervector description of $S(\varepsilon A^T, C^TQ, \varepsilon R^{-1}B^T, R^{-1}D^TQ)$ which is identical to that of $S(\varepsilon A^T, \varepsilon C^TQ, R^{-1}B^T, R^{-1}D^TQ)$. Note that

$$G_\varepsilon^* = \Theta_\varepsilon^{-1}G^*\Theta_\varepsilon, \quad \textit{and hence}, \quad G_{\varepsilon^2}^* = \Theta_\varepsilon^{-1}G_\varepsilon^*\Theta_\varepsilon. \tag{7.130}$$

Now consider the ε-weighted, ℓ-input, m-output model $S(\varepsilon A, \varepsilon B, C, D)$ of the system $S(A, B, C, D)$ with supervector matrix $G_\varepsilon : \mathscr{U} \to \mathscr{Y}$. Let $K(z)$ be a compensator with state space model $S(A_K, B_K, C_K, D_K)$ and transfer function matrix $K(z)$. The equivalent ε-weighted model is $S(\varepsilon A_K, \varepsilon B_K, C_K, D_K)$. The resultant matrix in the

ε-weighted supervector description is $K_\varepsilon : \mathscr{U} \to \mathscr{Y}$. The general gradient algorithm for this description updates ε-weighted inputs base on ε-weighted errors as follows

$$u^\varepsilon_{k+1} = u^\varepsilon_k + \beta K^*_\varepsilon e^\varepsilon_k \tag{7.131}$$

where $K^*_\varepsilon : \mathscr{Y} \to \mathscr{U}$ is the adjoint. That is, the signal $\psi^\varepsilon_k(t)$ defined by the supervector $\psi^\varepsilon_k = K^*_\varepsilon e^\varepsilon_k$ is generated from the equations

$$\begin{aligned} z^\varepsilon_k(t+1) &= \varepsilon A^T z^\varepsilon_k(t) + \varepsilon C^T Q e^\varepsilon_k(N-t), \quad z^\varepsilon_k(0) = 0, \quad \textit{and} \\ \psi^\varepsilon_k(t) &= R^{-1}B^T z^\varepsilon_k(N-t) + R^{-1}D^T Q e^\varepsilon_k(t), \quad \textit{for } 0 \le t \le N. \end{aligned} \tag{7.132}$$

The update relationship for weighted errors in $\mathscr{Y}$ is just

$$e^\varepsilon_{k+1} = (I - \beta G_\varepsilon K^*_\varepsilon) e^\varepsilon_k, \quad \textit{for } k \ge 0. \tag{7.133}$$

It follows that the convergence and robustness analysis of previous sections applies here with G, K, U etc. replaced by G_ε, K_ε, U_ε etc. and their associated transfer function matrices. In particular, the frequency domain conditions previously derived for convergence and robustness are of the form of inequalities involving z and z^{-1} that must hold on the unit circle $\{z : |z| = 1\}$. If the transfer function matrices $G_\varepsilon(z)$ etc. are replaced by their original (un-weighted) descriptions $G(\varepsilon^{-1}z)$ etc., and subsequently by $G(z)$ etc. the conditions take a simple form involving values at z and $\bar{z}$ with $|z| = \varepsilon^{-1}$. To illustrate the change, note that Theorem 7.16 describing one of the robustness conditions for the steepest descent algorithm can be extended to the case of weighted norms as follows,

Theorem 7.18 (Robustness and ε-norms when $m \le \ell$) *Under the assumptions of Theorem 7.16, let $\varepsilon \in (0, 1]$ be such that $G_\varepsilon(z) = G(\varepsilon^{-1}z)$ is asymptotically stable and $U_\varepsilon(z) = U(\varepsilon^{-1}z)$ is minimum-phase. Suppose that the chosen input update rule is now expressed in the form*

$$u^\varepsilon_{k+1} = u^\varepsilon_k + \beta G^*_\varepsilon e^\varepsilon_k. \tag{7.134}$$

Then, a sufficient condition for the monotonic convergence (with respect to the ε-weighted norm in $\mathscr{U}_\varepsilon$) of the subsidiary sequence $\{w^\varepsilon_k\}_{k\ge 0}$ to a limit $w^\varepsilon_\infty = 0$ (and hence the convergence of the error sequence $\{e_k\}_{k\ge 0}$ to $e_\infty = 0$) is that,

$$\hat{U}^T(z)Q + Q\hat{U}(\bar{z}) > \beta Q G(\bar{z}) R^{-1} G^T(z) Q \quad \textit{whenever } |z| = \varepsilon^{-1}. \tag{7.135}$$

The result is based on using weighted supervectors. At first sight, it extends the range of modelling errors that can be considered and, by using small values of ε, can make it possible, in principle, to remove assumptions requiring the stability of G and minimum-phase properties of U. The price that is paid for this, more general,

approach is that monotonicity of the ε-weighted norm sequence in $\mathscr{U}_\varepsilon$ may allow large, and possibly very large, increases in norms in $\mathscr{U}$ as, at best,

$$\|w\|_{\mathscr{U}} \leq \varepsilon^{-N}\|w\|_{\mathscr{U}_\varepsilon} \quad for \ \ all\, w \in \mathscr{U}. \tag{7.136}$$

Monotonicity for the weighted norm sequence hence could allow substantial "up and down" variations of the un-weighted norm sequence. As stated therefore, the result may only have practical value for cases where $U(z)$ is minimum-phase.

Proof of Theorem 7.18 The evolution of the error sequence $\{e_k\}_{k\geq 0}$ in $\mathscr{Y}_\varepsilon$ is described by the supervector recursion $e^\varepsilon_{k+1} = (I - \beta U_\varepsilon G_\varepsilon G^*_\varepsilon)e^\varepsilon_k$ in $\mathscr{Y}$. From previous results, a sufficient condition for monotonicity of the weighted norms and convergence of the subsidiary sequence $\|w^\varepsilon_k\|_{\mathscr{U}}$ is that, whenever $|z| = 1$,

$$\hat{U}^T_\varepsilon(z)Q + Q\hat{U}_\varepsilon(z^{-1}) > \beta Q G_\varepsilon(z^{-1})R^{-1}G^T_\varepsilon(z)Q. \tag{7.137}$$

which is just

$$\hat{U}^T(\varepsilon^{-1}z)Q + Q\hat{U}(\varepsilon^{-1}z^{-1}) > \beta Q G(\varepsilon^{-1}z^{-1})R^{-1}G^T(\varepsilon^{-1}z)Q. \tag{7.138}$$

The result follows by replacing $\varepsilon^{-1}z$ by z and noting that $\overline{\varepsilon^{-1}z} = \varepsilon^{-1}z^{-1}$. □

In general, the consequences of the use of ε-weighted norms are that

1. Monotonic convergence and robustness can be analysed (using the results of this chapter) for some choice of $\varepsilon \in (0, 1]$ using weighted signals and models.
2. It is always possible to satisfy any necessary theoretical assumptions of stability and/or minimum-phase properties using a suitable choice of $\varepsilon > 0$. This observation release the results of this chapter for application to almost all discrete systems by focussing attention on monotonicity of ε-weighted norms.
3. For implementation purposes, the update relationship (7.131) for the weighted input can be expressed in terms of an update relationship for the un-weighted input by replacing

$$u^\varepsilon_{k+1} = u^\varepsilon_k + \beta K^*_\varepsilon e^\varepsilon_k \quad by \quad u_{k+1} = u_k + \beta K^*_{\varepsilon^2} e_k, \tag{7.139}$$

where $K^*_{\varepsilon^2} : \mathscr{Y} \to \mathscr{U}$ is the adjoint of the operator $K_{\varepsilon^2} : \mathscr{U} \to \mathscr{Y}$ corresponding to the state space model

$$S(\varepsilon^2 A_K, \varepsilon^2 B_K, C_K, D_K). \tag{7.140}$$

The proof of this statement follows using Θ_ε and the connection, $f^\varepsilon = \Theta_\varepsilon f$, between a supervector f and its weighted form f^ε and the relationship $K^*_{\varepsilon^2} = \Theta^{-1}_\varepsilon K^*_\varepsilon \Theta_\varepsilon$. The details are left as an exercise for the reader.
4. If $\varepsilon \to 0+$, then continuity of the eigenvalues of $GK^*_{\varepsilon^2}$ at $\varepsilon = 0$ implies that convergence is always achieved for small enough values of $\varepsilon > 0$ if the $m \times m$ matrix $I_m - \beta D R^{-1} D^T_K Q$ has eigenvalues in the unit circle only.

7.5.6 A Steepest Descent Algorithm Using ε-Norms

The use of ε-weighted norms also open up the possibility of new algorithms. For example, consider the steepest descent algorithm for $S(A, B, C, D)$ in the absence of any modelling error. Choosing the gradient compensator $K = G_{\varepsilon^2}$ with state space model $S(\varepsilon^2 A, \varepsilon^2 B, C, D)$ suggests the algorithm

Algorithm 7.11 (*Steepest Descent with ε-Weighting*) Suppose that $\varepsilon \in (0, 1]$. Then the modified steepest descent algorithm

$$u_{k+1} = u_k + \beta G^*_{\varepsilon^2} e_k, \quad for\, k \geq 0 \tag{7.141}$$

has an equivalent ε-weighted description in $\mathscr{U}$ and $\mathscr{Y}$ of the form

$$u^\varepsilon_{k+1} = u^\varepsilon_k + \beta G^*_\varepsilon e^\varepsilon_k, \quad with \quad e^\varepsilon_{k+1} = (I - \beta G_\varepsilon G^*_\varepsilon) e^\varepsilon_k \, k \geq 0. \tag{7.142}$$

Then, Theorem 7.1 and related results indicate that there are now two possibilities describing convergence in the absence of modelling errors.

1. Suppose that $m \leq \ell$ and that $\mathscr{R}[G] = \mathscr{Y}$. Then, $\mathscr{R}[G] = \mathscr{R}[G_\varepsilon]$ and the algorithm generates a monotonically decreasing sequence of ε-weighted error norms $\|e_k\|_{\mathscr{Y}_\varepsilon}$ for all β in the range
$$0 < \beta \|G^*_\varepsilon\|^2 < 2 \tag{7.143}$$
Moreover, the sequence converges to the limit $e_\infty = 0$.
2. Let $m \geq \ell$. Suppose that $ker[G] = \{0\}$ and write $e_0 = Gw_0 + \xi_0$ with $\xi_0 \in ker[G^*]$. Then $e_k = Gw_k + \xi_0$ where $w_{k+1} = (I - \beta G^*_{\varepsilon^2} G) w_k$, $k \geq 0$. As a consequence $w^\varepsilon_{k+1} = (I - \beta G^*_\varepsilon G_\varepsilon) w^\varepsilon_k$. The algorithm therefore generates a monotonically decreasing sequence of ε-weighted norms $\|w^\varepsilon_k\|_{\mathscr{U}} = \|w_k\|_{\mathscr{U}_\varepsilon}$ for all β in the range $0 < \beta \|G_\varepsilon\|^2 < 2$ so that

$$\lim_{k \to \infty} w_k = 0 \quad and, \, also, \quad \lim_{k \to \infty} e_k = \xi_0. \tag{7.144}$$

The bounding of the norms $\|G_\varepsilon\|$ and $\|G^*_\varepsilon\|$ of $G_\varepsilon : \mathscr{U} \to \mathscr{Y}$ and its adjoint $G^*_\varepsilon : \mathscr{Y} \to \mathscr{U}$ can be done using frequency domain properties of the state space model $S(\varepsilon A, \varepsilon B, C, D)$ and its associated transfer function matrix $G_\varepsilon(z) = G(\varepsilon^{-1} z)$.

7.6 Discussion, Comments and Further Generalizations

Classically, control systems design has been underpinned by a number of principles that guide the design engineer and offer choices that benefit both the design process and the performance of the control system that results from their application. The ideas of gain and phase compensation increase the prospect of obtaining greater

performance using relatively simple control elements including, for example, phase-lead (or lead-lag) networks. In this text the ideas of feedforward and feedback iterative control have been introduced but, within these principles, there are many options yet to be explored.

7.6.1 Bringing the Ideas Together?

Guided by an assumption that monotonic reduction of quadratic measures (norms) of the tracking error is of practical benefit in both the design process and achievable performance of Iterative Control systems, the analysis has shown the great benefits inherent in the use of compensators that are adjoints K^* of operators K of linear dynamical systems. For linear, continuous time or discrete time, state space systems, input update computations are done off-line, between iterations, and take the form of reverse time simulations from zero terminal conditions. Such computations have been associated with $R^{-1}K^T(z^{-1})Q$. As a consequence, they have a natural "phase lead" structure that can be applied to create the positivity conditions on GK^* necessary for monotonic convergence of iterations of the form $e_{k+1} = Le_k$ where $L = (I - \beta GK^*)$. The case of $m = \ell$ is the simplest case. The cases of $m > \ell$ and $m \leq \ell$ differ in the introduction of subsidiary signals related to input signals. The reader should take note of the difference in convergence properties that may result.

The use of steepest descent is simply to set K to be a model G of the plant. In this case, L is self adjoint if there is no modelling error. Preliminary results in this area were derived by the author and co-workers in [94] and, adding a flavour of inverse algorithm, in [85]. Related ideas are presented in [115].

The approach presented in this text generalizes those results substantially. The most general convergence theorem is Theorem 7.1 which considers quite general plant operators including, for typical applications, discrete and continuous state space systems. Note however that, as G need only be linear and bounded, other possibilities include the application to multi-rate, discrete, sample-data systems. For applications where the input generating the desired output $y = r$ is non unique, Theorem 7.2 relates the converged solution to the optimal control concepts of minimum energy/minimum norm inputs. Despite the power of this result, there is always the need to consider the details. The issues that have a direct impact on practical applications are those of realizing the update relation $u_{k+1} = u_k + \beta G^* e_k$ as a reasonable computational procedure, identifying the role of parameters in the performance of the algorithms and the nature of the limit and, finally, characterizing the effect of modelling errors on the convergence that might be expected in practice.

The main parameters in steepest descent algorithms are the "gain" β and the weighting matrices Q and R (which define the topologies in output and input spaces but also arise as parameters in the adjoint operator). The matrix Q may be specified by the designer for reasons unrelated to the performance of the algorithm or to reflect the relative importance of the accuracies required in each control loop. In contrast, the matrix R^{-1} either defines an "energy function" (minimized in the limit)

or performs the role of a "gain matrix" that gives the engineer some control over relative magnitudes of control input changes in each control loop. Finally, β is the tuning parameter and is available as a final influence on convergence rates with a suitable range defined by an H_∞-norm or equivalent spectral radius condition. Monotonicity does not define the *rate* of convergence which depends very much on controller and plant structures and, indeed, the reference signal. Insight into the nature of the convergence has been obtained using the eigenstructure of GG^*. In principle, it offers the possibility of convergence to zero error in a finite number of iterations but a lack of information about the eigenvalues limits the practical use of this idea. It does however suggest that systematic variations in β from iteration to iteration may improve convergence properties (an idea linked to that of Parameter Optimal Iterative Learning Control as introduced in Chap. 14). Eigenstructure analysis also shows that the reference signal itself will influence convergence rates with slow convergence being expected if it is dominated by eigenvectors related to very small eigenvalues. These ideas are intuitively linked to the use of frequency domain analysis using Fourier series representations of r on an interval $[0, T]$.

A more familiar description of algorithm properties comes when the robustness of the steepest descent approach is considered. Conditions for monotonic convergence in the presence of modelling errors take the form of inequalities relating the necessary properties of the modelling error $U(z)$ and the plant transfer function matrix $G(z)$. In general, there does not appear to be a graphical interpretation of the condition except in the case of single-input, single-output systems when the need for positivity properties and magnitude constraints are expressed in terms of the behaviour of the Nyquist plot of $U(z)|G(z)|^2$ in the complex plane. This condition suggests the possibility for the development of frequency domain design techniques specifically for gradient-based Iterative Control. This idea is supported by the section on more general gradients where frequency domain conditions suggest that anti-causal compensators can provide essential phase lead and positivity properties. Further research is needed to make this work easier to apply in practice. For example, regarding K as a simplified model of G, how can a suitable choice be made and what is the simplest such choice? Part of the answer to this question can be deduced from the section on robustness of steepest descent algorithms using a change in notation. More precisely, if the compensator K is a simplified model G of the actual plant model and the actual plant model is the simplified model G modified by a multiplicative modelling error, U, the robustness theorems provide a formal relationship between G and U.

Given the practical uncertainty in the choice of Q and R, it can be argued that monotonicity could usefully be relaxed to allow some increase in norm provided that it is limited in magnitude. There is no general analysis available for what might be seen by some readers as an ill-defined concept. However, the use of ε-weighted norms provides a useful special case by transferring the monotonicity requirement to a simple norm that weights the earlier parts of the time interval more than later parts using a geometric weighting ε^t with $0 < \varepsilon \leq 1$. This changes the frequency domain analysis by replacing the unit circle by the circle $|z| = \varepsilon^{-1}$. The results extend (a) the range of compensators that guarantee convergence and/or (b) the range of uncertainties that can be tolerated. An important byproduct of the idea is

that, with a suitable choice of ε, the model used to describe weighted inputs, outputs and states will be both "minimum-phase" and "asymptotically stable". As many of the theoretical results depend on pole and zero positions, the technique provides a mechanism for transforming a system $S(A, B, C, D)$ into a form $S(\varepsilon A, \varepsilon B, C, D)$ where they can be applied. The reader should note however that, if ε is small, the monotonicity of the weighted norm may allow large increases in un-weighted norm before convergence is finally achieved. A simple approach to choice of an acceptable range of ε could be based on examination of the highest power ε^N. This quantity describes the weights at the final time relative to the weighting given to $t = 0$. If this relative weight is defined by the user to be R_W, then ε can be computed, in principle, from the formula $N^{-1} \log_e R_W = \log_e \varepsilon$.

7.6.2 Factors Influencing Achievable Performance

The dynamical structure of the plant $S(A, B, C, D)$ and the choice of Q and R will influence achievable performance through pole and zero positions, the frequency domain properties of matrices such as $G(z)R^{-1}K^T(z^{-1})Q$ and its interaction structure. From the derived frequency domain conditions, the presence of resonances may both limit the gains β that can be used even in the absence of modelling errors whilst the presence of zeros close to the stability boundary will guarantee slow convergence of frequency components in the reference that lie close to those zeros. Non-minimum-phase zeros will also limit performance. This issue will be addressed in more detail later in the text where eigenvalue analysis links the property to a tendency of the signal norms to "plateau" (sometimes called "flat-lining"). This is a situation where monotonicity is retained but, after an initial period of faster convergence, the algorithm may converge infinitesimally slowly. This final convergence is so slow that, for all practical purposes, no further improvement in tracking accuracy is achievable.

Although many of the issues associated with non-minimum-phase properties can be eliminated from the mathematics by the use of ε-weighted signals and norms, the nature of the monotonicity also changes and this approach is not the dominant design issue. The real choices are those of ensuring that the plant itself has acceptable properties and of choosing $K(z)$ to meet design requirements for convergence and robustness. In Chap. 8, a generic design approach is suggested that provides a systematic way of structuring the decision-making process. Not every aspect of what will be suggested will be relevant or possible in any particular application but the aim is to open "design doors" by providing details of possibilities for immediate use and further development.

7.6.3 Notes on Continuous State Space Systems

The form of the algorithm has been defined for both continuous time and discrete time systems. The analysis has, however, focussed mainly on the case of discrete time systems as the methodology has a strong link to the ideas of matrix algebra. The analysis of continuous time systems has a similar structure with properties covered by the general Theorems 7.1 and 7.2 but full details have been omitted from the text for reasons of available space. The generalization to continuous time state space systems $S(A, B, C, D)$ is relatively straightforward as its adjoint is easily computed and typically takes the form $S(A^T, C^T Q, R^{-1}B^T, R^{-1}D^T Q)$ with zero terminal condition at the final time T. For cases where D loses rank, the range of G, at best, is dense in $\mathscr{Y}$ but, as continuous systems are effectively discrete systems with an "infinitely high" sampling rate, the essential structure of the theory will be very similar.

Finally, readers may wish to consider the extension of the ideas of weighted norms to continuous time, state space systems, with inner products in $\mathscr{Y}_\varepsilon = L_2^m[0, T]$ and $\mathscr{U}_\varepsilon = L_2^\ell[0, T]$,

$$\langle y, w\rangle_{\mathscr{Y}_\varepsilon} = \int_0^T e^{-2\varepsilon t} y^T(t) Q w(t) dt \text{ and } \langle u, v\rangle_{\mathscr{U}_\varepsilon} = \int_0^T e^{-2\varepsilon t} u^T(t) R v(t) dt \tag{7.145}$$

with the choice of $\varepsilon > 0$ clearly indicating a higher weighting on the early parts of the interval $[0, T]$. The weighted signals $r^\varepsilon, y^\varepsilon, e^\varepsilon, u^\varepsilon, x^\varepsilon, \ldots$ are computed using the generic "exponentially weighted" map $f(t) \mapsto f^\varepsilon(t) = e^{-\varepsilon t} f(t)$. The continuous system $S(A, B, C, D)$ mapping u into y then becomes the systems $S(A - \varepsilon I, B, C, D)$ that maps u^ε into y^ε. As the transfer function matrix changes from $G(s)$ to $G_\varepsilon(s) = G(s + \varepsilon)$, this model becomes both asymptotically stable and minimum -phase if ε is large enough. In addition,

1. in frequency domain analysis, the imaginary axis must be replaced by the vertical line $\{s : Re[s] = \varepsilon\}$.
2. The weighted input update relationship is defined by

$$u_{k+1}^\varepsilon(t) = u_k^\varepsilon(t) + \beta G_\varepsilon^* e_k^\varepsilon(t), \quad for \quad t \in [0, T], \tag{7.146}$$

where $G_\varepsilon^* : \mathscr{Y} \to \mathscr{U}$ is the adjoint operator for the system $S(A - \varepsilon I, B, C, D)$. The un-weighted signal update relationship is then easily derived to be

$$u_{k+1}(t) = u_k(t) + \beta G_{2\varepsilon}^* e_k(t), \quad for \quad t \in [0, T], \tag{7.147}$$

where $G_{2\varepsilon}^*$ is the adjoint operator for the system $S(A - 2\varepsilon I, B, C, D)$.

The reader may wish to explore previously defined uses of "exponentially weighted norms" in nonlinear Iterative Control by examination of publications such as [113, 17, 25, 78] and [114] and the references therein.

Chapter 8
Combined Inverse and Gradient Based Design

Preceding chapters have shown the theoretical power of inverse model-based and gradient-based Iterative Control algorithms for linear systems using an operator formulation that highlights the great generality of the concepts. The general theory inevitably needs more detail to be added depending on the form of plant model to be used. Linear, time invariant state space models (and their transfer function matrix equivalents) formed the focus of the discussion with emphasis on the case of discrete/sampled data systems as the finite dimensional nature of the resultant problem simplified the presentation considerably. The purpose of this chapter is to explore the links between inverse model and gradient approaches and bring them together in the form of a structured design approach. It will not be claimed that the presentation answers all the questions posed by the wide range of applications but it is hoped that the issues described and answers considered will help the reader in the process of tailoring the ideas to meet his or her needs.

The benefits of bringing inverse model and gradient concepts together is first described by a section on filtering to improve the robustness of inverse model algorithms. This is followed by a summary of issues that may be faced in design and a range of compensation strategies.

8.1 Inverse Algorithms: Robustness and Bi-directional Filtering

The purpose of this section is use gradient concepts to improve the robustness of the inverse model algorithms discussed in Chap. 6 using bi-directional ("zero phase change") filtering techniques. The approach is capable of achieving some compensation of the effects of modelling error gains without the introduction of the phase lag associated with "uni-directional" filters. Controlling phase lag is essential as it

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_8

could lead to the possible consequence that the required positivity conditions for the modelling error are violated.

The operator notation is used throughout and specific detailed results for linear, time invariant, discrete time state space models stated in some detail.

The Case of $m \leq \ell$ Suppose that G_R is a right inverse of a model G of a plant UG with left multiplicative modelling error U. Then the inverse model Algorithm 6.1 has update formula $u_{k+1} = u_k + \beta G_R e_k$ and is monotonically convergent if U is asymptotically stable and the relevant operator or frequency domain conditions are satisfied (Theorem 6.6). Compensation for any difficult gain characteristics of $U(z)$ such as high frequency parasitic resonances clearly must take into account the need to avoid introducing additional phase lag into the system whilst simultaneously providing appropriate gain changes over chosen frequency bands. In what follows this is achieved using an algorithm based on the use of a filter $F : \mathscr{F} \to \mathscr{Y}$ where, as used previously, $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ is regarded as a Hilbert space created from time series in $\mathscr{R}^m$ of length $N+1$ in supervector form. The chosen inner product and norm are

$$\langle y, w\rangle_{\mathscr{Y}} = \sum_{t=0}^{N} y^T(t) Q w(t) \quad \textit{and} \quad \|y\|_{\mathscr{Y}} = \sqrt{\langle y, y\rangle_{\mathscr{Y}}} \quad \textit{with} \quad Q = Q^T > 0. \tag{8.1}$$

The space $\mathscr{F} = \mathscr{R}^{m_F(N+1)}$ is regarded as a Hilbert space created from time series in $\mathscr{R}^{m_F}$ of length $N+1$ in supervector form. The chosen inner product and norm are written as follows

$$\langle f, g\rangle_{\mathscr{F}} = \sum_{t=0}^{N} f^T(t) R_F g(t) \quad \textit{and} \quad \|f\|_{\mathscr{F}} = \sqrt{\langle f, f\rangle_{\mathscr{F}}} \quad \textit{where} \quad R_F = R_F^T > 0. \tag{8.2}$$

As usual, the filter is assumed to have a state space model $S(A_F, B_F, C_F, D_F)$. The following algorithm is bi-directional (or bi-causal) as it includes the operation $FF^* e_k$ which is realized by *two* simulations. The first filtering computation, $\psi_k = F^* e_k$, is approached using the "adjoint model"

$$S(A_F^T, C_F^T Q, R_F^{-1} B_F^T, R_F^{-1} D_F^T Q)$$

and time reversal operator techniques and takes the form

$$\psi_k = \mathscr{T}(m_F, N) G(A_F^T, C_F^T Q, R_F^{-1} B_F^T, R_F^{-1} D_F^T Q) \mathscr{T}(m, N) e_k, \tag{8.3}$$

An equivalent statement is as follows,

$$\begin{aligned} z_k(t+1) &= A_F^T z_k(t) + C_F^T Q e_k(N-t), \quad z_k(0) = 0, \qquad \textit{and} \\ \psi_k(t) &= R_F^{-1} B_F^T z(N-t) + R_F^{-1} D_F^T Q e_k(t), \ 0 \leq t \leq N. \end{aligned} \tag{8.4}$$

The second simulation, $F(F^*e_k) = F\psi_k$ uses the state space model $S(A_F, B_F, C_F, D_F)$ to compute its response, from zero initial conditions, to the time series $\psi_k(t),\ 0 \leq t \leq N$.

Algorithm 8.1 (*Right Inverse Algorithm with Bi-directional Filtering*) Suppose that a model G of the plant UG has a right inverse G_R and U is invertible. Suppose also that the design process has specified an $m \times m_F$ filter $F(z)$ that is asymptotically stable and minimum-phase with the property that $ker[F^*] = \{0\}$ (that is, $rank[D_F] = m$ and hence $m \leq m_F$ and $\mathscr{R}[F] = \mathscr{Y}$). The Right Inverse Model Algorithm with Bi-directional Filtering is defined by the input update formula

$$u_{k+1} = u_k + \beta G_R F F^* e_k, \quad k \geq 0. \tag{8.5}$$

The resultant error evolution is described by the equation

$$e_{k+1} = (I - \beta U F F^*) e_k, \quad k \geq 0. \tag{8.6}$$

The error evolution equation has the formal structure of a perturbed steepest descent algorithm with G replaced by F and hence

1. A sufficient condition for convergence to be achieved is obtained from Theorem 7.16. In terms of the proof of that Theorem, the error was written as $e = UG(u_\infty - u)$ and monotonic convergence defined in terms of the subsidiary time series $w_k = u_\infty - u_k,\ \ k \geq 0$, which has an obvious interpretation. For Algorithm 8.1, the variable w_k defined by $e_k = UFw_k$ does not have this interpretation but it can still be used in the proof as it is uniquely defined if values are taken to be in the closed (Hilbert) subspace $\mathscr{R}[F^*]$. With this understanding, a sufficient condition for monotonic convergence is that $U(z)$ is minimum-phase and also that,

$$\hat{U}^T(z)Q + Q\hat{U}(z^{-1}) > \beta Q F(z^{-1}) R_F^{-1} F^T(z) Q \quad \textit{whenever} \ \ |z| = 1\,. \tag{8.7}$$

The left-hand-side of this expression depends primarily on the modelling error $U(z)$ whilst the right-hand-side depends on the chosen filter $F(z)$ and the gain β. This observation is a key link between the properties of U and the choice of filter. Conversely, for a given filter, the expression provides data on the class of modelling errors U that can be tolerated whilst retaining monotonic convergence.
2. In the "nominal" case of no modelling error, $U = I$ and the condition reduces to $2Q > \beta Q F(z^{-1}) R_F^{-1} F^T(z) Q$ whenever $|z| = 1$, which, in operator terms, is just $0 < \beta \|F^*\|^2 < 2$ if N is large.
3. Both of the previous two observations have meaning if ε-weighted models and norms are used as described in Sect. 7.5.5, simply by replacing $U(z)$ by $U_\varepsilon(z) = U(\varepsilon^{-1}z)$ and $F(z)$ by $F_\varepsilon(z) = F(\varepsilon^{-1}z)$ from which monotonic convergence with respect to the weighted norm is ensured if

$$\hat{U}^T(z)Q + Q\hat{U}(\overline{z}) > \beta Q F(\overline{z}) R_F^{-1} F^T(z) Q \quad \textit{whenever} \ \ |z| = \varepsilon^{-1}. \tag{8.8}$$

The choice of $F(z)$ is an open question that is strongly related to the design issues pertinent to the problem/application being considered. The need for a form of positive real modelling errors is still essential but, using the techniques of Sect. 7.2.3, an analysis of the first iteration $e_1 = (I - \beta FF^*)e_0$ where e_0 is the supervector corresponding to the time series $\{e_0(t, z, \alpha)\}_{0\le t\le N} = \{z^t\alpha\}_{0\le t\le N}$ gives the approximate time series

$$e_1(t) \approx (I - \beta U(z)F(z)R_F^{-1}F^T(z^{-1})Q)e_0(t, z, \alpha), \quad 0 \le t \le N, \text{ if } N \text{ is large.} \tag{8.9}$$

This indicates that (a) the use of low pass filters with a specified bandwidth will help to suppress the effects of significant high frequency properties of $U(z)$ or (b) the use of a notch filter could reduce the effects of problematic frequency ranges. The choice should reflect the need to reduce problems in algorithm performance and also allow increased values of gain β to improve convergence rates.

Special Case of $m = \ell = 1$: Insight into possibilities is clearer in the SISO case. Take, for example,$Q = 1$ and $R_F = I_{m_F}$, and set $F(z)$ to be the row vector

$$F(z) = \left[F_1(z), \ldots, F_{m_F}(z)\right] \tag{8.10}$$

where $F_j(z),\ 1 \le j \le m_F$, are scalar, asymptotically stable filters. The condition $ker[F^*] = \{0\}$ simply requires that at least one such filter has a non-zero "D" term in its state space model. With this construction, the convergence condition becomes

$$\left|\frac{1}{\beta} - U(z)\sum_{j=1}^{m_F}|F_j(z)|^2\right| < \frac{1}{\beta} \quad \textit{whenever} \quad |z| = 1\ . \tag{8.11}$$

This clearly reveals the role of F in shaping the gain characteristics of U and their consequences for convergence and gain selection. The flexibility implicit in the role of m_F is also revealed. Note the separate effects of each filter in the summation. This structure opens up the possibility of filter design using simple component elements F_j designed for different purposes over different bandwidths. For example, let $m_F = 2$ and consider the 1×2 filter structure

$$F(z) = [F_1(z), F_2(z)] = \left[(1-\lambda)^{0.5}, \lambda^{0.5}\left(\frac{1-p}{z-p}\right)\right], \quad \textit{with}\ \ \lambda \in (0, 1), \tag{8.12}$$

gives

$$\sum_{j=1}^{m_F}|F_j(z)|^2 = (1-\lambda) + \lambda\frac{(1-p)^2}{|z-p|^2} \tag{8.13}$$

and provides separate influence over high frequency and low frequencies (by choice of λ) and bandwidth compensation (by choice of p).

The Case of $m \geq \ell$: The algorithm in this case takes the form,

Algorithm 8.2 (*Left Inverse Model Algorithm with Bi-directional Filtering*) Suppose that $m \geq \ell$ and that the model G of the plant GU has a left inverse G_L (and hence that $ker[G] = \{0\}$ and $\mathscr{Y} = ker[G_L] \oplus \mathscr{R}[G]$). Suppose also that U is invertible and that the design process has specified an $\ell \times m_F$ filter $F(z)$ that is asymptotically stable with the property that $ker[F^*] = \{0\}$ (that is $rank[D_F] = \ell$). The Left Inverse Model Algorithm with Bi-directional Filtering is defined by the update formula

$$u_{k+1} = u_k + \beta FF^* G_L e_k, \quad k \geq 0. \tag{8.14}$$

The resultant error evolution is described by $e_{k+1} = (I - \beta GUFF^* G_L)e_k, \ \ k \geq 0$.

Each error e_k can be written in the unique form $e_k = Gw_k + \xi_k$ with $\xi_k \in ker[G_L]$. A simple calculation gives the result that $\xi_k = \xi_0$ for all $k \geq 0$, and hence that the component of e_0 in $ker[G_L]$ remains unchanged from iteration to iteration. Significant error evolution only occurs, therefore, in $\mathscr{R}[G]$ and is described by the formula

$$w_{k+1} = (I - \beta UFF^*)w_k, \quad k \geq 0 \, . \tag{8.15}$$

Again, the equation has the structure of a perturbed steepest descent algorithm with G replaced by F and hence a sufficient condition for convergence of $\{w_k\}_{k\geq 0}$ to a limit $w_\infty = 0$ is that given for the case of $m \leq \ell$. The corresponding error evolution satisfies the condition

$$\lim_{k\to\infty} e_k = \xi_0 \, . \tag{8.16}$$

Conclusions and Comments: The material described above provides a partial unification of the inverse model and gradient paradigms for Iterative Control. The outcome is aimed at providing some control over the robustness of the inverse model approach using bi-directional filtering operations that mimic the structure of gradient methodologies. In the form described, the need for positivity conditions to be satisfied is retained but, where this causes difficulties, the use of ε-weighted signals can help considerably. With these constraints in mind however, the approach is suitable for implementation as a conditioning tool for inverse model algorithms although the choice of filter F will need careful consideration to meet the needs and concerns of the particular application being considered.

8.2 General Issues in Design

Although much of the general theory and underlying concepts and ideas presented in the text applies to a wide class of dynamical systems, the detailed analysis of the case of linear, time invariant, discrete time, state space systems provides some tools for analysis and design and indicates that the properties of the plant model G and the

form of any multiplicative modelling error U play an important role in the design process and behaviours. A review of the material of previous chapters suggests some general design guidelines. For example, specific issues that can be identified include:

1. The application of fixed (iteration independent) control loops to "pre-condition" the plant may be essential as a means of improving the potential for successful Iterative Control. The resultant controller is implemented on the plant. The original plant model is then replaced by a model of the pre-conditioned plant (and again denoted by G).
2. A preliminary analysis of the resultant conditioned model is useful and perhaps essential.
3. The design of of compensation schemes for the pre-conditioned plant to achieve the desired Iterative Control performance and robustness are the final off-line design stage.

8.2.1 Pre-conditioning Control Loops

Given an ℓ-input, m-output, state space model $S(A, B, C, D)$ (and associated transfer function matrix $G(z)$) of the plant to be controlled, the stability and robustness conditions for gradient-based Iterative Control depend explicitly on the the transfer function matrix $G(z)$ of the plant model and model error $U(z)$. In particular there is a need to ensure stability of the plant (essential for feedforward Iterative Control algorithms) and, where possible to eliminate or, at least reduce, the effects of non-minimum-phase zeros, resonances (small damping) and zeros close to the stability boundary. With this in mind, it is reasonable to assert that control loops should be added to the plant before Iterative algorithms are considered.

There are a large number of candidates for such pre-conditioning control loops and the pre-conditioning can be applied as "analogue control" to the underlying continuous system before sampling or to the sampled system itself. They, of course, include standard output feedback and feedforward controllers designed for stability, robustness and applications specific reasons. From a scientific point of view however, it is of interest to assess the potential full power of feedback by investigating what the use of full state feedback can offer. The full analysis of such an idea is beyond the scope of this text but an insight into possibilities can be obtained from the following examples.

Deadbeat Control Suppose that $m = \ell$, that $D = 0$ and that the model has uniform rank $k^* \geq 1$. That is, $CA^{j-1}B = 0$ for $1 \leq j < k^*$ with $\det[CA^{k^*-1}B] \neq 0$. Then the state feedback control law

$$u(t) = v(t) - (CA^{k^*-1}B)^{-1}CA^{k^*}x(t), \quad t \geq 0 \tag{8.17}$$

replaces the input u by the new input v and generates the closed loop plant model $S(A - B(CA^{k^*-1}B)^{-1}CA^{k^*}, B, C)$. A simple calculation then yields the following simple equation for closed loop output dynamics,

$$y(t + k^*) = CA^{k^*-1}Bv(t), \quad t \geq 0. \tag{8.18}$$

This model has a minimal realization of state dimension k^*m defined by the matrices

$$A = \begin{bmatrix} 0 & I_m & 0 & \cdots & \cdots & 0 \\ 0 & 0 & I_m & 0 & \cdots & 0 \\ \vdots & & & & & \vdots \\ \vdots & & & & & I_m \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ CA^{k^*-1}B \end{bmatrix}, \quad C = [I_m \ 0 \ \cdots 0] \ . \tag{8.19}$$

The state vector is $x(t) = \left[y^T(t + k^*), y^T(t + k^* - 1), \ldots, y^T(t)\right]^T$. The remaining $n - k^*m$ unobservable states are associated with poles equal to the zeros of the plant and hence are stable if, and only if, the original plant was minimum-phase.

State feedback can simplify the nature of plant dynamics although the range of its successful application may be influenced by the position of plant zeros. An interesting observation here is that the resultant closed loop system has transfer function matrix equal to $z^{-k^*}CA^{k^*-1}B$ and hence, assuming zero initial conditions, that the shifted output $y^e(t)$ defined by $y^e(t) = y(t + k^*)$ for $t \geq 0$ is described by the simple algebraic expression $y^e(t) = CA^{k^*-1}Bv(t)$. It is a simple matter now to show that the update rule $v_{k+1}(t) = v_k(t) + \beta(CA^{k^*-1}B)^{-1}e_k^e(t)$ where $e_k^e(t) = r^e(t) - y_k^e(t)$ and $r_k^e(t) = r(t + k^*)$ gives the recursion $e_{k+1}^e(t) = (1 - \beta)e_k^e(t)$, $t \geq 0$. This converges if $\beta \in (0, 2)$, a direct parallel to the expressions appearing in the inverse model algorithm. Clearly the use of state feedback can create inverse model behaviour exactly. More generally, it may create useful approximations to inverse algorithms.

Pole Allocation Loops: State feedback can also be used to allocate the poles of the preconditioned plant using the more general state feedback rule

$$u(t) = v(t) - Fx(t), \quad t \geq 0, \tag{8.20}$$

where F denotes an $\ell \times n$ state feedback matrix. If $S(A, B, C)$ is controllable, the eigenvalues/poles of the closed loop system $S(A - BF, B, C)$ can be allocated to any desired locations. The deadbeat control example is just one choice but leads to an unobservable closed loop system. Consider the special case of a single-input, single-output system ($m = \ell = 1$) of relative degree $k^* = 1$ and with $n - 1$, distinct, real zeros $0 < z_j < 1$ ordered as follows $z_1 > z_2 > \cdots > z_{n-1}$. Choose F to allocate the closed loop poles to real values $0 < p_j < 1, \ \leq j \leq n$ satisfying the interlacing condition

$$p_n < z_{n-1} < p_{n-1} < \cdots < z_1 < p_1. \tag{8.21}$$

The interesting properties of such "interlaced systems" include the facts that

1. The residues of the poles of $G(z)$ in its partial fraction expansion all take the same sign. Physically, if they are positive for example, the step response from zero initial conditions is monotonically increasing in the sense that $y(t_2) \geq y(t_1)$ whenever $t_2 > t_1$.
2. Also the H_∞ norm $\|G(z)\|_\infty = \sup_{|z|=1} |G(z)|$ of the closed loop transfer function $G(z)$ (and its shifted version $zG(z)$) is exactly the steady state value $|G(1)|$. This value can be used to compute a good upper bound for the induced norm of the matrix operator in the supervector description of $S(A - BF, B, C)$ and its shifted version $S(A - BF, B, CA, CB)$.

Within the monotonicity property, there is an infinity of design opportunities to influence closed loop response speed and bandwidth. Once achieved, the computation of a suitable range of β for use in steepest descent-based Iterative Control can then be computed from steady state step response characteristics. The approach also creates opportunities for simplification of the algorithm. For example, if the interlacing condition is satisfied and the differences $|z_j - p_j|$, $1 \leq j \leq n-1$ are small, then the closed loop system can be thought of as being the product of a first order model $G(z)$ and a model error $U(z)$ described by,

$$G(z) = G(1)\frac{(1-p_n)}{(z-p_n)} \quad \textit{and} \quad U(z) = \prod_{j=1}^{n-1}\left(\frac{(z-z_j)(1-p_j)}{(z-p_j)(1-z_j)}\right). \tag{8.22}$$

Noting that $U(1) = 1$, it is a simple matter to prove that $U(z)$ is also strictly positive real. The use of the first order model (in shifted form) as the model to be used for steepest descent Iterative Control will therefore be successful with a suitable value of β. Using the techniques of Sect. 7.5.3, the permissible range of β is computed from the graphical condition that, whenever $|z| = 1$,

$$\left|\frac{1}{\beta} - U(z)\,(zG(z))\left(z^{-1}G(z^{-1})\right)\right| = \left|\frac{1}{\beta} - U(z)\frac{|G(1)|^2(1-p_n)^2}{(z-p_n)(z^{-1}-p_n)}\right| < \frac{1}{\beta} \tag{8.23}$$

8.2.2 Compensator Structures

Given the final, pre-conditioned, asymptotically stable, plant model G, the design of steepest descent algorithms opens up a wide range of possibilities. Based on the successful analysis of inverse and gradient-based algorithms, the following general structures could form the basis of systematic design. The two compensator characterizations considered below contain elements of approximate inversion aimed at producing rapid and controlled convergence rates but also make use of the "zero-phase" aspects of steepest descent. The details are as follows and are based on the use of an "approximate right inverse" G_{aR} of a discrete state space system G,

Algorithm 8.3 (*Combined Inverse and Gradient-based Control when $m \leq \ell$*) Suppose that $m \leq \ell$ with $\mathscr{R}[G] = \mathscr{Y}$. Suppose also that $\mathscr{R}[G_{aR}] \cap ker[G] = \{0\}$ and $ker[G_{aR}] = \{0\}$. In particular it follows then that $\mathscr{U} = \mathscr{R}[G_{aR}] \oplus ker[G]$. A combined inverse and gradient-based input update rule could have the form

$$u_{k+1} = u_k + \beta G_{aR}(GG_{aR})^* e_k, \quad k \geq 0, \tag{8.24}$$

where G_{aR} is an asymptotically stable compensation network. Conceptually, it can be regarded as an approximate right inverse of G. The idea that G_{aR} is an approximate right inverse is not interpreted in a narrow sense and allows the design engineer freedom to roam over a wide range of possibilities.

The resultant error evolution is represented by the supervector recursion

$$e_{k+1} = (I - \beta GG_{aR}(GG_{aR})^*)e_k \tag{8.25}$$

and the choice of β can be based on the conditions for the steepest descent algorithm for the m-input, m-output system GG_{aR}. Analysis of robustness to left multiplicative model errors can be approached using the techniques of Sect. 7.5.3.

The following is a similar result using "approximate left inverse" systems G_{aL} of G.

Algorithm 8.4 (*Combined Inverse and Gradient-based Control when $m \geq \ell$*) Suppose that $m \geq \ell$, that $ker[G] = \{0\}$, that $\mathscr{R}[G_{aL}] = \mathscr{U}$ and that $ker[G_{aL}] \cap \mathscr{R}[G] = \{0\}$. In particular, it follows that $\mathscr{Y} = ker[G_{aL}] \oplus \mathscr{R}[G]$. The compensator structure naturally takes the form

$$u_{k+1} = u_k + \beta(G_{aL}G)^* G_{aL} e_k, \quad k \geq 0, \tag{8.26}$$

where G_{aL} represents an approximate left inverse of G. The error evolution is

$$e_{k+1} = ((I - \beta G(G_{aL}G)^* G_{aL}))e_k \tag{8.27}$$

Defining $w_k = G_{aL}e_k, \quad k \geq 0$, then

$$w_{k+1} = (I - \beta G_{aL}G(G_{aL}G)^*)w_k \tag{8.28}$$

and choice of β can be based on the conditions for the steepest descent algorithm for the ℓ-input, ℓ-output system $G_{aL}G$. Monotonic convergence of $\{w_k\}_{k\geq 0}$ to zero then ensures that $e_k \to e_0^{(2)}$ where the initial error has the unique decomposition $e_0 = e_0^{(1)} + e_0^{(2)}$ with $e_0^{(1)} \in \mathscr{R}[G]$ and $e_0^{(2)} \in ker[G_{aL}]$.

Finally, analysis of robustness to right multiplicative model errors can be approached using the techniques of Sect. 7.5.3.

Note that in the case when the approximate inverse is an exact inverse, the algorithms reduce to the simple inverse model algorithms described in Chap. 6.

8.2.3 Stable Inversion Algorithms

Combined approximate inverse and gradient concepts by adding bi-directional filtering offers a possible means of coping with inverse algorithms when the plant model G is non-minimum-phase.

The Case of $m = \ell = 1$: Consider the case of SISO systems with n_z zeros $z_1, \ldots, z_{n_z}$, the first n_+ of which, $\{z_j\}_{1 \le j \le n_+}$, have a modulus $|z| > 1$ with all remaining zeros satisfying $|z| < 1$. The transfer function of this system can be written as a product $G = G_{ap} G_m$. That is,

$$G(z) = G_{ap}(z) G_m(z) \quad \textit{where} \quad G_{ap}(z) = \prod_{j}^{n_+} \left(\frac{z - z_j}{1 - z z_j} \right) . \tag{8.29}$$

Here G_m is minimum-phase as the zeros of G at the points $z = z_j, \quad 1 \le j \le n_+$ have been replaced by zeros at points $z = z_j^{-1}, \quad 1 \le j \le n_+$. The factor $G_{ap}(z)$ is an asymptotically stable, all-pass network in the sense that

$$G_{ap}(z) G_{ap}(z^{-1}) \equiv 1, \quad \textit{and hence} \quad |G_{ap}(z)| \equiv 1 \quad \textit{whenever} \quad |z| = 1. \tag{8.30}$$

Following the application of the relevant shift, the system G_m has a stable inverse G_m^{-1}. Consider the Iterative Algorithm

Algorithm 8.5 (*Stable Inverse Algorithm when* $m = \ell = 1$) Using the notation defined above, suppose that the input and output spaces have the Euclidean norms $\|f\| = \sqrt{f^T f}$ (so that $Q = R = 1$). Suppose also that the plant has dynamics represented by UG where the multiplicative modelling error $U(z)$ is asymptotically stable. Consider the Iterative Algorithm input update rule

$$u_{k+1} = u_k + \beta G_m^{-1} F F^* G_{ap}^* e_k \tag{8.31}$$

where F is a filter as described in Sect. 8.1. The resultant error evolution is described by

$$e_{k+1} = (1 - \beta U G_{ap} F F^* G_{ap}^*) e_k, \quad k \ge 0. \tag{8.32}$$

The error sequence hence has the structure of steepest descent control for a process with model $G_{ap}F$ and modelling error $U(z)$. Taking $m_F = 1$, application of the robustness conditions for steepest descent iterations then proves that $\{e_k\}_{k \ge 0}$ converges to zero for all choice of β satisfying the graphical condition

$$\left| \frac{1}{\beta} - U(z) |F(z)|^2 \right| < \frac{1}{\beta}, \quad \textit{whenever} \quad |z| = 1. \tag{8.33}$$

The algorithm properties follow from discussion of robustness of steepest descent algorithms in Sect. 7.5.3. In particular it follows as the all-pass structure of G_{ap}

implies that $U(z)|G_{ap}(z)F(z)|^2 \equiv |F(z)|^2 U(z)$. The interesting conclusion is that the convergence condition is precisely the condition derived for robustness of the inverse algorithm using the full (right) inverse G^{-1} of the model G of UG. The replacement of this inverse by the defined stable inverse hence has no effect on the choice of gain required for monotonic convergence.

Note: There is a significant algorithm behaviour hidden behind this analysis. In the next subsection, an eigenvalue analysis indicates that convergence rates have special properties that can severely limit convergence rates in certain subspaces.

The Case of $m = \ell \geq 1$: Finally, the supervector descriptions may apply , in principle, to the case of multi-input, multi-output systems. The theoretical and computational aspects of this observation are beyond the scope of this text but it is worthwhile stating the natural structures that could be needed. More precisely, assuming $m = \ell$ and the topologies in $\mathscr{Y}$ and $\mathscr{U}$ defined by weighting matrices Q and R respectively, the natural model G should have a decomposition $G = G_{ap}G_m$ where both elements have asymptotically stable, discrete, state space models. The model of G_m should also be minimum phase whilst that of G_{ap} should be all-pass in the sense that its transfer function matrix $G_{ap}(z)$ satisfies the identity

$$G_{ap}(z)Q^{-1}G_{ap}^T(z^{-1})Q \equiv I_m. \tag{8.34}$$

In operator terms, this simply a condition that ensures that $G_{ap}G_{ap}^* = I$.

8.2.4 All-Pass Networks and Non-minimum-phase Systems

In what follows, the mathematical structure of $G_{ap}^*G_{ap}$ and $G_{ap}G_{ap}^*$ is described and related to slow algorithm convergence in certain subspaces of $\mathscr{Y}$. The SISO case of $m = \ell = 1$ is considered using eigenvalue analysis. Without loss of generality, it is assumed that $Q = R = 1$. The main theorem is now stated,

Theorem 8.1 (All-pass System Singular Value Distribution)
Suppose that an all pass system $G_{ap}(z)$ has the form

$$G_{ap}(z) = \frac{\prod_{j=1}^{n_+}(z - z_j)}{\prod_{j=1}^{n_+}(1 - z_j z)} = \frac{a_0 + a_1 z + \cdots + a_{n_+}z^{n_+}}{a_{n_+} + a_{n_+-1}z + \cdots + a_0 z^{n_+}}, \tag{8.35}$$

where $|z_j| > 1, i = 1, \ldots, n_+$, $a_{n_+} = 1$ and $a_0 \neq 0$. If the system operates on the time interval $0 \leq t \leq N$, then there are two cases possible for the distribution of the singular values of the supervector model matrix operator $G_{ap} : \mathscr{R}^{N+1} \to \mathscr{R}^{N+1}$:

1. *if the time interval length $N+1 \le n_+$, then all singular values $\sigma_j < 1$.*
2. *Alternatively, there are n_+ singular values $\sigma_j < 1$, $1 \le j \le n_+$ whilst all remaining singular values take the value $\sigma = 1$.*

Note: the eigenvalues of $G_{ap}^T G_{ap}$ and $G_{ap} G_{ap}^T$ are identical.

Proof For any vector $u = [u(0), u(1), \ldots, u(N)]^T$ define $u(z) = \sum_{j=0}^{N} u(j) z^{-j}$, and $y(z) = G_{ap}(z)u(z)$. Note that

$$u^T G_{ap}^T G_{ap} u \le \frac{1}{2\pi i} \oint_{|z|=1} \left| G_{ap}(z)u(z) \right|^2 \frac{dz}{z} = \frac{1}{2\pi i} \oint_{|z|=1} |u(z)|^2 \frac{dz}{z} = u^T u. \tag{8.36}$$

so that $G_{ap}^T G_{ap} \le I$ and all the eigenvalues of $G_{ap}^T G_{ap}$ are less than or equal to one. That is, all the singular values of G_{ap} are less than or equal to one. Now, consider the equation

$$u^T (I - G_{ap}^T G_{ap}) u = 0. \tag{8.37}$$

All the non-zero solutions u are eigenvectors of $G_{ap}^T G_{ap}$ corresponding to eigenvalues equal to one, and they span a subspace of $\mathscr{R}^{N+1}$ whose dimension is identical to the number of singular values satisfying $\sigma_i = 1$. Examination of the frequency domain inequality indicates that the equality condition holds if, and only if, in the expansion $y(z) = \sum_{j=0}^{\infty} y(j) z^{-j}$, the coefficients $y(j)$ are zero for $j > N$. That is $y(z)$ must be a polynomial in z^{-1} of degree $\le N$. There are two cases to consider, the first being the most important as, typically, N is large.

Case 1: Suppose that $N + 1 \ge n_+$. Consider the choice $u(z) = a_0 + a_1 z^{-1} + \cdots + a_{n_+} z^{-n_+}$, then

$$y(z) = G_{ap}(z)u(z) = a_{n_+} + a_{(n_+-1)} z^{-1} + \cdots + a_0 z^{-n_+}. \tag{8.38}$$

Clearly $y(j) = 0$ for $j > n_+$ and hence for $j > N$. Therefore the equality in Eq. (8.36) holds, which means $u = [a_0, a_1, \ldots, a_{n_+}, 0, \ldots 0]^T$ is a solution of (8.37). There are $N + 1 - n_+$ solutions of this type defined by the "shifted" polynomials

$$\begin{gathered} u_{(n_+p)}(z) = z^{-(p-1)} u(z) = a_0 z^{-(p-1)} + a_1 z^{-(p-1)-1} + \cdots + a_{n_+} z^{-(p-1)-n_+}, \\ \textit{where} \quad p = 1, \ldots, N+1-n_+, \end{gathered} \tag{8.39}$$

that also yield equality in Eq. (8.36) and the corresponding supervectors,

$$\begin{aligned} u_{n_++1} &= [a_0,\ a_1,\ \cdots,\ a_{n_+},\ \ 0,\ \ \cdots,\ \ 0,\ \ \ 0]^T \\ u_{n_++2} &= [\ 0,\ a_0,\ \ a_1,\ \ \cdots,\ a_{n_+},\ \ 0,\ \ \cdots,\ \ 0]^T \\ &\vdots \qquad\qquad\qquad\qquad\qquad \vdots \\ u_{N+1} &= [\ 0,\ 0,\ \ \cdots\ \ 0,\ \ \ a_0,\ \ \ a_1,\ \ \cdots,\ a_{n_+}]^T \end{aligned} \tag{8.40}$$

are also solutions of (8.37). This procedure generates $N+1-n_+$ linearly independent supervector solutions $\{u_p\}$ of (8.37). There are no other linearly independent solutions. To prove this, suppose there exists such a solution supervector $v=[v(0),\ldots,v(N)]^T$ represented by $v(z)=\sum_{j=0}^{N} v(j)z^{-j}$. Then there must exist a non-zero $v_0(z)=\sum_{j=N-n_++1}^{N} v_0(j)z^{-j}$ which is also a solution and can be constructed by subtracting linear combinations of u_p from v. Note that

$$G_{ap}(z)v_0(z)=\left(\frac{a_0+a_1z+\cdots+a_{n_+}z^{n_+}}{a_{n_+}+a_{(n_+-1)}+\cdots+a_0z^{n_+}}\right)\sum_{j=N-n_++1}^{N} v_0(j)z^{-j} \tag{8.41}$$

is a polynomial, if and only if, $v_0(z)$ cancels all the poles of $G_{ap}(z)$. This, however, cannot be satisfied because $v_0(z)$ has n_+-1 non-zero roots whilst $G_{ap}(z)$ has n_+ poles. Hence, there are exactly $N-n_+$ linearly independent solutions of (8.37) and these correspond to $N-n_+$ singular values equal to one. This also proves that all other singular values are strictly less than unity.

Case 2: $N+1<n_+$ In this case, it can be easily seen from the discussion above that there are no solutions satisfying (8.37). Hence all the singular values should be less than one. That completes the proof of the Theorem. □

Theorem 8.2 (Eigenvectors of $G_{ap}G_{ap}^T$) *Assume the notation of the preceding result and suppose that $N+1>n_+$. Then the eigenstructure of $G_{ap}G_{ap}^T$ has the form of*

1. *$N+1-n_+$ linear independent eigenvectors corresponding to the $N+1-n_+$ eigenvalues $\sigma_j^2=1$, $n_++1\le j\le N+1$:*

$$\begin{aligned}
\tilde{u}_{n_++1} &= [a_{n_+},\ a_{(n_+-1)},\ \cdots\ a_0,\ \ 0,\ \ \cdots,\ \ 0,\ \ 0]^T\\
\tilde{u}_{n_++2} &= [\ 0,\ \ a_{n_+},\ \ \cdots,\ a_1,\ \ a_0,\ \ 0,\ \ \cdots,\ 0]^T\\
&\ \ \vdots \qquad\qquad\qquad\qquad\quad \vdots\\
\tilde{u}_N &= [\ 0,\ \ 0,\ \ \cdots\ 0,\ \ a_{n_+},\ a_{(n_+-1)},\ \cdots,\ a_0]^T
\end{aligned} \tag{8.42}$$

   The span of these vectors, denoted $\mathscr{E}_1$, has the property that $G_{ap}G_{ap}^Ty=y$ whenever $y\in\mathscr{E}_1$.
2. *The n_+ eigenvectors of $G_aG_a^T$ corresponding to the n_+ eigenvalues that are less than one span the subspace $\mathscr{E}_{a+}=\mathscr{E}_1^{\perp}$.*

Proof Using time reversal matrices, note that $G_{ap}G_{ap}^T=\mathscr{T}(1,N)G_{ap}^TG_{ap}\mathscr{T}(1,N)$ and hence the eigenvectors u_p of $G_{ap}^TG_{ap}$ and $\tilde{u}_p$ of $G_{ap}G_{ap}^T$ are related by $\mathscr{T}(1,N)$. The proof of Theorem 8.1 then completes the proof of the first part of the result as a simple calculation gives $\tilde{u}_p=\mathscr{T}(1,N)u_{N+2+n_+-p}$, $p=n_++1,\ldots,N+1$. The form of $\mathscr{E}_{a+}$ follows as all other eigenvalues are less than unity and the eigenspaces of symmetric matrices corresponding to different eigenvalues are orthogonal. □

The two issues of importance to what follows are the magnitude of the non-unit eigenvalues and the structure of the subspace $\mathscr{E}_{a+}$. The simplest case is that when $n_+ = 1$ when the zero z_1 is real. In this case $\mathscr{E}_{a+}$ has dimension equal to one. Using the orthogonality property of $\mathscr{E}_{a+}$ and the definition of the zero, a simple calculation shows that eigenvector is expressed simply as

$$\alpha_1 = \left[1, z_1^{-1}, z_1^{-2}, \ldots, z_1^{-N}\right]^T \tag{8.43}$$

as $\tilde{u}_p^T \alpha_1 = 0, \quad n_+ + 1 \leq p \leq N + 1$. Noting that, in any state space realization of $U(z)$, the "D" matrix is simply the scalar $-z_1^{-1}$, and that the determinant is the product of the eigenvalues, write

$$\det[G_{ap} G_{ap}^T] = \det[G_{ap}]^2 = z_1^{-2(N+1)} = \prod_{j=1}^{N+1} \sigma_j^2 = \sigma_1^2. \tag{8.44}$$

which proves the result that

Theorem 8.3 (Non-unit Singular Values and Eigenvectors when $n_+ = 1$) *If $n_+ = 1$, then the only non-unit eigenvalue of $G_{ap} G_{ap}^T$ is $\sigma_1^2 = z_1^{-2(N+1)}$ with eigenvector α_1. In particular, as $|z_1| > 1$,*

$$\lim_{N \to \infty} \sigma_1^2 = 0 \tag{8.45}$$

and hence this eigenvalue can be made to be arbitrarily small if $z_1^{2(N+1)}$ is sufficiently large.

Note: For a given value of z_1, increasing the length of the interval reduces the magnitude of the eigenvalue.

It is useful at this stage to discuss the implications of this result for the stable inverse Algorithm of Sect. 8.2.3 in the special case of $F = I$ (no filtering) and the use of Euclidean norms. The resultant error evolution $e_{k+1} = (I - \beta G_{ap} G_{ap}^T) e_k$ can then be described by decomposing e_0 into the sum $e_0 = e_0^{(1)} + e_0^{(2)}$ where $e_0^{(1)} \in \mathscr{E}_{a+}$ and $e_0^{(2)} \in \mathscr{E}_1$, from which

$$e_k = (1 - \beta \sigma_1^2)^k e_0^{(1)} + (1 - \beta)^k e_0^{(2)}, \quad k \geq 0\,. \tag{8.46}$$

If z_1^N is large, then, whereas the second term in $\mathscr{E}_1$ becomes small quickly (depending on the choice of β), it is clear that the first term in $\mathscr{E}_{a+}$ decays infinitesimally slowly. This is clearly demonstrated in the special case of $\beta = 1$ when the second term is removed in one iteration. Suppose that $z_1 = 1.1$ and $N = 127$, then $\sigma_1^2 \approx 0.5 \times 10^{-4}$ and it is clear that $(1 - \beta 0.5 \times 10^{-4})^k$ reduces very slowly. This situation is further worsened as N increases. The theoretical convergence to the limit $e_\infty = 0$ remains true but the practical implication of the zero z_1 is that the user, over the finite number

of iterations possible in the real world, will observe an apparent convergence to a "pseudo-limit" vector

$$e_{\infty}^{pseudo} = e_0^{(1)} = \frac{\langle \alpha_1, e_0 \rangle}{\|\alpha_1\|^2} \alpha_1 \quad (\textit{The Apparent Limit Error}). \tag{8.47}$$

Graphically, this will observed as monotonic reductions in error norm until an apparent "plateauing" or "flat-lining" of the plot of $\|e_k\|$ against k will appear at a value very close to that computed from the formula

$$\begin{aligned} \|e_{\infty}^{pseudo}\| = \|e_0^{(1)}\| &= \tfrac{|\langle \alpha_1, e_0 \rangle|}{\|\alpha_1\|} = \sqrt{\tfrac{1-z_1^{-2}}{1-z_1^{-2(N+1)}}}\, |\langle \alpha_1, e_0 \rangle| \\ &= \sqrt{\tfrac{1-z_1^{-2}}{1-z_1^{-2(N+1)}}} \left| \textstyle\sum_{t=0}^{N} z_1^{-t} e_0(t) \right| \quad (\textit{The Apparent Limit Norm}). \end{aligned} \tag{8.48}$$

These equations shed light on the relationship between the magnitude of the apparent limit, the initial error and the value of the zero. In particular,

1. Error norms continue to reduce monotonically from iteration to iteration.
2. The apparent limit error is proportional to α_1 which can be identified with the time series $\{1, z_1^{-1}, z_1^{-2}, \ldots, z_1^{-(N-1)}, z_1^{-N}\}$.
   a. This signal decays geometrically with the most significant values being at the beginning of the time interval.
   b. The larger the value of z_1, the smaller the size of this interval.
   c. If $|z_1|$ is very large, the decay z_1^{-t} indicates that the apparent limit has significant values only in the first few samples.
3. The Cauchy-Schwarz inequality proves that $\|e_{\infty}^{pseudo}\| \leq \|e_0\|$. As a consequence, the smaller the magnitude of the initial error is, the smaller the magnitude of the apparent limit. In particular, the magnitude can be reduced by ensuring that the initial errors $e_0(t)$ are very small at the beginning of the time interval.
4. For a given norm of e_0, the norm of the apparent converged error is maximized when it is proportional to α_1. If $e_0 = \alpha_1$, then $e_0^{pseudo} = e_0$ and the convergence observed in practice will immediately become problematic.
5. Given the above and assuming zero initial conditions, the choice of $u_0 = 0$ will produce the initial error $e_0 = r$. This suggests that
   a. a reference signal $r(t)$ that is small or zero at sample points when z_1^{-t} is significant relative to unity will produce small apparent limit vector magnitudes.
   b. If, however, $|z_1|$ is close to unity then e_0^{pseudo} can still be small but the value will depend on the details of the time series $\{r(t)\}$. As a guide to what may be required, consider the case where z_1 is sufficiently close to unity so that $z_1^t \approx 1$ for $0 \leq t \leq N$. In this circumstance, the plateau will occur at a small value of the norm only if the arithmetical mean of the elements of r is around zero.

The final observation suggests that highly "active" reference signals will tend to reduce the plateau value. Also, when made possible by the conditions required by the application, it might be a good design strategy to lengthen the interval $0 \le t \le N$ to include the addition of N_I sample points to form an initial period where the reference signal is taken to be zero. The tracking of r on the original interval may then be replaced by the tracking of the new reference signal $\{0, 0, \ldots, 0, r(0), r(1), \ldots, r(N)\}$ on $0 \le t \le N + N_I$. The simple calculation with e_0 equal to the new reference gives

$$\sum_{t=0}^{N+N_I+1} z_1^{-t} e_0(t) = z_1^{-N_I} \left(\sum_{t=0}^{N+1} z_1^{-t} r(t) \right), \tag{8.49}$$

which indicates that the observed limit is reduced by a factor of approximately $z_1^{-N_I}$, a factor that can be small if N_I is large enough. For example, if $N_I = 8$ and $z_1 = 2$, then the reduction factor is $2^{-8} \approx 4 \times 10^{-3}$ which is substantial.

Intuitively, the addition of this interval adds degrees of freedom that the algorithm can use to create inputs that reduce the pseudo-limit error to smaller magnitudes more rapidly. This will be manifested by input activity on $0 \le t \le N_I$. In effect, a non-zero "initial condition" for the remaining $N+1$ samples is created to produce the practical error reduction sought.

The general ideas described above also apply when $n_+ > 1$. The following results consider the eigenvalues and eigenvectors as N increases.

Theorem 8.4 (Distinct Zeros and a Basis for $\mathscr{E}_{a+}$) *Suppose that the zeros* $\{z_j\}_{1 \le j \le n_+}$ *are distinct and that* $N \ge n_+ + 1$. *Then* $\mathscr{E}_{a+}$ *is spanned by the vectors* $\alpha_j = \alpha(z_j^{-1})$, $1 \le j \le n_+$, *where*

$$\alpha(z) = \left[1, z, z^2, \ldots, z^N\right]^T, \quad 1 \le j \le n_+. \tag{8.50}$$

Moreover, the Euclidean norm of each α_j *remains bounded as* $N \to \infty$.

Note: This defines α_j *if* z_j *is real but, if complex with* $z_k = z_j$, *the pair* α_j *and* α_k *should be replaced by the real and imaginary parts of* α_j.

Proof From Theorem 8.2 and the definition of the zeros, it is a simple matter to demonstrate that $\tilde{u}_p^T \alpha_j = 0$ so that $\alpha_j \in \mathscr{E}_{a+}$. That the $\{\alpha_j\}$ span $\mathscr{E}_{a+}$ follows as they are linearly independent when the zeros are distinct. Finally, a simple calculation gives $\overline{\alpha}_j^T \alpha_j < (1 - |z_j|^{-2})^{-1}$ for all N. □

Theorem 8.5 (Non-unit Eigenvalues when $n_+ \ge 1$) *Suppose that the zeros* $\{z_j\}_{1 \le j \le n_+}$ *of the all-pass system* G_{ap} *are distinct. Then, all non-unit eigenvalues* σ_j^2, $1 \le j \le n_+$ *of* $G_{ap} G_{ap}^T$ *(and hence* $G_{ap}^T G_{ap}$*) converge to zero as* $N \to \infty$. *In particular, defining* $z_0 = \min\{|z_1|, \ldots, |z_{n_+}|\}$, *there exists a scalar* $M_{(+)}$ *such that*

$$\max\{\sigma_1^2, \ldots, \sigma_{n_+}^2\} \le M_{(+)} z_0^{-2N} \quad \textit{for all} \quad N \ge n_+ + 1. \tag{8.51}$$

Proof It is sufficient to show that $\alpha_j^T G_{ap} G_{ap}^T \alpha_j \leq (|z_j| - 1)^{-1} |z_j|^{-2N}$, $1 \leq j \leq n_+$. Without loss of generality, take $j = 1$ and consider the input time series defined by $\alpha_1 = \{1, z_1^{-1}, \ldots, z_1^{-N}\}$. Using time reversal operators $\mathscr{T}(1, N)$, $\alpha_1^T G_{ap} G_{ap}^T \alpha_1 = (\mathscr{T}\alpha_1)^T G_{ap}^T G_{ap} (\mathscr{T}\alpha_1)$. The $\mathscr{Z}$-transform of the time series $\mathscr{T}\alpha_1$is just

$$(\mathscr{T}\alpha_1)(z) = \sum_{j=0}^{N} z^{-j} z_1^{j-N} = z_1^{-N} \sum_{j=0}^{N} z^{-j} z_1^j = z_1^{-N} \left(\frac{(1 - (\frac{z_1}{z})^{N+1})}{1 - \frac{z_1}{z}} \right). \quad (8.52)$$

and $G_{ap}\mathscr{T}\alpha_1$ is the restriction of the infinite time series on $0 \leq t \leq \infty$ generated by $G_{ap}(z)(\mathscr{T}\alpha_1))(z)$ to the interval $0 \leq t \leq N$. Defining $\tilde{G}_{ap}(z) = G_{ap}(z)\frac{z}{z - z_1}$, the pole-zero cancelation produces a stable transfer function with H_∞ norm

$$\|\tilde{G}_{ap}(z)\|_\infty = \sup_{|z|=1} \left| \tilde{G}_{ap}(z) \right| = \sup_{|z|=1} \left| \frac{z}{z - z_1} \right| = (|z_1| - 1)^{-1}. \quad (8.53)$$

Noting that the term $(\frac{z_1}{z})^{N+1}$ in $(\mathscr{T}\alpha_1)(z)$ does not contribute to the time series on $0 \leq t \leq N$, the use of discrete time reversal operators gives

$$\begin{aligned} \alpha_1^T G_{ap} G_{ap}^T \alpha_1 &= (\mathscr{T}\alpha_1)^T G_{ap}^T G_{ap} (\mathscr{T}\alpha_1) \\ &\leq z_1^{-2N} \tfrac{1}{2\pi i} \oint_{|z|=1} \left| \tilde{G}_{ap}(z) \right|^2 \tfrac{dz}{z} \leq \|\tilde{G}_{ap}(z)\|_\infty^2 z_1^{-2N} \end{aligned} \quad (8.54)$$

which is the desired result. □

The immediate conclusion from the above is that the consequences, for convergence, of having $n_+ > 1$ distinct zeros are the same as those already described for the case of $n_+ = 1$. The changes are in the technical details, but the general conclusion, for the stable inverse algorithm, is that, for large values of N (measured by the need for z_0^{-2N} to be very small), the algorithm will appear to converge to the orthogonal projection $e_0^{(1)}$ of e_0 onto $\mathscr{E}_{a+}$. If E_+ is an $(N + 1) \times n_+$ real matrix whose columns span $\mathscr{E}_{a+}$, then

$$e_0^{pseudo} = e_0^{(1)} = E_+(E_+^T E_+)^{-1} E_+^T e_0 \quad with \quad \|e_0^{(1)}\|^2 = e_0^T E_+ (E_+^T E_+)^{-1} E_+^T e_0. \quad (8.55)$$

Using the vectors $\{\alpha_j\}$ as models of such a basis, the reader can easily verify that initial errors e_0 that take small values on some interval close to the origin will have apparent convergence but to a plateau where norm values are, potentially, small.
The section is concluded by two observations

1. **Repeated Zeros**: The most useful conclusion from the analysis is expected to carry over to the more general case when some or all of the zeros in an all-pass system are repeated. It is left as an exercise for the reader to verify that, if the system has q distinct zeros z_j, one or more of which has multiplicity $n_j \geq 1$, then

$\mathscr{E}_{a+}$ is spanned by vectors, $1 \le p \le n_j$, $1 \le j \le q$, computed from the (scaled) time reversal $z_j^{-N}\mathscr{T}(1,N)\gamma_{jp}$ of the supervectors

$$\gamma_{jp} = \frac{d^{p-1}}{dz^{p-1}}\alpha(z) \quad evaluated\ \ at \quad z = z_j. \tag{8.56}$$

2. **Gradient Algorithms**: Many of the results have implications for other algorithms where the plant has non-minimum-phase zeros. For example, if $m = \ell = 1$, write the transfer function $G(z) = G_m(z)G_{ap}(z)$ where G_m is minimum-phase and G_{ap} is all-pass. In the supervector form, $G = G_m G_{ap}$. Write $GG^T = G_m G_{ap} G_{ap}^T G_m^T$ and note that, if $v_0 \in \mathscr{E}_{a+}$ and $e_0 = (G_m^T)^{-1}v_0$, then, if N is large, the norm of the product $GG^T e_0$ will be infinitesimally small relative to $\|e_0\|$. As a consequence $e_1 = (I - \beta GG^T)e_0 \approx e_0$ and convergence will be infinitesimally slow.
Note: The issue of slow convergence in $(G_m^T)^{-1}\mathscr{E}_{a+}$ will be compounded, for low pass systems by the inevitable slow convergence of the high frequency components of the error. However, whereas high frequencies are associated with rapidly varying time series such as $\{1, -1, 1, -1, \ldots\}$, the consequences of non-minimum-phase zeros are associated with slowly varying signals such as α_1 which contains substantial *low* (but still slowly converging) frequency components.

8.3 Gradients, Compensation and Feedback Design Methods

The previous sections have indicated the potentially powerful role of compensation in steepest descent algorithms and the inhibiting role of non-minimum-phase zeros on convergence rates. It also demonstrated the natural place of inverse models in Iterative Control and has suggested that good choices of compensator will, in some way, have a role that mirrors that of good approximate models of an inverse of G. Feedback design methodologies can be regarded as methods of designing structured approximations to inverse systems by ensuring good tracking of a specified reference signal. By their very nature, they must recognize the very significant place of non-minimum-phase zeros in the design process.

The following discussion is formulated using language relevant to applications to linear, time-invariant, discrete time state space systems $S(A, B, C, D)$ and, in the following section, to the continuous time case. It describes Iterative Control design as a two stage process beginning with feedback system design for the plant followed by its conversion into a steepest descent-like algorithm. Note, however, that, being formulated in operator notation, it also applies more generally although the design of feedback systems in such cases may need further development.

8.3.1 Feedback Design: The Discrete Time Case

Algorithm 8.6 (*Compensators based on Feedback Design Principles*) Suppose that Q is a constant matrix (independent of time) defining the topology in $\mathscr{Y}$. Suppose also that K_c is the forward path compensator in a unity negative feedback system for the plant G. The output from this closed-loop system is the plant output y whilst its inputs (equivalent to reference signals) are denoted by $\tilde{u}$.

The design criteria for K_c is assumed to include closed loop stability and familiar performance objectives in the frequency domain with the aim of providing some ability to track, albeit approximately, the reference signal r. Implicitly, it is assumed that the tracking is good and that the compensator ameliorates the effects of plant properties such as oscillation and loop interaction, the effects of non-minimum-phase zeros and, where thought to be of significance, to incorporate issues such as sensitivity, robustness and disturbance rejection.

Let $\mathscr{V}$ denote the input space for the closed loop system and assume that this is the Hilbert space $\mathscr{R}^{m(N+1)}$ with the same inner product as $\mathscr{Y}$ but with Q replaced by a constant, $m \times m$, symmetric, positive definite matrix Q_{T}. The closed loop operator T and sensitivity operator S of the resultant closed loop system will be denoted by

$$\mathsf{T} = (I + GK_c)^{-1}GK_c : \mathscr{V} \to \mathscr{Y} \quad \textit{and} \quad \mathsf{S} = I - \mathsf{T} = (I + GK_c)^{-1} : \mathscr{V} \to \mathscr{Y}. \tag{8.57}$$

The compensator for the compensated steepest descent algorithm is then constructed using

$$K = K_c\mathsf{S} \textit{ and } GK = GK_c\mathsf{S} = \mathsf{T}, \tag{8.58}$$

and the input update formula

$$u_{k+1} = u_k + \beta K_c\mathsf{S}\mathsf{T}^*e_k, \quad k \geq 0. \tag{8.59}$$

The resultant error evolution has the form

$$e_{k+1} = (I - \beta\mathsf{T}\mathsf{T}^*)e_k, \quad k \geq 0, \tag{8.60}$$

which takes the form of the error evolution obtained when the steepest descent algorithm is applied to the "plant" model $y = \mathsf{T}\tilde{u} + d$ tracking the reference r with initial values $\tilde{u}_0$. The consequent Iterative algorithm is convergent for all β in the range $0 < \beta\|T^*\|^2 < 2$.

The interpretation of the symbol T^* contains additional flexibility in the design process in the form of Q_{T} which plays the role of R as it was used in previous sections. With this in mind, $\|\mathsf{T}\| = \|\mathsf{T}^*\|$ and it can be bounded by

$$\|T\| \leq \|Q^{1/2}\mathsf{T}(z)Q_{\mathsf{T}}^{-1/2}\|_\infty = \sqrt{\sup_{|z|=1} r(\mathsf{T}(z)Q_{\mathsf{T}}^{-1}\mathsf{T}^{\,T}(z^{-1})Q)}. \tag{8.61}$$

The upper bound is an increasingly accurate estimate of the norm as N increases.

Design Notes: An insight into the potential benefits of a combination of feedback design and gradient iteration can be obtained by a consideration of the input-output equation $y = \mathsf{T}r$ of the feedback system. Perfect tracking requires that $\mathsf{T} = I$ which is unattainable using feedback but a well-designed feedback system typically ensures that y tracks r well except, possibly, at high frequencies. Expressed in terms of bandwidth for discrete time systems where G and K_c have state space models and associated transfer function matrices $G(z)$ and $K_c(z)$, the transfer function matrix associated with T can be denoted by $\mathsf{T}(z) = (I + G(z)K_c(z))^{-1}G(z)K_c(z)$. Good tracking often means that $\mathsf{T}(z)$ is close to the identity I over a desired bandwidth and, in the absence of inhibiting factors such as substantial high frequency resonances, it is likely that the associated operator norm $\|\mathsf{T}\| \approx 1$. A clearer picture of what is required is obtained using the frequency domain descriptions of Sect. 7.2.3. More precisely, the supervector response $(I - \beta\mathsf{T}\mathsf{T}^*)W_j(z_k)$ to the time series $z_k^t w_j(z_k)$, $0 \le t \le N$, represents the approximate time series

$$(I - \beta\mathsf{T}(z)Q_{\mathsf{T}}^{-1}\mathsf{T}^T(z^{-1})Q)z_k^t w_j(z_k), \quad 0 \le t \le N. \tag{8.62}$$

The effect of the operator $\mathsf{T}\mathsf{T}^*$ is closely linked to the values of transfer function matrix $\mathsf{T}(z)Q_{\mathsf{T}}^{-1}\mathsf{T}^T(z^{-1})Q$. The closer this matrix is to the identity and the wider the bandwidth over which this property can be achieved, the more likely it will be that frequency components in that bandwidth will be reduced by a factor of $1 - \beta$.

The discussion therefore suggests that, if K_c provides excellent feedback control of G and the dominant frequency content of the reference r lies in the bandwidth of the closed loop system T, then rapid convergence could be attained with a choice of gain $\beta \approx 1$.

Finally, the introduction of the use of feedback design potentially puts the proposed iterative compensation technique into the familiar area of classical multi-loop control systems design. Using the notation of discrete systems, the approach to algorithm design then includes the computational stages:

1. Firstly, consider the properties of $G(z)$ and any potential benefits of replacing it by and ε-weighted model.
2. Next, design a forward path compensator $K_c(z)$ for the plant $G(z)$ based on perceived needs for stability, the desired bandwidth and robustness.
3. Next, construct state space models of the systems $K(z) = K_c(z)(I + G(z)K_c(z))^{-1}$ and $\mathsf{T}(z) = G(z)K(z) = (I + G(z)K_c(z))^{-1}GK_c(z)$ and, adding in the choices of Q and Q_{T}, obtain a state space model of T^*.
4. Analyze the properties of $\mathsf{T}(z)$ and, if appropriate, re-assess the benefits of using an ε-weighted model in algorithm construction.
5. If required, assess the monotone robustness of the anticipated Iterative Control algorithm using the techniques of Sect. 7.5.3 and known, or assumed, data on anticipated multiplicative modelling errors U.
6. Finally, choose a value of β and construct the input update rule as a computational procedure that computes the response T^*e_k using reverse time simulation from zero terminal conditions and then sets $u_{k+1} - u_k$ equal to the response, from zero initial conditions, of βK to that output.

Note: The summary above could also include the use of relaxation factors as discussed in Sect. 5.2.3.

8.3.2 Feedback Design: The Continuous Time Case

The observant reader will note that the basic idea of feedback design in Iterative Control generalizes quite naturally to the case of continuous time, state space systems. The changes required lie in the detail of the feedback design and the need to use the more general Theorem 7.1 to assess convergence properties. The main issue here is that, for convergence of errors to zero, the closure $\overline{\mathscr{R}[TT^*]}$ of the range of TT^* must contain the reference signal otherwise a residual error equal to the orthogonal projection of the reference signal r onto $\overline{\mathscr{R}[TT^*]}^{\perp} = ker[T^*]$ will remain.The relevant modifications needed to include relaxation factors can be deduced from Algorithm 7.7.

8.4 Discussion and Further Reading

The concepts of inversion and gradient-based Iterative Control, although seemingly based on different basic principles, have in common the monotonic reduction of chosen signal norms. It is not surprising, therefore, that there is the possibility of bringing them together in the development of feedforward algorithms (see, for example, [85]). The key is the use of gradient methodologies as applied to compensated plant models, the benefits being that the convergence and robustness results available for steepest descent algorithms can be used on the compensated process. This compensation has two general forms: the first is the pre-conditioning of the plant by control loops that address issues of plant input-output performance such as resonance whilst the second is the (series) compensator in the Iterative Control algorithm.

Pre-conditioning seems to be essential even from the point of view of ensuring acceptable plant behaviour during each iteration, plant operation and test. However, taking the characteristics of the reference signal r into account may help in guiding the process. The form of the pre-conditioning is not specified precisely by algorithmic considerations so almost all of the many control systems design techniques developed over the past eight decades could, in principle, be used. They include classical Nyquist, Bode and root-locus analysis tools for output feedforward and feedback design but also include state feedback design approaches and many of the frequency domain approaches to the design of MIMO (multi-input, multi-output, multi-loop, multivariable, multi-channel) feedback systems. Such approaches should be influenced by the nature of the reference signal r which may allow the application of the Internal Model Principle to improve tracking performance.

The pre-conditioned plant becomes the new plant to which Iterative Control will be applied. Inverse algorithms could be applied to these without modification but could suffer from robustness issues and/or instability of the inverse (and associated sensitivity and computational problems) if the plant is non-minimum-phase. For minimum-phase systems, the simple modification described in Sect. 8.1 shows how bi-directional filtering can be included in the inverse algorithm. This leads to error evolution which has the structure of a gradient algorithm for the filter. As the filter will, typically, have simple structure, the convergence conditions will be simple to deduce as $\|F\|$ will be computable exactly off-line. The filter provides a mechanism for affecting and improving the robustness of the inverse algorithm.

Simple filters offer the potential for substantial benefits but are only the tip of the iceberg when it comes to discussing the choice of feedforward compensator. The next easily constructed step in complexity considers non-minimum-phase systems. The computational aspects for multi-loop systems is not included in the text but the benefits are clearly seen by considering the single-input, single-output case. By factoring the model transfer function into the series connection of a minimum-phase system and an all-pass network, the inverse algorithm can be modified as in Sect. 8.2.3 (see also [61]). More precisely, by only inverting the minimum-phase factor, the algorithm reduces to that of a steepest descent algorithm for the all-pass system (plus any filter included). This change removes unstable computations from the algorithm and, in the absence of filtering, leaves the range of permissible gains β unchanged. The hidden issue here is that of the rate of convergence and its relationship to the zeros and the reference signal. Section 8.2.4 provides theoretical evidence that there is, indeed, a significant effect on convergence rates. The analysis takes the form of proving that the space of error signals can be decomposed into two components: in one, the error converges to zero as expected from the inverse algorithm but, in a subspace of dimension n_+ defined by the n_+ non-minimum-phase zeros, error convergence in practice will be infinitesimally slow if N is large. This may or may not be a problem because the error norm reduces to what appears, in practice, as a plateau beyond which further reductions are so slow as to be virtually unseen. This effective minimum achievable tracking error depends on the initial error e_0 and time series/vectors defined by the zeros. This relationship throws light on the form of reference signal that reduces the norm value on the plateau and hence makes improved tracking possible. It suggests that a good strategy could be to use, when possible, reference signals augmented by extending the time interval to include an initial "lead-in phase". This adds degrees of freedom whose objective is to attain reduced plateau errors.

The non-minimum-phase analysis in this chapter has focussed on applications to stable inverse algorithms but was originally addressed for Norm Optimal Iterative Learning Control (see Chap. 9). The approach used is based on results in the text [51] on Toeplitz matrices and applied to discrete systems in [84]. A generalization to continuous systems is also available in [86] although this paper is technically more complex as the underlying spaces are infinite dimensional. The papers include comprehensive simulation and experimental results which provide a convincing verification of the physical reality of the phenomenon in practice.

More general compensation structures may suit or be necessary in some applications and further research may be needed in each case. The general principle of using compensators that are structured approximations to inverse systems naturally suggests the use of feedback methodologies. The general class of algorithms described in Sect. 8.3 have potential value as they link closely to the output feedback design techniques that may have been used in the pre-conditioning phase. In effect, the compensator is constructed from a feedback controller designed for the pre-conditioned plant. The reader is reminded that this model will often be the original plant model modified by output shifts.

MIMO control systems design methodologies are many [71, 81] and the author emphasizes his belief that there is no answer to the question "which is the best?". Each reader will have their own favorite ranging from simple process controllers, [65], to the complexities of H_∞ control [21, 23]. The development of good practice in this area will depend on the applications sector and the education, skills and preferences of the design team.

Chapter 9
Norm Optimal Iterative Learning Control

Gradient methodologies aim to reduce the magnitude of the tracking error from iteration to iteration, the ultimate objective being to minimize that tracking error. They are, as a consequence, strongly related to the idea of optimization regarded as the minimization of some objective function subject to defined constraints. This chapter creates a new perspective on Iterative Learning Control and monotonicity beginning from the construction of optimization problems that ensure monotonic reductions of errors from iteration to iteration. Emphasis is placed on wide applicability, establishing convergence properties and convergence rates, assessing the effects of parameters and characterizing robustness to modelling errors.

The approach is structured so that familiar optimization algorithms including those used in optimal control theory can be used to provide implementable solutions. For linear, discrete or continuous, state space systems, the control laws have both feedforward and state feedback representations related to the structures seen in quadratic optimal control. However, the chapter develops and describes the approach by writing the models and algorithms in operator notation and by regarding signals as points in defined Hilbert spaces. The approach is notationally compact and provides a clear, geometrical interpretation of algorithm behaviour. In addition, although it is algebraically similar to the use of transfer function notation, it has wider applicability and demonstrates that the algorithms apply to a wide class of model types and tracking objectives. The power of the approach will be seen in the following sections and chapters by their application to

1. tracking of a specified reference signal at each point on a finite time interval or
2. finding a suitable input signal to ensure that the plant output "passes through" specified points (that is, takes required values) at selected points on the time interval of interest (the, so-called, *Intermediate Point problem*).
3. The two ideas briefly described above can be combined to solve *Multi-task Tracking* problems where, together with the need to pass through intermediate points, the output is required to track reference signals on one or more subintervals and plant motion may be free on other subintervals.

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_9

These applications are possible as the plant operators G are associated with adjoints G^* with simple, state space structure and realizations that can be implemented as iterative feedback and feedforward controls.

The reader will see that this chapter describes a benchmark solution to many linear tracking problems but does not necessarily include all aspects of interest in applications. The addition of more complex design requirements such as satisfying auxiliary design objectives and "hard" input/output constraints is left for later chapters including Chaps. 11 and 12.

9.1 Problem Formulation and Formal Algorithm

In what follows, input-output dynamics are described by the familiar relationship $y = Gu + d$ with input $u \in \mathscr{U}$ and output $y \in \mathscr{Y}$ where $\mathscr{U}$ and $\mathscr{Y}$ are real Hilbert spaces. The control objective is to track a specified reference signal $r \in \mathscr{Y}$ using an Iterative Control procedure. The tracking error is $e = r - y$ and, as in previous chapters, signals f on iteration k are denoted by f_k. Iterations are initiated by an input signal u_0 generating an initial error $e_0 = r - y_0$ with output $y_0 = Gu_0 + d$ being obtained either from simulation or from experiments on the physical plant.

9.1.1 The Choice of Objective Function

Consider the problem of reducing the magnitude of a non-zero, initial error e_0 by a change $u_1 - u_0$ in u_0. The gradient methodologies of Chap. 7 produce reductions in error norm by choosing a suitable descent direction G^*e_0 and ensuring, by choice of β, that the magnitude of the change βG^*e_0 in input signal is not too large. A similar effect is obtained naturally by minimizing an objective function

$$\underbrace{J(u, u_0) = \|e\|^2_{\mathscr{Y}} + \varepsilon^2\|u - u_0\|^2_{\mathscr{U}}}_{(Objective \quad Function)}, \quad with \quad e = r - y, \quad where \quad \varepsilon^2 > 0, \tag{9.1}$$

subject to the dynamical system constraint $y = Gu + d$.

Note: *The function contains error magnitudes and the magnitude of the change in input signal. It is expressed in a quadratic form because this will lead to solutions that depend linearly on data* u_0, e_0. *The parameter* ε^2 *is added to provide a simple mechanism for varying the relative contributions of* $\|e\|^2$ *and* $\|u - u_0\|^2$ *to the objective function. Intuitively, if it is large, then the optimization will place greater emphasis on making* u *very similar to* u_0. *If it is small, greater variation in* u *is permitted to achieve a greater reduction in the error norm* $\|e\|$.

The formal way of representing this problem is to write the minimizing input u_1 as

$$u_1 = \arg\min_{u\in\mathscr{U}}\{\, J(u,u_0) \;:\; e = r - y,\;\; y = Gu + d\} \tag{9.2}$$

This problem is essentially the problem discussed in Sect. 2.6 and, in particular, using the techniques of Theorem 2.18, gives the solution $u_1 = u_0 + \varepsilon^{-2}G^*e_1$ where $e_1 = r - y_1$ and $y_1 = Gu_1 + d$. As $J(u_1,u_0) \le J(u,u_0)$ for all $u \in \mathscr{U}$, it follows, choosing $u = u_0$, that

$$\|e_1\|^2_{\mathscr{Y}} + \varepsilon^2\|u_1 - u_0\|^2_{\mathscr{U}} \le \|e_0\|^2_{\mathscr{Y}} \quad \textit{and hence} \quad \|e_1\|^2_{\mathscr{Y}} \le \|e_0\|^2_{\mathscr{Y}}, \quad \textit{for all } \varepsilon > 0, \tag{9.3}$$

equality holding if, and only if, $u_1 = u_0$. In such a situation $G^*e_0 = 0$, a condition that cannot hold if $ker[G^*] = \{0\}$.

In summary, by formulating the issue of reducing error norms as a quadratic optimal control problem in Hilbert space, the input update rule deduced from the optimization guarantees a reduction in error norm and removes the need for the use of a gain parameter β by replacing e_0 by e_1.

Regarding u_1 and e_1 as new initial signals, application of the same process will lead to a further reduction in error norm $\|e_2\|^2_{\mathscr{Y}} \le \|e_1\|^2_{\mathscr{Y}}$. This process can be continued indefinitely and, using Theorem 2.18, leads to the following general algorithm and associated properties,

Algorithm 9.1 (*Norm Optimal Iterative Learning Control*) Using the notation and terminology of the preceding discussion, the Norm Optimal Iterative Learning Control algorithm initiated by the input u_0 generates a sequence of inputs $\{u_k\}_{k\ge0}$ (and associated errors $\{e_k\}_{k\ge0}$) by computing

$$u_{k+1} = \arg\min_{u\in\mathscr{U}}\{\, J(u,u_k) \;:\; e = r - y,\;\; y = Gu + d\} \quad \textit{for} \quad k \ge 0. \tag{9.4}$$

That is, u_{k+1} is the the input that satisfies the dynamical relationships and also minimizes

$$J(u,u_k) = \|e\|^2_{\mathscr{Y}} + \varepsilon^2\|u - u_k\|^2_{\mathscr{U}}, \quad \textit{with} \quad e = r - y. \tag{9.5}$$

The inputs and errors are related by the formulae

$$\begin{aligned} u_{k+1} = u_k + \varepsilon^{-2}G^*e_{k+1},\ \textit{and hence} \quad e_{k+1} = Le_k,\ k \ge 0,\\ \textit{where the operator} \quad L = (I + \varepsilon^{-2}GG^*)^{-1}, \end{aligned} \tag{9.6}$$

is self-adjoint and positive definite with $0 < L \le I$ (a fact that follows from the monotonicity properties). In addition,

1. The error sequence $\{e_k\}_{k\ge0}$ has norms that satisfy the monotonicity conditions,

$$\|e_{k+1}\|_{\mathscr{Y}}^2 + \varepsilon^2 \|u_{k+1} - u_k\|_{\mathscr{U}}^2 \leq \|e_k\|_{\mathscr{Y}}^2, \quad \textit{and hence}, \quad \|e_{k+1}\|_{\mathscr{Y}}^2 \leq \|e_k\|_{\mathscr{Y}}^2 \tag{9.7}$$

with equality holding if, and only if, $u_{k+1} = u_k$.

2. If equality holds for some value of $k = \tilde{k}$, then $u_k = u_{\tilde{k}}$ for all $k \geq \tilde{k}$ and the algorithm converges in a finite number of iterations with $\lim_{k\to\infty} e_k = e_\infty = e_{\tilde{k}}$.
3. As $\{\|e_k\|\}_{k\geq 0}$ is positive and reducing, the limit $\lim_{k\to\infty} \|e_k\|_{\mathscr{Y}} = E_\infty$ exists with $E_\infty \geq 0$. The error converges to zero if, and only if, $E_\infty = 0$.
4. If $ker[G^*] = \{0\}$ and $e_0 \neq 0$, then $0 < \|e_{k+1}\|_{\mathscr{Y}}^2 < \|e_k\|_{\mathscr{Y}}^2$ for all $k \geq 0$.
   Proof If $\|e_{k+1}\|_{\mathscr{Y}}^2 = \|e_k\|_{\mathscr{Y}}^2$ holds for some index k, then $u_{k+1} = u_k$ and hence $G^* e_{k+1} = 0$. It follows that $e_{k+1} = e_k = 0$ and hence $u_k = u_{k-1}$ which leads to the conclusion that $e_{k-1} = 0$. An inductive argument then leads to the conclusion that $e_j = 0, \ \ 0 \leq j \leq k+1$ which contradicts the assumption that $e_0 \neq 0$. □
5. Writing $\|e_k\|_{\mathscr{Y}}^2 - \|e_{k+1}\|_{\mathscr{Y}}^2 \geq \varepsilon^2 \|u_{k+1} - u_k\|_{\mathscr{U}}^2$ for all $k \geq 0$ and adding gives

$$\|e_0\|_{\mathscr{Y}}^2 - E_\infty^2 \geq \varepsilon^2 \sum_{k=0}^{\infty} \|u_{k+1} - u_k\|_{\mathscr{U}}^2, \tag{9.8}$$

   so that $\lim_{\to\infty} \|u_{k+1} - u_k\| = 0$. That is, ultimately, the changes in input signal become infinitesimally small.
6. Finally, the minimum value of $J(u, u_k)$ is $J(u_{k+1}, u_k)$ which takes the form

$$J(u_{k+1}, u_k) = \langle e_k, (I + \varepsilon^{-2} GG^*)^{-1} e_k \rangle_{\mathscr{Y}} = \langle e_k, Le_k \rangle_{\mathscr{Y}} = \langle e_0, L^{2k+1} e_0 \rangle_{\mathscr{Y}}. \tag{9.9}$$

   Proof The derivation uses the defining relationships, the algebra

$$J(u_{k+1}, u_k) = \|e_{k+1}\|_{\mathscr{Y}}^2 + \varepsilon^2 \|\varepsilon^{-2} G^* e_{k+1}\|_{\mathscr{U}}^2 = \langle e_{k+1}, (I + \varepsilon^{-2} GG^*) e_{k+1} \rangle_{\mathscr{Y}} \tag{9.10}$$

   and the substitution $e_{k+1} = Le_k$ and $e_k = L^k e_0$. □

Note: For simplicity of presentation, the use of the label "Norm Optimal Iterative Learning Control" Algorithm will be abbreviated to the **NOILC** Algorithm.

9.1.2 Relaxed Versions of NOILC

The NOILC algorithm provides both a descent direction for $\|e\|$ and, implicitly, a step size. Relaxation techniques retain this structure but use a variety of modifications to influence algorithm properties such as convergence rates and robustness. A relaxed version of NOILC that retains the feedback and optimization structure of the basic Algorithm 9.1 has the following form

Algorithm 9.2 (*NOILC with Relaxation: Feedback Version*) Using the notation of Algorithm 9.1, the relaxed version uses the modified objective function

$$J(u, \alpha u_k) = \|e_{k+1}\|^2_{\mathscr{Y}} + \varepsilon^2 \|u - \alpha u_k\|^2_{\mathscr{U}} \quad with \quad 0 < \alpha \leq 1, \tag{9.11}$$

obtained by replacing u_k by αu_k. Minimizing this objective function leads to the relaxed input update formula

$$u_{k+1} = \alpha u_k + \varepsilon^{-2} G^* e_{k+1}. \tag{9.12}$$

Using the dynamics $y = Gu + d$ results in the error evolution

$$e_{k+1} = L\,(\alpha e_k + (1-\alpha)(r-d)) \quad where,\ again, \quad L = (I + \varepsilon^{-2} GG^*)^{-1}. \tag{9.13}$$

The feedback interpretation is motivated by the presence of e_{k+1} in the formula for u_{k+1} whilst relaxation is represented by replacing u_k by a scaled version αu_k.

An alternative approach to relaxed algorithm development is as follows,

Algorithm 9.3 (*NOILC with Relaxation: Feedforward Version*) Using the notation of Algorithm 9.1, a feedforward, relaxed version, on iteration $k+1$, computes a preliminary input signal $\tilde{u}_{k+1}$ by minimizing the NOILC objective function

$$J(u, u_k) = \|e\|^2_{\mathscr{Y}} + \varepsilon^2 \|u - u_k\|^2_{\mathscr{U}}. \tag{9.14}$$

Following this minimization, the new input u_{k+1} to be applied to the plant is computed from the relaxed, feedforward, input update formula

$$u_{k+1} = \alpha u_k + \beta(\tilde{u}_{k+1} - u_k) \ with \quad 0 < \alpha \leq 1, \ \ and \ \ 0 < \beta \leq 1 \tag{9.15}$$

incorporating relaxation defined by the parameter α plus an additional "gain" β. Using the dynamics $y = Gu + d$ yields

$$\tilde{u}_{k+1} - u_k = \varepsilon^{-2} G^* (I + \varepsilon^{-2} GG^*)^{-1} e_k \tag{9.16}$$

and hence that

$$e_{k+1} = ((\alpha - \beta)I + \beta L)e_k + (1-\alpha)(r-d) \quad where,\ again, \quad L = (I + \varepsilon^{-2} GG^*)^{-1}. \tag{9.17}$$

The relaxation parameter α plays its usual role whilst the introduction of the parameter β is motivated by its use in, for example inverse model and gradient algorithms as described in Chaps. 6 and 7. This link can be reinforced by writing

$$(\alpha - \beta)I + \beta L = \alpha I - \beta \varepsilon^{-2} GG^* (I + \varepsilon^{-2} GG^*)^{-1} \tag{9.18}$$

which is precisely the error evolution operator expected from the relaxed algorithm $u_{k+1} = \alpha u_k + \beta K_0 e_k$ with feedforward compensator $K_0 = \varepsilon^{-2} G^* (I + \varepsilon^{-2} GG^*)^{-1}$.

The behaviour of the three algorithms defined above is governed by the spectrum of the three operators L, αL and $(\alpha - \beta)I + \beta L$ which, using the Spectral Mapping Theorem, can be computed from the spectral properties of L.

The next two sections underpin the algorithm descriptions given above by putting them into a familiar state space system context. This is then followed by a general analysis of the properties of the algorithms using operator theory methods including an eigenstructure interpretation that sheds some light on the role and choice of ε.

9.1.3 NOILC for Discrete-Time State Space Systems

With the objective of setting the algorithms of the previous section into a more familiar context, the realization of Algorithm 9.1 when the plant model G takes the form of an m-output, ℓ-input, linear, time-invariant, discrete time system $S(A, B, C)$ of state dimension n, operating on a finite time interval $0 \leq t \leq N$, is considered. The model has the form

$$\begin{aligned} &x(t+1) = Ax(t) + Bu(t), \quad 0 \leq t \leq N-1, \quad \textit{with} \quad x(0) = x_0, \\ &\textit{and} \quad y(t) = Cx(t), \quad 0 \leq t \leq N. \end{aligned} \tag{9.19}$$

The initial condition x_0 is assumed to be iteration independent and the objective of Iterative Control is to find an input u_∞ that tracks, exactly, a specified reference signal r defined by the time series $r(t)$, $0 \leq t \leq N$. The model is most easily discussed by using supervector terminology and identifying the input and output spaces as $\mathscr{U} = \mathscr{R}^{\ell(N+1)}$ and $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ respectively with inner products

$$\langle u, v\rangle_{\mathscr{U}} = \sum_{j=0}^{N} u^T(t)R(t)v(t) \quad \textit{and} \quad \langle y, w\rangle_{\mathscr{Y}} = \sum_{j=0}^{N} y^T(t)Q(t)w(t) \tag{9.20}$$

where the time varying weight matrices $R(t) = R^T(t) > 0$ and $Q(t) = Q^T(t) > 0$ for $t = 0, 1, 2, \ldots, N$.
With the above definitions, Algorithm 9.1 becomes

Algorithm 9.4 (*NOILC for Discrete Time, State Space Systems*) Using the notation given above, suppose that, on iteration k, the input u_k was used and generated output and error time series y_k and $e_k = r - y_k$. NOILC Algorithm 9.1 then constructs the input time series $u_{k+1}(t)$, $0 \leq t \leq N$, to be used on iteration $k+1$ as the one that minimizes the quadratic objective function

$$\begin{aligned} &J(u, u_k) \\ &= \sum_{j=0}^{N} \Big((r(t) - y(t))^T Q(t)(r(t) - y(t)) + \varepsilon^2 (u(t) - u_k(t))^T R(t)(u(t) - u_k(t)) \Big). \end{aligned} \tag{9.21}$$

This input is applied to the plant to generate data y_{k+1} and e_{k+1}, the index is updated and the process repeated indefinitely.

The optimal control problem is that of Sect. 4.7 with $R(t)$ replaced by $\varepsilon^2 R(t)$. The solution can be implemented in either a feedback or feedforward form.

Two Feedback Implementations: Two feedback implementations can be considered. In the feedback *Implementation One*, the control signal has the form

$$u_{k+1}(t) = u_k(t) + \varepsilon^{-2} R^{-1}(t) B^T \left(-K(t) x_{k+1}(t) + \xi_{k+1}(t)\right), \quad 0 \le t \le N, \quad (9.22)$$

where $x_{k+1}(t)$ is the measured state vector at time t on iteration $k+1$. In addition,

1. the (iteration independent) $n \times n$ state feedback matrices $K(t)$, $0 \le t \le N$ are computed off-line before initialization of the iterative procedure using the nonlinear recursion

$$\begin{aligned} \tilde{K}(t+1) &= A^T K(t+1) + C^T Q(t+1) C, \\ K(t) &= \left(I + \varepsilon^{-2} \tilde{K}(t+1) B R^{-1}(t) B^T\right)^{-1} \tilde{K}(t+1) A \end{aligned} \quad (9.23)$$

   starting from the terminal condition $K(N) = 0$.
2. The predictive term $\xi_{k+1}(t)$, $0 \le t \le N$, is computed from the recursion, $0 \le t \le N-1$,

$$\begin{aligned} \xi_{k+1}(t) &= \left(I + \varepsilon^{-2} \tilde{K}(t+1) B R^{-1}(t) B^T\right)^{-1} \psi_{k+1}(t), \qquad \textit{where}, \\ \psi_{k+1}(t) &= A^T \xi_{k+1}(t+1) - \tilde{K}(t+1) B u_k(t) + C^T Q(t+1) r(t+1), \end{aligned} \quad (9.24)$$

   beginning with the terminal condition $\xi_{k+1}(N) = 0$.

Note that, whereas the state feedback matrices are computed only once and used on all iterations, it is necessary to compute the predictive term ξ_{k+1} for each iteration. It is computed off-line, using a plant model, in the time between iteration k and $k+1$ when the system is being reset for the next iteration. The feedback term provides performance data from the current state of the system on iteration $k+1$ whilst the term ξ_{k+1} feeds information forward from iteration k.

A alternative approach uses e_k in the calculations. An approach to this is deduced by writing the input update formula in the form

$$u_{k+1} - u_k = \varepsilon^{-2} G^* e_{k+1} = \varepsilon^{-2} G^* L e_k. \quad (9.25)$$

The second term $\varepsilon^{-2} G^* L e_k$ is identical to the control input computed using one iteration of NOILC for the system $y - y_k = G(u - u_k)$ from the starting condition of zero input ($u = u_k$) and a reference signal equal to e_k. Noting that the state for the system is $x(t) - x_k(t)$, this leads to what will be called the feedback *Implementation Two* which generates the input signal using

$$u_{k+1}(t) = u_k(t) + \varepsilon^{-2} R^{-1}(t) B^T \left(-K(t) \underbrace{(x_{k+1}(t) - x_k(t))}_{} + \xi_{k+1}(t) \right), \quad (9.26)$$

where the feedback gain $K(t)$ now operates on the difference $x_{k+1}(t) - x_k(t)$ in state vectors between iterations. The matrices $\{K(t)\}_{0 \le t \le N}$ remain unchanged, but the predictive term $\xi_{k+1}(t)$, $0 \le t \le N$, is computed from the modified recursion, $0 \le t \le N-1$,

$$\begin{aligned} \xi_{k+1}(t) &= \left(I + \varepsilon^{-2} \tilde{K}(t+1) B R^{-1}(t) B^T \right)^{-1} \psi_{k+1}(t), \qquad \textit{with}, \\ \psi_{k+1}(t) &= A^T \xi_{k+1}(t+1) + C^T Q(t+1) \underbrace{e_k(t+1)}_{}, \qquad \xi_{k+1}(N) = 0. \end{aligned} \quad (9.27)$$

The two feedback implementations differ in their use of data. Both implementations use current iteration data in the form of $x_{k+1}(t)$ with previous iteration performance being represented by $u_k(t)$ and, in Implementation Two, via the state $x_k(t)$ and the presence of the error $e_k(t)$ in the equations for ψ_{k+1}. Intuitively, Implementation Two contains the greatest link to iteration k but, mathematically, the two are identical in the absence of modelling errors.

Feedforward Implementation: Given the data r, u_k, e_k, the *feedforward implementation* of NOILC for discrete time state space systems is simply stated as the use of models to calculate the new input u_{k+1}, off-line, in the time between iterations k and $k+1$. These calculations can be approached using the formulae given for the feedback case above. Following its evaluation, u_{k+1} is applied to the plant and new data e_{k+1} obtained from sensors to allow iterations to continue.

There is, however, an important practical issue implicit in these comments, namely, that computations using the first feedback implementation in the form described above use only r and u_k to generate u_{k+1}. In a feedforward computation, this calculation therefore ignores the actual behaviour of the plant as there is no feedback from the previously observed error e_k. In effect, the algorithm is ignoring the external reality which plays no role in the computations at all! In contrast, the use of equations for *Implementation Two* includes measurements of the error data $e_k(t)$ and provides the necessary link.

9.1.4 Relaxed NOILC for Discrete-Time State Space Systems

Continuing with the discussion of the previous section, the relaxed versions of the algorithm has two forms.

Feedback Relaxation: Using the feedback relaxation Algorithm 9.2, a relaxed version of Algorithm 9.4 gives the input update in the form $u_{k+1} = \alpha u_k + \varepsilon^{-2} G^* e_{k+1}$.

1. Using the equations for *Implementation One*, a feedback implementation is obtained by modifying the equations for ψ_{k+1} by the substitution

$$u_k(t) \;\mapsto\; \alpha u_k(t). \tag{9.28}$$

2. For *Implementation Two*, the dynamics are written in the form $y - \alpha y_k = G(u - \alpha u_k) + (1-\alpha)d$ which has a state space model $S(A, B, C)$ with input $u(t) - \alpha u_k(t)$, output $y(t) - \alpha y_k(t)$, state $x(t) - \alpha x_k(t)$ and initial condition $(1 - \alpha)x_0$. Writing the performance index in the form

$$\begin{aligned} J(u, u_k) &= \|r - \alpha y_k - (y - \alpha y_k)\|_{\mathscr{Y}}^2 + \varepsilon^2 \|u - \alpha u_k\|_{\mathscr{U}}^2, \quad \textit{gives} \\ u_{k+1}(t) - u_k(t) &= \varepsilon^{-2} R^{-1}(t) B^T \left(-K(t) \underbrace{(x_{k+1}(t) - \alpha x_k(t))}_{} + \xi_{k+1}(t) \right), \end{aligned} \tag{9.29}$$

where $\{\xi_{k+1}(t)\}$ comes from the equations for ψ_{k+1} using the substitutions

$$u_k(t) \mapsto 0 \quad \textit{and} \quad r(t) \mapsto r - \alpha y_k. \tag{9.30}$$

This gives the modified equation, with $\xi_{k+1}(N) = 0$,

$$\begin{aligned} \xi_{k+1}(t) &= \left(I + \varepsilon^{-2} \tilde{K}(t+1) B R^{-1}(t) B^T \right)^{-1} \psi_{k+1}(t), \qquad \textit{with}, \\ \psi_{k+1}(t) &= A^T \xi_{k+1}(t+1) + C^T Q(t+1) \underbrace{(r(t+1) - \alpha y_k(t+1))}_{}. \end{aligned} \tag{9.31}$$

Feedforward Relaxation Using the feedback relaxation Algorithm 9.3, a relaxed version of Algorithm 9.4 is obtained by computing, off-line, the feedforward implementation value of the new input, denoting it by $\tilde{u}_{k+1}$, and subsequently computing the new input to be applied to the plant from the formula

$$u_{k+1}(t) = \alpha u_k(t) + \beta \left(\tilde{u}_{k+1}(t) - u_k(t) \right), \quad \textit{for} \quad 0 \le t \le N. \tag{9.32}$$

9.1.5 A Note on Frequency Attenuation: The Discrete Time Case

Error evolution is governed by the properties of the operator L. The most precise physical interpretation of this uses eigenvector and eigenvalue analysis but the robustness analysis of later sections will link robustness properties to the frequency domain and the transfer function matrix $G(z)$. The methodology and notation of Sect. 7.2.3 can be applied to the equation $e_{k+1} = Le_k$ in the frequency domain by supposing that

$e_k = W_j(z_k)$ and noting that e_{k+1} can then be approximated, if N is large, by

$$e_{k+1} \approx (1 + \varepsilon^{-2}\sigma_j^2(z_k))^{-1} W_j(z_k), \tag{9.33}$$

which links the evolution of individual frequency components in the error in terms of the eigenstructure of $G(z)R^{-1}G^T(z^{-1})Q$. It is an approximation as initial and terminal conditions for G and G^* are neglected. It does however provide support for the intuitive idea that individual frequency components are influenced by the frequency domain properties of $G(z)$. In particular, it suggests that

1. for low pass systems, high frequency error components are attenuated slowly,
2. frequency content in the vicinity of resonances are attenuated more severely than other frequency ranges and,
3. if, $G(z)$ has a zero close to the unit circle, frequencies close to that zero will be attenuated very slowly.

Future development to choice of weights Q and R could be based on their effects on the eigenvalues $\sigma_j^2(z_k)$ (see Theorem 9.20).

9.1.6 NOILC: The Case of Continuous-Time State Space Systems

The form of the NOILC algorithm for a linear, time-invariant, m-output, ℓ-input, continuous time system $S(A, B, C)$ is constructed in much the same way as that for the discrete time case in Sects. 9.1.3 and 9.1.4. The dynamics on $[0, T]$ are

$$\begin{aligned} &\dot{x}(t) = Ax(t) + Bu(t), && \textit{with} \quad x(0) = x_0, \\ \textit{and} \quad &y(t) = Cx(t), && 0 \le t \le T. \end{aligned} \tag{9.34}$$

The initial condition x_0 is assumed to be iteration independent and the objective of Iterative Control is to find an input u_∞ that tracks, exactly, a specified reference signal $r \in \mathscr{Y}$ defined by the vector function $r(t) \in \mathscr{R}^m$, $0 \le t \le T$. The input and output spaces are $\mathscr{U} = L_2^\ell[0, T]$ and $\mathscr{Y} = L_2^m[0, T]$ with inner products

$$\langle u, v\rangle_{\mathscr{U}} = \int_0^T u^T(t)R(t)v(t)dt \quad \textit{and} \quad \langle y, w\rangle_{\mathscr{Y}} = \int_0^T y^T(t)Q(t)w(t)dt \tag{9.35}$$

where the matrices $R(t) = R^T(t) > 0$ and $Q(t) = Q^T(t) > 0$ for all $t \in [0, T]$.

Algorithm 9.5 (*NOILC for Continuous Time, State Space Systems*) Using the notation given above, suppose that, on iteration k, the input u_k was used and generated a measured output and error signals y_k and $e_k = r - y_k$. NOILC Algorithm 9.1 then constructs the input signal $u_{k+1}(t)$, $0 \le t \le T$, to be used on iteration $k + 1$ as the one that minimizes the quadratic objective function

$$J(u, u_k) = \int_0^T ((r(t) - y(t))^T Q(t)(r(t) - y(t)) + \varepsilon^2 (u(t) - u_k(t))^T R(t)(u(t) - u_k(t))dt. \quad (9.36)$$

This input is then applied to the plant to generate data y_{k+1} and e_{k+1}, the index is updated and the process repeated indefinitely.

The computations are precisely those needed to solve the optimization problem discussed in Sect. 3.10 with $R(t)$ replaced by $\varepsilon^2 R(t)$. By transforming the input update rule $u_{k+1} = u_k + \varepsilon^{-2} G^* e_{k+1}$ into a two-point boundary value problem, the solution can be implemented in either a feedback or feedforward form.

Feedback Implementation: There are two feedback implementations. In the feedback *Implementation One*, the results of Sect. 3.10.4 indicate that the control signal has the form

$$u_{k+1}(t) = u_k(t) + \varepsilon^{-2} R^{-1}(t) B^T \left(-K(t) x_{k+1}(t) + \xi_{k+1}(t)\right), \quad 0 \le t \le T, \quad (9.37)$$

where $x_{k+1}(t)$ is the measured state vector at time t on iteration $k + 1$. In addition,

1. the (iteration independent) $n \times n$, time varying, state feedback matrix $K(t),\ 0 \le t \le T$, is computed off-line before initialization of the iterative procedure by solving the nonlinear matrix differential equation

$$\frac{dK(t)}{dt} + A^T K(t) + K(t)A - \varepsilon^{-2} K(t) B R^{-1}(t) B^T K(t) + C^T Q(t) C = 0 \quad (9.38)$$

   with the terminal condition $K(T) = 0$.
2. The predictive term $\xi_{k+1}(t),\ 0 \le t \le T$, is computed from the terminal condition $\xi_{k+1}(T) = 0$ and the differential equation on $[0, T]$,

$$\frac{d\xi_{k+1}(t)}{dt} = -\left(A^T - \varepsilon^{-2} K(t) B R^{-1}(t) B^T\right) \xi_{k+1}(t) - C^T Q(t) r(t) + K(t) B u_k(t). \quad (9.39)$$

Again, $K(t)$ is computed only once and used on each and every iteration but it is necessary to recompute the predictive term ξ_{k+1} for each iteration. It is computed off-line in the time between iteration k and $k + 1$ when the system is being reset.

In feedback *Implementation Two*, the ideas expressed in Sect. 9.1.3 can again be applied to replace the control law by

$$u_{k+1}(t) = u_k(t) + \varepsilon^{-2} R^{-1}(t) B^T \left(-K(t)(x_{k+1}(t) - x_k(t)) + \xi_{k+1}(t)\right), \quad (9.40)$$

where, using the same terminal condition,

$$\frac{d\xi_{k+1}(t)}{dt} = -\left(A^T - \varepsilon^{-2} K(t) B R^{-1}(t) B^T\right) \xi_{k+1}(t) - C^T Q(t) e_k(t). \quad (9.41)$$

Intuitively, this version will suit applications better as ξ_{k+1} responds directly to the measured error data rather than the input used.

Feedforward Implementation: Given the data r, u_k, e_k, the feedforward implementation of NOILC for continuous time state space systems is simply stated as the use of models to calculate the new input u_{k+1}, off-line, in the time between iterations k and $k+1$. These calculations can be approached using the formulae given for the feedback case above. Following its evaluation, u_{k+1} is applied to the plant and new data e_{k+1} obtained from sensors to allow iterations to continue. The discussion of the discrete case in Sect. 9.1.3 applies here. That is, in order to make the algorithm respond, explicitly, to the previously observed error e_k, the computations associated with Implementation Two seem to be the most appropriate.

Finally, the relaxed versions of the algorithm are simply stated as follow:

Feedback Relaxation: Using the feedback relaxation Algorithm 9.2 and a similar analysis to that of Sect. 9.1.4, a relaxed version of Algorithm 9.5 is obtained in two forms. *Implementation One* replaces u_k by αu_k to give

$$\begin{aligned} u_{k+1}(t) &= \alpha u_k(t) + \varepsilon^{-2} R^{-1}(t) B^T \left(-K(t) x_{k+1}(t) + \xi_{k+1}(t)\right), \quad \textit{and} \\ \frac{d\xi_{k+1}(t)}{dt} &= -\left(A^T - \varepsilon^{-2} K(t) B R^{-1}(t) B^T\right) \xi_{k+1}(t) - C^T Q(t) r(t) + \alpha K(t) B u_k(t) \end{aligned} \tag{9.42}$$

with the terminal condition $\xi_{k+1}(T) = 0$. In contrast, *Implementation Two* uses

$$\begin{aligned} u_{k+1}(t) &= \alpha u_k(t) + \varepsilon^{-2} R^{-1}(t) B^T \left(-K(t)(x_{k+1}(t) - x_k(t)) + \xi_{k+1}(t)\right), \quad \textit{and} \\ \frac{d\xi_{k+1}(t)}{dt} &= -\left(A^T - \varepsilon^{-2} K(t) B R^{-1}(t) B^T\right) \xi_{k+1}(t) - C^T Q(t) \left(r(t) - \alpha y_k(t)\right) \end{aligned} \tag{9.43}$$

which is driven by the measured output y_k.

Feedforward Relaxation: Using the feedback relaxation Algorithm 9.3, a relaxed version of Algorithm 9.5 is obtained by computing, off-line, the input $\tilde{u}_{k+1}$ generated by Implementation Two of Algorithm 9.5. The new input to be applied to the plant is then obtained from the formula

$$u_{k+1}(t) = \alpha u_k(t) + \beta \left(\tilde{u}_{k+1}(t) - u_k(t)\right), \quad \textit{for} \quad 0 \le t \le T. \tag{9.44}$$

9.1.7 Convergence, Eigenstructure, ε^2 and Spectral Bandwidth

Sections 9.1.3 and 9.1.6 provide simple computations for practical implementations. They carry no obvious information on the effects of the reference signal r or the

objective function weights Q and R on algorithm performance. There is no explicit relationship available but a useful insight can be obtained by assuming an eigenstructure for GG^* and examining the convergence in terms of the evolution of the eigenvector components of the error. A sufficient condition for such an eigenstructure to exist is that $\mathscr{Y}$ is finite dimensional as, for example, in the case of discrete state space models.

Section 5.2.4 has revealed the potential power of eigenstructure in iterative analysis. The main value of the results was in increasing the understanding of iteration behaviour. The techniques are not aimed at practical computation as, in practice, computation of the eigenvalues and eigenvectors of high or infinite dimensional operators is a difficult or impossible task. In this section, the assumption of an eigenstructure for GG^* is used to create a greater understanding of the "internal dynamics" of the NOILC algorithms. Note that

$$\mathscr{Y} = ker[G^*] \oplus \overline{\mathscr{R}[GG^*]} \tag{9.45}$$

is an orthogonal subspace decomposition of $\mathscr{Y}$ and $ker[G^*]$ is the eigenspace of GG^* corresponding to zero eigenvalues. By construction, GG^* maps $\overline{\mathscr{R}[GG^*]}$ into the dense subspace $\mathscr{R}[GG^*]$ of $\overline{\mathscr{R}[GG^*]}$. Assume that $GG^* : \overline{\mathscr{R}[GG^*]} \to \overline{\mathscr{R}[GG^*]}$ has eigenvalues $\{\sigma_j^2\}_{j\geq 1}$ with corresponding orthonormal eigenvectors $\{v_j\}_{j\geq 1}$ that span $\overline{\mathscr{R}[GG^*]}$. Then, by construction, $\sigma_j^2 > 0, \;\; j \geq 1$ and, by reordering,

$$\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2 \geq \cdots \quad (\textit{where multiplicities are permitted}). \tag{9.46}$$

As GG^* is self-adjoint,

$$r(GG^*) = \|GG^*\| = \|G^*\|^2 = \sigma_1^2. \tag{9.47}$$

Consider now the operator $L = (I + \varepsilon^{-2}GG^*)^{-1}$ and note that $Le = e$ for all $e \in ker[G^*]$ and that $Lv_j = (1 + \varepsilon^{-2}\sigma_j^2)^{-1}v_j$ for all $j \geq 1$.

Consider Algorithm 9.1 with initial error e_0 decomposed into the sum $e_0 = e_0^{(1)} + e_0^{(2)}$ with $e_0^{(1)} \in ker[G^*]$ and $e_0^{(2)} \in \overline{\mathscr{R}[GG^*]}$. Then, writing $e_0^{(2)} = \sum_{j\geq 1} \gamma_j v_j$,

$$\begin{aligned} &\|e_0^{(2)}\|_{\mathscr{Y}}^2 = \sum\nolimits_{j\geq 1} \gamma_j^2, \quad \textit{so that} \quad \lim_{j\to\infty} \gamma_j = 0 \\ &\quad \textit{and} \qquad e_k = L^k e_0 = e_0^{(1)} + \sum\nolimits_{j\geq 1} \underbrace{(1 + \varepsilon^{-2}\sigma_j^2)^{-k}\gamma_j v_j}. \end{aligned} \tag{9.48}$$

The simple observations suggested by this analysis include,

1. The component of e_0 in $ker[G^*]$ remains unchanged from iteration to iteration.
2. The contribution of the eigenvector v_j decreases in significance from iteration to iteration at a rate governed by the power rule $(1 + \varepsilon^{-2}\sigma_j^2)^{-k}, \;\; k \geq 0$. Hence, defining $\sigma_\infty^2 = \inf_{j\geq 1} \sigma_j^2$,

a. if $\sigma_\infty^2 > 0$, then $\|e_k - e_0^{(1)}\| \leq (1+\varepsilon^{-2}\sigma_\infty^2)^{-k}\|e_0 - e_0^{(1)}\|$ so that e_0 converges to the orthogonal projection $e_0^{(1)}$ of e_0 onto $\overline{\mathscr{R}[GG^*]}$.
b. If $\sigma_\infty^2 = 0$, the convergence properties are retained but the magnitude of the contribution of eigenvectors corresponding to eigenvalues $\sigma_j^2 \ll \varepsilon^2$ reduces infinitesimally slowly. This observation provides insight into the effect of the reference signal. More precisely, rapid convergence (faster than some power rule $(1+\delta^2)^{-k}$, with $\delta^2 > 0$) to small errors will only occur if $e_0^{(1)} = 0$ and r is dominated by the contribution of eigenvectors with eigenvalues σ_j^2 similar in magnitude to or greater than $\varepsilon^2\delta^2$. It is important to note that the number of reference signals in this set increases as ε^2 reduces suggesting that fast convergence is then achieved for a wider class of reference signals.
c. In many situations, the eigenvectors corresponding to very small eigenvalues will be associated with "high frequency" characteristics of the plant. This idea is supported by the state space example in Sect. 5.2.5 where eigenfunctions are associated with terms in a Fourier series representation. This can be a good guide to physical behaviour but it does not tell the whole story as is seen by noting, Sect. 8.2.4, that non-minimum-phase behaviours are associated with infinitesimally small eigenvalues but are not a high frequency phenomenon.
d. The signal with the fastest convergence to zero are those in the subspace spanned by eigenvectors with eigenvalues equal to $\sigma_1^2 = \|GG^*\|$. That is, the choice of inner products in $\mathscr{Y}$ and $\mathscr{U}$ (which influence the form of G^*) also influence convergence rates.

3. Finally, the norm and spectral radius can be computed as follows,

a. $r(L) = \|L\| = 1$ if $ker[G^*] \neq \{0\}$.
b. If $ker[G^*] = \{0\}$, then $r(L) = \|L\| = (1+\varepsilon^{-2}\sigma_\infty^2)^{-1}$.

In case (b), if $\sigma_\infty^2 > 0$, the convergence of the algorithm to zero error is guaranteed. If, however, $\sigma_\infty^2 = 0$ then the reader can verify that $e_k \to 0$ in the weak topology in $\mathscr{Y}$. A proof of convergence in norm is left for the next section.

These observations indicate a link between any eigenstructure of GG^* and convergence rates. In the following paragraphs, a simple parametric characterization of the link between convergence and the spectrum of GG^* is suggested as an aid to design discussions and the choice of ε^2.

Definition 9.1 (*The Concept of Spectral Bandwidth*) Suppose that GG^* has an eigenstructure as discussed in the preceding paragraphs and that $e_0 = \sum_{j\geq 1} \gamma_j v_j$. Then, given two real numbers λ and μ in the half-open interval $(0, 1]$, the NOILC Iterative Control algorithm is said to have a Spectral Bandwidth $S_{BW}(\lambda, \mu)$ if, and only if, the contribution, to the error signal, of all eigenvectors v_j with eigenvalues $\sigma_j^2 \geq \lambda\|G^*\|^2$ decay at a rate bounded from above by the geometric sequence $\mu^k\gamma_j$.

As $e_k = L^k e_0$, it follows that $e_k = \sum_{j\geq 1} (1+\varepsilon^{-2}\sigma_j^2)^{-k}\gamma_j v_j$ and hence a spectral bandwidth $S_{BW}(\lambda, \mu)$ is achieved if

$$(1 + \varepsilon^{-2}\lambda\|G^*\|^2)^{-1} \leq \mu \quad \textit{for the chosen} \quad \varepsilon^2 > 0. \tag{9.49}$$

This condition has application to the choice of ε^2 for a given choice of λ and μ.

1. If attention is focussed on the convergence rate of σ_1^2 only, then $\lambda = 1$. If it is seen as desirable that this eigen-component at least halves in magnitude from iteration to iteration, then it must converge faster than $\left(\frac{1}{2}\right)^k$ and the choice of $\mu = 0.5$ is appropriate. The spectral bandwidth $S_{BW}(1, 0.5)$ is then achieved for any choice of $0 < \varepsilon^2 \leq \|G^*\|^2$.
2. If attention is focussed on the convergence rate of eigenvectors with eigenvalues greater than $\frac{1}{2}\sigma_1^2$ only, then $\lambda = 0.5$. If it is seen as desirable that all such eigen-components converge faster than $\left(\frac{1}{2}\right)^k$, choose $\mu = 0.5$. The spectral bandwidth $S_{BW}(0.5, 0.5)$ is then achieved for any choice of $0 < \varepsilon^2 \leq 0.5\|G^*\|^2$.
3. If attention is focussed on the convergence rate of eigenvectors with eigenvalues greater than $\frac{1}{2}\sigma_1^2$ only, then $\lambda = 0.5$. If it is seen as desirable that all such eigen-components converge faster than $\left(\frac{1}{3}\right)^k$, choose $\mu = 0.33$. The spectral bandwidth $S_{BW}(0.5, 0.33)$ is then achieved for any choice of $0 < \varepsilon^2 \leq 0.25\|G^*\|^2$.

Note that achieving a specific spectral bandwidth $S_{BW}(\lambda, \mu)$ does not imply that error convergence satisfies $\|e_k\| \leq \mu^k\|e_0\|$. The precise form of convergence depends on the initial error e_0 and hence r and u_0. Loosely speaking, such convergence is achieved approximately for any e_0 whose eigenvector representation is dominated by eigenvectors with eigenvalues in the range $\lambda\|G^*\|^2 \leq \sigma_j^2 \leq \|G^*\|^2$. A more precise description is obtained by assuming that a spectral bandwidth $S_{BW}(\lambda, \mu)$ has been achieved and then writing

$$e_0 = \sum_{\sigma_j^2 \geq \lambda\|G^*\|^2} \gamma_j v_j + \sum_{\sigma_j^2 < \lambda\|G^*\|^2} \gamma_j v_j = e_0^{(1)} + e_0^{(2)} \tag{9.50}$$

with the natural identification of terms. It follows that,

$$\|e_k\|_{\mathscr{Y}} \leq \mu^k\|e_0^{(1)}\|_{\mathscr{Y}} + \|e_0^{(2)}\|_{\mathscr{Y}} \tag{9.51}$$

and hence that the error sequence convergence can be thought of as an initial convergence following the power law μ^k to a closed ball centred on the origin of radius $\|e_0^{(2)}\|_{\mathscr{Y}}$. This is followed by slower convergence to its final limit. If $\|e_0^{(2)}\|_{\mathscr{Y}} \ll \|e_0\|_{\mathscr{Y}}^2$ then the algorithm achieves accurate tracking quickly. Note that reducing λ with μ fixed will (a) require a reduction in the value of ε^2 and (b) include more terms in the expression for $e_0^{(1)}$ and hence reduce $\|e_0^{(2)}\|_{\mathscr{Y}}$.

Finally, a good choice of ε^2 is clearly related to desired convergence properties and the norm $\|G^*\|^2$. For asymptotically stable, linear, time invariant, discrete, state space systems $S(A, B, C, D)$ for example, previous calculations in Sect. 4.8.2 have proved that $\|G^*\|^2 = \|G\|^2$ and provided a bound described by the spectral radius $r\left(R^{-1}G^T(z^{-1})QG(z)\right)$ that is accurate if N is large.

9.1.8 Convergence: General Properties of NOILC Algorithms

The previous section has demonstrated the power of an eigenstructure interpretation of algorithm behaviour. However, the form of L and the fact that it is self adjoint makes it possible to make useful statements about its spectral radius and norm and hence the convergence properties of the algorithm without appealing to or assuming any eigenstructure. These relationships are stated as formal theorems in what follows. The first defines useful properties of L as follows,

Theorem 9.1 (General Properties of $L = (I+\varepsilon^{-2}GG^*)^{-1}$) *With the notation defined above, the operator L in the NOILC Algorithm 9.1 satisfies the conditions*

$$ker[L] = \{0\} \quad and \quad 0 < (1 + \varepsilon^{-2}\|G^*\|^2)^{-1}I \leq L \leq I. \tag{9.52}$$

In particular,

1. *If there exists a scalar $\varepsilon_0 > 0$ such that $GG^* \geq \varepsilon_0^2 I$, then*

$$(1 + \varepsilon^{-2}\|G^*\|^2)^{-1}I \leq L \leq (1 + \varepsilon^{-2}\varepsilon_0^2)^{-1}I \ < I. \tag{9.53}$$

2. *If no such value of ε_0 exists, then $L < I$ if, and only if, $ker[G^*] = \{0\}$.*
3. *$Le_0 = e_0$ if, and only if, $e_0 \in ker[G^*]$. That is, there exists a non-zero initial error e_0 such that no improvement in tracking error is possible using NOILC if, and only if, $ker[G^*] \neq \{0\}$.*

Proof Using the NOILC interpretation, denote $e_1 = Le_0$. If $e_0 \neq 0$ and $e_1 = Le_0 = 0$, then $u_1 \neq u_0$ and $J(u_1, u_0) = \langle e_0, Le_0\rangle = \varepsilon^2\|u_1 - u_0\|^2 > 0$ which is not possible. Hence $e_0 = 0$ and $ker[L] = \{0\}$ as required. Next, note that $\|e_1\| \leq \|e_0\|$ for all $e_0 \in \mathscr{Y}$. It follows that $\|L\| \leq 1$. That is, as L is self-adjoint, $L \leq \|L\|I \leq I$ as required. Also, $J(u_1, u_0) = \langle e_0, Le_0\rangle > 0$ for all $e_0 \neq 0$ and hence $L > 0$. Next, let $H = H^*$ be a self-adjoint, positive operator and denote the unique positive-definite, self-adjoint, square root of $(I + H)^{-1}$ by X. Then

$$\begin{aligned}(I + H)^{-1} - (I + \|H\|)^{-1}I &= (I + H)^{-1}(I + \|H\|)^{-1}\left(\|H\|I - H\right) \\ &= (I + \|H\|)^{-1}X\left(\|H\|I - H\right)X \geq 0\end{aligned} \tag{9.54}$$

as $\|H\|I - H \geq 0$. Choosing $H = \varepsilon^{-2}GG^*$ (respectively, $H = \varepsilon^{-2}G^*G$) then gives

$$\begin{array}{rl} & (I + \varepsilon^{-2}GG^*)^{-1} - (I + \varepsilon^{-2}\|G^*\|^2)^{-1}I \geq 0 \\ (resp. & (I + \varepsilon^{-2}G^*G)^{-1} - (I + \varepsilon^{-2}\|G\|^2)^{-1}I \geq 0) \end{array} \tag{9.55}$$

as $\|GG^*\| = \|G^*\|^2$ and $\|G^*G\| = \|G\|^2$. Now, consider $I - L$ in the form

$$\begin{aligned} &I - L = (I + \varepsilon^{-2}GG^*)^{-1}\varepsilon^{-2}GG^* = G(I + \varepsilon^{-2}G^*G)^{-1}\varepsilon^{-2}G^*, \quad and \ hence \\ &\varepsilon^{-2}\|G^*e_0\|_{\mathscr{U}}^2 \geq \langle e_0, (I - L)e_0\rangle_{\mathscr{Y}} \geq (I + \varepsilon^{-2}\|G\|^2)^{-1}\varepsilon^{-2}\|G^*e_0\|_{\mathscr{U}}^2 \end{aligned} \tag{9.56}$$

which proves that $Le_0 = e_0$ if, and only if, $e_0 \in ker[G^*]$ and also that $L < I$ if $ker[G^*] = \{0\}$.
Finally, suppose that $GG^* \geq \varepsilon_0^2 I$ and that $e_0 \neq 0$. Again writing $H = GG^*$,

$$(I + \varepsilon^{-2}H)^{-1} - (I + \varepsilon^{-2}\varepsilon_0^2)^{-1}I = (I + \varepsilon^{-2}H)^{-1}(I + \varepsilon^{-2}\varepsilon_0^2)^{-1}\varepsilon^{-2}\left(\varepsilon_0^2 I - H\right)$$
$$\leq 0 \tag{9.57}$$

which completes the proof of the result. □

Discussion: The result has a number of useful consequences,

1. The property $ker[L] = \{0\}$ indicates that, if $e_0 \neq 0$, all following $e_k \neq 0$.
2. The subspace $ker[G^*]$ plays an important role in convergence properties. Monotonicity of the error norm sequence becomes strict monotonicity, $\|e_{k+1}\| < \|e_k\|, \quad k \geq 0$ if $ker[G^*] = \{0\}$. However, if $ker[G^*] \neq \{0\}$, then there exists initial errors for which no change in error can be achieved.
3. As $ker[G^*] = \mathscr{R}[G]^{\perp}$, the condition $ker[G^*] = \{0\}$ implies that the range of G is dense in $\mathscr{Y}$. More generally, using the fact that $ker[G^*] = ker[GG^*]$,

$$\mathscr{Y} = \overline{\mathscr{R}[G]} \oplus ker[G^*] \quad and \quad \overline{\mathscr{R}[GG^*]} = \overline{\mathscr{R}[G]}. \tag{9.58}$$

   from which it is concluded that $\mathscr{R}[GG^*]$ is dense in $\mathscr{R}[G]$.
4. The positivity properties of GG^* play an important role in the characterization of $\|L\|$ and situations when $L < I$. These properties, and the results in Sect. 5.2 are central to the important convergence properties discussed in the next theorem.
5. Using the Spectral Mapping Theorem gives the inclusion conditions

$$\begin{aligned} if \;\; GG^* \geq \varepsilon_0^2 I > 0, \;\; then, \quad spec[L] &\subset [(1 + \varepsilon^{-2}\|G^*\|^2)^{-1}, (1 + \varepsilon^{-2}\varepsilon_0^2)^{-1}], \\ and \quad spec[L] &\subset [(1 + \varepsilon^{-2}\|G^*\|^2)^{-1}, 1] \;\; otherwise. \\ Also \quad spec[\alpha L] &= \alpha spec[L] \\ and \quad spec[(\alpha - \beta)I + \beta L] &= (\alpha - \beta) + \beta spec[L]. \end{aligned} \tag{9.59}$$

   These relations can be used to bound operator norms as the operators are self adjoint and hence the spectral radius is equal to the operator norm.

Turning now to the issues of convergence in NOILC algorithms,

Theorem 9.2 (Error Convergence in NOILC Algorithm 9.1) *Application of the NOILC Algorithm 9.1 to the dynamics $y = Gu + d$ with reference r and initial input u_0 has the following error convergence properties*

1. $\|e_{k+1}\|_{\mathscr{Y}} \leq \|e_k\|_{\mathscr{Y}}$ *for all* $k \geq 0$ *with strict inequality holding if* $ker[G^*] = \{0\}$.
2. *If* $\|L\| < 1$, *then the error sequence converges to zero as* $k \to \infty$.
3. *If* $\|L\| = 1$, *then the error sequence converges to zero as* $k \to \infty$ *if the initial error* $e_0 \in \overline{\mathscr{R}[I - L]} = \overline{\mathscr{R}[G]}$. *If, in addition,* $ker[G^*] = \{0\}$, *then* $L < I$ *and convergence to zero is achieved for all* $e_0 \in \mathscr{Y}$.

4. *In general, the error sequence converges to* $P_{ker[G^*]}e_0$ *as* $k \to \infty$ *where* $P_{ker[G^*]}$ *is the self adjoint, positive operator defining the orthogonal projection of a vector onto the closed subspace* $ker[G^*] \subset \mathscr{Y}$.

Proof The monotonicity properties have been discussed previously and need no further comment. If $\|L\| < 1$, then convergence to zero follows from Theorem 5.4. Next, if $\|L\| = 1$, then Theorem 5.9 shows that $e_k \to 0$ whenever $e_0 \in \overline{\mathscr{R}[I-L]}$ which is just $e_0 \in \overline{\mathscr{R}[GG^*]}$ as $I-L = GG^*(I+GG^*)^{-1}$ and hence for any $e_0 \in \overline{\mathscr{R}[G]}$ as $\overline{\mathscr{R}[GG^*]} = \overline{\mathscr{R}[G]}$. If, in addition, $ker[G^*] = \{0\}$, then Theorem 9.1 proves that $I-L > 0$. Theorem 5.10 then proves convergence to zero for all $e_0 \in \mathscr{Y}$. Finally, the projection characterization of the limit follows from Theorem 5.9 using the identity $ker[I-L] = ker[G^*]$. □

As in Theorem 7.2, the convergence of the input signal sequence $\{u_k\}_{k\geq 0}$ requires an additional condition, namely that $e_0 \in \mathscr{R}[I-L] = \mathscr{R}[GG^*]$. The following result has a similar structure and demonstrates that NOILC retains many of the desirable properties of steepest descent algorithms.

Theorem 9.3 (Input Convergence in NOILC Algorithm 9.1) *With the notation of Theorem 9.2 suppose that* $e_0 \in \mathscr{R}[I-L] = \mathscr{R}[GG^*]$. *Under these conditions, the sequence* $\{u_k\}_{k\geq 0}$ *converges in norm to the unique input* $u_\infty \in \mathscr{U}$ *satisfying the tracking relationship* $r = Gu_\infty + d$ *whilst simultaneously minimizing the norm* $\|u-u_0\|^2_{\mathscr{U}}$. *That is,*

$$u_\infty = \arg\min\{\ \|u-u_0\|^2_{\mathscr{U}}\ : r = Gu+d\ \}. \tag{9.60}$$

Proof Convergence of the error to $e_\infty = 0$ follows from Theorem 9.2. Apply induction to the input update equation to show that

$$u_{k+1} = u_0 + \varepsilon^{-2}G^*L\sum_{j=0}^{k} L^j e_0. \tag{9.61}$$

As $e_0 \in \mathscr{R}[I-L]$, write $e_0 = (I-L)v_0$ for some $v_0 \in \mathscr{Y}$ and, using the orthogonal subspace decomposition $\mathscr{Y} = ker[I-L] \oplus \overline{\mathscr{R}[I-L]}$, take, without loss of generality $v_0 \in \overline{\mathscr{R}[I-L]}$. Theorem 5.9 then proves that $L^k v_0 \to 0$ as $k \to \infty$ and

$$\begin{aligned} u_{k+1} &= u_0 + \varepsilon^{-2}G^*L\left(\sum\nolimits_{j=0}^{k} L^j\right)(I-L)v_0 \ = \ u_0 + \varepsilon^{-2}G^*L(I-L^{k+1})v_0 \\ &\to u_\infty \ = \ u_0 + \varepsilon^{-2}G^*Lv_0 \qquad (as\ \ k\to\infty) \end{aligned} \tag{9.62}$$

and, as $e_\infty = 0$, it follows that $r = Gu_\infty + d$ as required. Finally, the equations define the minimum norm solution as, for any u in the closed linear variety $S = \{\ u\ :\ r = Gu+d\} \subset \mathscr{U}$, the inner product $\langle u-u_\infty, u_\infty - u_0\rangle_{\mathscr{U}} = \langle u-u_\infty, G^*Lv_0\rangle_{\mathscr{U}} = \langle G(u-u_\infty), Lv_0\rangle_{\mathscr{Y}} = 0$ as $G(u-u_\infty) = 0$. This is precisely the condition defining

the orthogonal projection of u_0 onto S which, from the Projection Theorem 2.17, defines the unique solution of the minimum norm problem. □

The inclusion of relaxation simplifies the result considerably but requires a characterization of the non-zero limit error,

Theorem 9.4 (Error Convergence in Relaxed NOILC: The Feedback Case) *Application of the relaxed feedback NOILC Algorithm 9.2 with $0 \leq \alpha < 1$ to the dynamics $y = Gu + d$ with reference r and initial input u_0 has the following convergence properties*

$$\lim_{k\to\infty} e_k = e_\infty \quad \textit{where} \quad e_\infty = L\left[\alpha e_\infty + (1-\alpha)(r-d)\right]. \tag{9.63}$$

In particular, this equation has the unique solution

$$e_\infty = (I + (1-\alpha)^{-1}\varepsilon^{-2}GG^*)^{-1}(r-d), \tag{9.64}$$

and hence, $\lim_{k\to\infty} u_k = u_\infty$, which is the input signal that minimizes the quadratic performance index

$$J(u) = \|e\|^2_{\mathscr{Y}} + (1-\alpha)\varepsilon^2\|u\|^2_{\mathscr{U}} \quad \textit{with} \quad e = r - y, \tag{9.65}$$

subject to the constraints $y = Gu + d$.

The result has an interesting performance and design interpretation,

1. The result ensures convergence to a limit that has a clear interpretation as the solution of a quadratic optimization problem where the input weighting is reduced by a factor of $1-\alpha$.
2. If α is close to unity, the weighting of the input is very close to zero—a so-called "cheap" optimal control problem. In essence, this problem tries to minimize the error norm with only a small penalty on large deviations of the input signal from zero. As $\varepsilon^2(1-\alpha)$ increases, the penalty for using significant input magnitudes increases, leading to smaller control magnitudes and larger limit errors.
3. If the minimization of the objective function $J(u)$ is set as the asymptotic objective of the iterations, the properties and magnitude of the limit error e_∞ is defined by the parameter $\gamma = (1-\alpha)\varepsilon^2$. The value of this product is chosen by the user and is likely to be small. It does not define α or ε uniquely leaving room for design considerations. For example, if $\varepsilon^2 \geq \gamma$ is chosen to provide desirable convergence properties of the operator L, the value of α (and hence the magnitude of the limit e_∞) is computed from $\alpha = 1 - \varepsilon^{-2}\gamma$. For discrete or continuous state space systems, desirable convergence properties may include a need to avoid high feedback gains in the Riccati matrix.

Proof of Theorem 9.4 Using the error evolution in Algorithm 9.2, note that $r(\alpha L) = \|\alpha L\| \leq \alpha < 1$ and hence the algorithm converges to a limit (in the norm topology) as required. The value of e_∞ is obtained by replacing e_{k+1} and e_k by e_∞ which gives the required result. As L is invertible,

$$\begin{array}{rl} & (I+\varepsilon^{-2}GG^*)e_\infty \;=\; \alpha e_\infty + (1-\alpha)(r-d) \\ so\ \ that & ((1-\alpha)I+\varepsilon^{-2}GG^*)e_\infty = (1-\alpha)(r-d) \end{array} \tag{9.66}$$

which provides the desired unique solution as $(1-\alpha)I+\varepsilon^{-2}GG^*$ is invertible. Finally, using the techniques of Theorem 2.18, the input minimizing $J(u)$ is the solution of the equation $\varepsilon^2(1-\alpha)u = G^*e$ so that $\varepsilon^2(1-\alpha)Gu = GG^*e$. That is, $\varepsilon^2(1-\alpha)(r-d-e) = GG^*e$ which gives the required solution. □

Finally, the feedforward version 9.3 of the relaxed algorithm has the following convergence properties.

Theorem 9.5 (Error Convergence in Relaxed NOILC: The Feedforward Case) *Application of the relaxed feedforward NOILC Algorithm 9.3 with* $0 < \alpha < 1$ *and* $0 < \beta \leq 1$ *to the dynamics* $y = Gu + d$ *with reference* r *and initial input* u_0 *has the following convergence properties*

$$\lim_{k\to\infty} e_k = e_\infty \quad where \quad e_\infty = [(\alpha-\beta)I+\beta L]\,e_\infty + (1-\alpha)(r-d). \tag{9.67}$$

In particular, if $\alpha = 1$ *and* $0 < \beta \leq 1$*, then the error sequence converges to the orthogonal projection of* e_0 *onto the closed subspace* $ker[G^*]$.

Proof The proof when $\alpha < 1$ follows immediately as $(\alpha-\beta)I \leq (\alpha-\beta)I+\beta L = \alpha I - \beta(I-L) \leq \alpha I$ and hence the spectral radius $r\,((\alpha-\beta)I+\beta L) < 1$. The details are left as an exercise for the reader. If $\alpha = 1$ and $0 < \beta < 1$, then convergence is guaranteed as,

$$\begin{array}{c} L_\beta = (1-\beta)I+\beta L = I - \beta\varepsilon^{-2}GG^*(I+\varepsilon^{-2}GG^*)^{-1} \\ with \quad \left(1-\frac{\beta\varepsilon^{-2}\|G^*\|^2}{1+\varepsilon^{-2}\|G^*\|^2}\right) I \leq L_\beta \leq I. \end{array} \tag{9.68}$$

If $\|L_\beta\| < 1$, the proof follows as $\|e_{k+1}\| \leq \|L_\beta\|\|e_k\|$ for all $k \geq 0$ and hence $e_\infty = 0$. If $\|L_\beta\| = 1$, then Theorem 5.9 indicates that $e_\infty = 0$ if $e_0 \in \overline{\mathscr{R}[I-L_\beta]} = \overline{\mathscr{R}[GG^*]}$. The orthogonal complement of this closed subspace is $ker[I-L_\beta] = ker[GG^*] = ker[G^*]$ and it is a simple calculation to show that $L_\beta v = v$ for all $v \in ker[G^*]$. The result follows by writing $e_0 = e_0^{(1)} + e_0^{(2)}$ with $e_0^{(1)} \in \overline{\mathscr{R}[I-L_\beta]}$ and $e_0^{(2)} \in ker[G^*]$. □

9.2 Robustness of NOILC: Feedforward Implementation

The reader will note that, being monotonic, NOILC Algorithm 9.1 shares many properties with gradient-based and inverse-model-based algorithms. This section considers the feedforward implementation of NOILC and demonstrates that it also has robustness properties that are strongly related to those of gradient and inverse

algorithms. The model of process dynamics is, again, $y = Gu + d$ and the underlying feedforward input update equation used for off-line computation is

$$u_{k+1} = u_k + \varepsilon^{-2} G^*(I + \varepsilon^{-2} GG^*)^{-1} e_k = u_k + \varepsilon^{-2} G^* L e_k. \tag{9.69}$$

where e_k is the measured error signal on iteration k. The following analysis firstly considers computation of u_{k+1}. Robustness analysis is separated into two cases, namely the case of left and the case of right multiplicative modelling errors. It is based on monotonicity of carefully selected quadratic forms and again introduces a need for positivity of multiplicative modelling errors. As was the case in inverse and gradient algorithms, the geometry of the input and output spaces is central to the analysis. For example, $\mathscr{Y}$ has an orthogonal subspace decomposition of the form

$$\begin{array}{c} \mathscr{Y} = \overline{\mathscr{R}[G]} \oplus ker[G^*], \quad with \;\; ker[G^*] = ker[G^* L], \\ and \quad \overline{\mathscr{R}[G^* G]} = \overline{\mathscr{R}[G^* L G]} = \overline{\mathscr{R}[G^*]}, \end{array} \tag{9.70}$$

the second equality follows from $G^* L = (I + \varepsilon^{-2} G^* G)^{-1} G^*$ and the final equality from $ker[G^* G] = ker[G]$ and $G^* L G = G^* G (I + \varepsilon^{-2} G^* G)^{-1}$.

9.2.1 Computational Aspects of Feedforward NOILC

e_k is the observed/measured tracking error on iteration k and not necessarily equal to the error predicted by the plant model. Updating of the input using the model can be achieved based on the observation that the input change $u_{k+1} - u_k = \varepsilon^{-2} G^* L e_k$ can be computed off-line as the input generated by the model in one iteration of NOILC from a starting condition of zero input using the value $d = 0$ and the "reference signal" equal to the measured error e_k. For example,

1. for discrete state space Algorithm 9.4 for the model $S(A, B, C)$, the update $\Delta u_{k+1} = u_{k+1} - u_k = \varepsilon^{-2} G^* L e_k$ is computed from the equations used in feedback *Implementation Two* of Sect. 9.1.3 written in the form,

$$\begin{aligned} z_{k+1}(t+1) &= A z_{k+1}(t) + B \Delta u_{k+1}(t), \qquad \underbrace{z_{k+1}(0) = 0}, \\ \Delta u_{k+1}(t) &= R^{-1}(t) B^T \left(-K(t) z_{k+1}(t) + \xi_{k+1}(t) \right), \\ \xi_{k+1}(t) &= \left(I + \varepsilon^{-2} \tilde{K}(t+1) B R^{-1}(t) B^T \right)^{-1} \psi_{k+1}(t), \qquad with, \\ \psi_{k+1}(t) &= A^T \xi_{k+1}(t+1) + C^T Q(t+1) \underbrace{e_k(t+1)}, \qquad \xi_{k+1}(N) = 0. \end{aligned} \tag{9.71}$$

   where $K(t)$ remains unchanged and is that defined by model data A, B, C, D plus $Q(t)$, $R(t)$ and ε^2.

2. For continuous state space Algorithm 9.5 for the model $S(A, B, C)$, the update $\Delta u_{k+1} = u_{k+1} - u_k = \varepsilon^{-2} G^* L e_k$ is computed from the equations used in feedback *Implementation Two* of Sect. 9.1.6 written in the form,

$$\begin{aligned}
\dot{z}_{k+1}(t) &= A z_{k+1}(t) + B\Delta u_{k+1}(t), \qquad \underbrace{z_{k+1}(0) = 0}, \\
\Delta u_{k+1}(t) &= R^{-1}(t) B^T \left(-K(t) z_{k+1}(t) + \xi_{k+1}(t)\right), \\
\dot{\xi}_{k+1}(t) &= \left(A^T - \varepsilon^{-2} K(t) B R^{-1}(t) B^T\right) \xi_{k+1}(t) - C^T Q(t) \underbrace{e_k(t)}.
\end{aligned} \tag{9.72}$$

where $K(t)$ remains unchanged and is that defined by model data.

9.2.2 The Case of Right Multiplicative Modelling Errors

Now suppose that G is a model of the actual plant and that the plant is represented by the operator GU where $U : \mathscr{U} \to \mathscr{U}$ represents a right multiplicative modelling error. The case when $U = I$ is that when the model is capable of predicting plant behaviour exactly. Using the plant input/output relationship $y = GUu + d_U$ then gives the error update equation

$$e_{k+1} = L_U e_k, \quad \textit{for all } k \geq 0, \quad \textit{where } L_U = \left(I - \varepsilon^{-2} GUG^* L\right) \tag{9.73}$$

noting that, in general, $L_U \neq L_U^*$ and that $L_U e = e$ if $e \in ker[G^*]$. Note, in particular, that any component of $e_0 \in ker[G^* L] = ker[G^*]$ is unchanged by the iterative process. The only significant evolution occurs in $\overline{\mathscr{R}[G]}$, which is both L-invariant and L_U-invariant and is a Hilbert space in its own right. This can be expressed more clearly by writing $e_0 = e_0^{(1)} + e_0^{(2)}$ with $e_0^{(1)} \in \overline{\mathscr{R}[G]}$ and $e_0^{(2)} \in ker[G^*]$. A simple calculation gives,

$$e_k = L_U^k e_0 = L_U^k e_0^{(1)} + e_0^{(2)} \quad \textit{and} \quad \|e_k\| \geq \|e_0^{(2)}\|, \quad \textit{for all } k \geq 0. \tag{9.74}$$

Convergence is hence described entirely by the properties of the restriction of L_U to the closed subspace $\overline{\mathscr{R}[G]}$. The best that can be achieved is that $\lim_{k\to\infty} e_k = e_0^{(2)}$.

Convergence in the presence of the modelling error U is defined by the construction of a topologically equivalent norm in $\mathscr{Y}$ (and hence $\overline{\mathscr{R}[G]}$). Noting that $L = L^* > 0$, Theorem 9.1 then indicates that the inner product

$$\langle e, w\rangle_0 = \langle e, Lw\rangle_{\mathscr{Y}} \quad \textit{generates a norm} \quad \|e\|_0 = \langle e, Le\rangle_{\mathscr{Y}}^{1/2} \tag{9.75}$$

that is topologically equivalent to $\|e\|_{\mathscr{Y}}$.

The approach to robustness analysis is based on a consideration of the possibility of ensuring the monotonicity property $\|e_{k+1}\|_0 < \|e_k\|_0$ for all $k \geq 0$. That is,

$$(\textit{Robust Monotonicity}) \quad \langle e_{k+1}, Le_{k+1}\rangle_{\mathscr{Y}} < \langle e_k, Le_k\rangle_{\mathscr{Y}}, \quad \textit{for all } k \geq 0, \tag{9.76}$$

which, using the error evolution equation, can be written in the form

$$\begin{gathered}\|e_{k+1}\|_0^2 = \|e_k\|_0^2 + \varepsilon^{-2}\langle \eta_k, \left[\varepsilon^{-2}U^*G^*LGU - (U+U^*)\right]\eta_k\rangle_{\mathscr{U}} \\ \textit{where} \qquad \eta_k = G^*Le_k \in \mathscr{R}[G^*] \subset \mathscr{U}.\end{gathered} \tag{9.77}$$

Note that $\eta_k = 0$ if, and only if, $e_k \in ker[G^*L] = ker[G^*]$ i.e. $e_k^{(1)} = 0$. Also, as L has a bounded inverse, $\mathscr{R}[G^*L] = \mathscr{R}[G^*]$ and η_k can take any value in $\mathscr{R}[G^*]$.

The following Theorem describes robust monotonic convergence in the new norm topology.

Theorem 9.6 (Robustness of NOILC with Right Multiplicative Errors) *Consider NOILC Algorithm 9.1 in its feedforward implementation using the measured data e_k. Using the notation defined above, a necessary and sufficient condition for the norm sequence $\{\|e_k\|_0\}_{k\geq 0}$ to satisfy the monotonicity condition $\|e_{k+1}\|_0 < \|e_k\|_0$, for all $k \geq 0$, and for all e_0 with $e_0^{(1)} \neq 0$, is that*

$$\textit{Condition One} \quad -U + U^* > \varepsilon^{-2}U^*G^*LGU, \qquad \textit{on } \mathscr{R}[G^*]. \tag{9.78}$$

In these circumstances,

1. *The modelling error satisfies the condition $ker[U] \cap \mathscr{R}[G^*] = \{0\}$.*
2. *For any initial error e_0, the norm sequence $\{\|e_k\|_0\}_{k\geq 0}$ is bounded and the limit $\lim_{k\to\infty}\|e_k\|_0 = E_0 \geq 0$ exists. The limit $e_\infty = \lim_{k\to\infty} e_k = 0$ is therefore achieved if, and only if, $E_0 = 0$.*
3. *A sufficient condition for Condition One to be satisfied is that the operator inequality holds either on the closure $\overline{\mathscr{R}[G^*]}$ or on the full input space $\mathscr{U}$. If $ker[G] = \{0\}$, then, as $\overline{\mathscr{R}[G^*]} = \mathscr{U}$, these two alternatives are identical.*

Finally, in the new topology,

1. *the induced norm of L_U in $\mathscr{Y}$ satisfies $\|L_U\|_0 \leq 1$ and, more specifically,*
2. *the induced norm of the restriction of L_U to $\overline{\mathscr{R}[G]}$ in the new topology also satisfies*

$$\|L_U\|_0 \leq 1. \tag{9.79}$$

In particular, if $\mathscr{R}[G]$ is finite dimensional, then this norm is strictly less than unity and,

$$\textit{for all starting conditions} \quad e_0 \in \mathscr{Y}, \qquad \lim_{k\to\infty} e_k = e_0^{(2)} \in ker[G^*]. \tag{9.80}$$

Note: If $ker[G^*] = \{0\}$, $\mathscr{R}[G]$ is dense in $\mathscr{Y}$ and monotonicity occurs for all $e_0 \in \mathscr{Y}$.

Proof The first condition follows by noting that $U + U^* > 0$ on $\mathscr{R}[G^*]$ by assumption as $U^*G^*LGU \geq 0$. The assumed inequality would then be violated if there existed a non-zero $u \in \mathscr{R}[G^*]$ satisfying $Uu = 0$. For the remainder of the proof, it is only necessary to consider the case when $k = 0$. The necessity of the conditions for monotonicity follows as, if $e_0^{(1)} \neq 0$, then $\eta_0 \neq 0$ and, by suitable choice of $e_0^{(1)}$, η_0 can take any value in $\mathscr{R}[G^*]$. Condition One then follows from Eq. (9.77) which then requires that the second term is strictly negative for all non-zero $\eta_0 \in \mathscr{R}[G^*]$. Sufficiency follows easily from Eq. (9.77). The existence of $E_0 \geq 0$ follows from positivity of norms and monotonicity whilst the comment on the case when $E_0 = 0$ is a consequence of the definition of convergence in norm to zero. Next the inequality holds on $\mathscr{R}[G^*]$ if it holds on any subspace that contains it. The proofs that $\|L_U\|_0 \leq 1$ are a direct consequence of the monotonicity of the norms on $\mathscr{Y}$ and hence on any L_U-invariant subspace. Finally, if $\mathscr{R}[G]$ is finite dimensional, then $\mathscr{R}[G] = \overline{\mathscr{R}[G]}$. Choose a non-zero $e_0 \in \overline{\mathscr{R}[G]}$ arbitrarily. Then $\eta_0 \neq 0$ and the resultant inequality $\|e_1\|_0 < \|e_0\|_0$ plus the compactness of the unit sphere leads to the strict bound $\|L_U\|_0 < 1$ on the norm of the restriction of L_U to $\overline{\mathscr{R}[G]}$ for, otherwise, there would exist a non-zero $e \in \overline{\mathscr{R}[G]}$ such that $\|L_U e\|_0 = \|e\|_0$. This contradicts the proven monotonicity. The proof of convergence to $e_0^{(2)}$ follows. □

In principle, this result provides a good description of robustness in the presence of the modelling error. The practical problems implicit in checking Condition One lie in the presence of the operator L, which depends on GG^* in a "nonlinear" way. The following result simplifies the condition by providing two simpler, but more conservative, alternative sufficient conditions.

Theorem 9.7 (Robust Convergence and Boundedness: Alternative Conditions) *Using the assumptions of Theorem 9.6, its conclusions remain valid if (a sufficient condition) Condition One is replaced by either*

$$\begin{aligned} &\textit{Condition Two} - U + U^* > \frac{\varepsilon^{-2}\|G\|^2}{1+\varepsilon^{-2}\|G\|^2} U^*U, \qquad \textit{on } \mathscr{R}[G^*], \\ \textit{or}, \quad &\textit{Condition Three} - U + U^* > \varepsilon^{-2}U^*G^*GU, \qquad \textit{on } \mathscr{R}[G^*]. \end{aligned} \tag{9.81}$$

A sufficient condition for each Condition to be satisfied is that the associated operator inequality holds either on $\overline{\mathscr{R}[G^*]}$ or on the full space $\mathscr{U}$.

Proof The sufficiency of Condition Two follows from Condition One and the identity $\varepsilon^{-2}G^*LG = (I+\varepsilon^{-2}G^*G)^{-1}\varepsilon^{-2}G^*G = I-(I+\varepsilon^{-2}G^*G)^{-1} \leq \left(\frac{\varepsilon^{-2}\|G\|^2}{1+\varepsilon^{-2}\|G\|^2}\right) I$ from Theorem 9.1 (with G replaced by G^*). Condition Three also follows from Theorem 9.1 as $L \leq I$. □

Note: Of the three given, Condition One is clearly the least conservative, Condition Two adds conservatism by replacing G^*LG by a constant upper bound whilst Condition Three replaces L by a constant upper bound and leaves the factor G^*G in place. The replacement of $\mathscr{R}[G^*]$ by $\overline{\mathscr{R}[G^*]}$ or $\mathscr{U}$ also adds conservatism but may make the

checking of the condition easier in practice. A useful example of this relaxation also allows Conditions Two and Three to be combined into one, parameterized condition by writing $L = \theta L + (1-\theta)L$ and noting that, for all $\theta \in [0, 1]$,

$$\varepsilon^{-2} G^* L G \leq \theta \frac{\varepsilon^{-2}\|G\|^2}{1+\varepsilon^{-2}\|G\|^2} I + (1-\theta)\varepsilon^{-2} G^* G \tag{9.82}$$

in the $\mathscr{Y}$ topology. Applications of this bound include,

Theorem 9.8 (Robust Convergence and Boundedness: Conditions Combined) *Using the notation of the discussion above, suppose that U has a bounded inverse $\hat{U}$ on $\mathscr{U}$ and that, for some $\theta \in [0, 1]$,*

$$\begin{array}{c} \textit{Condition Four} - \hat{U} + \hat{U}^* > \theta\beta_I I + (1-\theta)\beta_G G^* G \quad \textit{on} \quad \mathscr{U} \\ \textit{where} \quad \beta_I = \frac{\varepsilon^{-2}\|G\|^2}{1+\varepsilon^{-2}\|G\|^2} \quad \textit{and} \quad \beta_G = \varepsilon^{-2}. \end{array} \tag{9.83}$$

Then Condition One holds and the monotonicity and convergence predictions of Theorem 9.6 are guaranteed.

Proof Consider Condition one with $\mathscr{R}[G^*]$ replaced by $\mathscr{U}$ and write it in the form $\hat{U} + \hat{U}^* > G^* L G$ on $\mathscr{U}$. Using the mixed bound for L then gives the result. □

Some insight into the implications of these NOILC robustness conditions is obtained by considering the case of no modelling error when $U = I$ and linking the results to conditions for robustness of the inverse model and gradient algorithms in Chaps. 6 and 7. More precisely,

1. Condition Two is always satisfied when $U = I$ as it then reduces to $2I > \beta_I I$ as the "gain" parameter $\beta_I = \frac{\varepsilon^{-2}\|G\|^2}{1+\varepsilon^{-2}\|G\|^2} < 1$.
2. More generally, Condition Two is algebraically identical to the robustness condition for the left inverse model algorithm with right multiplicative perturbation U. In particular, it is satisfied if,

$$\|(I - \beta_I U)\|^2 = r\left((I - \beta_I U)^*(I - \beta_I U)\right) < 1. \tag{9.84}$$

3. If $U = I$, then Condition Three reduces to $2I > \varepsilon^{-2} G^* G$ which is satisfied if $\varepsilon^2 > \frac{1}{2}\|G\|^2$. That is, the weighting on the control component of the NOILC objective function is limited by this condition. The eigenstructure ideas of Sect. 9.1.7, then imply that the achievable spectral bandwidth is limited as is the achievable convergence rate.
4. More generally, Condition Three is identical to that derived for the gradient algorithm with right multiplicative perturbation U. In particular, it is satisfied if,

$$r\left((I - \beta_G G U G^*)^*(I - \beta_G G U G^*)\right) < 1. \tag{9.85}$$

5. The mixed condition of Theorem 9.8 provides a continuous link between the inverse and gradient conditions and will be seen to have possible value in the later development of frequency domain robustness criteria.

Finally, the nature of the convergence can, in general, be refined by considering the detailed form of U and G in the application of the results. The monotonicity of the error sequence measured by the $\|\cdot\|_0$ norm implies that the induced operator norm $\|L_U\|_0 \leq 1$ in the topology induced by the inner product $\langle e, w\rangle_0 = \langle e, Lw\rangle_{\mathscr{Y}}$. In particular, $\|L_U\|_0 = 1$ if $ker[G^*] \neq \{0\}$. More generally, its restriction to $\overline{\mathscr{R}[G]}$ can have unit norm even if $ker[G^*] = \{0\}$, a property that occurs only in the infinite dimensional case and leads to technical problems beyond the chosen scope of this text. The result has shown that $\|L_U\|_0 < 1$ on $\overline{\mathscr{R}[G]}$ if it is finite dimensional. The following result describes a case that is particularly relevant to finite or (a class of) infinite dimensional problems where $ker[G] = \{0\}$.

Theorem 9.9 (Robust Convergence to $e_0^{(2)} \in ker[G^*]$) *Suppose that the plant GU has model G and right multiplicative modelling error U. Suppose also that that there exists a real number $\varepsilon_0^2 > 0$ such that*

$$G^*G \geq \varepsilon_0^2 I \quad \text{in} \quad \mathscr{R}[G^*]. \tag{9.86}$$

Suppose also that either

$$\begin{aligned} &\textit{Condition Two} - (A) \quad U + U^* \geq \frac{\varepsilon^{-2}\|G\|^2}{1+\varepsilon^{-2}\|G\|^2} U^*U + \varepsilon_0^2 I \qquad \text{in} \quad \mathscr{R}[G^*], \\ \text{or} \quad &\textit{Condition Three} - (B) \quad U + U^* \geq \varepsilon^{-2} U^*G^*GU + \varepsilon_0^2 I \qquad \text{in} \quad \mathscr{R}[G^*], \end{aligned} \tag{9.87}$$

or, U has a bounded inverse on $\mathscr{U}$, $\theta \in [0, 1]$ and

$$\textit{Condition Four} \quad - \quad (C) \quad \hat{U} + \hat{U}^* > \theta\beta_I I + (1-\theta)\beta_G G^*G + \varepsilon_0^2 I \ \text{ on } \ \mathscr{U} \tag{9.88}$$

holds, then $\|L_U\|_0 < 1$ on $\overline{\mathscr{R}[G]}$ and the error sequence converges to the component $e_0^{(2)} \in ker[G^*]$.

Proof Note that a proof that $\|L_U\|_0 < 1$ on $\overline{\mathscr{R}[G]}$ proves the convergence to $e_0^{(2)}$. Next consider the use of (A). From (9.77) with $\eta_k = G^*Le_k$ and e_k arbitrary,

$$\begin{aligned} \|e_{k+1}\|_0^2 &= \|e_k\|_0^2 + \varepsilon^{-2}\langle \eta_k, \left[\varepsilon^{-2}U^*G^*LGU - \left(U + U^*\right)\right]\eta_k\rangle_{\mathscr{U}} \\ &\leq \|e_k\|_0^2 + \varepsilon^{-2}\langle \eta_k, \left[\frac{\varepsilon^{-2}\|G\|^2}{1+\varepsilon^{-2}\|G\|^2} U^*U - \left(U + U^*\right)\right]\eta_k\rangle_{\mathscr{U}} \\ &\leq \|e_k\|_0^2 - \varepsilon^{-2}\varepsilon_0^2\|\eta_k\|_{\mathscr{U}}^2. \end{aligned} \tag{9.89}$$

Now set $e_k = Gw_k \in \mathscr{R}[G]$ and note that, using Theorem 9.1 with G replaced by G^*, and the inequality $\|e\| \leq \|G\|\|w\|$, it follows that

$$\|\eta_k\|_{\mathscr{U}}^2 = \|G^*LGw_k\|_{\mathscr{U}}^2 = \langle w_k, G^*LGG^*LGw_k\rangle_{\mathscr{U}}$$
$$\geq \left(\frac{\varepsilon_0^2}{1+\varepsilon^{-2}\|G\|^2}\right)^2 \|w_k\|_{\mathscr{U}}^2 \geq \left(\frac{\|G\|^{-1}\varepsilon_0^2}{1+\varepsilon^{-2}\|G\|^2}\right)^2 \|e_k\|_{\mathscr{Y}}^2. \tag{9.90}$$

Using this relation and the topological equivalence of the norms implies that, for any $e_k \in \mathscr{R}[G]$, the inequality $\|e_{k+1}\|_0^2 \leq (1-\lambda)\|e_k\|_0^2$ holds for some $\lambda > 0$. The relation therefore is also valid in the closure $\overline{\mathscr{R}[G]}$. The required inequality $\|L_U\|_0 < 1$ then follows. Finally, for (B), note that $L \leq I$ and write

$$\|e_{k+1}\|_0^2 = \|e_k\|_0^2 + \varepsilon^{-2}\langle \eta_k, \left[\varepsilon^{-2}U^*G^*LGU - \left(U+U^*\right)\right]\eta_k\rangle_{\mathscr{U}}$$
$$\leq \|e_k\|_0^2 + \varepsilon^{-2}\langle \eta_k, \left[\varepsilon^{-2}U^*G^*GU - \left(U+U^*\right)\right]\eta_k\rangle_{\mathscr{U}}$$
$$\leq \|e_k\|_0^2 - \varepsilon^{-2}\varepsilon_0^2\|\eta_k\|_{\mathscr{U}}^2. \tag{9.91}$$

The proof is then concluded using a similar argument to that used for (A). (C) is proved in a similar way noting that

$$\|e_{k+1}\|_0^2 \leq \|e_k\|_0^2 - \varepsilon^{-2}\varepsilon_0^2\|U\eta_k\|_{\mathscr{U}}^2 \tag{9.92}$$

and, using invertibility of U, the existence of a real number $\alpha > 0$ such that $U^*U \geq \alpha I$. The details are left as an exercise for the reader. □

The robustness results are reassuring in that they indicate substantial robustness of the algorithms whenever the defined positivity conditions are satisfied. To be useful in practice, the conditions must be converted into checkable conditions, preferably written in terms of quantities familiar to practicing engineers. One such example is now described.

9.2.3 Discrete State Space Systems with Right Multiplicative Errors

Consider the case when G can be represented by a discrete time, linear, time invariant, state space model $S(A, B, C, D)$ (or its equivalent supervector description) on the interval $0 \leq t \leq N$ and that it has the transfer function matrix $G(z)$. Consider Algorithm 9.4 where the weights Q and R are taken to be independent of sample number "t". The actual plant model is assumed to be expressed in the form GU with right multiplicative modelling error $U : \mathscr{U} \to \mathscr{U}$ which has its own state space model $S(A_U, B_U, C_U, D_U)$ and $\ell \times \ell$ transfer function matrix $U(z)$. In this case, $\mathscr{Y}$ and $\mathscr{U}$ are finite dimensional and hence every vector subspace is closed.

In what follows, Conditions Two and Three of Theorem 9.7 are converted into frequency domain tests for robust convergence of the feedforward implementation.

Theorem 9.10 (Checking Condition Two for Discrete, State Space Systems) *Let the model G, the actual plant GU and the NOILC objective function be as described above. Then a sufficient condition for Condition Two of Theorem 9.7 to hold is that $U(z)$ is asymptotically stable and also that*

$$RU(z) + U^T(z^{-1})R > \beta_I U^T(z^{-1})RU(z), \text{ for all } |z| = 1, \tag{9.93}$$

where the "gain" parameter

$$\beta_I = \frac{\varepsilon^{-2}\|G\|^2}{1 + \varepsilon^{-2}\|G\|^2}. \tag{9.94}$$

In these circumstances, the Feedforward NOILC algorithm converges to the component $e_0^{(2)}$ of e_0 in $ker[G^*]$, this convergence being monotonic in the norm $\|e\|_0 = \sqrt{\langle e, Le\rangle}$. Finally, for computational purposes, $\|G\|^2$ can be replaced in the formula for β_I by its upper bound

$$\|G\|^2 \leq \sup_{|z|=1} r\left(R^{-1}G^T(z^{-1})QG(z)\right), \tag{9.95}$$

a relationship that links the frequency domain condition to the weighting matrices Q and R.

Proof First note that continuity and compactness of the unit sphere implies that there must exist a real number $\underline{g} > 0$ such that $RU(z) + U^T(z^{-1})R \geq \underline{g}R$ whenever $|z| = 1$ and hence $U(z)$ is strictly positive real in the sense defined by Theorem 4.7. The same theorem can then be used, with a change in notation, to prove the operator inequality $U + U^* \geq \underline{g}I > 0$ on $\mathscr{U}$ and hence that $ker[U] = \{0\}$. The proof that $U + U^* > \beta U^* U$ on $\overline{\mathscr{U}}$ using the frequency domain condition follows by applying Theorem 4.9 with a change in notation, replacing both $G(z)$ and $K(z)$ by $U(z)$ and noting that $\beta^* > \beta_I$ and $Uu \neq 0$ for all non-zero $u \in \mathscr{U}$. Using the finite dimensional nature of $\mathscr{U}$ and $\mathscr{Y}$ then indicates that L_U has norm strictly less than unity on the closed subspace $\mathscr{R}[G]$ which proves the convergence statement. Finally, as $\frac{a}{1+a}$ is monotonically increasing in a, β can be replaced by the value obtained when $\|G\|$ is substituted by any upper bound. The bound stated in the theorem was derived in Chap. 4 in Theorem 4.5. □

Theorem 9.11 (Checking Condition Three for Discrete State Space Systems) *A sufficient condition for Condition Three of Theorem 9.7 to hold is that both $G(z)$ and $U(z)$ are asymptotically stable and also that*

$$RU(z) + U^T(z^{-1})R > \beta_G U^T(z^{-1})G^T(z^{-1})QG(z)U(z), \text{ for all } |z| = 1, \tag{9.96}$$

where the scalar $\beta_G = \varepsilon^{-2}$. *The Feedforward NOILC algorithm then converges to the component* $e_0^{(2)}$ *of* e_0 *in* $ker[G^*]$. *Convergence is monotonic in the norm* $\|e\|_0$.

Proof As in the proof of the preceding result, $U + U^* \geq \underline{g}I$ for some $\underline{g} > 0$. The same theorem can then be used, with a change in notation, to prove the operator inequality $U + U^* \geq \underline{g}I > 0$ on $\mathscr{U}$ and hence that $ker[U] = \{0\}$. The proof that $U + U^* > \beta U^* G^* GU$ on $\mathscr{R}[G^*]$ using the frequency domain condition then follows by applying Theorem 4.9 with a change in notation, replacing $G(z)$ by $G(z)U(z)$ and $K(z)$ by $U(z)$, noting that $\beta^* > \beta_G$ and that $GUu \neq 0$ for all non-zero $u \in \mathscr{R}[G^*]$. This last statement is proved by writing any non-zero $u \in \mathscr{R}[G^*]$ as $u = G^* w$, from which the equality $2\langle w, GUu\rangle_{\mathscr{Y}} = 2\langle G^* w, Uu\rangle_{\mathscr{U}} = \langle u, (U + U^*)u\rangle_{\mathscr{U}} \geq \underline{g}\|u\|^2_{\mathscr{U}} > 0$. Using the finite dimensional nature of $\mathscr{U}$ and $\mathscr{Y}$ then indicates that L_U has norm strictly less than unity on $\mathscr{R}[G]$ which proves the convergence statement. □

The following result is proved in a similar way,

Theorem 9.12 (Condition Four in Discrete State Space System Applications) *A sufficient condition for Condition Four of Theorem 9.7 to hold is that* $U(z)$ *is both invertible and minimum-phase, that* $G(z)$ *is asymptotically stable and also that, for some* $\theta \in [0, 1]$,

$$R\hat{U}(z) + \hat{U}^T(z^{-1})R > \theta\beta_I R + \beta_G(1-\theta)G^T(z^{-1})QG(z), \quad \textit{for all } |z| = 1. \tag{9.97}$$

The Feedforward NOILC algorithm then converges to the component $e_0^{(2)}$ *of* e_0 *in* $ker[G^*]$, *this convergence being monotonic in the norm* $\|e\|_0$.

The results provide considerable insight into robustness of the NOILC procedure and some reassurance that robustness is an inherent property of the algorithmic process. Some areas of uncertainty do remain, as reflected by the observations that,

1. as presented, the conditions require the existence of a strictly positive real error $U(z)$. When $\ell > m$ such a description requires the definition and the use of redundant components in the kernel of $G(z)$.
2. The assumption of asymptotic stability of $U(z)$ suggests that the non-minimum-phase zeros of G must be those of GU.
3. The conditions require that $ker[U] = \{0\}$ and hence that D_U is nonsingular. The single-input, single-output case leads to the interpretation that, in effect, the modelling error cannot change the relative degree of the model.

Theorems 9.10 and 9.11 provide frequency domain checks for the use of the model G in the presence of U but also link the robustness to the chosen value of ε^2 and the choice of Q and R (which appear explicitly and also, implicitly, in the value of $\|G\|$). These relationships are complex, but they can be simplified for the single-input, single-output case ($m = \ell = 1$) to the conditions, respectively,

$$\left|\beta^{-1} - U(z)\right| < \beta^{-1} \quad \text{with "gain"} \quad \beta = \frac{\varepsilon^{-2}\|G\|^2}{1+\varepsilon^{-2}\|G\|^2},$$
$$\text{and} \quad \|G\|^2 = R^{-1}Q\|G(z)\|_\infty^2$$
$$\text{or,} \quad \left|\beta^{-1} - |G(z)|^2\, U(z)\right| < \beta^{-1} \quad \text{with "gain"} \quad \beta = \varepsilon^{-2}QR^{-1}, \tag{9.98}$$

for all z satisfying $|z| = 1$. These are precisely the robustness conditions for inverse and steepest descent algorithms in Chaps. 6 and 7 and provide a link between these algorithms and NOILC. Both have graphical interpretations and underline both the need for $U(z)$ to be strictly positive real and, also, the benefits (to robustness) of using smaller values of β. Within this constraint, note that the different criteria provide different characterizations of the range of $U(z)$ that can be tolerated.

1. In both cases, the modelling error can vary considerably, particularly if β is small or, equivalently, ε^2 is sufficiently large.
2. Although the second condition does not permit small values of ε^2, it may be of more practical use when $G(z)$ is low pass but $U(z)$ has high frequency resonances of large magnitude. The first condition will limit the value of gain β in such circumstances but, as the product $|G(z)|^2U(z)$ attenuates the high frequency values of U, the second test may permit larger gain values.
3. The mixed robustness criterion of Theorem 9.12 represents a compromise solution where the frequency domain content of the right hand side of the test can be manipulated, in a limited way, by a choice of $\theta \in [0, 1]$. Conceptually, varying θ changes the predicted permissible range of ε^2 that can be used. The natural choice is the value which maximizes this range.

9.2.4 The Case of Left Multiplicative Modelling Errors

This case is described by the use of a model G in the situation where the actual plant is described by UG where the left multiplicative modelling error operator $U : \mathscr{Y} \to \mathscr{Y}$ is linear and bounded. As with right multiplicative modelling errors, the error evolution in the feedforward implementation is described by a recursion

$$e_{k+1} = L_U e_k \quad \text{where} \quad L_U = I - \varepsilon^{-2}UGG^*L = I - U(I-L),$$
$$\text{with} \quad L = (I + \varepsilon^{-2}GG^*)^{-1} \quad \text{and} \quad I - L = \varepsilon^{-2}G(I + \varepsilon^{-2}G^*G)^{-1}G^*. \tag{9.99}$$

A simple calculation shows that errors in $ker[G^*]$ are unchanged by the iterative process and that error changes occur only in the subspace $\mathscr{R}[UG]$. The difference between left and right multiplicative modelling errors therefore is that the subspace where dynamics is concentrated depends on the modelling error itself. The technical issue that arises from this fact can be expressed in terms of the characterization of all signals into components in $ker[G^*]$ and $\mathscr{R}[UG]$ or its closure $\overline{\mathscr{R}[UG]}$.

Theorem 9.13 (Properties of U, $ker[G^*]$, $\mathscr{R}[UG]$ and $\overline{\mathscr{R}[UG]}$) *Assuming the notation described above, the condition* $ker[G^*] \cap \mathscr{R}[UG] = \{0\}$ *is satisfied if*

$$U + U^* > 0 \ \ on \ \ \mathscr{R}[G]. \tag{9.100}$$

In addition, if, for some scalar $\varepsilon_0^2 > 0$,

$$U + U^* \geq \varepsilon_0^2 I \ \ on \ \ \mathscr{Y}, \tag{9.101}$$

then U *has a bounded inverse* $\hat{U}$ *on* $\mathscr{Y}$ *and the following relationships are satisfied*

$$ker[G^*] \cap \overline{\mathscr{R}[UG]} = \{0\} \ \ and \ \ \mathscr{Y} = ker[G^*] \oplus \overline{\mathscr{R}[UG]}. \tag{9.102}$$

Notes:

1. *A continuity argument also indicates that the existence of* $\varepsilon_0 > 0$ *implies that the positivity condition* $U + U^* \geq \varepsilon_0^2 I$ *remains valid on* $\overline{\mathscr{R}[G]}$.
2. *The simplest situation is when* $ker[G^*] = \{0\}$ *when* $\mathscr{R}[G]$ *is then dense in* $\mathscr{Y}$.

Proof of Theorem If $ker[G^*] \cap \mathscr{R}[UG] \neq \{0\}$, there exists a non-zero $y \in \mathscr{R}[G]$ and a $u \in \mathscr{U}$ such that $y = Gu$ and $G^*Uy = G^*UGu = 0$. It follows that $\langle Gu, (U + U^*)Gu\rangle_{\mathscr{Y}} = \langle y, (U+U^*)y\rangle_{\mathscr{Y}} = 0$ which contradicts the assumed strict positivity of $U + U^*$ on $\mathscr{R}[G]$. Next, given the existence of $\varepsilon_0^2 > 0$, the invertibility of U follows from Theorem 2.9. If $\varepsilon_0^2 > 0$ exists, then suppose that there exists a non-zero vector $y \in \overline{\mathscr{R}[UG]}$ such that $G^*y = 0$. Let $\delta > 0$ be arbitrary and choose $w \in \mathscr{R}[UG]$ such that $\|y-w\| < \delta$. Note that $w \neq 0$ if $\delta < \|y\|_{\mathscr{Y}}$ is sufficiently small as $\|w\| > \|y\|-\delta$. Write $w = UGv$ and $G^*y = G^*\left(UGv + (y - w)\right) = 0$ and use the Cauchy Schwarz inequality to deduce that

$$\varepsilon_0^2\|Gv\|_{\mathscr{Y}}^2 \leq \langle Gv, (U + U^*)Gv\rangle_{\mathscr{Y}} \leq 2|\langle v, G^*(y - w)\rangle_{\mathscr{U}}| < 2\|Gv\|_{\mathscr{Y}}\delta. \tag{9.103}$$

As $w \neq 0$, $Gv \neq 0$ and hence $\varepsilon_0^2\|Gv\| < 2\delta$. Writing $\|w\| \leq \|U\|\|Gv\|$ then produces

$$\|y\| - \delta < \|w\| \leq \|U\|\|Gv\| < 2\varepsilon_0^{-2}\|U\|\delta \ \ for \ \ all \ \ \delta > 0. \tag{9.104}$$

That is $Gv = 0$ and $y = 0$, which is a contradiction. This proves that $ker[G^*] \cap \overline{\mathscr{R}[G]}$. Next, suppose that there exists a non-zero vector y that is orthogonal to the vector subspace $\mathscr{S} = ker[G^*] \oplus \overline{\mathscr{R}[UG]} \subset \mathscr{Y}$. It follows that it is also orthogonal to the subset $\mathscr{S}_0 = ker[G^*] \oplus U\overline{\mathscr{R}[G]} \subset \mathscr{S}$. Writing an arbitrarily chosen $x \in \mathscr{Y}$ in the form $x = g^* + g$ with $g^* \in ker[G^*]$ and $g \in \overline{\mathscr{R}[G]}$, the elements in $\mathscr{S}_0$ then take the form $g^* + Ug$. Examination of the orthogonality condition $\langle y, g^* + Ug\rangle = 0$ for all x implies that $\langle y, g^*\rangle = 0$ and hence that $y \in ker[G^*]^\perp = \overline{\mathscr{R}[G]}$. Choosing $g = y$ then gives $0 = \langle y, Uy\rangle \geq \frac{1}{2}\varepsilon_0^2\|y\|^2$ yielding $y = 0$ which is a contradiction. Hence $\mathscr{S}^\perp \subset \mathscr{S}_0^\perp = \{0\}$ proving denseness and $\overline{\mathscr{S}}_0 = \overline{\mathscr{S}} = \mathscr{Y}$.

Finally, note that $\mathscr{S}_0$ is the range of the bounded linear operator $\Omega : \mathscr{Y} \to \mathscr{Y}$ defined by $\Omega = (I - P) + UP$ where P is the self-adjoint, positive, orthogonal projection operator onto $\overline{\mathscr{R}[G]}$. Note that $P^2 = P, ker[P] = ker[G^*], \mathscr{R}[P] = \overline{\mathscr{R}[G]}$ and the restriction of P to $\overline{\mathscr{R}[G]}$ is the identity. To prove that $\mathscr{S}_0 = \mathscr{S} = \mathscr{Y}$ it is sufficient to prove that the range of Ω is closed in $\mathscr{Y}$. From the Closed Range Theorem 2.11, it is sufficient to prove that the range of Ω^* is closed. Note that $\Omega^* = (I - P) + PU^*$ and choose $y \in \overline{\mathscr{R}[\Omega^*]}$ and a sequence $\{x_k\}_{k\geq 1}$ in $\mathscr{Y}$ with the properties that $\|y - \Omega^* x_k\| < k^{-1}$. Using orthogonality,

$$k^{-2} > \|(I - P)(y - x_k)\|^2 + \|P(y - U^* x_k)\|^2, \quad for\ all\ k \geq 1. \tag{9.105}$$

In particular, $\lim_{k\to\infty} (I-P)x_k = (I-P)y$ and $\lim_{k\to\infty} P(y-U^*x_k) = 0$. Writing

$$\begin{array}{c} P(y - U^* x_k) = P(y - U^*(I - P)x_k) - PU^* P x_k \quad and \quad Px_k = P(Px_k) \\ then\ gives \quad \lim_{k\to\infty} PU^* P(Px_k) = Py - PU^*(I - P)y. \end{array} \tag{9.106}$$

Writing $x_k = (I - P)x_k + Px_k$, the convergence of $\{x_k\}$ to a limit $x_\infty \in \mathscr{Y}$ (which automatically satisfies $y = \Omega^* x_\infty$) then follows by regarding $\overline{\mathscr{R}[G]}$ as a Hilbert space with the inner product and norm inherited from $\mathscr{Y}$, regarding PU^*P as a map from $\overline{\mathscr{R}[G]}$ into itself and proving that it has a bounded inverse. First note that PU^*P has an adjoint PUP. Then use the inequalities,

$$0 \leq P(I - \lambda U)P(I - \lambda U^*)P \quad and \quad 0 \leq P(I - \lambda U^*)P(I - \lambda U)P, \tag{9.107}$$

noting that $P^2 = P$, and that $PUP + PU^*P \geq \varepsilon_0^2 I$ on $\overline{\mathscr{R}[G]}$, to give (by suitable choice of λ) the existence of a scalar $\alpha > 0$ such that $(PUP)PU^*P \geq \alpha I$ and $(PU^*P)PUP \geq \alpha I$ on $\overline{\mathscr{R}[G]}$. The existence of a bounded inverse then follows from Theorem 2.9. This completes the proof. □

The result will be used to derive robustness conditions. In particular, the characterization $\mathscr{Y} = ker[G^*] \oplus \overline{\mathscr{R}[UG]}$ creates a property that is a parallel to the condition $\mathscr{Y} = ker[G^*] \oplus \overline{\mathscr{R}[G]}$ used in the right multiplicative modelling error case. Before this is done, a suitable norm with which to model robust monotonicity and convergence is needed. First note that any initial error e_0 can be expressed, uniquely, as $e_0 = e_0^{(1)} + e_0^{(2)}$ with $e_0^{(1)} \in \overline{\mathscr{R}[UG]}$ and $e_0^{(2)} \in ker[G^*]$ so that $L_U e_0^{(2)} = e_0^{(2)}$ and hence

$$e_k = L_U^k e_0^{(1)} + e_0^{(2)} \quad for\ all\ k \geq 0. \tag{9.108}$$

Errors therefore evolve, essentially, in $\overline{\mathscr{R}[UG]}$. For the purposes of analysis, this closed subspace is regarded as a Hilbert space with inner product $\langle \cdot, \cdot \rangle_0$ defined by

$$\langle y, w \rangle_0 = \langle y, (I - L)w \rangle_{\mathscr{Y}} \quad for\ all\ y, w\ in\ \overline{\mathscr{R}[UG]}. \tag{9.109}$$

The inner product satisfies the linearity requirement with respect to each argument and also satisfies the required positivity properties as, for any non-zero $y \in \mathscr{Y}$,

$$\begin{aligned}\|y\|_0^2 &= \langle y, (I-L)y\rangle_{\mathscr{Y}} = \varepsilon^{-2}\langle G^*y, (I+\varepsilon^{-2}G^*G)^{-1}G^*y\rangle_{\mathscr{U}} \\ &\geq \frac{\varepsilon^{-2}}{1+\varepsilon^{-2}\|G\|^2}\|G^*y\|_{\mathscr{U}}^2 \end{aligned} \tag{9.110}$$

and the necessary positivity follows as $G^*y \neq 0$ if $y \in \overline{\mathscr{R}[UG]}$.

Note: The norm $\|\cdot\|_0$ on $\overline{\mathscr{R}[UG]}$ is not a norm on $\mathscr{Y}$ if $ker[G^*] \neq \{0\}$. More generally, it is not necessarily topologically equivalent to the norm $\|\cdot\|_{\mathscr{Y}}$ applied to $\overline{\mathscr{R}[UG]}$ but it will be

1. if $ker[G^*] = \{0\}$ and $\mathscr{R}[UG]$ is finite dimensional, as is the case when G represents a discrete time state space system $S(A, B, C, D)$ on a finite time interval with $m \leq \ell$ and $rank[D] = m$,
2. or, if G^* has positivity properties such as $GG^* \geq \varepsilon_1^2 I$ on $\mathscr{R}[G]$ for some $\varepsilon_1 > 0$.

In such circumstances, convergence is guaranteed with respect to the norm $\|\cdot\|_{\mathscr{Y}}$. More generally, the nature of any convergence could be complex and it will be good practice to consider the nature of convergence with respect to this norm for the model class under consideration. The following result does not resolve the issue but it provides some insight into the issues that may arise. In particular, a proof of boundedness will imply *weak* convergence.

Theorem 9.14 (Robustness, Boundedness and Weak Convergence) *Using the notation defined above,* $\lim_{k\to\infty}\|e_k\|_0 = 0$ *if, and only if, the condition* $\lim_{k\to\infty}\|G^*e_k\|_{\mathscr{U}} = 0$ *is satisfied. In addition, if* $\lim_{k\to\infty}\|e_k\|_0 = 0$*, then*

$$\lim_{k\to\infty}\langle f, e_k - e_o^{(2)}\rangle_{\mathscr{Y}} = 0, \quad \textit{for all} \quad f \in \mathscr{R}[G] \oplus \mathscr{R}[UG]^{\perp}. \tag{9.111}$$

If $\mathscr{R}[G]$ is closed, then $\{e_k\}_{k\geq 0}$ converges to zero in the weak topology in $\mathscr{Y}$ defined by the inner product $\langle\cdot,\cdot\rangle_{\mathscr{Y}}$. Otherwise weak convergence is assured if the sequence $\{\|e_k\|_{\mathscr{Y}}\}_{k\geq 0}$ is bounded.

Proof The first statement follows as the two norms $\|e\|_0$ and $\|G^*e\|_{\mathscr{U}}$ in $\mathscr{Y}$ are topologically equivalent. To prove the second, note that it is necessary only to prove the statement for $e_k - e_0^{(2)} \in \overline{\mathscr{R}[UG]}$. The first step is to prove that

$$\mathscr{Y} = \overline{\mathscr{R}[G]} \oplus ker[(UG)^*] = \overline{\mathscr{R}[G]} \oplus \mathscr{R}[UG]^{\perp} \tag{9.112}$$

by using Theorem 9.13 and noting that the invertibility of U implies that $\hat{U} + \hat{U}^* \geq \varepsilon_0^2\hat{U}^*\hat{U} \geq \tilde{\varepsilon}_0^2 I$ for some $\tilde{\varepsilon}_0^2 > 0$. Again using Theorem 9.13 with the replacements $G \mapsto UG$, $U \mapsto \hat{U}$ and $\varepsilon_0 \mapsto \tilde{\varepsilon}_0$, it is then deduced that $\mathscr{Y} = \overline{\mathscr{R}[G]} \oplus ker[(UG)^*] = \overline{\mathscr{R}[G]} \oplus \mathscr{R}[UG]^{\perp}$ as required. Let $f = f_1 + f_2$ with $f_1 = Gf_3 \in \mathscr{R}[G]$ and $f_2 \in \mathscr{R}[UG]^{\perp}$ so that $\langle f, e_k - e_0^{(2)}\rangle_{\mathscr{Y}} = \langle f_1, e_k - e_0^{(2)}\rangle_{\mathscr{Y}} = \langle f_3, G^*(e_k - e_0^{(2)})\rangle_{\mathscr{Y}} \to 0$ as

$k \to \infty$ if $\lim_{k\to\infty} \|e_k\|_0 = 0$. Weak convergence then follows if $\mathscr{R}[G] = \overline{\mathscr{R}[G]}$. Otherwise, boundedness of the sequence $\{\|e_k\|_{\mathscr{Y}}\}_{k\geq 0}$ permits the extension of this result to include cases where $f_1 \in \overline{\mathscr{R}[G]}$. □

Using the new topology, $\{\|e_k\|_{\mathscr{Y}}\}_{k\geq 0}$ will typically not be monotonic. However, the monotonicity of $\{\|e_k - e_0^{(2)}\|_0\}_{k\geq 0}$ can be proved in the following form,

Theorem 9.15 (Monotonicity and Robustness with Left Multiplicative Errors) *Consider NOILC Algorithm 9.1 in its feedforward implementation. Using the notation defined above, a sufficient condition for the norm sequence* $\{\|e_k - e_0^{(2)}\|_0\}_{k\geq 0}$ *to satisfy the monotonicity condition* $\|e_{k+1} - e_0^{(2)}\|_0 < \|e_k - e_0^{(2)}\|_0$, *for all* $k \geq 0$, *in the presence of the left multiplicative modelling error* U, *is that* $e_0^{(1)} \neq 0$ *and that there exists a real number* $\varepsilon_0^2 > 0$ *such that*

$$\textit{Condition One} - U + U^* \geq U^*(I-L)U + \varepsilon_0^2 I, \qquad \textit{on } \mathscr{R}[G]. \tag{9.113}$$

In these circumstances the subspace decomposition $\mathscr{Y} = ker[G^*] \oplus \overline{\mathscr{R}[UG]}$ *is valid and, in the new topology,*

1. *the induced norm of the restriction of* L_U *to the* L_U*-invariant subspace* $\overline{\mathscr{R}[UG]}$ *satisfies*
$$\|L_U\|_0 \leq 1. \tag{9.114}$$
2. *In particular, if* $\mathscr{R}[G]$ *is finite dimensional, then this norm is strictly less than unity and, in the original norm topology of* $\mathscr{Y}$,
$$\textit{for all starting conditions } e_0 \in \mathscr{Y}, \qquad \lim_{k\to\infty} e_k = e_0^{(2)} \in ker[G^*]. \tag{9.115}$$

Proof The subspace representation follows from Theorem 9.13 and it is clear that $e_{k+1} - e_0^{(2)} = L_U(e_k - e_0^{(2)})$ for $k \geq 0$. Next, note that, if $e_0^{(1)} = 0$, then $e_k = e_0$ for all $k \geq 0$. Suppose therefore that $e_0^{(1)} \neq 0$ and note that $ker[I-L] = ker[G^*]$ so that $\overline{\mathscr{R}[I-L]} = ker[G^*]^{\perp} = \overline{\mathscr{R}[G]}$. Examine the operator $\Lambda : \mathscr{Y} \to \mathscr{Y}$ defined by

$$\begin{aligned}\Lambda &= (I-L) - L_U^*(I-L)L_U = (I-L)\left(U + U^* - U^*(I-L)U\right)(I-L)\\ &\geq \varepsilon_0^2 (I-L)^2\end{aligned} \tag{9.116}$$

and note that, as a consequence, $\langle e, \Lambda e\rangle_{\mathscr{Y}} \geq \varepsilon_0^2 \|(I-L)e\|_{\mathscr{Y}}^2$. In particular, $\langle e, \Lambda e\rangle_{\mathscr{Y}} > 0$ on any subset that intersects with $ker[G^*]$ at $e = 0$ only. The relevant example of such a set is $\overline{\mathscr{R}[UG]}$ from which, for all $e \in \overline{\mathscr{R}[UG]}$,

$$\langle e, \Lambda e\rangle_{\mathscr{Y}} > 0 \quad \textit{which is just} \quad \|e\|_0^2 > \|L_U e\|_0^2 \quad \textit{on } \overline{\mathscr{R}[UG]} \tag{9.117}$$

which proves monotonicity. The observation that $\|L_U\|_0 \leq 1$ on $\overline{\mathscr{R}[UG]}$ follows from the definition of operator norms whilst strict inequality $\|L_U\|_0 < 1$, and the

convergence property, when $\overline{\mathscr{R}[UG]}$ is finite dimensional, follows from the compactness of the unit sphere in any finite dimensional space Hilbert space. □

The following result follows using similar techniques to Theorem 9.8,

Theorem 9.16 (Alternative, Simplified Robustness Conditions) *Two sufficient conditions for Condition One of Theorem 9.15 to be valid for some choice of $\varepsilon_0 > 0$, are as follows*

$$\begin{array}{l} \textit{Condition Two} \ : \quad U + U^* \geq \frac{\varepsilon^{-2}\|G^*\|^2}{1+\varepsilon^{-2}\|G^*\|^2} U^*U + \varepsilon_0^2 I \ \textit{ on } \ \mathscr{R}[G] \\ \textit{Condition Three} \ : \ U + U^* \geq \varepsilon^{-2} U^*GG^*U + \varepsilon_0^2 I \quad \textit{ on } \ \mathscr{R}[G] \end{array} \tag{9.118}$$

Finally, another sufficient condition is that U has a bounded inverse $\hat{U}$ on $\mathscr{Y}$, and, with a suitable choice of $\varepsilon_0 > 0$ and $\theta \in [0, 1]$,

$$\begin{array}{c} \textit{Condition Four} \ : \hat{U} + \hat{U}^* \geq \theta\beta_I I + (1-\theta)\beta_G GG^* + \varepsilon_0^2 I \ \textit{ on } \ \mathscr{Y} \\ \textit{where} \quad \beta_I = \frac{\varepsilon^{-2}\|G^*\|^2}{1+\varepsilon^{-2}\|G^*\|^2} \quad \textit{and} \quad \beta_G = \varepsilon^{-2}. \end{array} \tag{9.119}$$

Note that Condition Four is related to conditions Two and Three by choice of $\theta = 0$ or $\theta = 1$. A simple illustration of this fact uses algebraic manipulation to give $U+U^* \geq \theta\beta_I U^*U+(1-\theta)\beta_G U^*GG^*U+\varepsilon_0^2 U^*U$ on $\mathscr{Y}$ and, noting that $U^*U \geq \alpha I$ for some $\alpha > 0$, then replacing ε_0^2 by $\alpha\varepsilon_0^2$.

9.2.5 Discrete Systems with Left Multiplicative Modelling Errors

The results derived for the case of right multiplicative perturbations carry over to discrete state space systems with a few minor changes. More precisely, consider the case when G can be represented by a discrete time, linear, time invariant, state space model $S(A, B, C, D)$ (or its equivalent supervector description) on the interval $0 \leq t \leq N$ and that it has the transfer function matrix $G(z)$. Using the notation of Sect. 9.2.3, the actual plant model is now assumed to take the form UG with left multiplicative modelling error $U : \mathscr{Y} \to \mathscr{Y}$ with state space model $S(A_U, B_U, C_U, D_U)$ and $m \times m$ transfer function matrix $U(z)$. Again, $\mathscr{Y}$ and $\mathscr{U}$ are finite dimensional and hence every vector subspace is closed.

The following results can be proved by requiring the operator inequalities to be valid in the whole space $\mathscr{Y}$ or $\mathscr{U}$ as appropriate. Also note that the finite dimensional nature of the spaces and the compactness of the unit circle in the complex plane make it possible to replace $\varepsilon_0 > 0$ by $\varepsilon_0 = 0$ provided that the symbol $\geq$ is replaced by the strict inequality $>$.

Theorem 9.17 (Condition Two for Discrete, State Space System Applications) *Let the model G, the actual plant UG and the NOILC objective function be as described*

above. Then a sufficient condition for Condition Two of Theorem 9.16 to hold is that $U(z)$ is asymptotically stable and also that, in the complex Euclidean topology in $\mathscr{C}^m$, the matrix inequality

$$QU(z) + U^T(z^{-1})Q > \beta_I U^T(z^{-1})QU(z), \text{ for all } |z| = 1, \tag{9.120}$$

is satisfied where the "gain" parameter

$$\beta_I = \frac{\varepsilon^{-2}\|G^*\|^2}{1 + \varepsilon^{-2}\|G^*\|^2}. \tag{9.121}$$

In these circumstances, the Feedforward NOILC algorithm converges to the component $e_0^{(2)}$ of e_0 in $ker[G^*]$. The convergence of $e_k - e_0^{(2)}$ to zero is monotonic in the norm $\|e\|_0 = \sqrt{\langle e, (I-L)e\rangle}$ on $\mathscr{R}[UG]$.
Finally, for computational purposes, $\|G^*\|$ can be replaced in the formula for β_I by its upper bound

$$\|G^*\| = \|G\| \leq \sup_{|z|=1} r\left(R^{-1}G^T(z^{-1})QG(z)\right), \tag{9.122}$$

a relationship that links the frequency domain condition to the weighting matrices Q and R.

The proof is left as an exercise for for the reader to fill in. The following result is also proved in a similar matter to Theorem 9.12,

Theorem 9.18 (Condition Four in Discrete State Space System Applications) *A sufficient condition for Condition Four to hold is that $G(z)$ is asymptotically stable, $U(z)$ is invertible and minimum-phase and also that there exists a $\theta \in [0, 1]$ such that, on $\mathscr{C}^m$, for all $|z| = 1$,*

$$Q\hat{U}(z) + \hat{U}^T(z^{-1})Q > \theta\beta_I Q + (1-\theta)\beta_G QG(z)R^{-1}G^T(z^{-1})Q. \tag{9.123}$$

In these circumstances, the Feedforward NOILC algorithm converges to the component $e_0^{(2)}$ of e_0 in $ker[G^*]$. The convergence of $e_k - e_0^{(2)}$ is monotonic in the norm $\|e\|_0 = \sqrt{\langle e, (I-L)e\rangle}$ on $\mathscr{R}[UG]$.

9.2.6 Monotonicity in $\mathscr{Y}$ with Respect to the Norm $\|\cdot\|_{\mathscr{Y}}$

This short section is relevant to the situation where the model G is related to the plant by two possible uncertainty descriptions $U_L G$ and GU_R where both the left (respectively, right) multiplicative perturbation U_L (respectively, U_R) satisfy the conditions of the previous sections for monotonicity in the relevant norm topology. It is assumed here that $\overline{\mathscr{R}[G]} = \mathscr{Y}$ and $\overline{\mathscr{R}[G^*]} = \mathscr{U}$ so that $ker[G^*] = \{0\}$ and hence $e_0^{(2)} = 0$.

The monotonicity conditions become, for all $e_0 \in \mathscr{Y}$,

$$\langle e_{k+1}, Le_{k+1}\rangle_{\mathscr{Y}} < \langle e_k, Le_k\rangle_{\mathscr{Y}} \quad \textit{and} \quad \langle e_{k+1}, (I-L)e_{k+1}\rangle_{\mathscr{Y}} < \langle e_k, (I-L)e_k\rangle_{\mathscr{Y}}, \tag{9.124}$$

so that monotonicity with respect to the norm $\|\cdot\|_{\mathscr{Y}}$ follows by adding to give

$$\|e_{k+1}\|^2_{\mathscr{Y}} < \|e_k\|^2_{\mathscr{Y}} \quad \textit{for all} \quad k \geq 0. \tag{9.125}$$

The reader will also note that this ensures boundedness but, if $\mathscr{R}[G]$ is finite dimensional, it also ensures that $\lim_{k\to\infty} e_k = 0$. Finally,

Theorem 9.19 (Monotonicity of $\|e_k\|_{\mathscr{Y}}$ with Commuting Error Models) *If U_L commutes with G (as is the case with linear, single-input, single-output, time-invariant, state space systems), $U_R = U_L$ and satisfaction of any of the robustness conditions guarantees monotonicity of the norm sequence $\{\|e_k\|_{\mathscr{Y}}\}_{k\geq 0}$.*

9.3 Non-minimum-phase Properties and Flat-Lining

Previous chapters (Sect. 8.2.4) have noted that, for single-input, single-output, discrete time state space systems, the presence of non-minimum-phase zeros has a very specific effect on Iterative Control convergence if the number of samples $N+1$ is large. It takes the form of initially good norm reductions followed by a "plateauing" (or "flat-lining") phenomenon where the norm $\|e_k\|_{\mathscr{Y}}$ is non-zero but begins to reduce infinitesimally slowly. Convergence to zero is still guaranteed theoretically but this slow convergence, in practical terms, means that no further realistic improvement in tracking error is possible using the chosen algorithm. Indeed, tens of thousands of iterations may be needed to achieve even small further improvements. This can be understood, conceptually, by taking an error norm sequence of the form

$$\|e_k\|_{\mathscr{Y}} = \frac{1}{2}\|e_0\|_{\mathscr{Y}}\left(\lambda_1^k + \lambda_2^k\right) \tag{9.126}$$

where the two positive scalars λ_1 and λ_2 are strictly less than one but $\lambda_1 < \lambda_2$ and λ_2 is very close to unity. If, for example, $\lambda_1 = 0.1$ and $\lambda_2 = 1 - 10^{-8} < 1$, then, after only a few iterations, the first term becomes very small leaving a residual convergence represented by $\|e_k\|_{\mathscr{Y}} = \frac{1}{2}\|e_0\|_{\mathscr{Y}}\lambda_2^k$ which remains very close to $\frac{1}{2}\|e_0\|_{\mathscr{Y}}$ for many thousands of iterations.

In what follows, a brief discussion of the same phenomenon is given with the conclusion that NOILC and gradient algorithms have identical properties in this case. The approach taken is that of constructing a model of the convergence that reveals the plateauing phenomenon by, firstly, identifying subspaces associated with normal convergence rates and slow convergence and then modelling the plateauing effect by setting the slow convergence rate to precisely zero.

The analysis is restricted to the case of single-input, single-output, asymptotically stable, discrete time state space systems $S(A, B, C, D)$ described by the transfer function $G(z) = G_m(z)G_{ap}(z)$ where $G_{ap}(z)$ is all pass and $G_m(z)$ is asymptotically stable and minimum phase. It is assumed that $D \neq 0$ by using the shift techniques of Chap. 4 and hence that the supervector model matrices G and G_m are invertible. The notation of Sect. 8.2.4 is used and the simplest case of $n_+ = 1$ and a single real zero with modulus $|z_1| > 1$ is assumed to simplify the discussion. Theorem 8.3 in Sect. 8.2.4 is particularly relevant to the development.

Using supervector descriptions, GG^* can be identified with $\varepsilon^{-2}R^{-1}QGG^T$ which takes the form of the symmetric matrix $\varepsilon^{-2}R^{-1}QG_mG_{ap}G_{ap}^TG_m^T$. The factor $G_{ap}G_{ap}^T$ is the key to understanding the phenomenon. More precisely, its eigenvalues are all equal to unity apart from a single eigenvalue σ_1^2 which takes the value $z_1^{-2(N+1)} \to 0$ as $N \to \infty$. This eigenvalue has eigenvector $\alpha_1 = [1, z_1^{-1}, \ldots, z_1^{-N}]^T$ spanning the subspace $\mathscr{E}_{a+}$. The orthogonal complement of this subspace is denoted by $\mathscr{E}_1$ and, if $V : \mathscr{R}^N \to \mathscr{R}^{N+1}$ is an $(N+1) \times N$ matrix whose columns span $\mathscr{E}_1$, then $V^T\alpha_1 = 0$. Without loss of generality, V can be constructed to satisfy $V^TV = I_N$.

The key to understanding the behaviour when $N \gg 0$ is obtained by using the approximation $\sigma_1 = 0$ and using the orthogonal subspace decomposition

$$\mathscr{Y} = G_m\mathscr{E}_1 \oplus (G_m^T)^{-1}\mathscr{E}_{a+}. \tag{9.127}$$

Note that, with these assumptions, $GG^T\left((G_m^T)^{-1}\alpha_1\right) = \sigma_1^2 G_m\alpha_1 = 0$. Also, note that the matrix $(G_mV)^TG_mV \geq \varepsilon_0^2 I$ for all $N \geq 1$ and some $\varepsilon_0^2 > 0$. This suggests the approximation $G_{ap}G_{ap}^T = VV^T$ and hence

$$L = (I + GG^*)^{-1} \approx L_A = (I_{N+1} + \varepsilon^{-2}R^{-1}QG_mVV^TG_m^T)^{-1} \tag{9.128}$$

will be accurate for a large number of iterations when N is very large. In particular, $L_A(G_m^T)^{-1}\alpha_1 = (G_m^T)^{-1}\alpha_1$ and hence, if e_0 is written in the form $e_0 = e_0^{(1)} + e_0^{(2)}$ with $e_0^{(1)} = G_mV\gamma_1$ and $e_0^{(2)} = (G_m^T)^{-1}\alpha_1\gamma_2$ for some scalar γ_2 and vector $\gamma_1 \in \mathscr{R}^N$, a simple computation gives

$$\begin{aligned} e_k = L^k e_0 &\approx L_A^k G_mV\gamma_1 + e_0^{(2)} \\ &= G_mV(I_N + \varepsilon^{-2}R^{-1}QV^TG_m^TG_mV)^{-k}\gamma_1 + e_0^{(2)}, \end{aligned} \tag{9.129}$$

and the approximation predicts the convergence to the apparent limit $e_\infty^{pseudo} = e_0^{(2)}$ as $k \to \infty$ as $\|(I_N + \varepsilon^{-2}R^{-1}QV^TG_m^TG_mV)^{-1}\| \leq (1 + \varepsilon^{-2}R^{-1}Q\varepsilon_0^2)^{-1} < 1$. This apparent limit vector is a close approximation to the errors that will be observed as the flat-lining phenomenon begins. That is, it is closely associated with the plateau which

consists of slow convergence from the value $\|e_0^{(2)}\|_{\mathscr{Y}}$ to zero. A simple calculation using orthogonality of $\mathscr{E}_1$ and $\mathscr{E}_{a+}$ gives

$$e_\infty^{pseudo} = \frac{\alpha_1^T G_m^{-1} e_0}{\alpha_1^T G_m^{-1} (G_m^T)^{-1} \alpha_1} (G_m^T)^{-1} \alpha_1 = \frac{\alpha_1^T G_m^{-1} e_0}{\alpha_1^T (G_m^T G_m)^{-1} \alpha_1} (G_m^T)^{-1} \alpha_1. \quad (9.130)$$

Its value is linked to the vector α_1 but is also influenced by the nature of the minimum-phase component G_m. In this simple case, the nature of the signal can be computed by using the properties of time reversal operators $\mathscr{T}$ (defined in Sect. 4.3) to write $G_m^T = \mathscr{T} G_m \mathscr{T}$ and hence $(G_m^T)^{-1} = \mathscr{T} G_m^{-1} \mathscr{T}$. That is, the elements of $(G_m^T)^{-1}\alpha_1$ can be computed from the time reversal of the signal obtained from the first $N+1$ components of the signal $G_m^{-1}(z)\hat{\alpha}_1(z)$ where $\hat{\alpha}_1(z)$ is the $\mathscr{Z}$-transform of the signal obtained from the time reversal of α_1, on $0 \le t \le N$, which is then extended to the infinite interval. As the time reversal of α_1 is just $z_1^{-N}\left[1, z_1, \ldots, z_1^N\right]^T$, the natural extension is $\hat{\alpha}_1(t) = z_1^{-N} z_1^t$ for $t \ge N+1$ which gives $\hat{\alpha}_1(z) = z_1^{-N}\,(z/(z-z_1))$. Partial fraction methods indicate that

$$G_m^{-1}(z)\hat{\alpha}_1(z) = G_m^{-1}(z_1)\hat{\alpha}_1(z) + z_1^{-N}\psi(z, z_1) \quad (9.131)$$

where $\psi(z, z_1)$ is uniformly bounded on the unit circle $|z| = 1$ and for all N. If N is large, z_1^{-N} is very small and it follows that the time reversal of $G_m^{-1}\hat{\alpha}_1$ on $0 \le t \le N$ is very close to the time series $G_m(z_1)^{-1}\alpha_1$ which is just

$$(G_m^T)^{-1}\alpha_1 \approx G_m(z_1)^{-1}\left[1, z_1^{-1}, \ldots, z_1^{-N}\right]^T \quad when \quad N \gg 0. \quad (9.132)$$

This time series is defined by the inverse of the zero. Finally, using this approximation, the plateau is characterized by the limit

$$e_\infty^{pseudo} = \frac{\alpha_1^T G_m^{-1} e_0}{\alpha_1^T G_m^{-1} (G_m^T)^{-1} \alpha_1} (G_m^T)^{-1} \alpha_1 \approx \frac{\alpha_1^T e_0}{\alpha_1^T \alpha_1} \alpha_1. \quad (9.133)$$

which is precisely the limit computed for the stable inverse algorithm in Sect. 8.2.4. This signal is proportional to the stable time series $\alpha_1 = \left[1, z_1^{-1}, \ldots, z_1^{-N}\right]^T$ with the constant of proportionality

$$\frac{\alpha_1^T e_0}{\alpha_1^T \alpha_1} = \left(\frac{1 - z_1^{-2(N+1)}}{1 - z_1^{-2}}\right) \sum_{t=0}^{N} z_1^{-j} e_0(t) \quad (9.134)$$

which is small only if z_1 is large and/or $e_0(t)$ is small on the initial time interval and gaining significant values only when $|z_1^{-t}| \ll 1$.

9.4 Discussion and Further Reading

9.4.1 Background Comments

The introduction of the use of optimization-based Iterative Learning Control was originally introduced in the general form described here in [6] by the author and co-workers and has led to a number of approaches and proposed generalizations (for example, [17, 44, 64]). Its aim was to create algorithms capable of achieving monotonic convergence for a wide class of linear systems. The use of operator notation in a Hilbert space setting makes this possible with the key assumptions of linearity, boundedness of the operator G and the use of objective functions that are quadratic in the error norm and input change. This structure requires the use of the adjoint operator to compute optimal minimizing solutions described by equations linear in the signals of interest. Crucially, it also permits significant generality whilst providing a natural link with linear, quadratic, optimal control theory for continuous systems [6] and discrete systems [5]. For notational convenience, only examples taken from linear time-invariant, state space systems have been considered but the reader will note that this assumption is easily removed to create convergent NOILC algorithms for "time varying" state space models illustrated by

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(0) = x_0, \quad y(t) = C(t)x(t) \tag{9.135}$$

using the minimization $u_{k+1} = \arg\min J(u, u_k)$ with weights $Q(t)$ and $R(t)$ as defined in Sect. 9.1.6.

The potentially wide application to different model types is a good feature of NOILC but it also acts as a source of new ideas. The next chapter considers examples of new algorithms where NOILC theory provides iterative control solutions of problems that introduce addition features to this basic idea and/or relax some of the assumptions. These include multi-rate sampling, variations of initial conditions, the notions of intermediate point and multi-task problems and the use of predictive control methodologies to improve convergence rates using multi-models. Other model types can be considered in these approaches including differential delay systems exemplified by the state equations

$$\begin{aligned} &\frac{dx(t)}{dt} = A(t)x(t) + A_0(t)x(t-\tau) + B(t)u(t), \\ &x(t) = x_0(t), \quad t \in [-\tau, 0] \quad \textit{and} \\ &y(t) = C(t)x(t) + D(t)u(t), \end{aligned} \tag{9.136}$$

where $\tau > 0$ is a state time delay. Further information on time delay systems can be found in, for example, [50]. Note the need for a function $x_0(t)$ to define the initial condition. This model can be identified with a bounded mapping G with the same input and output spaces as those used in the delay free case. The optimal solution

has the same characterization in terms of G^* but the conversion of the update law into a causal form is more complex.

9.4.2 Practical Observations

Before concluding this chapter, it is worthwhile noting the following points:

1. **Applications Potential**: The practical viability of NOILC has been verified in applications to physical plant as in [13, 29, 97, 100] and Freeman et al. [42].
2. **Nonlinearity**: The algorithm $u_{k+1} = \arg\min J(u, u_k)$ provides monotonic error reductions even if the plant model is nonlinear. The optimization problem is then more computationally demanding, its solution may be non-unique, convergence to the reference signal cannot be guaranteed in general and the analytical tools that permitted a detailed analysis of algorithm properties in the linear case are no longer available.
3. **Effect of Parameters**: The theory shows the place that Q, R and ε^2 play in NOILC algorithms and, in the SISO case, the choice of suitable values can be guided by concepts such as that of spectral bandwidth. In MIMO systems, the choice of the "internal structure" of the matrices Q and R is less clear at a theoretical level. Simple guidelines suggest that Q influences the relative convergence rates in the control loops whilst R provides some control over the changes in control signal at each iteration. Where solutions are non-unique, R can be used to reflect the preferred inputs to be used to solve the tracking problem. For example, if R is diagonal, increasing R_{jj} will penalize changes in a loop input $u_j(t)$ and hence encourage the algorithm to use inputs in the other loops to achieve the error reduction sought. These connections are, however, imprecise and detailed choices will inevitably often involve a degree of trial and error or experience with the domain of application.
4. **Exponentially Weighted Signals**: As in the case of inverse and gradient algorithms, the use of exponentially weighted signals can be included in the algorithm in both the continuous time and discrete time cases. For continuous time systems, the objective function, with $\lambda > 0$, is

$$\int_0^T e^{-2\lambda t} \times \Big((r(t) - y(t))^T Q(t)(r(t) - y(t)) + (u(t) - u_k(t))^T R(t)(u(t) - u_k(t))\Big)\, dt \tag{9.137}$$

and provides (additional) time dependent weighting $e^{-2\lambda t}$ for $S(A, B, C)$. Equivalently, the weighting can be removed and the system replaced by $S(A - \lambda I, B, C)$ with signals u, x, y, r, e replaced by weighted signals defined by the map $f(t) \mapsto e^{-\lambda t} f(t)$. This simple modification could, in principle, be used to reduce the

plateauing effect caused by non-minimum-phase properties but the nature of the convergence is changed as greater emphasis will be placed on error reductions early in the interval $[0, T]$ before later errors can take any priority.

5. **Iteration Variation of Parameters**: The parameters Q, R and ε could, in practice be varied from iteration to iteration to speed up or slow down convergence speeds. The theory still guarantees norm reductions at each iteration but, as that norm may be changing as Q changes, little can be said about the nature of the convergence without additional assumptions. Note that the Riccati solution will need to be recomputed whenever any one or more of Q, R and ε is changed. This idea will be revisited in Chap. 13 in the "Notch Algorithm" and in the "Parameter Optimal" methods in Chap. 14.

9.4.3 Performance

It can be argued that *NOILC is a benchmark ILC algorithm* in the sense that it has defined properties and guaranteed convergence for a wide class of linear applications. Its application to non-minimum-phase (NMP) systems is influenced by the emergence of two stage convergence. This consists of an initially good convergence rate followed by a rate that is so slow that, in practice, no further benefits can be achieved. For minimum-phase systems, the algorithm is therefore capable of excellent convergence properties but acceptable behaviour in the NMP case will depend on the zero positions and the reference signal in order to ensure that the "plateau" value for $\|e_k\|_{\mathscr{Y}}$ represents acceptable tracking accuracy. A more detailed analysis of these phenomena is given in the reference [84] for discrete state space systems and in [86] for the continuous case. The two analyses suggest that large zeros and/or reference signals that are small over a suitably large initial sub-interval can lead to outcomes where tracking accuracy is very good. The validity of the predictions of the analysis has been tested, successfully, in a laboratory environment [86] and also in unreported industrial tests, the outcomes of which are subject to commercial confidentiality agreements.

9.4.4 Robustness and the Inverse Algorithm

Next, as in the case of inverse model and gradient algorithms, modelling errors also influence the nature of the convergence. The interesting conclusion arising in this chapter is that, assuming that robustness will be assessed in terms of monotonicity of signal norms in the presence of multiplicative modelling errors U, the error norm $\|e\|_{\mathscr{Y}}$ is not necessarily the most appropriate norm for analysis. For the situations considered, the choice of norm seems to depend on the nature of the modelling error as exemplified by the choice of error norm $\|e\|_0 = \sqrt{\langle e, Le\rangle_{\mathscr{Y}}}$ with $L = (I + \varepsilon^{-2}GG^*)^{-1}$ for asymptotically stable right multiplicative perturbations. The nature of the new form of monotonicity will therefore, in general, need to be considered to make sure that it is acceptable for the application domain. A unifying theme is that,

the robustness of NOILC can be guaranteed by positivity tests that are algebraically identical to those that guarantee robust convergence of inverse model and/or gradient algorithms,

(Chapters 6 and 7). There seems to be no escaping the need for positivity of the multiplicative modelling error U although, being expressed as a positivity condition on $U + U^*$, the use of the topology in the relevant Hilbert space (in the form of Q and R for example) may be helpful. The analysis of robustness is technically complex and there may be room for improvements in and extensions to the results presented. The chosen material was aimed at three objectives, namely,

1. that of achieving a degree of generality at the operator theoretical level to illustrate and underline the robustness implicit in the NOILC paradigm and
2. providing guidance on the technical issues that separate the cases where the output space $\mathscr{Y}$ is either finite or infinite dimensional.
3. Crucially, the approach permits the construction of more familiar frequency domain tests for discrete time, state space systems and, in particular, linking these tests to structural properties of the plant model such as pole and zero positions.

9.4.5 Alternatives?

The reader may have a view on the choices made by the author. Alternatives could modify the form of the objective function or the characterization of the modelling errors, for example, using errors in the parameters in the model [3]. The choice of objective function has received attention [17, 64] with the change

$$J(u, u_k) \;\mapsto\; \|e\|^2_{\mathscr{Y}} + \varepsilon^2\|u - u_k\|^2_{\mathscr{U}} \;+\; \varepsilon_1^2\|u|^2_{\mathscr{U}}, \quad \varepsilon_1^2 > 0, \tag{9.138}$$

where the last term is included to constrain the magnitude of the inputs that are used. In this case $\|e\|^2_{\mathscr{Y}} + \varepsilon_1^2\|u|^2_{\mathscr{U}}$ reduces monotonically each iteration and the limit error cannot be zero. The optimal solution on iteration $k + 1$ is

$$u_{k+1} = \frac{\varepsilon^2}{\varepsilon^2 + \varepsilon_1^2}u_k + \frac{1}{\varepsilon^2 + \varepsilon_1^2}G^*e_{k+1} \tag{9.139}$$

which is a version of the feedback relaxed NOILC Algorithm 9.2 where ε^2 is replaced by $\varepsilon^2 + \varepsilon_1^2$ and $\alpha = \frac{\varepsilon^2}{\varepsilon^2+\varepsilon_1^2}$.

Another example, [102] added data from a number of iterations and an optimization over a number of parameters in the objective function

$$\|e\|^2_{\mathscr{Y}} \;+\; \|u - \sum_{j=1}^{n_p}\alpha_{k+1,j}u_{k+1-j}\|^2_{\mathscr{U}}, \quad \textit{with} \;\; \sum_{j=1}^{n_p}\alpha_{k+1,j} = 1 \tag{9.140}$$

representing linear combinations of n_p previous input signals. The options are many and include a multi-objective approach proposed by Tousain et al. [108].

9.4.6 Q, R and Dyadic Expansions

Finally, the evolution of individual frequency components in the error provides an interesting approximation, in the frequency domain, in terms of the eigenvalues $\sigma_j^2(z)$ of $G(z)R^{-1}G^T(z^{-1})Q$. It opens up the possibility of using it as a basis for the choice of Q and R, particularly if one particular frequency z_s is very important to design. For example, suppose that $m = \ell$. Using the concept of Dyadic Expansions [80, 81], consider the case where $G(z_s)G^{-1}(\overline{z}_s)$ only has eigenvalues $\{\zeta_j\}_{1\leq j\leq m}$ of unit modulus, then there exists real, nonsingular, $m \times m$ matrices P_1 and P_2 such that

$$G(z_s) = P_1 \Lambda P_2 \quad with \quad \Lambda = diag[\zeta_1, \zeta_2, \ldots, \zeta_m]. \tag{9.141}$$

Noting that $G(z)R^{-1}G^T(z^{-1})Q = P_1 \Lambda P_2 R^{-1} P_2^T \overline{\Lambda} P_1^T Q$ and $\Lambda\overline{\Lambda} = I_m$ then gives,

Theorem 9.20 (NOILC, *Q, R* and Dyadic Expansions) *Suppose that $m = \ell$, $|z_s| = 1$ and let P_0 be any choice of real, positive definite, diagonal $m \times m$ matrix. Using the notation and assumptions described above, then $\sigma_j^2(z_s) = 1,\ 1 \leq j \leq m$, and hence $G(z_s)R^{-1}G^T(z_s^{-1})Q = I_m$ if*

$$P_2 R^{-1} P_2^T = P_0 \quad and \quad P_1 P_0 P_1^T Q = I_m. \tag{9.142}$$

The result suggests that the frequency component of the error signal at $z = z_s$ will be attenuated at a rate $(1 + \varepsilon^{-2})^{-k}$. The degrees of freedom implicit in P_0 leave a number of open questions that would need resolution for wide application. The approach places constraints on the choice of Q and/or R as, for example, it is unlikely that the relations above will allow diagonal choices. Also, [80], it is known that, although Dyadic Expansions apply more generally, the assumptions of unit modulus eigenvalues for $G(z_s)G^{-1}(\overline{z}_s)$ does not describe every situation. Further research is needed to cover such cases.

Chapter 10
NOILC: Natural Extensions

Norm Optimal Iterative Learning Control has great generality in its formulation, convergence properties and robustness characterizations. The two examples of linear, continuous and discrete time, state space systems provide a link to applications and design and underline the algorithm as a viable approach for practice. The ideas apply much more generally but the user may have to think very clearly about how an Iterative Control problem can be formulated as a NOILC problem. The essential ingredients are the choice of $\mathscr{Y}$ and $\mathscr{U}$ (and their Hilbert space topologies), the construction of the operator G and its adjoint G^* and the solution of the optimality conditions to form an implementable, real-world solution. This leads to many possibilities, some of which are explored in this chapter. The theoretical content retains the, now familiar, NOILC structure including the objective function $J(u, u_k)$ and relies heavily on the update relation $u_{k+1} = u_k + \varepsilon^{-2}G^*e_{k+1}$ and error evolution characterization $e_{k+1} = Le_k$ where $L = (I + \varepsilon^{-2}GG^*)^{-1}$.

10.1 Filtering Using Input and Error Weighting

For systems evolving on the infinite discrete time interval $0 \leq t < \infty$, there may be benefits in measuring errors by a frequency domain-based norm

$$\|e\|^2 = \frac{1}{2\pi i}\oint_{|z|=1} e^T(z^{-1})F_e^T(z^{-1})QF_e(z)e(z)\frac{dz}{z} \tag{10.1}$$

where $F_e(z)$ is a minimum-phase, asymptotically stable, proper filter. The choice of F_e can be based on a desire to place greater importance on some frequency ranges. For example, in the single-input, single-output case, ensuring that its low frequency gains are higher than its high frequency gains will place emphasis on reducing the low frequency components of the tracking error. Iterative Control, however, is, inevitably,

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_10

on a finite interval and hence the frequency domain interpretation (in terms of contour integrals) is imprecise. The use of such filters in the time domain can however still be considered. In such circumstances, an m-input, m-output state space model $S(A_e, B_e, C_e, D_e)$ can be used and F_e can be regarded as an operator on $\mathscr{Y}$. This observation motivates the general discussion that follows.

Suppose that F_e (respectively, F_u) is a bounded filter mapping $\mathscr{Y}$ (respectively, $\mathscr{U}$) onto itself and that it has a bounded inverse. A norm on $\mathscr{Y}$ (respectively $\mathscr{U}$) can be defined by

$$\|e\|_{0e}^2 = \|F_e e\|_{\mathscr{Y}}^2 \quad (\textit{respectively} \quad \|u\|_{0u}^2 = \|F_u u\|_{\mathscr{U}}^2) \tag{10.2}$$

The norm $\|\cdot\|_{0e}$ (respectively, $\|\cdot\|_{0u}$) is topologically equivalent to the norm $\|\cdot\|_{\mathscr{Y}}$ (respectively, $\|\cdot\|_{\mathscr{U}}$) and hence convergence with respect to one norm guarantees convergence with respect to the other.

Algorithm 10.1 (*A Filtered NOILC Algorithm*) A NOILC Algorithm 9.1 using the filtered signals can be stated as the computation of the input update u_{k+1} from the data u_k as the input minimizing the objective function

$$J_0(u, u_k) = \|e_{k+1}\|_{0e}^2 + \varepsilon^2 \|u - u_k\|_{0u}^2 \tag{10.3}$$

subject to the dynamics $y = Gu + d$.
This algorithm can be restated in an alternative, and more familiar, form by defining signals $w = F_e y$, $r_e = F_e r$ and $v = F_u u$ and updating the signal v at each iteration. More precisely, the update v_{k+1} of v_k minimizes the objective function

$$J(v, v_k) = \|r_e - w_{k+1}\|_{\mathscr{Y}}^2 + \varepsilon^2 \|v - v_k\|_{\mathscr{U}}^2 \tag{10.4}$$

subject to the dynamics represented by the composite operator $F_e G F_u^{-1}$,

$$w = F_e G F_u^{-1} v + F_e d. \tag{10.5}$$

That is,

$$v_{k+1} = v_k + (F_e G F_u^{-1})^*(r_e - w_{k+1}) \quad \textit{from which} \quad u_{k+1} = F_u^{-1} v_{k+1}. \tag{10.6}$$

The evolution of the filtered error has exactly the properties described for NOILC in the previous chapter with G replaced by $F_e G F_u^{-1}$. For state space systems, the construction of a state space representation of $F_e G F_u^{-1}$ by combining the state space models of its components again releases the use of optimal control methods to provide feedback or feedforward solutions.

Although the filtered algorithms inherit all of the beneficial properties of NOILC Algorithm 9.1, no additional general statements can be made as the physical performance of the algorithm will depend on the choice of filters. For discrete time state

space systems, the frequency domain intuition obtained from contour integral representations on infinite intervals can be expected to be a good guide for the choice if N is large as $J(u, u_k)$ can be interpreted as a finite time interval form of the frequency domain objective function

$$\begin{array}{l} \frac{1}{2\pi i}\oint_{|z|=1} e^T(z^{-1})F_e^T(z^{-1})QF_e(z)e(z)\frac{dz}{z} \\ +\varepsilon^2\frac{1}{2\pi i}\oint_{|z|=1} (u(z^{-1})-u_k(z^{-1}))^T F_u^T(z^{-1})RF_u(z)(u(z)-u_k(z))\frac{dz}{z}. \end{array} \tag{10.7}$$

10.2 Multi-rate Sampled Discrete Time Systems

Consider a linear, time invariant, state space system as discussed in Sect. 3.7. Regard the underlying dynamics as that of a uni-rate, discrete time, state space system $S(A, B, C, D)$ written as $y = Gu + d$ in $\mathscr{Y}$. The outputs can be represented by a selector operator $S_M : \mathscr{Y} \to \mathscr{Y}_e$ that maps the underlying output sequence $y = \{y(0), y(1), \ldots, y(N)\}$ into a sequence of measurements $y^e = \{y^e(0), y^e(2), \ldots\}$ with $y^e(t) = y(Mt),\ t \geq 0$. For simplicity, it is here assumed that N is an integer multiple of M so that $\mathscr{Y}_e = \mathscr{R}^{m(N/M+1)}$ and S_M has an $m(N/M+1) \times m(N+1)$ matrix representation consisting of the array of $m \times m$ blocks of the form

$$(S_M)_{ij} = I_m\delta_{M(i-1)+1,j}, \quad 1 \leq i \leq N/M+1, \quad 1 \leq j \leq N+1. \tag{10.8}$$

The resultant multi-rate dynamics are expressed in the form

$$y^e = S_M G u + S_M d. \tag{10.9}$$

Using the inner product and norm in $\mathscr{Y}$ and $\mathscr{U}$ as defined in Eq. (7.16), note that $\|y^e\|_{\mathscr{Y}_e} = \|S_M^T y^e\|_{\mathscr{Y}}$ is a norm in $\mathscr{Y}_e$. Given a reference signal r^e defined at the slow sampled rate, the NOILC problem for the multi-rate system is therefore based on the minimization of an objective function

$$J_e(u, u_k) = \|e^e_{k+1}\|^2_{\mathscr{Y}_e} + \varepsilon^2\|u - u_k\|^2_{\mathscr{U}} = \|S_M^T e^e_{k+1}\|^2_{\mathscr{Y}} + \varepsilon^2\|u - u_k\|^2_{\mathscr{U}} \tag{10.10}$$

where $e^e = r^e - y^e$ is the slow sampled error signal. The process of interpreting this as a recursive optimal control problem consists of regarding r^e as a sampled version of a signal $r \in \mathscr{Y}$ satisfying $r^e = S_M r$. Defining the error $e = r - y$ as the resultant error at the underlying (faster) sample rate, the optimal control computations used in Algorithm 9.4 will successfully solve this multi-rate case by working in terms of the signals u and e and, crucially, using the same objective function but setting $Q(t) = 0$ whenever the underlying sample rate index t is not equal to any of the values $0, M, 2M, \ldots$. Note that, although e contains signals that are not measured if $M > 1$, these signals disappear from the computations and are not required in control input evaluations.

Finally, two simple observations can be made

1. A simple extension of this multi-rate case is obtained if the sampling is non-uniform in the sense that the measured output $y^e = \{y(t_0), y(t_1), y(t_2), \ldots\}$ consists of the time series created when y is sampled at times $0 \leq t_0 < t_1 < t_2 < \cdots$. The outcome of the analysis is then that Algorithm 9.4 can again be used by setting $Q(t) = 0$ whenever the slow rate sample index t is not equal to any of the values $t_0, t_1, t_2, \ldots$. This idea will be discussed further when the Intermediate Point NOILC Algorithm is considered in Sect. 10.5.
2. If each output of a multi-input, multi-output system is sampled at a different rate (the case of non-uniform sampling across the outputs), the basic constructions used above can be modified by redefining the notion of a selector operator S. The details are omitted but readers now have the machinery to derive the resultant algorithm for themselves.

10.3 Initial Conditions as Control Signals

Consider a linear state space system $S(A, B, C)$ with initial condition for the state $x(0) \in \mathscr{R}^n$. In most applications, $x(0)$ is specified by a known initial state x_0 that is independent of iteration index k. In this section, motivated by the conceptual case of a mechanical system starting from rest but where the starting position can be changed by the user, it is assumed that this is not the case and that the user has some control over its choice. More precisely, the initial condition on iteration k is assumed to lie in a linear variety in $\mathscr{R}^n$ defined by

$$x_k(0) = x_0 + H_0 h_k, \quad k \geq 0, \tag{10.11}$$

where x_0 is fixed, H_0 is a $n \times p$ matrix defining a p-dimensional subspace within which $x_k(0) - x_0$ can vary and $h_k \in \mathscr{R}^p$ is the flexibility in changing the initial condition within that space. If the initial condition can be chosen arbitrarily, then $x_0 = 0$ and $H_0 = I_n$ are suitable choices. If $H_0 = 0$, then the initial condition is independent of iteration.

Taking the case of continuous time, the solution from the initial condition $x_0 + H_0 h$ at $t = 0$ is

$$x(t) = e^{At}(x_0 + H_0 h) + \int_0^t e^{A(t-t')} B u(t') dt', \ \textit{with } y(t) = Cx(t). \tag{10.12}$$

Being free variables, the mathematical "input" to the plant is the pair (u, h) and, in operator notation,

$$y = \mathscr{G}(u, h) + d, \ \textit{where } \mathscr{G}(u, h) = Gu + Hh \tag{10.13}$$

is the linear operator acting on (u, h). Again, G is the familiar convolution operator used in the case of fixed initial condition (when $H_0 = 0$) and $d(t) = Ce^{At}x_0$. The column matrix h is taken to lie in the Hilbert space $\mathscr{X}_0 = \mathscr{R}^p$ with inner product $\langle h_1, h_2\rangle_{\mathscr{X}_0} = h_1^T Q_0 h_2$ (where $Q_0 = Q_0^T > 0$). $H : \mathscr{X}_0 \to \mathscr{Y}$ is then the map

$$\begin{aligned} h \mapsto (Hh)(t) = Ce^{At}H_0 h \quad &\textit{with adjoint defined by} \\ w \mapsto H^* w = Q_0^{-1} \textstyle\int_0^T H_0^T e^{A^T t} C^T Q(t) w(t) dt. \end{aligned} \tag{10.14}$$

Defining the inner product in the "extended input", product space $\mathscr{U} \times \mathscr{X}_0$ by

$$\begin{aligned} \langle (u, h_1), (v, h_2)\rangle_{\mathscr{U}\times\mathscr{X}_0} &= \langle u, v\rangle_{\mathscr{U}} + \langle h_1, h_2\rangle_{\mathscr{X}_0} \\ &= \int_0^T u^T(t) R(t) v(t) dt \; + \; h_1^T Q_0 h_2, \end{aligned} \tag{10.15}$$

gives the adjoint of $\mathscr{G}$ as the operator defined by the relation

$$\begin{aligned} &\mathscr{G}^* e = (G^* e, H^* e) \in \mathscr{U} \times \mathscr{X}_0 \quad \textit{so that} \\ &\mathscr{G}\mathscr{G}^* = GG^* + HH^*. \end{aligned} \tag{10.16}$$

With this notation, a NOILC algorithm that uses both input signals u and initial condition updates to achieve tracking is as follows,

Algorithm 10.2 (*NOILC with Initial Conditions as Additional Controls*) Let the starting values u_0 and h_0 be arbitrarily chosen. Suppose that, on iteration k, the input u_k was used and that the initial condition was generated by h_k, then a convergent NOILC algorithm tracking a reference signal $r(t)$ on $[0, T]$ is obtained by choosing u_{k+1} and h_{k+1} to minimize

$$\begin{aligned} J((u, h), (u_k, h_k)) &= \|e\|_{\mathscr{Y}}^2 + \varepsilon^2 \|(u - u_k, h - h_k)\|_{\mathscr{U}\times\mathscr{X}_0}^2 \\ &= \|e\|_{\mathscr{Y}}^2 + \varepsilon^2 \left(\|(u - u_k\|_{\mathscr{U}}^2 + \|(h - h_k\|_{\mathscr{X}_0}^2 \right) \\ = \int_0^T e^T(t) Q(t) e(t) dt \; &+ \; \varepsilon^2 \int_0^T (u(t) - u_k(t))^T R(t) (u(t) - u_k(t)) dt \\ &+ \; \varepsilon^2 (h - h_k)^T Q_0 (h - h_k), \end{aligned} \tag{10.17}$$

In operator notation, the algorithm is described by the update relations $(u_{k+1}, h_{k+1}) = (u_k, h_k) + \varepsilon^{-2}\mathscr{G}^* e_{k+1}$ which has the familiar form

$$\begin{aligned} &u_{k+1} = u_k + \varepsilon^{-2} G^* e_{k+1} \quad \textit{plus the additional condition} \\ &h_{k+1} = h_k + \varepsilon^{-2} H^* e_{k+1}. \end{aligned} \tag{10.18}$$

The error evolution is described by $e_{k+1} = (I + \varepsilon^{-2}\mathscr{G}\mathscr{G}^*)^{-1} e_k$, $k \geq 0$ with convergence being described by Theorems 9.2 and 9.3 (with G replaced by $\mathscr{G}$).

Theorem 9.3 also provides conditions for the converged solution to minimize $\|(u - u_0, h - h_0)\|^2_{\mathscr{U}\times\mathscr{X}_0}$.

Finally, note that the fact that $\mathscr{G}\mathscr{G}^* \geq GG^*$ suggests that the following inequality holds,

$$(I + \varepsilon^{-2}\mathscr{G}\mathscr{G}^*)^{-1} \leq (I + \varepsilon^{-2}GG^*)^{-1} \tag{10.19}$$

That is, the initial condition variation (being additional degrees of freedom) will tend to improve error convergence speeds as compared to the convergence rates seen with the basic NOILC Algorithm 9.1.

The computational aspects of this algorithm are a little more complex than those of the case of fixed initial conditions. The Two-Point-Boundary-Value Problem defining the solutions is stated fully as follows,

$$\begin{aligned}
&\dot{x}_{k+1}(t) = Ax_{k+1}(t) + Bu_{k+1}(t) \quad \textit{with} \quad x_{k+1}(0) = x_0 + H_0h_{k+1}\\
&y_{k+1}(t) = Cx_{k+1}(t) \quad \textit{for} \quad t \in [0, T], \qquad \textit{and}\\
&e_{k+1}(t) = r(t) - y_{k+1}(t). \qquad\qquad \textit{In addition},\\
&u_{k+1}(t) = u_k(t) + \varepsilon^{-2}R^{-1}(t)B^Tp(t), \qquad \textit{with}\\
&\quad \dot{p}(t) = -A^Tp(t) - C^TQ(t)e_{k+1}(t), \quad \textit{and} \;\; p(T) = 0.
\end{aligned} \tag{10.20}$$

Added to these relations is complexity which comes from the additional update relationship for h which depends on e_{k+1}, namely,

$$\underbrace{h_{k+1} = h_k + \varepsilon^{-2}Q_0^{-1}\int_0^T H_0^T e^{A^Tt}C^TQ(t)e_{k+1}(t)dt.}_{\textit{Additional Initial Condition Update Equation}} \tag{10.21}$$

Analysis of these relationships suggests that the solution can be derived as for Algorithm 9.5 to be

$$u_{k+1}(t) = u_k(t) + \varepsilon^{-2}R^{-1}(t)B^T\left[-K(t)x_{k+1}(t) + \xi_{k+1}(t)\right] \tag{10.22}$$

where $K(t)$ and $\xi_{k+1}(t)$ are as defined in Sect. 9.1.6. $K(t)$ is iteration independent and $\xi_{k+1}(t)$ depends only on data from iteration k and can be computed off-line between the end of iteration k and the initiation of iteration $k + 1$. However, the initial condition h_{k+1} defined by Eq. (10.21) depends on future values of the output. This could make a feedback implementation of the algorithm difficult to realize but a feedforward version of the implementation can be viewed from two perspective, namely,

1. as the iterative, off-line solution of the three equations

$$\begin{aligned} e_{k+1} &= r - Gu_{k+1} - Hh_{k+1} - d, \\ u_{k+1} = u_k + \varepsilon^{-2} G^* e_{k+1},\ &\textit{and}\ h_{k+1} = h_k + \varepsilon^{-2} H^* e_{k+1}. \end{aligned} \tag{10.23}$$

The equations define closed linear varieties in $\mathscr{Y} \times \mathscr{U} \times \mathscr{X}_0$. Computation of $(e_{k+1}, u_{k+1}, h_{k+1})$ is hence the problem of finding a point in their intersection. The use of the ideas of successive projection introduced in Chap. 12 represent one possible approach.

2. Alternatively, a formula-based analytic solution can be obtained. To do this, write the dynamics of the implemented control in the form

$$\begin{aligned} \dot{x}_{k+1}(t) = \left(A - B\varepsilon^{-2} R^{-1}(t) B^T K(t)\right) x_{k+1}(t) + B\left(u_k(t) + \varepsilon^{-2} R^{-1}(t) B^T \xi_{k+1}(t)\right), \\ \textit{with}\quad y_{k+1}(t) = C x_{k+1}(t)\ \textit{and}\ e_{k+1}(t) = r(t) - y_{k+1}(t). \end{aligned} \tag{10.24}$$

   a. Let $e^{(0)}_{k+1}(t)$ be the error computed using the input update rule but using the initial condition $x_{k+1}(0) = x_0$, then

$$\begin{aligned} e_{k+1}(t) = e^{(0)}_{k+1}(t) - C\Phi_K(t) H_0 h_{k+1},\ \textit{where}\ \Phi_K(t)\ \textit{is}\ n \times n\ \textit{and} \\ \dot{\Phi}_K(t) = \left(A - B\varepsilon^{-2} R^{-1}(t) B^T K(t)\right) \Phi_K(t)\ \textit{with}\ \Phi_K(0) = I_n. \end{aligned} \tag{10.25}$$

   Using this in the update equation for h then yields the formula

$$\begin{aligned} h_{k+1} = (I_p + \varepsilon^{-2} W_H)^{-1} \left(h_k + \varepsilon^{-2} Q_0^{-1} \int_0^T H_0^T e^{A^T t} C^T Q(t) e^{(0)}_{k+1}(t) dt\right) \\ \textit{where} \qquad W_H = Q_0^{-1} \int_0^T H_0 e^{A^T t} C^T Q(t) C \Phi_K(t) H_0 dt \end{aligned} \tag{10.26}$$

   and the matrix inverse is assumed to exist.

   b. Alternatively, let $e^{(h_k)}_{k+1}(t)$ be the error computed using $x_{k+1}(0) = x_0 + H_0 h_k$, then

$$\begin{aligned} e_{k+1}(t) = e^{(h_k)}_{k+1}(t) - C\Phi_K(t) H_0 (h_{k+1} - h_k),\ \textit{so that} \\ h_{k+1} = h_k + (I_p + \varepsilon^{-2} W_H)^{-1} \varepsilon^{-2} Q_0^{-1} \int_0^T H_0^T e^{A^T t} C^T Q(t) e^{(h_k)}_{k+1}(t) dt. \end{aligned} \tag{10.27}$$

The reader should note that a feedback implementation of these formulae can be achieved by computing $e^{(\cdot)}_{k+1}$ either off-line or by experiment and following this by the evaluation of the required h_{k+1}. The generation of e_{k+1} is then achieved using the standard NOILC feedback Algorithm 9.5 implementation from the initial condition $x_{k+1}(0) = x_0 + H_0 h_{k+1}$.

10.4 Problems with Several Objectives

Although, implicitly, very general, the presentation has primarily illustrated the ideas using the tracking of functions or time series $r(t)$ on finite intervals. In the next sections, this idea will be refined to provide useful and substantial generalizations. This section presents the basic concepts to be used. The approach arises from a more detailed structural definition of the tracking task. More precisely, consider a system described by dynamic relationships $y = Gu + d$ with input $u \in \mathscr{U}$ and output $y \in \mathscr{Y}$. This underlying dynamics may take any form but, as an aid to visualization of the idea, the reader may wish to consider the case of a state space model where the tracking tasks may be written in the form of other derived variables such as a combination of the specification of

1. required values of outputs at isolated times in the interval,
2. periods where no tracking is required or
3. subintervals where tracking of a reference is required.

This list is only indicative but, in general, the problem is one of finding an input u to ensure that M tracking objectives are satisfied. Symbolically, this is represented by a set of linear equations

$$y^{(j)} = G_j u + d^{(j)}, \quad 1 \leq j \leq M, \tag{10.28}$$

where $y^{(j)} \in \mathscr{Y}_j$ (a real Hilbert space) is a signal or value of interest and $G_j : \mathscr{U} \to \mathscr{Y}_j$ is a bounded linear map. These objects can be combined to produce a signal y^e defined by the M-tuple

$$y^e = \left(y^{(1)}, y^{(2)}, \ldots, y^{(M)}\right) \in \mathscr{Y}_1 \times \mathscr{Y}_2 \times \cdots \times \mathscr{Y}_M = \mathscr{Y}_e \tag{10.29}$$

where the product space $\mathscr{Y}_e$ inherits a Hilbert space structure by using the inner product

$$\langle (y^{(1)}, y^{(2)}, \ldots, y^{(M)}), (w^{(1)}, w^{(2)}, \ldots, w^{(M)}) \rangle_{\mathscr{Y}_e} = \sum_{j=1}^{M} \langle y^{(j)}, w^{(j)} \rangle_{\mathscr{Y}_j}, \tag{10.30}$$

and associated induced norm. Note that y^e can also be regarded as the column vector $y^e = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(M)} \end{bmatrix}$. The control task is to find a suitable u to satisfy the requirement that $y^e = r^e = \begin{bmatrix} r_1^e \\ \vdots \\ r_M^e \end{bmatrix}$ where the $\{r_j^e\}_{1 \leq j \leq M}$ are the desired "values" of $\{y^{(j)}\}_{1 \leq j \leq M}$.

Bringing the individual elements together produces the model

$$y^e = G_e u + d^e \quad with \quad d^e = \left(d^{(1)}, \ldots, d^{(M)}\right) \quad and \tag{10.31}$$

the operator $G_e : \mathscr{U} \to \mathscr{Y}_e$ is defined by $G_e u = (G_1 u, G_2 u, \ldots, G_M u)$ with adjoint

$$G_e^* \left(y^{(1)}, y^{(2)}, \ldots, y^{(M)}\right) = \sum_{j=1}^{M} G_j^* y^{(j)}. \tag{10.32}$$

With this construction, the operator $G_e G_e^*$ can be identified with the block matrix

$$G_e G_e^* = \begin{bmatrix} G_1 G_1^* & G_1 G_2^* & \cdots & G_1 G_M^* \\ G_2 G_1^* & G_2 G_2^* & \cdots & G_2 G_M^* \\ \vdots & & & \vdots \\ G_M G_1^* & G_M G_2^* & \cdots & G_M G_M^* \end{bmatrix} \tag{10.33}$$

which, if $\mathscr{Y}_e$ is finite dimensional is simply a matrix of real scalars.

The NOILC algorithm that can solve the Iterative Control problem is as follows,

Algorithm 10.3 (*NOILC with Many Objectives*) Suppose that u_0 and r^e are specified for the process $y = Gu + d$ and that $y^e = G_e + d^e$ represents the dynamics of the variables to be controlled. Suppose that the input u_k produced the outcome y_k^e and consequent tracking error $e_k^e = \left(e_k^{(1)}, \ldots, e_k^{(M)}\right)$ on iteration k, then the NOILC algorithm that constructs u_{k+1} as that input that minimizes the objective function

$$J(u, u_k) = \|e_{k+1}^e\|_{\mathscr{Y}_e}^2 + \varepsilon^2 \|u - u_k\|_{\mathscr{U}}^2 \tag{10.34}$$

produces a monotonically reducing norm sequence $\|e_k^e\|_{\mathscr{Y}_e}$. The solution of the problem leads to the input and error update formulae

$$\begin{aligned} u_{k+1} \;&=\; u_k + \varepsilon^{-2} G_e^* e_{k+1}^e = u_k + \varepsilon^{-2} \sum\nolimits_{j=1}^{M} G_j^* e_{k+1}^{(j)} \quad and \\ e_{k+1}^e = L_e e_k^e, \quad &for \;\; k \geq 0, \quad where \quad L_e = (I + \varepsilon^{-2} G_e G_e^*)^{-1}. \end{aligned} \tag{10.35}$$

The convergence properties are described by Theorems 9.2 and 9.3 with G replaced by G_e.

The algorithm has structural similarity to that of NOILC as described in Chap. 9 but, again, the input update u_{k+1} depends on the outcome e_{k+1}^e. It is therefore necessary to use a feedforward implementation unless a causal feedback/feedforward realization can be derived.

10.5 Intermediate Point Problems

The problems discussed in Sects. 9.1.3, 9.1.6, 10.2 and 10.3 indicate the richness of the range of applications of Algorithm 9.1, even for linear state space systems. Some possibilities were summarized in Sect. 10.4. The problem considered here is a pre-cursor to that of Sect. 10.6.

Consider the objective of controlling a linear, continuous (or discrete), state space system $S(A, B, C)$ from an initial condition $x(0) = x_0$ on a defined finite interval $[0, T]$ (or $0 \leq t \leq N$). The objective of the control system is assumed to be that of using iteration aimed at ensuring that the output signals pass through specified *intermediate points* or linear varieties in $\mathscr{R}^m$ at specified times.

10.5.1 Continuous Time Systems: An Intermediate Point Problem

Assume that the input/output dynamics are represented by the model $y = Gu + d$ with $u \in \mathscr{U} = L_2^\ell[0, T]$ and $y \in \mathscr{Y} = L_2^m[0, T]$ but that the reference signal is not defined in $\mathscr{Y}$. Instead. let

$$0 < t_1 < t_2 \cdots < t_M \leq T \tag{10.36}$$

be chosen points on the interval $[0, T]$. For notational purposes, set $t_0 = 0$. Let $F_1, F_2, \ldots, F_M$ be $f_j \times m$ matrices of full row rank and set the control objective to that of ensuring that the output response $y(t)$ satisfies the M point value constraints

$$r_j^e = F_j y(t_j) \in \mathscr{R}^{f_j}, \quad \textit{for} \quad 1 \leq j \leq M. \tag{10.37}$$

where $\{r_j^e\}_{1 \leq j \leq M}$ define intermediate values specified by the user.

Example If $m = 3$ and $T = 10$, the objective might be to ensure that the following conditions are achieved

$$\begin{aligned} [1\ 1\ 1]y(1) = 2 = r_1^e, \quad y(4) = \begin{bmatrix} 0 \\ 0 \\ -6 \end{bmatrix} = r_2^e, \\ \begin{bmatrix} 0 & 1 & 1 \\ 2 & 0 & 1 \end{bmatrix} y(7) = \begin{bmatrix} 1 \\ -1 \end{bmatrix} = r_3^e, \quad y(10) = \begin{bmatrix} -1 \\ 2 \\ -4 \end{bmatrix} = r_4^e. \end{aligned} \tag{10.38}$$

Clearly, $M = 4$, $t_1 = 1$, $t_2 = 4$, $t_3 = 7$ and $t_4 = 10$ with

$$F_1 = [1\ \ 1\ \ 1\ ], \quad F_2 = I_3, \quad F_3 = \begin{bmatrix} 0 & 1 & 1 \\ 2 & 0 & 1 \end{bmatrix} \quad \textit{and} \quad F_4 = I_3 \tag{10.39}$$

so that $f_1 = 1, f_2 = 3, f_3 = 2$ and $f_4 = 3$. The aim of the control is to ensure that the output vector takes specified values at $t = 4$ and $t = 10$ whilst passing through the two-dimensional "plane" $F_1 y = r_1^e$ at $t = 1$ and hitting the one-dimensional "line" $F_3 y = r_3^e$ at $t = 7$. The precise places where that plane and line is intersected is not specified and is assumed to be unimportant to the problem.

Note: It is always possible to replace the plane and line intersection requirements by specified values of y at $t = 1$ and $t = 7$ but this would tend to lead to increases in the control energy used. □

In operator form, the reference signal can be regarded as a finite sequence $\left(r_1^e, \ldots, r_M^e\right)$ in the space $\mathscr{Y}_e$ with

$$\mathscr{Y}_e = \mathscr{Y}_1 \times \mathscr{Y}_2 \times \cdots \times \mathscr{Y}_M, \quad with \quad \mathscr{Y}_j = \mathscr{R}^{f_j}, \quad 1 \le j \le M, \tag{10.40}$$

which can also be identified with $\mathscr{R}^{\sum_{j=1}^{M} f_j}$ by writing r^e as a column vector $[(r_1^e)^T, (r_2^e)^T, \ldots, (r_M^e)^T]^T$. The topology in $\mathscr{Y}_j$ is assumed to be generated by the inner product $\langle z, w\rangle_{\mathscr{Y}_j} = z^T Q_j w$ where Q_j is $f_j \times f_j$, real, symmetric and positive definite.

The underlying dynamics $S(A, B, C)$ can be represented in the operator form

$$y^e = G_e u + d^e \tag{10.41}$$

where $G_e : \mathscr{U} \to \mathscr{Y}_e$ and associated operators $G_j : \mathscr{U} \to \mathscr{R}^{f_j}$ are defined by

$$G_e u = (G_1 u, \ldots, G_M u), \quad where \quad G_j u = \int_0^{t_j} F_f C e^{A(t_j - t)} B u(t) dt, \quad 1 \le j \le M, \tag{10.42}$$

and d^e is the column generated from the M-tuple $d^e = \left(F_1 C e^{A t_1} x_0, \ldots, F_M C e^{A t_M} x_0\right)$.

The topology in $\mathscr{U}$ is that given for NOILC in Sect. 9.1.6 whilst the norm topology in $\mathscr{Y}_e$ is generated from the inner product

$$\langle y^e, w^e\rangle_{\mathscr{Y}_e} = \sum_{j=1}^{M} \langle y^{(j)}, w^{(j)}\rangle_{\mathscr{Y}_j} = \sum_{j=1}^{M} (y^{(j)})^T Q_j w^{(j)}. \tag{10.43}$$

With these definitions, the *Intermediate Point Norm Optimal Iterative Learning Control Algorithm* (or, IPNOILC for simplicity) is defined as

Algorithm 10.4 (*The Intermediate Point NOILC Algorithm*) Let the starting values u_0 be arbitrarily chosen. Suppose that, on iteration k the input u_k was used, then a convergent IPNOILC algorithm tracking a reference signal $r^e \in \mathscr{Y}_e$ is obtained using Algorithm 9.1 and choosing u_{k+1} to minimize

$$J(u, u_k) = \|e_{k+1}^e\|_{\mathscr{Y}_e}^2 + \varepsilon^2 \|u - u_k\|_{\mathscr{U}}^2 \qquad (with \ \ error \ \ e^e = r^e - y^e) \tag{10.44}$$

subject to the constraint $y^e = G_e u + d^e$.

The convergence of the algorithm is governed by Theorems 9.2 and 9.3. As a consequence, a sufficient condition for the algorithm to converge to the limit error $e^e_\infty = 0$ is that (A, B) is controllable as then there exists at least one input that solves the tracking problem. The algorithm also inherits the NOILC property of converging to the input u_∞ minimizing the input energy function $\|u - u_0\|^2_{\mathscr{U}}$.

In operator notation, the details of the algorithm are described by the update relation $u_{k+1} = u_k + \varepsilon^{-2} G^*_e e^e_{k+1}$. The adjoint of G_e is defined by its operation on e^e_{k+1} as follows,

$$\begin{gathered} G^*_e e^e_{k+1} = \textstyle\sum_{j=1}^M \; G^*_j e^{(j)}_{k+1} \quad \textit{where} \;\; e^{(j)}_{k+1} = (r^e_j - F_j y_{k+1}(t_j)), \quad 1 \le j \le M, \\ (G^*_j e^{(j)}_{k+1})(t) = 0, \quad \textit{for} \;\; t_j < t \le T, \quad \textit{and} \\ (G^*_j e^{(j)}_{k+1})(t) = R^{-1}(t) B^T e^{A^T(t_j - t)} C^T F_j^T Q_j e^{(j)}_{k+1}, \;\; 0 \le t \le t_j, \end{gathered} \tag{10.45}$$

which can be written in the form

$$\begin{gathered} (G^*_e e^e_{k+1})(t) = R^{-1}(t) B^T \left(\textstyle\sum_{j=1}^M \; p_j(t) \right) \quad \textit{where} \;\; \dot{p}_j(t) = -A^T p_j(t) \\ \textit{with} \;\; p_j(t_j) = C^T F_j^T Q_j e^{(j)}_{k+1} \;\; \textit{and} \;\; p_j(t) = 0, \;\; \textit{for} \;\; t_j < t \le T. \end{gathered} \tag{10.46}$$

Using linearity and defining $p(t) = \sum_{j=1}^M \; p_j(t)$ then gives

$$\begin{gathered} (G^*_e e^e_{k+1})(t) = R^{-1}(t) B^T p(t) \quad \textit{where} \;\; \dot{p}(t) = -A^T p(t) \quad \textit{whenever} \quad t \ne t_j, \\ p(t) = 0 \;\; \textit{for} \;\; t_M < t \le T \;\; \textit{plus the "jump conditions"} \\ p(t_j-) - p(t_j+) = C^T F_j^T Q_j e^{(j)}_{k+1} \;\; \textit{with} \;\; p(t_M+) = 0. \end{gathered} \tag{10.47}$$

Feedback Solution: The input u_{k+1} is given by $u_{k+1}(t) = u_k(t) + \varepsilon^{-2} R^{-1}(t) B^T p(t)$. The state equations, together with the costate equation $\dot{p}(t) = -A^T p(t)$ are identical in form to the equations analyzed in Sect. 3.10 with $Q(t) = 0$ but with the addition of the "jump conditions". As a consequence, the implemented control takes the form

$$\begin{gathered} u_{k+1}(t) = u_k(t) + \varepsilon^{-2} R^{-1}(t) B^T \left(-K(t) x_{k+1}(t) + \xi_{k+1}(t) \right) \;\; \textit{by writing} \\ p(t) = -K(t) x_{k+1}(t) + \xi_{k+1}(t), \\ \textit{on each subinterval} \quad (t_{j-1}, t_j), \quad 1 \le j \le M. \end{gathered} \tag{10.48}$$

The reader can verify that both $K(t)$ and $\xi_{k+1}(t)$ satisfy the equations derived in Sect. 3.10 (and used for the Algorithm 9.5), with the only data changes that the weight $Q(t) = 0$, $r(t) = 0$ and $R(t)$ is replaced by $\varepsilon^2 R(t)$. That is

$$\begin{gathered} \dot{K}(t) + A^T K(t) + K(t) A - \varepsilon^{-2} K(t) B R^{-1}(t) B^T K(t) = 0 \quad \textit{and} \\ \dot{\xi}_{k+1}(t) = -(A^T - \varepsilon^{-2} K(t) B R^{-1}(t) B^T) \xi_{k+1}(t) + K(t) B u_k(t), \end{gathered} \tag{10.49}$$

In order for the jump conditions for $p(t)$ to be satisfied at the intermediate points, $K(t)$ and $\xi_{k+1}(t)$ must satisfy their own jump conditions, namely,

$$\begin{aligned} K(t_j-) - K(t_j+) &= C^T F_j^T Q_j F_j C, \quad 1 \le j \le M \quad \textit{with} \quad K(t_M+) = 0, \\ \xi_{k+1}(t_j-) - \xi_{k+1}(t_j+) &= C^T F_j^T Q_j r_j^e \quad \textit{with} \quad \xi_{k+1}(t_M+) = 0. \end{aligned} \tag{10.50}$$

Jumps are easily included in computations using simulations, sequentially, over the open intervals

$$(t_M, T), \ \textit{then} \ (t_{M-1}, t_M), \ \textit{then} \ (t_{M-2}, t_{M-1}) \ldots, \ \textit{and finally} \ (t_0, t_1), \tag{10.51}$$

including the jumps to generate the terminal condition for the next subinterval.

Error Evolution and Eigenstructure: As in the case of NOILC, the error evolution is given by the expression $e_{k+1}^e = (I + \varepsilon^{-2} G_e G_e^*)^{-1} e_k^e$ but, as the output space is finite dimensional (of dimension $\sum_{j=1}^M f_j$), $G_e G_e^*$ is a matrix with block elements consisting of $f_i \times f_j$ matrices

$$(G_e G_e^*)_{ij} = G_i G_j^* = \int_0^{\min(t_i, t_j)} F_i C e^{A(t_i - t)} B R^{-1}(t) B^T e^{A^T(t_j - t)} C^T F_j^T Q_j dt, \tag{10.52}$$

where i is the block row index and j is the block column index.

Theorem 10.1 (Invertibility of $G_e G_e^*$ and Controllability) *A sufficient condition for $G_e G_e^*$ to be nonsingular is that the pair (A, B) satisfies the state controllability conditions.*

Proof State controllability guarantees that all targets $r^e \in \mathscr{Y}_e$ can be achieved by suitable choice of input. IPNOILC Algorithm 10.4 hence converges for all references r^e. This requires that $G_e G_e^* > 0$ which proves nonsingularity. □

As $G_e G_e^*$ is self adjoint in $\mathscr{Y}_e$, it has real, positive eigenvalues. As a consequence, examination of $(I + \varepsilon^{-2} G_e G_e^*)^{-1}$ indicates that

1. convergence rates increase as ε gets smaller,
2. convergence is fastest in the subspace(s) of $\mathscr{Y}_e$ where those eigenvalues are largest and
3. convergence may be slow if any one, or more, of the eigenvalues is/are small relative to ε^2.

If the eigenvalues $0 < \sigma_1^2 \le \sigma_2^2 \le \cdots$ of $G_e G_e^*$ are computed, this information can guide the choice of ε^2 as the slowest convergence rate occurs in the subspace spanned by the eigenvectors corresponding to the eigenvalue σ_1^2. If convergence of error norms is to be guaranteed to satisfy $\|e_k^e\|_{\mathscr{Y}_e} \le \mu^k \|e_0^e\|_{\mathscr{Y}_e}$ for all $k \ge 0$, it is sufficient to ensure that

$$(1 + \varepsilon^{-2} \sigma_1^2)^{-1} < \mu. \tag{10.53}$$

Feedforward Solution: A feedforward solution can be constructed for off-line computation of u_{k+1} by computing the matrix $G_e G_e^*$ and using the error evolution formula to compute the predicted value of e^e_{k+1}. The required new input is then

$$u_{k+1} = u_k + \varepsilon^{-2} G_e^* e^e_{k+1} \tag{10.54}$$

with $G_e^* e^e_{k+1}$ written as $R^{-1}(t)B^T p(t)$ where $\dot{p}(t) = -A^T p(t)$ and the solution is generated using simulation and the jump conditions generated by e^e_k. It is left as an exercise for the reader to fill in the details.

10.5.2 Discrete Time Systems: An Intermediate Point Problem

The problem formulation for discrete systems is identical to the case of continuous systems with the time interval $[0, T]$ replaced by $N+1$ samples at $t = 0, 1, 2, \ldots, N$. The intermediate points $\{t_j\}_{1 \le j \le M}$ are then sample instants and the space $\mathscr{Y}_e$ is as described above. The main difference is that the operators G, G_e and $\{G_j\}_{1 \le j \le M}$ and the underlying input space $\mathscr{U}$ change.

The simplest approach to the solution of the problem is to note that the IPNOILC objective function minimization is essentially a linear quadratic optimal control problem. It is then possible to appeal to the optimal control solution described in Sect. 4.7 and set

$$Q(t) = 0 \quad \textit{whenever} \quad t \neq t_j, \quad Q(t_j) = F_j^T Q_j F_j \quad \textit{for} \quad 1 \le j \le M \tag{10.55}$$

and replacing the time series $r(t)$ by any sequence satisfying $F_j r(t_j) = r^e_j$, $1 \le j \le M$.

10.5.3 IPNOILC: Additional Issues and Robustness

The application of Intermediate Point Iterative Norm Optimal Iterative Learning Control in practice inevitably leads to other questions that users may ask or add to the design. Issues include:

1. **Nature of the Converged Solution**: The converged input $u_\infty(t)$ typically minimizes the quantity $\|u - u_0\|^2_{\mathscr{U}}$ and may have undesirable characteristics. If $u_0 = 0$, then it is the "minimum energy" solution. This solution has the typical property that, if the length $t_j - t_{j-1}$ of the subintervals is long, the exponential nature of the solution of $\dot{p}(t) = -A^T p(t)$ suggests that most of the control activity takes place in the vicinity of t_j–. If however $u_0 \neq 0$ then u_∞ is close to u_0 (in norm) so the choice of u_0 is one way of influencing its form and the distribution of control input activity. This choice could be made in many ways including,

a. **Parameterized Control Signals**: u_0 could be assumed to be close to a desired structural form. For example, a smoother change in control signal might be achieved using a continuous, piecewise linear form,

$$u(t) = u(t_{j-1}) + (u(t_j) - u(t_{j-1}))\left(\frac{t - t_{j-1}}{t_j - t_{j-1}}\right) \quad when \quad t_{j-1} \le t \le t_j, \tag{10.56}$$

which parameterizes the input in terms of the values $\{u(t_j)\}_{0 \le j \le M}$. The reader will recognize that the consequence of this assumption is that the dynamics $y^e = G_e u + d^e$ can be replaced by a relationship $y^e = G_p u^p + d^e$ where u^p is the supervector generated by the sequence $(u(t_0), u(t_1), \ldots, u(t_M))$ and G_p is a $mM \times M\ell$ matrix of scalar quantities. Using a few iterations of the NOILC Algorithm 9.1 to this model with the reference trajectory r^e would generate a piece-wise linear control signal which can then be used as the initial input u_0 for the use of IPNOILC Algorithm 10.4 which then allows full freedom of form for further input changes.

b. **Applying Input and Output Constraints**: Some control over the likely form of the converged input could be attempted using constraints on inputs and outputs. For example, suppose that the user specifies a desirable range of input and output values at each point of time $t \in [0, T]$ by set inclusion conditions

$$u(t) \in \Omega_u(t) \quad and \quad y(t) \in \Omega_y(t), \quad for \ \ 0 \le t \le T, \tag{10.57}$$

where $\Omega_u(t)$ (respectively, $\Omega_y(t)$) are closed, convex subsets of $\mathscr{R}^\ell$ (respectively, $\mathscr{R}^m$). Then the dynamics $y = Gu + d$ and the set $\{(y, u) \ : \ (y(t), u(t)) \in \Omega_y(t) \times \Omega_u(t), \ \ 0 \le t \le T\}$ define closed convex sets in $\mathscr{Y} \times \mathscr{U}$. Application of a few iterations of the successive projection algorithm of Chap. 12 should lead to a pair $(\hat{y}, \hat{u})$ that is close to the constraint set and the output will be close to the target r^e. Choosing $u_0 = \hat{u}$ may then be a suitable starting condition for the use of Algorithm 10.4.

There are many options here and further research can be merited. In particular, the constraint set in $\mathscr{Y} \times \mathscr{U}$ is usefully closed and convex but need not be expressed in terms of point-wise properties of y and u. For example, the constraint on the output could be expressed in a norm form $\|r - y\|_{\mathscr{Y}} \le M_y$ where M_y is a defined upper bound for the norm and $r(t)$ is a piecewise linear form

$$r(t) = r^e_{j-1} + (r^e_j - r^e_{j-1})\left(\frac{t - t_{j-1}}{t_j - t_{j-1}}\right) \quad when \quad t_{j-1} \le t \le t_j. \tag{10.58}$$

2. **Algorithm Robustness**: The behaviour of the algorithm depends on the form and properties of $G_e G_e^*$ whereas model mismatch is most easily represented for the underlying model $y = Gu + d$.

a. **Left Multiplicative Perturbations**: Supposing that $G_e G_e^*$ is nonsingular, the effect of error in G will affect G_e in the form of a left multiplicative perturbation U which is simply a matrix of scalar quantities. Using a feedforward implementation then gives the error evolution $e^e_{k+1} = (I - \varepsilon^{-2} U G_e G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1}) e^e_k$. Using exactly the same arguments as those used for robustness of NOILC in Sect. 9.2.4 provides the sufficient condition for robust monotonic convergence with respect to the norm

$$\|e^e\|_0 = \sqrt{\langle e^e, (I - L_e) e^e \rangle_{\mathscr{Y}_e}}, \quad \textit{where} \quad L_e = (I + \varepsilon^{-2} G_e G_e^*)^{-1} \tag{10.59}$$

in the form of positivity conditions, using Theorem 9.16,

$$U + U^* > \beta_1 U^* U, \quad \textit{where} \quad \beta_1 = \frac{\varepsilon^{-2} \|G_e^*\|^2}{1 + \varepsilon^{-2} \|G_e^*\|^2}, \tag{10.60}$$

which, as $0 < \beta_1 < 1$ includes the nominal case of $U = I$ and, particularly if β_1 is small, allows a wide range of additional perturbations.

b. **Right Multiplicative Perturbations**: If the underlying model G has a right multiplicative modelling error $U : \mathscr{U} \to \mathscr{U}$, then the plant has dynamics GU which, when transferred to G_e, replaces the model G_e by $G_e U$ and produces the error evolution (for a feedforward implementation)

$$e^e_{k+1} = \left(I - \varepsilon^{-2} G_e U G_e^* L_e \right) e^e_k, \quad k \geq 0. \tag{10.61}$$

The relevant norm to be used is now $\|e^e\|_0 = \sqrt{\langle e^e, L_e e^e \rangle_{\mathscr{Y}_e}}$ and, in the same way as that described in Theorem 9.6, monotonic convergence with respect to this norm is ensured if, using that Theorem,

$$U + U^* > \varepsilon^{-2} U^* G_e^* L_e G_e U, \quad \textit{on} \ \mathscr{R}[G_e^*]. \tag{10.62}$$

The techniques of Sect. 9.2.2 then apply to this equation with G and L replaced by G_e and L_e respectively. To obtain a frequency domain description of robustness, G_e must be removed from the problem as it is not easily incorporated into such an approach. The relevant result here is Theorem 9.10 which provides the robust monotonicity condition, for all $|z| = 1$,

$$RU(z) + U^T(z^{-1})R > \beta_I U^T(z^{-1}) R U(z), \quad \beta_I = \frac{\varepsilon^{-2} \|G_e\|^2}{1 + \varepsilon^{-2} \|G_e\|^2}. \tag{10.63}$$

The reader should note that, as both $\mathscr{Y}$ and $\mathscr{U}$ are finite dimensional, it follows that $\|G_e^*\| = \|G_e\|$ so the gain parameter appearing in both the left and right multiplicative error cases is identical. It cannot normally be computed, however, from the TFM $G(z)$ of the underlying discrete model

G. The natural approach is to note that $\|G^*\|^2$ is the largest eigenvalue of the matrix GG^*.

3. **Removal of Intermediate Points**: The number and values of the intermediate times $\{t_j\}_{1\leq j\leq M}$ will influence convergence rates and the control energy used.
   a. The impact on energy usage, as measured by $\|u_\infty - u_0\|_{\mathscr{U}}$, is simple to state as the solution for M specified, intermediate points $\{t_j\}_{1\leq j\leq M}$ is, for given values of $\{r_j^e\}_{1\leq j\leq M}$, suboptimal for any problem where some of the intermediate times (and the corresponding reference values) are omitted. It follows that the control energy needed for this second case will be smaller.
   b. The removal of intermediate points has the effect of removing the corresponding rows and columns of $G_e G_e^*$ and hence increases the smallest eigenvalue. In general terms, this indicates that faster convergence rates can be expected if intermediate points are removed from the problem.

10.6 Multi-task NOILC

The success of Norm Optimal Iterative Learning Control lies in its generality, the monotonic error norm property and the further details that can be derived for state space systems. The previous section has shown how the notion of tracking of a reference signal $r(t)$ can be relaxed to the problem of ensuring that the output trajectory passes through specified intermediate points or linear varieties in $\mathscr{R}^m$ at specified times. In this section, linear state space systems $S(A, B, C)$ are again considered with tracking objectives consisting of a mix of requirements of the general form of

1. achieving user-defined values of outputs at specified isolated times in the interval,
2. allowing periods where no tracking is required and
3. including requirements to track reference signals on subintervals.

The first two properties have already been seen in Sect. 10.5. The new ingredient, therefore, is the need to track on subintervals. In what follows, the problem is defined, set in a general NOILC context and the solutions obtained using Riccati methodologies enhanced by "jump conditions".

Illustrative Example: A simple illustration of the idea using a continuous time, single output system could specify the control objective to be as follows,

1. The system starts from a specified initial condition at $t = 0$ and evolves on the interval $0 \leq t \leq 10 = T$.
2. The output is required to satisfy point-wise values $y(1) = 1$, $y(9) = -1$ and $y(10) = 0$ at times $t_1 = 1, t_2 = 9$ and $t_3 = 10$ whilst
3. tracking the reference signal $\sin t$ on the subinterval $2 \leq t \leq 8$.

The range of possible tracking specifications are extensive and could include

1. a need to meet rate of change requirement using the simple mechanism of adding the required derivatives to the output specification and,
2. more generally, by setting $C = I_n$ and $m = n$ gives $y = x$ and point-wise values or tracking requirements can be specified for any linear combination of state variables. This general approach includes the fact that
3. linear trajectory constraints of the form $\alpha^T y(t) = r^\alpha(t)$ (where α is a constant vector and $r^\alpha(t)$ is a specified function) can be included by adding the quantity $\alpha^T y(t)$ to the list of outputs.

10.6.1 Continuous State Space Systems

Assume that the input/output dynamics are represented by the ℓ-input, m-output state space model $S(A, B, C)$ in the form $y = Gu + d$ with $u \in \mathscr{U} = L_2^\ell[0, T]$ and $y \in \mathscr{Y} = L_2^m[0, T]$. The inner product in $\mathscr{U}$ is that used in Sect. 9.1.6. The topology of $\mathscr{Y}$ plays no role in what follows as the reference signal is not defined in $\mathscr{Y}$. Instead, define the tracking problem as follows,

Intermediate Point Tracking: As in Sect. 10.5, let

$$0 = t_0 < t_1 < t_2 \cdots < t_M \leq T \tag{10.64}$$

be chosen points on the interval $[0, T]$. Let $F_1, F_2, \ldots, F_M$ be $f_j \times m$ matrices of full row rank and set part of the control objective to be that of ensuring that the output response $y(t)$ satisfies the M point value constraints

$$r_j^e = y^{(j)} \quad \textit{where} \quad y^{(j)} = F_j y(t_j) \in \mathscr{R}^{f_j}, \quad \textit{for} \quad 1 \leq j \leq M, \tag{10.65}$$

and $\{r_j^e\}_{1 \leq j \leq M}$ defines the desired intermediate values as specified by the user.

Subinterval Tracking Requirements: In addition, tracking requirements on subintervals are specified using $\hat{M}$ additional requirements on the values of output signals on intervals $[\hat{t}_{j-1}, \hat{t}_j] \subset [0, T]$ with $0 = \hat{t}_0 < \hat{t}_1 < \hat{t}_2 < \cdots < \hat{t}_{\hat{M}} = T$. The tracking requirement is that

$$P_j y(t) = r_j^P(t), \quad \textit{for} \quad t \in [\hat{t}_{j-1}, \hat{t}_j] \quad \textit{and} \quad 1 \leq j \leq \hat{M}, \tag{10.66}$$

where $r_j^P(t)$ defines the required form of $P_j y(t)$ on $[\hat{t}_{j-1}, \hat{t}_j]$. Note that either

1. each P_j is a $p_j \times m$ matrix of full row rank defining those outputs for which tracking is required,
2. or $P_j = 0$ (representing no tracking requirement on that interval), p_j is arbitrary and $r_j^P(t) = 0$.

The Model of Dynamic Behaviour: It is useful to use the notation, for any $f \in L_2^m[0,T]$,

$$f^e = \begin{bmatrix} F_1 f(t_1) \\ \vdots \\ F_M f(t_M) \\ Pf \end{bmatrix} \in \mathscr{Y}_e = \mathscr{R}^{f_1} \times \cdots \times \mathscr{R}^{f_M} \times L_2^{p_1}[0,\hat{t}_1] \times \cdots \times L_2^{p_{\hat{M}}}[\hat{t}_{\hat{M}-1}, \hat{t}_{\hat{M}}] \tag{10.67}$$

Equivalently, f^e can be written as a row $f^e = (F_1 f(t_1), \ldots, F_M f(t_M), Pf)$ in situations where it helps the presentation. The operator P is defined by functional relations as follows,

$$\begin{gathered} Pf = \left((Pf)_1, \ldots, (Pf)_{\hat{M}}\right), \quad with \quad (Pf)_j \in L_2^{p_j}[\hat{t}_{j-1}, \hat{t}_j],\ 1 \le j \le \hat{M}, \\ and \quad (Pf)_j(t) = P_j f(t), \quad t \in [\hat{t}_{j-1}, \hat{t}_j], \quad 1 \le j \le \hat{M}. \end{gathered} \tag{10.68}$$

Technical Note: The map $f \mapsto f^e$ has range dense in $\mathscr{Y}_e$.

The model is the means of predicting values of y^e from input signals. Combining these ideas with the notation of Sect. 10.5 yields a model of the form

$$y^e = G_e u + d^e, \quad G_e u = \begin{bmatrix} G_1 u \\ \vdots \\ G_M u \\ PGu \end{bmatrix}, \quad d^e = \begin{bmatrix} F_1 d(t_1) \\ \vdots \\ F_M d(t_M) \\ Pd \end{bmatrix} \tag{10.69}$$

where G_e is the bounded linear operator

$$G_e : L_2^\ell[0,T] \to \mathscr{R}^{f_1} \times \cdots \times \mathscr{R}^{f_M} \times L_2^{p_1}[0,\hat{t}_1] \times \cdots \times L_2^{p_{\hat{M}}}[\hat{t}_{\hat{M}-1}, \hat{t}_{\hat{M}}]. \tag{10.70}$$

Here, $G_j : L_2^\ell[0,T] \to \mathscr{R}^{f_j}$ is as defined in Sect. 10.5 and $(PG)_j : L_2^\ell[0,T] \to L_2^{p_j}[\hat{t}_{j-1}, \hat{t}_j]$ is defined by

$$((PG)_j u)(t) = (P_j(Gu))(t) = P_j \int_0^t C e^{A(t-t')} B u(t') dt', \quad t \in [\hat{t}_{j-1}, \hat{t}_j]. \tag{10.71}$$

The Reference Signal Characterization: The reference signal is constructed for this model by specifying the desired values of $F_j y(t_j)$, $1 \le j \le M$ and the desired form of the signals $P_j y(t)$, $t \in [\hat{t}_{j-1}, \hat{t}_j]$, $1 \le j \le \hat{M}$. The attainability of this reference by the system requires some consistency between these characterizations, particularly at the points $\hat{t}_j$ connecting the subintervals. In addition, the point-wise constraints need to be consistent with subinterval tracking objectives. The natural way to do this is to choose a continuous reference signal $r \in \mathscr{Y}$ for y such that the reference signal

for y^e is precisely r^e. This linking of the control objective in $\mathscr{Y}_e$ to an objective in the underlying space $\mathscr{Y}$ is crucial to the interpretation of the problem in NOILC and hence optimal control terms. More precisely, the tracking error $e^e = r^e - y^e$ can be associated with an underlying tracking error $e(t) = r(t) - y(t)$ used in NOILC as follows,

$$e^e = r^e - y^e = (r-y)^e = \begin{bmatrix} F_1 e(t_1) \\ \vdots \\ F_M e(t_M) \\ Pe \end{bmatrix} \tag{10.72}$$

It is useful to note the following:

1. The "virtual reference" r and the "virtual error" e play a role in the simplification of the theory and in relating the optimization problems used in the algorithm to more familiar optimal control problems. Neither r nor e are needed in the final computations as they are always multiplied by operations that convert them into components of the extended signals such as $P_j e$ that can be computed from the specified signal $P_j r = r_j^P$ and the measured plant output.
2. The pure intermediate point problem of Sect. 10.5 is obtained simply by removing Pf from the definition of f^e whilst the NOILC problem of Sect. 9.1.6 is obtained by removing the intermediate points and looking at the case of $\hat{M} = 1$, $\hat{t}_1 = T$ and $P = I$. In terms of defining f^e, this is just (respectively)

$$f^e = \begin{bmatrix} F_1 f(t_1) \\ \vdots \\ F_M f(t_M) \end{bmatrix} \in \mathscr{R}^{f_1} \times \cdots \times \mathscr{R}^{f_M}, \quad \textit{and} \quad f^e = f \in L_2^m[0,T] \tag{10.73}$$

   Note: In terms of parameter choice, the first is obtained in what follows by setting $P_j = 0$, $1 \le j \le \hat{M}$ whilst the second is obtained using $F_j = 0$, $1 \le j \le M$.

The Consequent Topology in $\mathscr{Y}_e$: The inner product proposed for $\mathscr{Y}_e$ is

$$\begin{aligned} &\langle (w_1, \ldots, w_M, h_1, \ldots, h_{\hat{M}}), (\tilde{w}_1, \ldots, \tilde{w}_M, \tilde{h}_1, \ldots, \tilde{h}_{\hat{M}}) \rangle_{\mathscr{Y}_e} \\ &\quad = \sum_{j=1}^{M} w_j^T Q_j \tilde{w}_j + \sum_{j=1}^{\hat{M}} \int_{\hat{t}_{j-1}}^{\hat{t}_j} h_j^T(t) \hat{Q}_j(t) \tilde{h}_j(t) dt \end{aligned} \tag{10.74}$$

where the Q_j are as used in Sect. 10.5 and the $p_j \times p_j$ matrices $\hat{Q}_j(t)$ are specified by the user as being symmetric and positive definite at each point in $[\hat{t}_{j-1}, \hat{t}_j]$.

The control problem defined above has many components and will be described as a *Multi-task Control Problem* to reflect this fact. In terms of Iterative Control,

Algorithm 10.5 (*Multi-task NOILC for Continuous State Space Systems*) Let $S(A, B, C)$ be a continuous linear state space system evolving on $[0, T]$ from a fixed, iteration independent, initial condition x_0. Let u_0 and the reference signal $r^e \in \mathscr{Y}_e$ be specified and that r^e can be generated using a tracking signal $r \in \mathscr{Y}$ defined on $[0, T]$. Then, given a control signal u_k used on iteration k, the control signal u_{k+1} to be used on iteration $k+1$ is computed as the unique solution of the NOILC optimization problem defined by the minimization of the objective function

$$\begin{aligned} J(u, u_k) &= \|e^e_{k+1}\|^2_{\mathscr{Y}_e} + \varepsilon^2 \|u - u_k\|^2_{\mathscr{U}} \\ &= \textstyle\sum_{j=1}^{M} e^T_{k+1}(t_j) F_j^T Q_j F_j e^T_{k+1}(t_j) + \sum_{j=1}^{\hat{M}} \int_{\hat{t}_{j-1}}^{\hat{t}_j} e^T_{k+1}(t) P_j^T \hat{Q}_j(t) P_j e_{k+1}(t) dt \\ &\qquad + \int_0^T (u(t) - u_k(t))^T R(t)(u(t) - u_k(t)) dt \end{aligned} \tag{10.75}$$

subject to the constraint $y^e = G_e u + d^e$.

The convergence of the algorithm is described by Theorems 9.2 and 9.3 with G replaced by G_e. In particular, if $r - d$ lies in the closure of the range of G, then the limit of the error sequence is precisely zero. If $\overline{\mathscr{R}[G]} = \mathscr{Y}$, the algorithm achieves perfect tracking for all references r^e.

A Feedback Implementation: The objective function J is reminiscent of a quadratic optimal control performance criterion. A feedback implementation of this algorithm is obtained by using NOILC concepts. More precisely,

$$\begin{gathered} u_{k+1} - u_k = \varepsilon^{-2} G_e^* e^e_{k+1} = \varepsilon^{-2} \left(\textstyle\sum_{j=1}^{M} G_j^* e^{(j)}_{k+1} + (PG)^* P e_{k+1} \right) \\ \text{with} \quad e^{(j)}_{k+1} = F_j e_{k+1}(t_j) = F_j \left(r(t_j) - y_{k+1}(t_j) \right) = r^e_j - F_j y(t_j), \quad 1 \le j \le M. \end{gathered} \tag{10.76}$$

Using the notation of Sect. 9.1.6, this formula can be converted into a more useful form. First note that the first term is precisely that seen in Sect. 10.5 and hence has the form

$$\begin{gathered} \left(\textstyle\sum_{j=1}^{M} G_j^* e_{k+1}\right)(t) = R^{-1}(t) B^T p(t) \quad \text{where} \quad \dot{p}(t) = -A^T p(t) \quad \text{whenever} \quad t \neq t_j \\ \text{and} \quad p(t_j-) - p(t_j+) = C^T F_j^T Q_j e^{(j)}_{k+1}, \quad 1 \le j \le M, \quad \text{with} \quad p(t_M+) = 0. \end{gathered} \tag{10.77}$$

The computation of the second term starts with the use of the notation $(Pe_{k+1})_j \in L_2^{p_j}[\hat{t}_{j-1}, \hat{t}_j]$ to denote the signal $(Pe_{k+1})_j(t) = P_j e_{k+1}(t)$ on $[\hat{t}_{j-1}, \hat{t}_j]$ and then to identify $(PG)^* P e_{k+1}$ as the sum $\sum_{j=1}^{\hat{M}} (PG)_j^* (Pe_{k+1})_j$ with

$$\begin{gathered} ((PG)_j^* (Pe_{k+1})_j)(t) = R^{-1}(t) B^T \hat{p}_j(t) \quad \text{on} \quad [0, T], \\ \hat{p}_j(t) \quad \text{is continuous on} \quad [0, T], \quad \hat{p}_j(t) = 0 \quad \text{when} \quad \hat{t}_j < t \le T, \\ \dot{\hat{p}}_j(t) = -A^T \hat{p}_j(t) - C^T P_j^T \hat{Q}_j(t) P_j e_{k+1}(t) \quad \text{on} \quad [\hat{t}_{j-1}, \hat{t}_j] \\ \text{with} \quad \hat{p}_j(\hat{t}_j) = 0 \quad \text{and} \quad \dot{\hat{p}}_j(t) = -A^T \hat{p}_j(t) \quad \text{for} \quad 0 \le t < \hat{t}_{j-1}. \end{gathered} \tag{10.78}$$

The computations can be simplified using

1. the $m \times m$, symmetric, positive semi-definite matrix $Q(t)$, $0 \leq t \leq T$, as the time-varying matrix taking the values $Q(T) = P_{\hat{M}}^T \hat{Q}_{\hat{M}}(T) P_{\hat{M}}$ and

$$Q(t) = P_j^T \hat{Q}_j(t) P_j, \quad \textit{for} \quad t \in [\hat{t}_{j-1}, \hat{t}_j), \quad 1 \leq j \leq \hat{M}. \tag{10.79}$$

2. Also define

$$p(t) = \sum_{j=1}^{M} p_j(t) + \sum_{j=1}^{\hat{M}} \hat{p}_j(t) \tag{10.80}$$

Linearity then gives $(G_e^* e_{k+1}^e)(t) = R^{-1}(t) B^T p(t)$ with $p(t)$ solving

$$\begin{array}{c} \dot{p}(t) = -A^T p(t) - C^T Q(t) e_{k+1}(t), \quad \textit{with} \quad p(T+) = 0 \\ \textit{and} \quad p(t_j-) - p(t_j+) = C^T F_j^T Q_j F_j \left(r(t_j) - y_{k+1}(t_j) \right) \end{array} \tag{10.81}$$

and the required input is defined by

$$u_{k+1}(t) = u_k(t) + \varepsilon^{-2} R^{-1}(t) B^T p(t), \quad 0 \leq t \leq T. \tag{10.82}$$

The feedback form of this input is obtained by writing $p(t) = -K(t) x_{k+1}(t) + \xi_{k+1}(t)$ and using the same techniques as those used in Sect. 10.5 to deduce that $K(t)$ is the solution of the matrix Riccati equation

$$\begin{array}{c} \dot{K}(t) + A^T K(t) + K(t) A - \varepsilon^{-2} K(t) B R^{-1}(t) B^T K(t) + C^T Q(t) C = 0 \\ \textit{on} \quad [0, T], \quad \textit{with jump conditions} \\ K(t_j-) - K(t_j+) = C^T F_j^T Q_j F_j C, \quad 1 \leq j \leq M, \quad \textit{where} \quad K(T+) = 0. \end{array} \tag{10.83}$$

The term $\xi_{k+1}(t)$ is computed by solving

$$\begin{array}{c} \dot{\xi}_{k+1}(t) = -(A^T - \varepsilon^{-2} K(t) B R^{-1}(t) B^T) \xi_{k+1}(t) - C^T Q(t) r(t) + K(t) B u_k(t) \quad \textit{with} \\ \xi_{k+1}(t_j-) - \xi_{k+1}(t_j+) = C^T F_j^T Q_j r_j^e, \quad 1 \leq j \leq M, \quad \textit{and} \quad \xi_{k+1}(T+) = 0. \end{array} \tag{10.84}$$

The impact of the choice of $\{P_j\}_{1 \leq j \leq \hat{M}}$ is hidden in the matrix $Q(t)$ which contains all of the information on intervals where tracking is needed together with the outputs relevant to those intervals.. The reader should note that the term $Q(t)r(t)$, although apparently requiring full details of the virtual reference $r(t)$ does, in fact, only use the originally specified functions $\{r_j^P(t)\}_{1 \leq j \leq \hat{M}}$ on intervals where tracking is needed as $Q(t)r(t)$ contains the product $P_j r(t)$ on $[\hat{t}_{j-1}, \hat{t}_j]$ which is just $r_j^P(t)$. If $P_j \neq 0$ then $P_j r(t)$ can be replaced by $r_j^P(t)$ whenever it appears. When $P_j = 0$, $P_j r(t) = 0$ and any consideration of a choice of $r(t)$ is irrelevant.

Algorithm 10.6 (*Multi-task Algorithm* 10.5 *in State Feedback Form*) Using the same construction as that stated in Algorithm 10.5 and the results of the discussion above, the iteration process generates the input signal $u_{k+1}(t)$ on iteration $k+1$ from the feedback formula

$$u_{k+1}(t) = u_k(t) + \varepsilon^{-2}R^{-1}(t)B^T\left(-K(t)x_{k+1}(t) + \xi_{k+1}(t)\right) \tag{10.85}$$

where $K(t)$ is computed before the iteration process is started whilst $\xi_{k+1}(t)$ is computed off-line between iterations k and $k+1$ using the reference signal and input data from iteration k.

This parallels Implementation One in Sect. 9.1.6. The parallel to Implementation Two is as follows,

Algorithm 10.7 (*Multi-task Algorithm: Alternative State Feedback Form*) Using the same construction as that stated in Algorithm 10.5 and the results of the discussion above, the iteration process generates the input signal $u_{k+1}(t)$ on iteration $k+1$ from the feedback formula

$$u_{k+1}(t) = u_k(t) + \varepsilon^{-2}R^{-1}(t)B^T\left(-K(t)\left(x_{k+1}(t) - x_k(t)\right) + \xi_{k+1}(t)\right) \tag{10.86}$$

where $K(t)$ is computed before the iteration process is started whilst $\xi_{k+1}(t)$ is computed off-line between iterations k and $k+1$ using error data as follows

$$\begin{gathered}\dot{\xi}_{k+1}(t) = -(A^T - \varepsilon^{-2}K(t)BR^{-1}(t)B^T)\xi_{k+1}(t) - C^TQ(t)e_k(t) \quad with \\ \xi_{k+1}(t_j-) - \xi_{k+1}(t_j+) = C^TF_j^TQ_j\left(r_j^e - F_jy_k(t_j)\right), \quad 1 \le j \le M, \\ and \quad \xi_{k+1}(T+) = 0.\end{gathered} \tag{10.87}$$

This version is suited to feedforward implementation as it uses measured error data.

10.6.2 Adding Initial Conditions as Controls

Using the notation of Sect. 10.3, the reader will see that, simply by replacing G by G_e, the use of initial condition variation defined by $x_k = x_0 + H_0h_k$ can be included in the multi-task NOILC algorithm. In this case, the map H and its adjoint H^* become, respectively,

$$\begin{gathered}h \;\mapsto\; Hh = \left(F_1Ce^{At_1}H_0h, \ldots, F_MCe^{At_M}H_0h, P\psi_h\right), \quad with \;\; (\psi_h)(t) = Ce^{At}H_0h, \\ and \quad \left(w_1, w_2, \ldots, w_M, v_1, \ldots, v_{\hat{M}}\right) \;\mapsto\; H^*\left(w_1, w_2, \ldots, w_M, v_1, \ldots, v_{\hat{M}}\right) \\ = \sum_{j=1}^{M} Q_0^{-1}H_0^Te^{A^Tt_j}C^TF_j^TQ_jw_j \;+\; \sum_{j=1}^{\hat{M}} Q_0^{-1}\int_{\hat{t}_{j-1}}^{\hat{t}_j} H_0^Te^{A^Tt}C^TP_j^T\hat{Q}_j(t)v_j(t)dt.\end{gathered} \tag{10.88}$$

Define, using the objects $\Phi_K(t)$ and $Q(t)$ defined in Sects. 10.3 and 10.6,

$$\begin{aligned} W_{He} &= Q_0^{-1} \sum_{j=1}^{M} H_0^T e^{A^T t_j} C^T F_j^T Q_j F_j C \Phi_K(t_j) H_0 \\ &\quad + Q_0^{-1} \sum_{j=1}^{\hat{M}} \int_{\hat{t}_{j-1}}^{\hat{t}_j} H_0^T e^{A^T t} C^T P_j^T \hat{Q}_j(t) P_j C \Phi_K(t) H_0 dt \\ &= Q_0^{-1} \sum_{j=1}^{M} H_0^T e^{A^T t_j} C^T F_j^T Q_j F_j C \Phi_K(t_j) H_0 \\ &\quad + Q_0^{-1} \int_0^T H_0^T e^{A^T t} C^T Q(t) C \Phi_K(t) H_0 dt. \end{aligned} \tag{10.89}$$

Algorithm 10.8 (*Multi-task NOILC with Initial Condition Variation*) Algorithm 10.6 is extended to include initial condition variation as described in Sect. 10.3 by initiating the algorithm by choice of u_0 and h_0 and, for following iterations, computing u_{k+1} and h_{k+1} for iteration $k+1$ by the two step process:

STEP ONE: Use the data from iteration k, use Algorithm 10.6 to compute the input and resultant underlying error $e_{k+1}^{(h_k)}$ that results from the use of the initial condition $x_0 + H_0 h_k$. The tracking error from the initial initial condition h_{k+1} can then be represented by

$$e_{k+1}^e = \left(e_{k+1}^{(h_k)}\right)^e - (C\Phi_K H_0 (h_{k+1} - h_k))^e. \tag{10.90}$$

STEP TWO: Use this result to compute u_{k+1} as the input created by again using Algorithm 10.6 from the initial condition $x_{k+1}(0) = x_0 + H_0 h_{k+1}$ where h_{k+1} is evaluated from the expression $h_{k+1} = h_k + H^* e_{k+1}^e$ to be

$$\begin{aligned} h_{k+1} &= h_k + \varepsilon^{-2}(I_m + \varepsilon^{-2} W_{He})^{-1} \gamma_{k+1} \quad with \\ \gamma_{k+1} &= \sum_{j=1}^{M} Q_0^{-1} H_0^T e^{A^T t_j} C^T F_j^T Q_j F_j e_{k+1}^{(h_k)}(t_j) \\ &\quad + Q_0^{-1} \int_0^T H_0^T e^{A^T t} C^T Q(t) e_{k+1}^{(h_k)}(t) dt. \end{aligned} \tag{10.91}$$

The proof of the form and properties of the algorithm are very similar to that given in Sect. 10.3 with the revised definitions of H and H^*. The details are left as an exercise for the reader. Convergence properties are described by Theorems 9.2 and 9.3 with G replaced by G_e.

10.6.3 Discrete State Space Systems

Multi-task NOILC with and without variation of initial conditions also applies to the case of discrete time state space models $S(A, B, C)$ using discrete time analogues to those used in Sects. 10.3 and 10.6. The necessary computations are relatively simple. The details are left for the reader to explore.

10.7 Multi-models and Predictive NOILC

Intuitively, a consideration of the effect of current decisions on future iterations whilst constructing the current input signal could provide improved control and better convergence rates. Formally, the optimization problem on iteration $k+1$ could be formulated over the $M \geq 1$ iterations with index $k+1, k+2, \ldots, k+M$. Each of the M iterations considered can be associated with a "prediction" of the consequences of current control signals on future behaviour. The following physical/computational realization of the idea takes the form of either

1. the use of the physical plant operating in parallel with $M-1$ models of the plant (a *feedback* implementation) or
2. M plant models used off-line to compute the desired control input u_{k+1} (a *feed-forward* implementation)

plus the interpretation of these "*multi-models*" as processes to which the Norm Optimal Iterative Learning Control ideas of Chap. 9 and its extensions can be applied.

10.7.1 Predictive NOILC—General Theory and a Link to Inversion

Using the notation of the previous chapters, the plant is assumed to be described by the model $y = Gu + d$ with input and output spaces $\mathscr{U}$ and $\mathscr{Y}$ respectively. For theoretical development, the situation of zero modelling error is assumed. Robustness is discussed in Sect. 10.7.6. The reference signal $r \in \mathscr{Y}$ is given. The Multi-model of the plant describes the effects of M input signals $\{u^{(j)}\}_{1\leq j\leq M}$ using M copies of the plant,

$$y^{(j)} = Gu^{(j)} + d \quad each \quad yielding \quad M \quad errors \quad e^{(j)} = r - y^{(j)}, \ 1 \leq j \leq M. \tag{10.92}$$

Let u_k and e_k be plant data recorded on iteration k. Consider iteration $k+1$ and the objective function,

$$\begin{aligned} &J_M(\lambda, u^{(1)}, \ldots, u^{(M)}, u_k) = \textstyle\sum_{j=1}^{M} \lambda^{2(j-1)} J(u^{(j)}, u^{(j-1)}) \\ &= J(u^{(1)}, u^{(0)}) + \lambda^2 J(u^{(2)}, u^{(1)}) + \cdots + \lambda^{2(M-1)} J(u^{(M)}, u^{(M-1)}) \\ &\qquad\qquad with \quad u^{(0)} = u_k. \end{aligned} \tag{10.93}$$

Here, M is termed the "prediction horizon", $\lambda > 0$ is a real scalar and $J(u^{(j)}, u^{(j-1)})$ is the NOILC objective function of Algorithm 9.1 in the form, for $1 \leq j \leq M$,

$$J(u^{(j)}, u^{(j-1)}) = \|e^{(j)}\|^2_{\mathscr{Y}} + \varepsilon^2 \|u^{(j)} - u^{(j-1)}\|^2_{\mathscr{U}}. \tag{10.94}$$

Clearly Eq. (10.93) represents a weighted sum of NOILC objective functions over M iterations and hence its minimization generates M inputs that could be used on those iterations. However, all of the input signals $\{u^{(j)}\}_{1\leq j\leq M}$ are computed on the current iteration $k+1$ and, as a consequence, can be interpreted as potential input signals to be used on that (rather than future) iteration(s). The parameter λ describes this weighting and the power law $\lambda^{2(j-1)}$ provides a simple structure for analysis. Note that the individual terms in $J_M(\lambda,\ldots)$ are linked as each $u^{(j)},\ 1\leq j<M-1$, occurs in both $J(u^{(j)},u^{(j-1)})$ and $J(u^{(j+1)},u^{(j)})$.

If the objective function is minimized with respect to the inputs and defined multi-model constraints, the unique minimizing solution consists of M input signals, denoted $\{u^{(j)}_{k+1}\}_{1\leq j\leq M}$ which generate M predicted errors $e^{(j)}_{k+1}=r-y^{(j)}_{k+1}$ from dynamics $y^{(j)}_{k+1}=Gu^{(j)}_{k+1}+d$. It is important to note that, if $\alpha_j,\ 1\leq j\leq M$, are real scalars satisfying the convex combination rule

$$\sum_{j=1}^{M}\alpha_j=1, \tag{10.95}$$

then the input $u^{\{\alpha_j\}}_{k+1}$ generates the output and tracking error $y^{\{\alpha_j\}}_{k+1}$ and $e^{\{\alpha_j\}}_{k+1}$ where

$$u^{\{\alpha_j\}}_{k+1}=\sum_{j=1}^{M}\alpha_j u^{(j)}_{k+1},\quad y^{\{\alpha_j\}}_{k+1}=\sum_{j=1}^{M}\alpha_j y^{(j)}_{k+1}\quad \textit{and}\quad e^{\{\alpha_j\}}_{k+1}=\sum_{j=1}^{M}\alpha_j e^{(j)}_{k+1}. \tag{10.96}$$

Algorithm 10.9 (*Predictive NOILC with Convex Combinations*) Suppose that the reference r and initial input signal u_0 are given and that the scalars $\{\alpha_j\}_{1\leq j\leq M}$ are chosen. Suppose also that, on iteration $k\geq 0$, the input u_k was used and resulted in the error signal e_k. Then the Predictive Norm Optimal Iterative Learning Control Algorithm with Convex Combinations computes the input to be used on iteration $k+1$ from the formula

$$u_{k+1}=u^{\{\alpha_j\}}_{k+1}=\sum_{j=1}^{M}\alpha_j u^{(j)}_{k+1}, \tag{10.97}$$

where $u^{\{\alpha_j\}}_{k+1}$ is the input signal generated from the sequence $\{u^{(j)}_{k+1}\}_{1\leq j\leq M}$ that minimizes the objective function of Eq. (10.93). That is,

$$\left(u^{(1)}_{k+1},\ldots,u^{(M)}_{k+1}\right)=\arg\min\ J_M(\lambda,u^{(1)},\ldots,u^{(M)},u_k). \tag{10.98}$$

Note: Simple extensions of the algorithm would allow $\{\alpha_j\}_{1\leq j\leq M}$ and/or the value of λ to change from iteration to iteration. These refinements are not considered here.

The properties of the algorithm are considered in the following sections. It is however possible, at this point to make certain general statements,

Theorem 10.2 (Predictive NOILC—General Properties) *Algorithm 10.9 has the following properties:*

1. *The algorithm reduces to the basic NOILC Algorithm 9.1 when* $M = 1$.
2. *Suppose that* $r - d \in \mathscr{R}[G]$. *Then, if* $u_\infty \in \mathscr{U}$ *is any input signal that generates zero tracking error, the M-tuple* $(u_\infty, \ldots, u_\infty)$ *is suboptimal and hence, for all* $M \geq 1$,

$$J_M(\lambda, u_{k+1}^{(1)}, \ldots, u_{k+1}^{(M)}, u_k) \leq J_M(\lambda, u_\infty, \ldots, u_\infty, u_k) = \varepsilon^2 \|u_\infty - u_k\|_{\mathscr{U}}^2. \tag{10.99}$$

3. *If* $\alpha_j = \delta_{j,j'}$ *(the Kronecker delta), then* $u_{k+1} = u_{k+1}^{(j')}$ *and*

$$\|e_{k+1}\|_{\mathscr{Y}} \leq \lambda^{-(j'-1)} \varepsilon \|u_\infty - u_k\|_{\mathscr{U}}. \tag{10.100}$$

Proof The first two statements follow from the definitions and optimality of $\{u_{k+1}^{(j)}\}_{1 \leq j \leq M}$. The third is a simple consequence of the inequality $\lambda^{2(j'-1)} \|e_{k+1}^{(j')}\|_{\mathscr{Y}}^2 \leq J_M(\lambda, u_{k+1}^{(1)}, \ldots, u_{k+1}^{(M)}, u_k)$ and noting that $u_{k+1} = u_{k+1}^{\{\alpha_j\}} = u_{k+1}^{(j')}$. □

This result suggests that choosing $j' > 1$ and $u_{k+1} = u_{k+1}^{(j')}$ whilst reducing ε and/or increasing the value of λ for fixed j' will tend to reduce the error e_{k+1}. In particular, supposing that ε is fixed, the following result applies to the first iteration,

Theorem 10.3 (Theoretical Accuracy of Predictive Control as $\lambda \to \infty$) *Suppose that* $k = 0$. *If* $1 < j' \leq M$, $r - d \in \mathscr{R}[G]$ *and* $u_0 \in \mathscr{U}$ *is arbitrary, then*

$$\lim_{\lambda \to \infty} e_1 = 0. \tag{10.101}$$

The general interpretation of the result is that,

1. for large values of λ, highly accurate tracking can be achieved in one iteration using any of the inputs $u_{k+1}^{(j)}$, $1 < j \leq M$, a property previously observed only for the Inverse Model Algorithms of Chap. 6.
2. In addition, the values of λ needed to generate this result depend on the magnitude of u_∞ (and hence r) and u_0. For an accuracy condition $\|e_1\|_{\mathscr{Y}} < E_1$ to hold, it is sufficient that E_1 satisfies $\varepsilon \|u_\infty - u_0\|_{\mathscr{U}} < \lambda^{j'-1} E_1$. In the case of non-minimum-phase (NMP) state space systems, the inverse system is unstable and hence u_∞ typically contains unstable exponentials. If the iteration time interval is long, then $\|u_\infty\|$ can be very large. The natural conclusion reached is that using $M > 1$ and increasing λ^M provides a general benefit and may be capable of improving the conditioning of NOILC algorithms, even in application to NMP systems.

Further information is also provided in Theorem 10.6 for situations where M is large.

10.7.2 A Multi-model Representation

From an optimization point of view, the Predictive NOILC Algorithm 10.9 is well-defined and provides a starting point for computational studies. A derivation of its properties requires a more detailed analysis. This is achieved starting from the construction of a "multi-model" of the form

$$y^e = G_e u^e + d^e, \quad y^e \in \mathscr{Y}_e, \ \ r^e \in \mathscr{Y}_e, \ \ d^e \in \mathscr{Y}_e, \ \ u^e \in \mathscr{U}_e, \ \ e^e = r^e - y^e \quad (10.102)$$

where $\mathscr{Y}_e = \mathscr{Y} \times \cdots \times \mathscr{Y}$ (M-times product) and $\mathscr{U}_e = \mathscr{U} \times \cdots \times \mathscr{U}$ (M-times product). The inner products (and associated norms) in $\mathscr{Y}_e$ and $\mathscr{U}_e$ are defined by

$$\begin{aligned} \langle (y^{(1)}, \ldots, y^{(M)}), (w^{(1)}, \ldots, w^{(M)}) \rangle_{\mathscr{Y}_e} &= \sum\nolimits_{j=1}^{M} \lambda^{2(j-1)} \langle y^{(j)}, w^{(j)} \rangle_{\mathscr{Y}} \quad \textit{and} \\ \langle (u^{(1)}, \ldots, u^{(M)}), (v^{(1)}, \ldots, v^{(M)}) \rangle_{\mathscr{U}_e} &= \sum\nolimits_{j=1}^{M} \lambda^{2(j-1)} \langle u^{(j)}, v^{(j)} \rangle_{\mathscr{U}}. \end{aligned} \quad (10.103)$$

Multi-model ONE: In a matrix-like format, the parallel, synchronized, off-line or on-line operation of plant models can be represented in the form

$$\begin{gathered} y^e = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(M)} \end{bmatrix}, \quad u^e = \begin{bmatrix} u^{(1)} \\ \vdots \\ u^{(M)} \end{bmatrix}, \quad d^e = \begin{bmatrix} d \\ \vdots \\ d \end{bmatrix}, \\ \textit{with reference} \quad r^e = \begin{bmatrix} r \\ \vdots \\ r \end{bmatrix} \quad \textit{and} \quad G_e = \begin{bmatrix} G & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & 0 & G \end{bmatrix}. \end{gathered} \quad (10.104)$$

Note: In feedback realizations, the multi-model uses $M-1$ "copies" of the plant G to construct G_e. Although it is not sensible to preclude the possibility that, in some applications, actual physical replication of the plant may be feasible, it is more likely that the plant cannot be duplicated in this way. In this case, $M-1$ of the terms G are models and only one term G is the actual physical plant. This still leaves open the possibility that any one of these copies could be the plant.

Multi-model TWO: For analysis purposes, it is convenient to write Multi-model One in terms of input differences $u^{(j)} - u^{(j-1)}$ to obtain the alternative definitions for iteration $k+1$,

$$u^e = \begin{bmatrix} u^{(1)} - u^{(0)} \\ \vdots \\ u^{(M)} - u^{(M-1)} \end{bmatrix}, \quad d^e = \begin{bmatrix} y_k \\ y_k \\ \vdots \\ y_k \end{bmatrix}, \quad G_e = \begin{bmatrix} G & 0 & 0 & \cdots & 0 \\ G & G & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ G & G & \cdots & G & G \end{bmatrix} \quad (10.105)$$

with $u^{(0)} = u_k$. It is simple matter to show that

$$\sum_{j=1}^{M} \lambda^{2(j-1)} \langle y^{(j)}, \sum_{i=1}^{j} Gw^{(i)} \rangle_{\mathscr{Y}} = \sum_{i=1}^{M} \lambda^{2(i-1)} \langle \sum_{j=i}^{M} \lambda^{2(j-i)} G^* y^{(j)}, w^{(i)} \rangle_{\mathscr{U}} \quad (10.106)$$

and hence that the operator $G_e : \mathscr{U}_e \to \mathscr{Y}_e$ has the adjoint (in "matrix of operators" form)

$$G_e^* = \begin{bmatrix} G^* & \lambda^2 G^* & \lambda^4 G^* & \cdots & \lambda^{2(M-1)} G^* \\ 0 & G^* & \lambda^2 G^* & \cdots & \lambda^{2(M-2)} G^* \\ \vdots & & & & \vdots \\ 0 & 0 & \cdots & 0 & G^* \end{bmatrix} \quad \textit{mapping} \quad \mathscr{Y}_e \to \mathscr{U}_e. \quad (10.107)$$

10.7.3 The Case of Linear, State Space Models

To illustrate the form of the computations that may be used, consider an ℓ-input, m-output, state dimension n, linear, continuous time, state space system $S(A, B, C)$ operating on $[0, T]$. The reader should note that the case of discrete time models can be considered in a similar way. The details are left as an exercise. Multi-model One takes the form of M copies of $S(A, B, C)$ written as

$$\dot{x}^e(t) = A_e x^e(t) + B_e u^e(t), \quad x^e(0) = x_0^e \quad \textit{with} \quad y^e(t) = C_e x^e(t) \ \textit{and}, \quad t \in [0, T],$$

$$y^e(t) = \begin{bmatrix} y^{(1)}(t) \\ \vdots \\ y^{(M)}(t) \end{bmatrix}, \quad x^e(t) = \begin{bmatrix} x^{(1)}(t) \\ \vdots \\ x^{(M)}(t) \end{bmatrix}, \quad A_e = \begin{bmatrix} A & 0 & 0 \cdots 0 \\ 0 & A & 0 \cdots 0 \\ \vdots & & \vdots \\ 0 \cdots 0 & 0 & A \end{bmatrix},$$

$$u^e(t) = \begin{bmatrix} u^{(1)}(t) \\ \vdots \\ u^{(M)}(t) \end{bmatrix}, \quad B_e = \begin{bmatrix} B & 0 & \cdots 0 \\ \vdots & & \vdots \\ 0 \cdots & 0 & B \end{bmatrix}, \quad x_0^e = \begin{bmatrix} x_0 \\ \vdots \\ x_0 \end{bmatrix},$$

$$C_e = \begin{bmatrix} C & 0 & \cdots 0 \\ \vdots & & \vdots \\ 0 \cdots & 0 & C \end{bmatrix}, \quad r^e(t) = \begin{bmatrix} r(t) \\ \vdots \\ r(t) \end{bmatrix}, \quad \textit{and} \quad e^e(t) = r^e(t) - y^e(t). \quad (10.108)$$

On iteration $k + 1$, write the input differences in the form, with $u^{(0)}(t) = u_k(t)$,

$$\begin{bmatrix} u^{(1)}(t) - u_k(t) \\ \vdots \\ u^{(M)}(t) - u^{(M-1)}(t) \end{bmatrix} = F_0 \left(u^e(t) - \tilde{u}_k(t)\right)$$

$$\text{where} \quad \tilde{u}_k(t) = \begin{bmatrix} u_k(t) \\ \vdots \\ u_k(t) \end{bmatrix} \quad \text{and} \quad F_0 = \begin{bmatrix} I & 0 & 0 \cdots 0 \\ -I & I & 0 \cdots 0 \\ \vdots & & \vdots \\ 0 & \cdots 0 & -I \; I \end{bmatrix}. \tag{10.109}$$

For generality, assume that basic NOILC Algorithm 9.1 is used in the form described in Sect. 9.1.6 and that $Q(t)$ and $R(t)$ are the, possibly time dependent, weights in the objective function $J(u^{(j)}, u^{(j-1)})$. It follows that Algorithm 10.9 computes u^e_{k+1} by minimizing $J_M(\lambda, u^{(1)}, \ldots, u^{(M)}, u_k)$ written in the form

$$J_M(\cdots) = \int_0^T \left((e^e(t))^T Q_M(t) e^e(t) + \varepsilon^2 (u^e(t) - \tilde{u}_k(t))^T R_M(t) (u^e(t) - \tilde{u}_k(t)) \right) dt \tag{10.110}$$

where $Q_M(t)$ is block diagonal of the form

$$Q_M(t) = \begin{bmatrix} Q(t) & 0 & 0 \cdots & 0 \\ 0 & \lambda^2 Q(t) & 0 \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & 0 \; 0 & \lambda^{2(M-1)} Q(t) \end{bmatrix} \tag{10.111}$$

and $R_M(t)$ has the form

$$R_M(t) = F_0^T \begin{bmatrix} R(t) & 0 & 0 \cdots & 0 \\ 0 & \lambda^2 R(t) & 0 \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & 0 \; 0 & \lambda^{2(M-1)} R(t) \end{bmatrix} F_0 \tag{10.112}$$

Application of the familiar solution procedures of Sect. 3.10 in the form used in Sect. 10.6,

$$u^e_{k+1}(t) = \tilde{u}_k(t) + \varepsilon^{-2} R_M^{-1}(t) B_e^T \left(-K^e(t) x^e_{k+1}(t) + \xi^e_{k+1}(t) \right) \tag{10.113}$$

where $K^e(t)$ is the $Mn \times Mn$ solution of the Riccati equation for the matrices A_e, B_e, C_e of the multi-model with weights $Q_M(t)$ and $R_M(t)$. That is,

$$\dot{K}^e(t) + A_e^T K^e(t) + K^e(t) A_e - \varepsilon^{-2} K^e(t) B_e R_M^{-1}(t) B_e^T K^e(t) + (C_e)^T Q_M(t) C_e = 0, \tag{10.114}$$

with $K^e(T) = 0$. In a similar manner, $\xi^e_{k+1}(t) \in \mathscr{R}^{Mn}$ is the solution of the predictive equation with $\tilde{u}_k(t) \in \mathscr{R}^{M\ell}$ replacing $u_k(t) \in \mathscr{R}^{\ell}$. That is, with $\xi^e_{k+1}(T) = 0$,

$$\dot{\xi}^e_{k+1}(t) = -(A^T_e - \varepsilon^{-2}K^e(t)B_eR^{-1}_M(t)B^T_e)\xi^e_{k+1}(t) - (C_e)^T Q_M(t)r^e(t) + K^e(t)B_e\tilde{u}_k. \tag{10.115}$$

The actual implemented input signal has the feedback/feedforward form

$$u_{k+1}(t) = \sum_{j=1}^{M} \alpha_j u^{(j)}_{k+1}(t) = \Gamma_\alpha u^e_{k+1}(t), \quad \textit{with} \;\; \Gamma_\alpha = [\alpha_1 I_\ell, \ldots, \alpha_M I_\ell]\,. \tag{10.116}$$

As $\Gamma_\alpha \tilde{u}_k(t) = u_k(t)$, this can be written as

$$u_{k+1}(t) = u_k(t) + \Gamma_\alpha \varepsilon^{-2} R^{-1}_M(t) B^T_e \left(-K^e(t)x^e_{k+1}(t) + \xi^e_{k+1}(t)\right). \tag{10.117}$$

This control law has the form of Riccati feedback with a predictive term but

1. the $\ell \times Mn$ state feedback matrix $\Gamma_\alpha R^{-1}_M(t)B^T_e K^e(t)$ operates on states from all M state components $x^{(j)}_{k+1}(t)$ of the state $x^e_{k+1}(t)$ of the multi-model, one of which may be the plant itself.
2. The predictive term $\Gamma_\alpha R^{-1}_M(t)B^T_e \xi^e_{k+1}(t)$ can be computed off-line in the usual way.

A useful simplification of $R^{-1}_M(t)$ is obtained by noting that

$$R^{-1}_M(t) = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ I & I & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ I & I & I & \cdots & I \end{bmatrix} \begin{bmatrix} R^{-1}(t) & 0 & 0 \cdots & 0 \\ 0 & \lambda^{-2}R^{-1}(t) & 0 \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & 0 \cdots & \lambda^{-2(M-1)}R^{-1}(t) \end{bmatrix}$$
$$\times \begin{bmatrix} I & I & I & \cdots & I \\ 0 & I & I & \cdots & I \\ \vdots & & & & \vdots \\ 0 & 0 & \cdots & 0 & I \end{bmatrix}. \tag{10.118}$$

Finally, by suitable choice of G, the ideas also apply to the Intermediate Point NOILC problem and the use of Multi-task NOILC with either fixed initial stage conditions or varying initial conditions as seen in Sects. 10.3, 10.5 and 10.6. The details are left for the reader to explore.

10.7.4 Convergence and Other Algorithm Properties

Using Multi-model Two, the objective function of Eq. (10.93) is

$$J_M(\lambda, u^{(1)}, \ldots, u^{(M)}, u_k) = \|e^e\|^2_{\mathscr{Y}_e} + \varepsilon^2 \|u^e\|^2_{\mathscr{U}_e}. \tag{10.119}$$

Standard NOILC methodologies with $u^{(0)} = u_k$ give the minimizing input signals and error

$$u^e_{k+1} = \varepsilon^{-2} G_e^* e^e_{k+1}, \ \textit{and} \ e^e_{k+1} = (I + \varepsilon^{-2} G_e G_e^*)^{-1}(r^e - d^e), \ \textit{with} \ r^e - d^e = \begin{bmatrix} e_k \\ \vdots \\ e_k \end{bmatrix}. \tag{10.120}$$

The simplest way of viewing this computation is as that resulting from one iteration of NOILC for this multi-model with $d = 0$ and the reference replaced by the previous error e_k observed on the plant. A general representation of error evolution is

$$\begin{aligned} e_{k+1} &= \textstyle\sum_{j=1}^M \alpha_j e^{(j)}_{k+1} = L_{M,\{\alpha_j\}} e_k, \\ \textit{with} \ L_{M,\{\alpha_j\}} &= \textstyle\sum_{j=1}^M \alpha_j L_{M,j} \ \textit{and} \ e^{(j)}_{k+1} = L_{M,j} e_k, \end{aligned} \tag{10.121}$$

for some operators $L_{M,j} : \mathscr{Y} \to \mathscr{Y}$ mapping e_k into $e^{(j)}_{k+1}$. Clearly each $L_{M,j'}$ is the value of $L_{M,\{\alpha_j\}}$ when $\alpha_j = \delta_{j,j'}$ and it is the choice of $\{\alpha_j\}_{1 \le j \le M}$ and the properties of these operators that govern the error evolution and convergence properties.

Theorem 10.4 (Construction and Properties of the Operators $\{L_{M,j}\}_{1 \le j \le M}$) *The sequence of operators $X_j : \mathscr{Y} \to \mathscr{Y}$ defined by*

$$X_{j+1} = (I + \varepsilon^{-2} GG^*(I + \lambda^2 X_j))^{-1}(I + \lambda^2 X_j), \ \ j \ge 0, \ \textit{with} \ \ X_0 = 0, \tag{10.122}$$

is well-defined and each operator is linear, bounded, self adjoint and has a bounded inverse. Moreover,

1. *they all commute with each other and with GG^* and have the property that*
2. *the minimum value of the objective function J_M used in Algorithm 10.9 on iteration $k+1$ is equal to $\langle e_k, X_M e_k \rangle_{\mathscr{Y}}$.*

In particular, the self-adjoint operator $X_1 = (I + \varepsilon^{-2} GG^*)^{-1} > X_0 = 0$ is the operator associated with the NOILC Algorithm 9.1. More generally,

$$X_{j+1} > X_j \ \ \textit{for all} \ \ j \ge 0, \tag{10.123}$$

and the errors $\{e_{k+1}^{(j)}\}_{1\le j\le M}$ *in Algorithm* 10.9 *are related by, for* $1 \le j \le M$,

$$\begin{aligned} e_{k+1}^{(j)} &= \tilde{L}_{j,j-1} e_{k+1}^{(j-1)}, \quad \text{with } e_{k+1}^{(0)} = e_k \\ \text{and } \quad \tilde{L}_{j,j-1} &= (I + \varepsilon^{-2} GG^*(I + \lambda^2 X_{M-j}))^{-1} = X_{M+1-j}(I + \lambda^2 X_{M-j})^{-1}, \end{aligned} \tag{10.124}$$

where each $\tilde{L}_{j,j-1}$ *is self adjoint and positive definite and commutes with all other* $\tilde{L}_{i,i-1}$ *and* GG^*. *In particular,* $X_1 = \tilde{L}_{M,M-1}$ *and*

$$\begin{aligned} I \ge (I + \varepsilon^{-2} GG^*)^{-1} \ge \tilde{L}_{j+1,j} \ge \tilde{L}_{j,j-1}, \text{ for } 1 \le j \le M-1, \\ \text{so that } \quad 1 \ge \|\tilde{L}_{j+1,j}\| \ge \|\tilde{L}_{j,j-1}\| \text{ for } 1 \le j \le M-1. \end{aligned} \tag{10.125}$$

In addition,

1. *for all indices* $j \ge 0$, $\tilde{L}_{j+1,j} e = e$ *if, and only if,* $e \in ker[G^*]$ *i.e.*

$$ker[I - \tilde{L}_{j+1,j}] = ker[G^*] \quad \text{and hence} \quad \overline{\mathscr{R}[I - \tilde{L}_{j+1,j}]} = \overline{\mathscr{R}[G]}, \tag{10.126}$$

   from which, using invertibility, $\tilde{L}_{j+1,j}\overline{\mathscr{R}[G]} = \overline{\mathscr{R}[G]}$.
2. *The errors* $\{e_{k+1}^{(j)}\}_{1\le j\le M}$ *satisfy the relations*

$$\|e_{k+1}^{(j)}\|_{\mathscr{Y}} \le \|e_{k+1}^{(j-1)}\|_{\mathscr{Y}}, \quad 1 \le j \le M, \tag{10.127}$$

   strict inequality holding if, and only if, the orthogonal projection of e_k *onto* $\overline{\mathscr{R}[G]}$ *is non-zero.*
3. *For all* M *and* $1 \le j \le M$, $L_{M,j}$ *is the composite operator*

$$L_{M,j} = \tilde{L}_{j,j-1} \dots \tilde{L}_{1,0}, \text{ written } \quad L_{M,j} = \prod_{i=1}^{j} \tilde{L}_{i,i-1}. \tag{10.128}$$

   These operators are self-adjoint and positive, form a commuting set and also commute with GG^*.
4. *Finally,* $L_{M,j+1} \le L_{M,j} \le I$, $\|L_{M,j+1}\| \le \|L_{M,j}\| \le 1$ *and, for* $1 \le j \le M$,

$$ker[I - L_{M,j}] = ker[G^*] \quad \text{and} \quad L_{M,j}\overline{\mathscr{R}[G]} = \overline{\mathscr{R}[G]}. \tag{10.129}$$

Proof First observe that $J_{M-j+1}(\lambda, u^{(j)}, \dots, u^{(M)}, u_{k+1}^{(j-1)})$ is minimized by the sequence $(u_{k+1}^{(j)}, \dots, u_{k+1}^{(M)})$ as, from the definitions, and taking $J_0(\cdots) = 0$,

$$\begin{aligned} &J_M(\lambda, u_{k+1}^{(1)}, \dots, u_{k+1}^{(j-1)}, u^{(j)}, \dots, u^{(M)}, u_k) \\ &\quad = J_{j-1}(\lambda, u_{k+1}^{(1)}, \dots, u_{k+1}^{(j-1)}, u_k) + \lambda^{2(j-1)} J_{M-j+1}(\lambda, u^{(j)}, \dots, u^{(M)}, u_{k+1}^{(j-1)}). \end{aligned} \tag{10.130}$$

If $j = M$, then $u^{(M)}_{k+1}$ minimizes the NOILC objective function $J(u^{(M)}, u^{(M-1)}_{k+1})$ giving

$$\begin{array}{c} u^{(M)}_{k+1} = u^{(M-1)}_{k+1} + \varepsilon^{-2}G^*e^{(M)}_{k+1}, \ \textit{and} \ \ e^{(M)}_{k+1} = \tilde{L}_{M,M-1}e^{(M-1)}_{k+1} \quad \textit{where} \\ \tilde{L}_{M,M-1} = X_1 = (I + \varepsilon^{-2}GG^*)^{-1} > 0 = X_0 \quad \textit{and} \\ J(u^{(M)}_{k+1}, u^{(M-1)}_{k+1}) = \langle e^{(M-1)}_{k+1}, X_1 e^{(M-1)}_{k+1}\rangle_{\mathscr{Y}}. \end{array} \tag{10.131}$$

More generally, the hypothesis that

1. X_{M-j} is self-adjoint and positive definite, and that $X_{M-j} > X_{M-j-1}$,
2. that X_{M-j} commutes with all X_i, $0 \le i \le M-j-1$, and with GG^*,
3. that the minimum value

$$\min\ J_{M-j}(\lambda, u^{(j+1)}, \ldots, u^{(M)}, u^{(j)}_{k+1}) = \langle e^{(j)}_{k+1}, X_{M-j}e^{(j)}_{k+1}\rangle_{\mathscr{Y}} \tag{10.132}$$

4. and that $e^{(j+1)}_{k+1} = \tilde{L}_{j+1,j}e^{(j)}_{k+1}$,

forms the basis of an inductive proof.

Note, in particular, that the first two statements imply that $\tilde{L}_{j,j-1}$ is well defined, self-adjoint and strictly positive as $GG^*(I + \lambda^2 X_{M-j}) \ge 0$. As a consequence, the inverse exists. Also, as $X_{M-j} > X_{M-j-1}$, so that $\tilde{L}_{j,j-1} \le \tilde{L}_{j+1,j}$ and hence the set $\{\tilde{L}_{j,j-1}\}$ is bounded above by $\tilde{L}_{M,M-1} = X_1 \le I$.

Returning to the inductive proof, the hypothesis is certainly true for $j = M-1$. If true for all $j \le M-1$, then the characterization of $L_{M,j}$ follows trivially noting that $e^{(0)}_{k+1} = e_k$. Consider now the minimization of $J_{M-j+1}(\lambda, u^{(j)}, \ldots, u^{(M)}, u^{(j-1)}_{k+1})$ by minimizing $J(u^{(j)}, u^{(j-1)}_{k+1}) + \lambda^2\langle e^{(j)}, X_{M-j}e^{(j)}\rangle_{\mathscr{Y}}$ with respect to $u^{(j)}$ and the dynamics $y^{(j)} = Gu^{(j)} + d$. This can be written in the form

$$\begin{array}{c} \langle e^{(j)}, (I + \lambda^2 X_{M-j})e^{(j)}\rangle_{\mathscr{Y}} + \varepsilon^2\|u^{(j)} - u^{(j-1)}_{k+1}\|^2_{\mathscr{U}} \ \textit{which is minimized when} \\ u^{(j)}_{k+1} = u^{(j-1)}_{k+1} + \varepsilon^{-2}G^*(I + \lambda^2 X_{M-j})e^{(j)}_{k+1} \quad \textit{and hence} \\ e^{(j)}_{k+1} = (I + \varepsilon^{-2}GG^*(I + \lambda^2 X_{M-j}))^{-1}e^{(j-1)}_{k+1} = \tilde{L}_{j,j-1}e^{(j-1)}_{k+1}. \end{array} \tag{10.133}$$

The inductive hypothesis is proved as a simple calculation yields the minimum value

$$\begin{array}{l} \langle e^{(j)}_{k+1}, (I + \lambda^2 X_{M-j})e^{(j)}_{k+1}\rangle_{\mathscr{Y}} + \varepsilon^2\|u^{(j)}_{k+1} - u^{(j-1)}_{k+1}\|^2_{\mathscr{U}} \\ \quad = \langle e^{(j)}_{k+1}, \left(I + \lambda^2 X_{M-j} + \varepsilon^{-2}(I + \lambda^2 X_{M-j})GG^*(I + \lambda^2 X_{M-j})\right)e^{(j)}_{k+1}\rangle_{\mathscr{Y}} \\ \quad = \langle e^{(j-1)}_{k+1}, X_{M+1-j}e^{(j-1)}_{k+1}\rangle_{\mathscr{Y}} \quad \textit{as} \ \ \tilde{L}_{j,j-1}(I + \lambda^2 X_{M-j}) = X_{M+1-j} \end{array} \tag{10.134}$$

from the definitions. That X_{M-j+1} is self adjoint and commutes with X_i with $i \le M-j$ follows easily whilst $X_{M+1-j} \ge 0$ as $\min J_{M+1-j} \ge 0$. In particular $X_{M+1-j} \ge X_{M-j}$ as

$$\min J_{M+1-j}(\lambda, u^{(j)}, \ldots, u^{(M)}, u_{k+1}^{(j-1)}) \;\geq\; \min J_{M-j}(\lambda, u^{(j)}, \ldots, u^{(M-1)}, u_{k+1}^{(j-1)}) + \lambda^{2(M-j)} \left(\|e_{k+1}^{(M)}\|_{\mathscr{Y}}^2 + \varepsilon^2 \|u_{k+1}^{(M)} - u_{k+1}^{(M-1)}\|_{\mathscr{U}}^2 \right). \tag{10.135}$$

It also follows that strict inequality holds otherwise, for some non-zero $e_{k+1}^{(j-1)}$, the final term is zero indicating that $e_{k+1}^{(M)} = 0$ and $u_{k+1}^{(M)} = u_{k+1}^{(M-1)}$. Invertibility of $\tilde{L}_{M,M-1}$ then shows that $e_{k+1}^{(M-1)} = 0$ and, using recursion, all $e_{k+1}^{(i)} = 0, i \geq j-1$ which provides a contradiction.

From the properties of NOILC Algorithm 9.1, $X_1 e = e$ if, and only if, $e \in ker[G^*]$ and hence $X_1 < I$ on the closed subspace $ker[G^*]^\perp = \overline{\mathscr{R}[G]}$ so that $\|e_{k+1}^{(M)}\| < \|e_{k+1}^{(M-1)}\|$ if, and only if, the orthogonal projection $P_{\overline{\mathscr{R}[G]}} e_{k+1}^{(M-1)}$ of $e_{k+1}^{(M-1)}$ onto $\overline{\mathscr{R}[G]}$ is non-zero. More generally, writing $\tilde{L}_{j,j-1} = I - \varepsilon^{-2} X_{M+1-j} G G^*$ leads to the conclusion that $ker[I - \tilde{L}_{j,j-1}] = ker[G^*]$ for $1 \leq j \leq M$ and hence that $P_{\overline{\mathscr{R}[G]}} e_{k+1}^{(M-1)} \neq 0$ only if $P_{\overline{\mathscr{R}[G]}} e_{k+1}^{(j-1)} \neq 0$ for $1 \leq j \leq M$. In particular, for strict inequality to hold for $1 \leq j \leq M$, it is necessary that $P_{\overline{\mathscr{R}[G]}} e_{k+1}^{(0)} = P_{\overline{\mathscr{R}[G]}} e_k \neq 0$. To prove that this is sufficient, note that $ker[\tilde{L}_{j,j-1}] = \{0\}$ and $\tilde{L}_{j,j-1} \overline{\mathscr{R}[G]} \subset \overline{\mathscr{R}[G]}$ so $P_{\overline{\mathscr{R}[G]}} e_{k+1}^{(j)} \neq 0$ for $1 \leq j \leq M$. The required monotonicity follows as $\tilde{L}_{j,j-1} < I$ on $\overline{\mathscr{R}[G]}$.

Finally, $L_{M,j+1} = L_{M,j} \tilde{L}_{j+1,j} \leq L_{M,j}$ and all $L_{M,j} \leq I$ as all $\tilde{L}_{j+1,j} \leq I$. From the properties of the $\{\tilde{L}_{j,j-1}\}$, $ker[G^*] \subset ker[I - L_{M,j}]$ for all M and j. If $j = 1$, then $L_{M,1} = \tilde{L}_{1,0}$ and the required result $ker[G^*] = ker[I - L_{M,1}]$ follows from the properties of $\tilde{L}_{1,0}$. More generally, if $ker[I - L_{M,j}] = ker[G^*]$, write

$$I - L_{M,j+1} = I - L_{M,j} \tilde{L}_{j+1,j} = (I - L_{M,j}) + L_{M,j}(I - \tilde{L}_{j+1,j}) \tag{10.136}$$

both terms on the right hand side being self-adjoint and positive. It follows that $ker[I - L_{M,j+1}] \subset ker[I - L_{M,j}] = ker[G^*]$ and the result is now proved using an induction process. □

The result has no impact on practical computation but has consequences for the convergence properties of the Predictive Algorithm 10.9. More precisely,

1. It is an easy matter to prove that

$$(i)\ L_{M,j+1} = \tilde{L}_{j+1,j} L_{M,j} \leq L_{M,j} \ \textit{so that}\ \ (ii)\ L_{M,j} \leq (I + \varepsilon^{-2} G G^*)^{-j} \tag{10.137}$$

and hence the use of $\alpha_j = \delta_{j,j'}$ (and hence $u_{k+1} = u_{k+1}^{(j')}$),

   a. will, in a single iteration, produce a norm reduction better than that obtained using "j'" iterations of NOILC Algorithm 9.1.
   b. In addition, increasing "j'" will improve convergence rates and the use of the control input $u_{k+1} = u_{k+1}^{(M)}$ will lead to the fastest convergence. It may, however, have implications for the robustness of the application.

2. Assuming that all $\alpha_j \geq 0$ and that $\sum_{j=1}^{M} \alpha_j = 1$, suppose that j^α is the index of the first non-zero value of α_j. Then the simple inequality

$$\|e_{k+1}^{\{\alpha_j\}}\| \leq \sum_{j=1}^{M} \alpha_j \|e_{k+1}^{(j)}\|_{\mathscr{Y}} \leq \|e_{k+1}^{(j^\alpha)}\|_{\mathscr{Y}} \tag{10.138}$$

bounds the error norm $\|e_{k+1}\|_{\mathscr{Y}}$ as a convex combination of the norms of the error signals $\{\|e_{k+1}^{(j)}\|_{\mathscr{Y}}\}_{1\leq j\leq M}$ and indicates that a consideration of the convex combinations always provides improved convergence when compared to that obtained using the choice $\alpha_1 = 1$ and all other $\alpha_j = 0$. For a feedforward implementation, it is a simple matter to show that the smallest value of $\|e_{k+1}^{\{\alpha_j\}}\|$ can be computed by evaluating $\{e_{k+1}^{(j)}\}_{1\leq j\leq M}$ off-line and then using the convex combination that minimizes

$$\|e_{k+1}^{\{\alpha_j\}}\|^2 \quad \textit{subject to} \quad \sum_{j=1}^{M} \alpha_j = 1. \tag{10.139}$$

This idea is linked to that of Parameter Optimal ILC (see Chap. 14).

Finally, convergence is described as follows,

Theorem 10.5 (Convergence of Predictive NOILC) *Suppose that* $\alpha_j \geq 0,\ 1 \leq j \leq M$ *and* $\sum_{j=1}^{M} \alpha_j = 1$, *then the Predictive Algorithm* 10.9 *has the properties that, for* $k \geq 0$, $\|e_{k+1}\|_{\mathscr{Y}} \leq \|e_k\|_{\mathscr{Y}}$. *In particular* $\{e_k\}_{k\geq 0}$ *converges monotonically to the orthogonal projection of* e_0 *onto* $ker[G^*]$ *and*

$$\lim_{k\to\infty} e_k = 0 \quad \textit{if} \quad r - d \in \overline{\mathscr{R}[G]}. \tag{10.140}$$

Proof The positive, self adjoint operator $L_{M,\{\alpha_j\}} = \sum_{j=1}^{M} \alpha_j\, L_{M,j}$ gives $L_{M,\{\alpha_j\}} \leq \sum_{j=1}^{M} \alpha_j\, I = I$ from the properties of the operators $\{L_{M,j}\}_{1\leq j\leq M}$ (Theorem 10.4). It follows that $\|e_{k+1}\| = \|e_{k+1}^{\{\alpha_j\}}\| \leq \|e_k\|$. Convergence properties then follows from Theorem 5.4 if $\|L_{M,\{\alpha_j\}}\| < 1$. If $\|L_{M,\{\alpha_j\}}\| = 1$, then the desired convergence properties follow from Theorem 5.9 and the easily proved observation that $I - L_{M,\{\alpha_j\}} = \sum_{j=1}^{M} \alpha_j (I - L_{M,j}) \geq 0$ for all permissible choices of $\{\alpha_j\}_{1\leq j\leq M}$. As a consequence, $ker[I - L_{M,\{\alpha_j\}}] = ker[I - L_{M,j}] = ker[G^*]$ for all j and hence the orthogonal complement $\overline{\mathscr{R}[I - L_{M,\{\alpha_j\}}]} = \overline{\mathscr{R}[G]}$. □

10.7.5 The Special Cases of $M = 2$ and $M = \infty$

Although Theorem 10.2 suggests that increasing λ^M and choosing u_{k+1} wisely from the available options $\{u_{k+1}^{(j)}\}$ will improve convergence rates, the question of whether or not it is best to increase λ and/or M is open for discussion. The simplest real-time implementation that retains the benefits of prediction whilst reducing the computational load is that achieved when $M = 2$ and λ is "large". In this case,

$$X_1 = \tilde{L}_{2,1} = (I + \varepsilon^{-2} GG^*)^{-1} \geq \tilde{L}_{1,0} = (I + \varepsilon^{-2} GG^*(I + \lambda^2 X_1))^{-1}. \quad (10.141)$$

The choice of α_1, α_2 and ε^2 also influence algorithm performance. The case of $\alpha_1 = 1$ and $\alpha_2 = 0$ is often called the *Receding Horizon* choice. Note that, in this case, the error evolution operator $L_{M,\{\alpha_j\}} = \tilde{L}_{1,0}$. Insight into the acceleration of convergence is obtained by comparing this with that seen in the NOILC Algorithm 9.1 and noting the presence of the additional term $\varepsilon^{-2}\lambda^2 GG^* X_1$ in the denominator. Using the eigenstructure assumptions of Sect. 9.1.7 for GG^*, $L_{M,\{\alpha_j\}}$ has eigenvalues

$$(1 + \varepsilon^{-2}\sigma_j^2 A_j)^{-1} \quad \textit{where} \quad A_j = (1 + \lambda^2 \left(\frac{1}{1 + \varepsilon^{-2}\sigma_j^2} \right)) > 1 \quad (10.142)$$

which can be interpreted as a formula similar to that obtained for NOILC but with an amplification of each eigenvalue σ_j^2 of GG^* by a factor A_j dependent on σ_j^2, ε^2 and λ^2. For a fixed value of ε^2, the amplification of very small eigenvalues is approximately $1 + \lambda^2$ whilst the amplification of the largest eigenvalues $\sigma_j^2 \approx \|G^*\|^2$ is approximately

$$A_1 = \left(1 + \lambda^2 \left(\frac{1}{1 + \varepsilon^{-2}\|G^*\|^2} \right)\right) < (1 + \lambda^2). \quad (10.143)$$

In terms of state space models and frequency domain properties, this suggests that

1. the Prediction methodology accelerates convergence at low frequencies as λ increases but that the accelerated reduction of high frequency components of the error is more significant.
2. At high frequencies, the error evolution is similar to that obtained in NOILC when ε^2 is replaced by the smaller value $\tilde{\varepsilon}^2 = \varepsilon^2(1 + \lambda^2)^{-1}$. If $\lambda = 4$, this gives the value $\varepsilon^2/17$.
3. If $\|G^*\|$ is computed, then the effect on the low frequency components is approximately equivalent to replacing ε^2 (in NOILC Algorithm 9.1) by a new, smaller, value $\tilde{\varepsilon}^2$ defined by the map

$$\varepsilon^2 \mapsto \tilde{\varepsilon}^2 = \varepsilon^2 \left(1 + \lambda^2 \left(\frac{1}{1 + \varepsilon^{-2}\|G^*\|^2} \right)\right)^{-1}. \quad (10.144)$$

For example, choosing $\varepsilon^2 = \|G^*\|^2/3$ and $\lambda = 4$ yields the new value $\tilde{\varepsilon}^2 = \varepsilon^2/5$.

Noting that non-minimum-phase components of the system are also associated with non-zero but extremely small eigenvalues, this suggests that there may be a useful impact of prediction on the convergence of non-minimum-phase behaviour. In reality, this will be limited unless λ is very large. This provides some motivation for the use of $M > 2$ but a precise theoretical relationship between M, λ and convergence speeds in this case is not available currently and may not be attainable. Some insight is gained however by again considering the "Receding Horizon" case when $M > 1$ and $\alpha_j = \delta_{j,1}$ (when the chosen input $u_{k+1} = u_{k+1}^{(1)}$) and letting the "Prediction Horizon" $M \to \infty$. Noting that,

$$L_{M,\{\alpha_j\}} = \tilde{L}_{1,0} = X_M(I + \lambda^2 X_{M-1})^{-1}. \tag{10.145}$$

it is clear that the convergence behaviour of the algorithm depends entirely on the behaviour of the sequence of operators $\{X_j\}_{j\geq 0}$ and the value of λ. The following theorem examines convergence as $M \to \infty$ in terms of the spectrum/eigenvalues.

Theorem 10.6 (The Behaviour of $\{X_j\}_{j\geq 0}$ and $\tilde{L}_{1,0}$ as $M \to \infty$) *Suppose that GG^* has strictly positive eigenvalues $\sigma_1^2 \geq \sigma_2^2 \geq \cdots$ associated with a complete set of orthonormal eigenvectors $\{v_i\}_{i\geq 1}$ that span $\overline{\mathscr{R}[G]}$. Then each X_j has an eigenvalue $\chi_0^{(j)} = 1 + \lambda^2 + \cdots + \lambda^{2(j-1)}$ with eigenspace $ker[G^*]$ if $ker[G^*] \neq \{0\}$ plus strictly positive eigenvalues $\{\chi_i^{(j)}\}_{i\geq 1}$ with associated eigenvectors $\{v_i\}_{i\geq 1}$. More precisely, using the notation $\sigma_0^2 = 0$,*

$$\begin{aligned} &\chi_i^{(j+1)} = (1 + \varepsilon^{-2}\sigma_i^2(1 + \lambda^2\chi_i^{(j)}))^{-1}(1 + \lambda^2\chi_i^{(j)}), \quad \chi_i^{(0)} = 0, \\ &\quad \textit{and} \quad \chi_i^{(j+1)} > \chi_i^{(j)} \quad \textit{for all} \quad j \geq 0 \quad \textit{and} \quad i \geq 0. \end{aligned} \tag{10.146}$$

In particular, for each index $i \geq 1$, the sequence $\{\chi_i^{(j)}\}_{j\geq 0}$ converges to a finite value $\chi_i^{\infty} > 0$. As a consequence, for all $e \in \overline{\mathscr{R}[G]}$,

$$\underbrace{\limsup_{M\to\infty} \|\tilde{L}_{1,0}e\|_{\mathscr{Y}} \leq \lambda^{-2}\|e\|_{\mathscr{Y}}}_{\textit{Plant Independent Convergence Rate as } M \to \infty} \tag{10.147}$$

Proof The proof of the eigenvalue formula follows from the relationship between X_{j+1} and X_j, the commutation property of all such operators and GG^* and the spectral mapping theorem. The increasing property of $\{\chi_i^{(j)}\}_{j\geq 0}$ follows from the corresponding property of the $\{X_j\}_{j\geq 0}$. For $i = 0$, the characterization of $\chi_0^{(j)}$ follows from the recursion formula with $\sigma_i^2 = 0$. For $i \geq 1$, the existence of χ_i^{∞} follows as $\sigma_i^2 > 0$ by assumption and hence the statement $\chi_i^{(j)} \to \infty$ as $j \to \infty$ leads to a contradiction. Next, note that, for a given M, the eigenvalues of $\tilde{L}_{0,1}$ are simply $\tilde{l}_{M,i} = \chi_i^{(M)}/(1+\lambda^2\chi_i^{(M-1)})$, with $i \geq 1$, and $\lim_{M\to\infty} \tilde{l}_{M,i} = \chi_i^{\infty}/(1+\lambda^2\chi_i^{\infty}) < \lambda^{-2}$. Let $\varepsilon_0 > 0$ and $e \in \overline{\mathscr{R}[G]}$ be arbitrarily chosen. It is always possible to write $e = v_1 + v_2$

with $\|v_2\| \leq \varepsilon_0$ and v_1 as a finite linear combination of the $\{v_j\}_{j\geq 1}$. It follows that, for large M, $\|\tilde{L}_{1,0}e\|_{\mathscr{Y}} \leq \|\tilde{L}_{1,0}v_1\|_{\mathscr{Y}} + \|\tilde{L}_{1,0}v_2\|_{\mathscr{Y}} \leq \lambda^{-2}\|v_1\|_{\mathscr{Y}} + \|v_2\|_{\mathscr{Y}} \leq \lambda^{-2}\|e\|_{\mathscr{Y}} + (1+\lambda^{-2})\varepsilon_0$ as $\tilde{L}_{1.0} \leq I$, $\|v_1\| \leq \|e\| + \|v_2\|$ and $\|v_2\| \leq \varepsilon_0$. The result is now proved as ε_0 is arbitrary. □

The result has a number of simple interpretations, namely that,

1. for any initial error $e_0 \in \overline{\mathscr{R}[G]}$, the choice of sufficiently high value of M and $\alpha_j = \delta_{j,1}$ will produce an error e_1 with norm reduced by a factor very close to, or better than, λ^{-2}. This emphasizes the need to choose $\lambda > 1$ and highlights the ability of the algorithm to produce arbitrarily small errors quickly by choosing $\lambda \gg 1$. For example, values of λ in the range $[\sqrt{2}, 4]$ would then produce reductions of the order of 50–94 % on each and every iteration.
2. Note that, if M is large, this contraction of the error is similar to that achieved in (right) inverse model control in Chap. 6 where contraction by a factor of $(1-\beta)$ is achieved and described by a parameter, β, chosen by the user.
3. The precise benefits that can be achieved in any particular application will depend on the details of the dynamics and the form of the reference signal r. In eigenstructure terms, the benefits may be most easily achieved using modest values of the prediction horizon M if the eigenfunction expansion of r is dominated by the largest eigenvalues of GG^*.

10.7.6 A Note on Robustness of Feedforward Predictive NOILC

In principle, techniques similar to those used for analysis of NOILC robustness can be used to provide insight into robustness of the Predictive Algorithm 10.9. The details are different however as there is added complexity because of (*i*) the effects of the choice of parameters in the convex combinations and (*ii*) because, in a feedback implementation, the use of the Multi-model implies that $M-1$ copies of G are models and hence the modelling error only applies to the remaining copy. This case presents considerable challenges for theoretical analysis and hence, in what follows, the feedforward implementation is considered.

Consider the feedforward implementation of Algorithm 10.9 based on the use of a model G of plant dynamics. Using M copies of G in a multi-model, off-line computation of the input signal u_{k+1} to be used on iteration $k+1$ can be undertaken using the previously used input u_k, the measured tracking error e_k and chosen $\{\alpha_j\}_{1\leq j\leq M}$. More precisely, one iteration of Predictive NOILC is undertaken using the multi-model with zero initial conditions $d = 0$, a reference signal e_k and zero initiating input.

It is assumed that $\sum_{j=1}^{M} \alpha_j = 1$. A formal description of u_{k+1} takes the form $u_{k+1} = u_k + \Omega_{\{\alpha_j\}}e_k$ for some linear operator $\Omega_{\{\alpha_j\}} : \mathscr{Y} \to \mathscr{U}$. Assuming a left multiplicative modelling error U, the actual plant has dynamics represented by

the operator UG. A simple calculation then gives $e_{k+1} = (I - UG\Omega_{\{\alpha_j\}})e_k$. This description applies for all U including $U = I$ and hence, using the notation of Sect. 10.7.4, $I - G\Omega_{\{\alpha_j\}} = L_{M,\{\alpha_j\}}$ so that

$$e_{k+1} = (I - U(I - L_{M,\{\alpha_j\}}))e_k, \quad k \geq 0. \tag{10.148}$$

As for NOILC in Sect. 9.2.4, error evolution occurs in the subspace $\mathscr{R}[UG]$ whilst errors in $ker[G^*]$ remain unchanged. Following the approach of that section, suppose that

$$\mathscr{Y} = ker[G^*] \oplus \overline{\mathscr{R}[UG]}. \tag{10.149}$$

The results of Theorem 10.4 indicate that $ker[I - L_{M,\{\alpha_j\}}] = ker[G^*]$ and its range is dense in $\overline{\mathscr{R}[G]}$. Using the bilinear form and associated semi-norm

$$\langle y, w\rangle_0 = \langle y, (I - L_{M,\{\alpha_j\}})w\rangle_{\mathscr{Y}} \quad and \quad \|y\|_0 = \sqrt{\langle y, y\rangle_0} \tag{10.150}$$

on $\mathscr{Y}$ also generates an inner product and norm on $\overline{\mathscr{R}[UG]}$ if $\alpha_j \geq 0, \quad 1 \leq j \leq M$. Robust monotonic convergence of the algorithm in the presence of the modelling error can then be defined as a need to satisfy, for all e_0, the following monotonicity condition

$$\|e_{k+1}\|_0 < \|e_k\|_0 \quad on \quad \overline{\mathscr{R}[UG]}, \quad for \ all \ \ k \geq 0. \tag{10.151}$$

Note that, if $e_0 \in \mathscr{Y}$ is arbitrary, then its component in $ker[G^*]$ remains unchanged from iteration to iteration whilst its component in $\overline{\mathscr{R}[UG]}$ then reduces in norm from iteration to iteration. From this the reader should be able to prove that, whatever the initial error, the resultant error sequence is bounded in norm. Applying a similar approach to that of Sect. 9.2.4,

Theorem 10.7 (Robustness of Predictive NOILC) *A sufficient condition for the robust monotonicity defined above to be present is that, for some* $\varepsilon_0 > 0$,

$$U + U^* > U^* \left(I - L_{M,\{\alpha_j\}}\right) U + \varepsilon_0^2 I \quad on \quad \mathscr{R}[G] \tag{10.152}$$

in the original inner product topology of $\mathscr{Y}$.

This condition is a parallel to that provided in Theorem 9.15 for the basic NOILC algorithm. In a similar manner it can be relaxed by

1. replacing $\mathscr{R}[G]$ by $\mathscr{Y}$ and/or,
2. by constructing positive, self-adjoint bounds $I - L_{M,\{\alpha_j\}} \leq \Gamma_{M,\{\alpha_j\}} \leq I$ to produce simplified sufficient conditions generated from

$$U + U^* > U^* \Gamma_{M,\{\alpha_j\}} U + \varepsilon_0^2 I \quad on \quad \mathscr{R}[G]. \tag{10.153}$$

For example, using only positive values of α_j yields $0 \leq I - L_{M,\{\alpha_j\}} \leq I$ and the robustness condition

$$U + U^* > U^*U + \varepsilon_0^2 I \quad on \quad \mathscr{R}[G] \tag{10.154}$$

which is independent of prediction horizon M, $\{\alpha_j\}_{1\leq j\leq M}$ and λ. For linear, discrete, state space models $S(A, B, C)$ of G, the value $\varepsilon_0 = 0$ can be used and the condition reduces to that seen in the robustness analysis of the Right Inverse Model Algorithm of Sect. 6.2 with learning gain $\beta = 1$.

A comparison of the robustness of the Predictive Algorithm 10.9 with that of the NOILC Algorithm 9.1 can be undertaken by considering the Receding Horizon case when $\alpha_1 = 1$ and all other $\alpha_j = 0$. This gives $L_{M,\{\alpha_j\}} = \tilde{L}_{1,0}$ (which decreases as M increases) and hence

$$I - L_{M,1} \leq I - L_{M+1,1} \tag{10.155}$$

with strict inequality on $\overline{\mathscr{R}[G]}$. That is, for the special case of Receding Horizon prediction, increasing the prediction horizon M will tend to reduce the robustness of the Algorithm 10.9 as the set of permissible modelling errors satisfying the robust monotonicity condition of Theorem 10.7 is reduced in size. It is concluded that the use of $M = 1$ (NOILC Algorithm 9.1) is likely to be the most robust case.

Given the above, the reader may wish to consider the possibility that the use of $M = 2$ could be the best choice if a balance between improving convergence rates and retaining robustness is required? Choosing $\alpha_1 = 1$ and $\alpha_2 = 0$ provides the characterization

$$\begin{aligned} L_{2,\{\alpha_j\}} &= \left(I + \varepsilon^{-2}GG^*(I + \lambda^2(I + \varepsilon^{-2}GG^*)^{-1})\right)^{-1} \quad so\ \ that \\ I - L_{2,\{\alpha_j\}} &= \varepsilon^{-2}GG^*X_2 \leq \beta_P I \quad with \\ \beta_P &= \frac{\varepsilon^{-2}\|GG^*(I + \lambda^2 X_1)\|}{1 + \varepsilon^{-2}\|GG^*(I + \lambda^2 X_1)\|}. \end{aligned} \tag{10.156}$$

A simpler upper bound is obtained as

$$\beta_P = \frac{\varepsilon^{-2}\|G^*\|^2(1 + \lambda^2)}{1 + \varepsilon^{-2}\|G^*\|^2(1 + \lambda^2)}. \tag{10.157}$$

A sufficient condition for robust monotone convergence in the $\|\cdot\|_0$ norm is hence that

$$U + U^* > \beta_P U^*U + \varepsilon_0^2 I \quad on \quad \mathscr{R}[G]. \tag{10.158}$$

which, when $\lambda = 0$, is Condition Two for robustness of NOILC as derived in Theorem 9.16. For $\lambda > 0$, β_P lies in the range $0 < \beta_P < 1$, the values increasing as ε^2 decreases and/or λ increases. Hence,

1. this condition reduces the robustness analysis to that of the Right Inverse Model Algorithm of Sect. 6.2 with learning gain $\beta = \beta_P$.
2. It is also the condition for robustness of NOILC given in Theorems 9.15 and 9.16 with ε^2 replaced by the value $\varepsilon^2/(1 + \lambda^2)$. That is, the effect of prediction on robustness can be visualized as robustness of the basic NOILC Algorithm 9.1 with reduced input weighting.

Both interpretations suggest that the algorithm robustness decreases as λ increases but a substantial degree of robustness remains even if λ is very large (and hence $\beta_P \to 1-$). The reader will note that, with $\varepsilon^2 = 0.5\|G^*\|^2$, β_P is close to one for "small" values of λ e.g. choosing $\lambda = 2$ (respectively, $\lambda = 5$) gives a value of $\beta_P = \frac{10}{11} \approx 0.9$ (respectively, $\beta_P = \frac{52}{53} \approx 0.98$).

Finally, in principle, there are many other choices of $\Gamma_{M,\{\alpha_j\}}$ but their application to robustness analysis depends crucially on the detailed properties of the operators $\{X_j\}_{j\geq 0}$ and the choice of M and $\{\alpha_j\}_{1\leq j\leq M}$. Again assuming the receding horizon case of $\alpha_j = \delta_{j,1}$ and $M \geq 2$ so that $L_{M,\{\alpha_j\}} = L_{M,1} = \tilde{L}_{1,0}$ gives

$$L_{M,1} = (I + \varepsilon^{-2}GG^*(I + \lambda^2 X_{M-1}))^{-1} \geq (I + \varepsilon^{-2}GG^*(1 + \lambda^2\|X_{M-1}\|))^{-1} \tag{10.159}$$

where $\|X_{M-1}\|$ is the norm of X_{M-1} regarded as an operator on the subspace $\overline{\mathscr{R}[G]}$. This formula indicates that robustness is again related to that of NOILC but, now, with ε^2 replaced by the smaller value $\varepsilon^2(1+\lambda^2\|X_{M-1}\|)^{-1}$ which reduces monotonically as M increases. A further simplification, the details of which are left to the reader, returns an inverse-model-like robustness test with

$$\beta_P = \frac{\varepsilon^{-2}\|G^*\|^2(1 + \lambda^2\|X_{M-1}\|)}{1 + \varepsilon^{-2}\|G^*\|^2(1 + \lambda^2\|X_{M-1}\|)}. \tag{10.160}$$

Note that $\|X_{M-1}\|$ can be replaced by any computable upper bound. The calculation of a bound for $\|X_{M-1}\|$ has already been seen for $M = 2$ but, for $M > 2$ a more detailed analysis is needed. For example, an examination of the formula for X_M indicates that, in $\mathscr{Y}$, $X_M \leq I + \lambda^2 X_{M-1}$ and hence, by induction, using $X_0 = 0$, $X_j \leq \sum_{i=1}^{j} \lambda^{2(j-1)} I$ for all $j \geq 1$. That is,

$$\|X_j\| \leq \sum_{i=1}^{j} \lambda^{2(i-1)} \quad \textit{for all} \quad j \geq 1. \tag{10.161}$$

This bound is easily computed but, for large M with $\lambda > 1$, produces increasingly large values and, ultimately, the limit $\beta_P \to 1-$.

10.8 Discussion and Further Reading

The chapter underlines the large number of problems that can be formulated and solved using NOILC-like algorithms. The presentation is not exhaustive but includes the use of state space models whether continuous or sampled. The general approach is identical in each case and is based on the identification of suitable spaces and operators, computation of adjoint operators and direct appeal to NOILC formulae derived in Chap. 9. The details depend on the problem considered but, in principle, the relevant equations and algorithms for differential-algebraic systems and differential delay systems could be derived. Systems described by partial differential equations are not easily covered (unless approximated by finite difference or other numerical techniques) as the operators involved are unbounded in general. The existence of an adjoint is essential as is the solution of the (implicit) formula for u_{k+1} to provide an implementable solution.

At a more practical level, the use of filters in Sect. 10.1 provides practical ways of enhancing and/or suppressing the effect of NOILC in selected frequency ranges. This allows the frequency content of the reference to be reflected in algorithm design and may increase robustness if the frequency ranges where model errors are largest are known and the filters are chosen to reduce the importance of this range on input updates. Typically low pass filters might be used as reference signals are often dominated by low frequency content. The use of notch frequencies to reduce the effects of specified (possibly narrow) frequency ranges can also be considered.

Much of the rest of the chapter is based on more complex input and/or reference signal specifications. The idea of the initial condition being a control variable has a physical motivation and may help in some applications. The solution relies heavily on the feedback form of the basic Algorithm 9.1 for state space systems as seen in Sect. 9.1.6. Its effect is to split each NOILC iteration into two stages, the first being a NOILC iteration from the previous initial condition followed by a NOILC iteration from a new initial condition computed from that data. This idea extends to the case of intermediate point and multi-task ILC as seen in Sects. 10.5, 10.6 and 10.6.2.

Multi-task and Intermediate Point algorithms open up possibilities for wider application of NOILC methodologies. The IPNOILC Algorithm 10.4 was first introduced in [41] with a solution provided in [92]. Its performance was assessed experimentally with considerable success. The results underlined the need for further consideration of both the initial input u_0 and the form of the converged output y_∞, both issues that may be of importance in some applications. The use of constraints is suggested in Sect. 10.5 but other approaches include the addition of auxiliary optimization objectives as in [90, 91], references that provide both a theoretical and experimental assessment. These algorithms are described in Chap. 11.

The multi-task Algorithms are a new addition to the algorithm set for ILC. They arise from the general study in Sect. 10.4 but have specific forms for each class of model and tracking objective. For linear state space systems, Sect. 10.6 provides an optimal control framework to include a combination of intermediate point require-

ments, tracking on subintervals and intervals with no tracking. This allows application to many new situations including

1. initial intervals $[0, \hat{t}_1]$ with no intermediate points or tracking requirements. This freedom could allow the use of control activity to produce an "initial condition" $x(\hat{t}_1)$ for the remaining interval $[\hat{t}_1, T]$ that makes tracking easier and, hopefully, convergence faster.
2. If higher derivatives are important to the tracking objective, the inclusion of rates of change in the measured output may allow intermediate points to be specified. More generally, by putting $C = I_n$, the choice of $\{F_j\}_{1\leq j\leq M}$ and $\{P_j\}_{1\leq j\leq \hat{M}}$ will permit control over any state variables that can be measured or accurately estimated.
3. For mechanical systems where outputs can represent spatial variables, choice of $\{F_j\}_{1\leq j\leq M}$ and $\{P_j\}_{1\leq j\leq \hat{M}}$ could be based on a need to ensure that the system satisfies planar constraints.

The general theory provides the required knowledge of convergence properties in terms of the self-adjoint, positive operator $(I + \varepsilon^{-2} G_e G_e^*)^{-1}$ that defines error updates. This knowledge is extremely important but, in the end, the solution for u_{k+1} reduces to familiar linear quadratic optimal control problems using Riccati equation solutions with "jump" conditions at intermediate points.

The underlying plant G may be non-minimum-phase (NMP). It is natural to ask whether or not the "plateauing" effects noted in Sect. 9.3 and the associated limitations imposed on achievable convergence will also have effects on the algorithms described in this chapter. There is no precise answer to this question but, intuitively, IPNOILC will not be overly sensitive provided that the number of intermediate points is not too large as NMP properties are typically associated with frequency domain characteristics, characteristics that have no obvious relevance to IPNOILC. Multi-task algorithms may suffer from NMP convergence issues but the degree to which this is a practical problem will depend on details. In particular, suppose that there is no subinterval tracking requirement on a sufficiently long initial period $[0, \hat{t}_1]$ in the Multi-task Algorithms of Sect. 10.6. The effects of NMP characteristics will then, intuitively, be reduced as the input on this interval is free to ensure that the state $x(\hat{t}_1)$ could reduce their impact. This is consistent with the notion, mentioned in Sect. 9.3, that $e_0(t)$ should be small around $t = 0$. It is conceivable that the best approach to control of NMP systems lies in a combination of multi-task algorithms with the ideas of Predictive NOILC described in Sect. 10.7, the use of a model relating exponentially weighted signals in a similar manner to that described in Chaps. 6 and 7 and the removal of a tracking requirement on $[0, \hat{t}_1]$.

Finally, predictive NOILC [7] has structural connections to the ideas of Model Predictive Control (see, for example, [22, 72, 98, 110]), particularly in the case when the "Receding Horizon" choice $u_{k+1} = u_{k+1}^{(1)}$ is made, but it has the important additional property that physical interactions between iterations are not present. They occur only after the Iterative Control algorithm is implemented! The parameters added to the design process are the prediction horizon M, the weight $\lambda > 1$ and the parameters $\{\alpha_j\}_{1\leq j\leq M}$, as each iteration generates M possible input functions,

the choice of which to use and in what (convex) combination [56] is a decision that must be made in the context of the application. Increasing the value of λ makes the algorithm closer, in principle, to the inverse algorithm but may make the Riccati solution badly conditioned. Increasing M adds to the computational complexity but, if large, has the benefit, ultimately, of producing reductions by a factor better than λ^{-2}. However, the reader may take the view that this added complexity is best avoided and that it is difficult to devise an argument that justifies the use of $M > 2$. This view has merit and is perhaps strengthened by the robustness analysis which suggests that robustness is strongly related to the robustness of inverse model algorithms and that the predicted robustness reduces as λ increases.

Chapter 11
Iteration and Auxiliary Optimization

The optimization paradigm discussed in previous chapters is focussed on the prime objective of reducing the tracking error norm from iteration to iteration. However, if $ker[G] \neq \{0\}$, the input achieving the desired tracking is non-unique. In state space terms, these ideas apply to cases when $\ell > m$. In such situations a useful by-product of using NOILC methodologies is that convergence also finds the input signal that minimizes the "control energy" measure $\|u - u_0\|^2_{\mathscr{U}}$. It is natural to ask whether or not it is possible to select one such input to satisfy other performance requirements. In what follows, this question is expressed in terms of the construction of iterative algorithms that converge to the desired tracking condition using an input that minimizes a secondary quadratic optimization objective. This is seen to be possible using (right) inverse models or a modified form of a NOILC algorithm but involves an increase in algorithm complexity either in the evaluation of a suitable right inverse operator or through the addition of "inner" iteration loops.

11.1 Models with Auxiliary Variables and Problem Formulation

The approach taken in this chapter is quite general and continues with the ideas and operator formulation of previous chapters. The plant model does, however, need a more detailed definition as it has three components—the plant dynamics, the tracking dynamics and the dynamics of *auxiliary variables*.

Underlying Plant Dynamics: plant dynamics has the form used in NOILC,

$$y = Gu + d \qquad (\textit{Plant Dynamics}) \tag{11.1}$$

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_11

with input u and output y being vectors in the real Hilbert spaces $\mathscr{U}$ and $\mathscr{Y}$ respectively. The operator $G : \mathscr{U} \to \mathscr{Y}$ is linear and bounded and d represents the effects of initial conditions or known disturbances.

Tracking Dynamics: the tracking problem is described by dynamics

$$y^e = G_e u + d^e \qquad (\textit{Tracking Dynamics}) \tag{11.2}$$

with output y^e in the real Hilbert space $\mathscr{Y}_e$. The operator $G_e : \mathscr{U} \to \mathscr{Y}$ is linear and bounded and d^e represents the effects of initial conditions or known disturbances.

Note: this notation allows application of the ideas to signal tracking, intermediate point tracking, multi-task control and other examples considered previously in this text (Sects. 10.5, 10.6 and elsewhere). With initial conditions as "control inputs", the description also includes problems such as that described in Sect. 10.3.

Auxiliary Variables: It is assumed that the plant can be associated with auxiliary signals z regarded as vectors in a real Hilbert space $\mathscr{Z}$ with inner product $\langle \cdot, \cdot \rangle_{\mathscr{Z}}$. These auxiliary variables are constructed from the input signal u from the relation

$$z = G_1 u + d_1 \qquad (\textit{Auxiliary Variable Dynamics}). \tag{11.3}$$

The operator $G_1 : \mathscr{U} \to \mathscr{Z}$ is linear and bounded and d_1 represents the effects of initial conditions or known disturbances.

For continuous state space models $S(A, B, C)$ with state vector $x(t)$ and initial condition x_0,

1. Auxiliary variables can be seen as additional measurements that do not form part of the tracking requirement but over which some control would be desirable.
2. For example, the auxiliary variables could be linear combinations of state variables written in the form $z = Fx$ for some matrix F. The operator G_1 is simply the state space model $S(A, B, F)$ and $d_1(t) = Fe^{At}x_0$.
3. $z(t)$ could be the derivative $\dot{y}(t)$ generated from the model $S(A, B, CA, CB)$. If $CB = 0$, then the second derivative can be constructed from $S(A, B, CA^2, CAB)$.
4. Alternatively, z could be a (causal or non-causal) filtered version of the input u and G_1 the operator associated with a model of that filter. A non-causal filter in this context will typically be the adjoint of a state space system $S(A_1, B_1, C_1, D_1)$.

The Iterative Control problem considered is expressed as follows,

Definition 11.1 *The Iterative Control Problem with Auxiliary Optimization* The Objective of the Iterative Control with Auxiliary Optimization Problem is to find an input signal u_∞ that ensures accurate tracking of a reference signal $r^e \in \mathscr{Y}_e$ whilst also minimizing the objective function

$$J_{\mathscr{Z}}(u, z_0, u_0) = \|z - z_0\|^2_{\mathscr{Z}} + \|u - u_0\|^2_{\mathscr{U}} \tag{11.4}$$

Here u_0 is the initial control signal and $z_0 = G_1 u_0 + d_1$ is the associated auxiliary signal. Formally the objective of iteration is to converge to the input

$$u_\infty = \arg\min_{u \in \mathscr{U}} \{ J_{\mathscr{Z}}(u, z_0, u_0) \ : \quad r^e = G_e u + d^e, \quad z = G_1 u + d_1 \} \tag{11.5}$$

Minimization of the auxiliary objective function $J_{\mathscr{Z}}$ represents a desire to keep z close to z_0 whilst using controls that do not differ too much from u_0. The relative weights given to each of these objectives is reflected in the choice of inner products and norms in $\mathscr{Z}$, $\mathscr{U}$ and $\mathscr{Y}$.

The interpretation of this problem differs from application to application. Some interpretations of auxiliary variables have been given above. They could be physical variables but it is worth noting that they could be introduced to add additional degrees of freedom to NOILC iteration. For example, consider the choice of $z = u$ (and hence $G_1 = I$ and $d_1 = 0$). This apparently trivial definition does have potential value as, if $J_{\mathscr{Z}} = \|u - u_0\|^2_{\mathscr{Z}} + \|u - u_0\|^2_{\mathscr{U}}$ represents the control energy function to be minimized, then changes to the inner products in $\mathscr{Z}$ and $\mathscr{U}$ that leave this unchanged may allow the inner product in $\mathscr{U}$ to be chosen to improve conditioning or convergence of an algorithm. For linear state space systems, this flexibility is made clearer using the inner products

$$\begin{aligned} &\langle u, v \rangle_{\mathscr{U}} = \gamma^2 \int_0^T u^T(t) R v(t) dt, \quad \langle u, v \rangle_{\mathscr{Z}} = \gamma^2 \int_0^T u^T(t) R_{\mathscr{Z}} v(t) dt, \quad \gamma^2 > 0, \\ &\textit{so that} \quad J_{\mathscr{Z}}(u, z_0, u_0) = \gamma^2 \int_0^T (u(t) - u_0(t))^T (R_{\mathscr{Z}} + R)(u(t) - u_0(t)) dt. \end{aligned} \tag{11.6}$$

If the weight defining the chosen control energy is $R_E = \gamma^2 (R_{\mathscr{Z}} + R)$ and this is kept constant, then the two weights $R_{\mathscr{Z}}$ and R and scaling parameter γ^2 can be varied to influence algorithm performance. The only constraints are that they should be symmetric and positive definite.

In what follows, the existence of at least one iterative solution of the problem is demonstrated using an appropriate right inverse algorithm in a similar manner to that seen in Chap. 6. This is followed by a more detailed study of an algorithm that avoids the use of inversion by using a modification of the Norm Optimal Control Algorithm and its extensions described in Chaps. 9 and 10. The presentation assumes that the reader is familiar with that material.

11.2 A Right Inverse Model Solution

The solution of the optimization problem defined by Eq. (11.5) can be written in the form of an equation whenever G_e has a bounded right inverse. More precisely,

Theorem 11.1 (An Important Right Inverse) *Suppose that, for some scalar $g_e \neq 0$, $G_e G_e^* \geq g_e^2 I$. Then, for any bounded linear operator $G_1 : \mathscr{U} \to \mathscr{Z}$,*

1. $G_e(I + G_1^* G_1)^{-1} G_e^*$ *has a bounded inverse and*
2. *the bounded operator* $\hat{G}_{eR} = (I + G_1^* G_1)^{-1} G_e^* (G_e (I + G_1^* G_1)^{-1} G_e^*)^{-1}$ *is well-defined and satisfies* $G_e \hat{G}_{eR} = I$. *It is hence a right inverse of* G_e.

Moreover, the input u_∞ *solving the problem defined by Eq.* (11.5) *is uniquely defined by*

$$u_\infty = u_0 + \hat{G}_{eR} e_0^e \quad \textit{where} \quad e_0^e = r^e - y_0^e \quad \textit{and} \quad y_0^e = G_e u_0 + d^e. \tag{11.7}$$

Proof The existence of a bounded inverse follows as $G_e(I + G_1^* G_1)^{-1} G_e^*$ is self adjoint with spectrum in the closed interval $[(1 + \|G_1\|^2)^{-1} g_e^2, \|G_e^*\|^2]$ that excludes the origin. The right inverse property of $\hat{G}_{eR}$ follows by computing $G_e \hat{G}_{eR}$. The characterization of u_∞ follows by writing $z - z_0 = G_1(u - u_0)$ so that $J_{\mathscr{Z}} = \langle u - u_0, (I + G_1^* G_1)(u - u_0) \rangle_{\mathscr{U}}$. Constructing the Lagrangian

$$\mathscr{L}\left[u, \gamma\right] = \langle u - u_0, (I + G_1^* G_1)(u - u_0) \rangle_{\mathscr{U}} + 2\langle \gamma, r^e - G_e u - d^e \rangle_{\mathscr{Y}_e} \tag{11.8}$$

with Lagrange Multiplier $\gamma \in \mathscr{Y}_e$, the Lagrangian is stationary if the following two conditions are satisfied,

$$(I + G_1^* G_1)(u_\infty - u_0) - G_e^* \gamma = 0 \quad \textit{and} \quad r^e - G_e u_\infty - d^e = 0. \tag{11.9}$$

The result follows by constructing the solution algebraically. □

The following algorithm is a direct parallel to the Right Inverse Algorithm of Chap. 6 but, with a careful selection of a right inverse, error convergence to zero is guaranteed with convergence to the solution of the auxiliary optimization problem being the added bonus.

Algorithm 11.1 (*Inverse Algorithm with Auxiliary Optimization*) With the notation given above and the right inverse defined by Theorem 11.1, let β be a real "learning gain" in the range $0 < \beta < 2$. Let u_0 be the initial control input. Then the Iterative Algorithm defined by the update equation

$$u_{k+1} = u_k + \beta \hat{G}_{eR} e_k^e \quad (\textit{where} \quad e_k^e = r^e - y_k^e \quad \textit{and} \quad y_k^e = G_e u_k + d^e) \tag{11.10}$$

generates a tracking error sequence $\{e_k^e\}_{k \geq 0}$ satisfying the equations

$$e_{k+1}^e = (1 - \beta) e_k^e, \quad \textit{for} \quad k \geq 0 \quad \textit{and} \quad \lim_{k \to \infty} e_k^e = 0. \tag{11.11}$$

Moreover, the input sequence $\{u_k\}_{k \geq 0}$ converges to the limit u_∞ solving the auxiliary optimization problem of Eq. (11.5).

Proof The proof is very similar to that for the right inverse algorithm in Chap. 6. The details are left as an exercise for the reader. The last statement follows from

$$u_{k+1} = u_0 + \sum_{j=0}^{k} (u_{j+1} - u_j) = u_0 + \beta \hat{G}_{eR} \sum_{j=0}^{k} e_j^e = u_0 + \beta \hat{G}_{eR} \sum_{j=0}^{k} (1-\beta)^j e_0^e, \tag{11.12}$$

noting that $|1 - \beta| < 1$, letting $k \to \infty$ and evaluating the summation. □

In summary, Algorithm 11.1 demonstrates the possibility of converging efficiently to the desired solution. The auxiliary objective is embedded in the algorithm by the choice of an appropriate right inverse of G_e. Many of the design comments made in Chap. 6 apply to this case. In particular, the rate of convergence can be controlled by the choice of β and, if the algorithm based on a model G_e is applied to a plant where G_e actually takes the form UG_e with right multiplicative modelling error U, robust monotonic convergence is achieved if

$$U + U^* > \beta U^* U \quad \textit{in the tracking space } \mathscr{Y}_e. \tag{11.13}$$

In the case of the Intermediate Point NOILC problem of Sect. 10.5, $\mathscr{Y}_e$ is finite dimensional and both $G_e G_e^*$ and U are matrices. The robustness condition is a matrix inequality that demonstrates the robustness of the algorithm for some range of β. The only constraint is the familiar positivity requirement that $U + U^* > 0$.

11.3 Solutions Using Switching Algorithms

The Right Inverse Algorithm 11.1 requires (right) invertibility and computation of (right) inverses of operators. Invertibility cannot be guaranteed in many situations and, even if present, may be computationally difficult (particularly in the situation when $\mathscr{Y}_e$ is infinite dimensional). In such cases, an alternative approach is needed or may be useful. In what follows, inversion is replaced by "switching" mechanisms based on a modified sequence of NOILC computations.

11.3.1 Switching Algorithm Construction

Each iteration of the following switching algorithm consists of two steps, the first being on-line or off-line use of NOILC Algorithm 9.1 and the second being an off-line or on-line optimization based on $J_{\mathscr{Z}}$.

Algorithm 11.2 (*A Switching Algorithm for Auxiliary Optimization*) With the notation of previous sections, suppose that the initial input $u_0 \in \mathscr{U}$ is given and that the associated z_0 and error $e_0^e = r^e - y_0^e$ has been computed/measured. Each iteration with index $k + 1$ then consists of TWO steps:

STEP ONE: Given the input u_k, use one (on-line or off-line) iteration of NOILC Algorithm 9.1 to create the input $\tilde{u}_{k+1}$ that minimizes $J(u, u_k)$ for the tracking system $y^e = G_e u + d^e$. That is,

$$\tilde{u}_{k+1} = \arg\min_{u \in \mathscr{U}} J(u, u_k) \quad \textit{subject to} \quad y^e = G_e u + d^e. \tag{11.14}$$

This input can be represented by the feedforward formula

$$\tilde{u}_{k+1} = u_k + \varepsilon^{-2} G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1} e_k^e \quad \textit{where} \quad e_k^e = r^e - y_k^e \tag{11.15}$$

or can be generated by a suitable feedback realization. This step always produces an error that is smaller in norm than that of e_k. However, the input u_{k+1} is, in general, not equal to $\tilde{u}_{k+1}$ and is constructed in the next step.

STEP TWO: Compute the signal z_k associated with u_k and use (off-line or on-line) NOILC techniques to compute the control signal u_{k+1} as the control signal that minimizes

$$J_{\mathscr{Z}}(u, z_k, \tilde{u}_{k+1}) = \|z_k - z\|_{\mathscr{Z}}^2 + \|u - \tilde{u}_{k+1}\|_{\mathscr{U}}^2 \tag{11.16}$$

subject to the auxiliary dynamical constraint $z = G_1 u + d_1$. That is,

$$u_{k+1} = \tilde{u}_{k+1} + G_1^* (z_k - z_{k+1}) . \tag{11.17}$$

Note: This problem can be interpreted as a NOILC problem for the auxiliary system that computes the input u_{k+1} that "optimally" improves tracking of the "reference" z_k using an input that remains close to $\tilde{u}_{k+1}$. Intuitively, this step is best suited to off-line evaluation as the tracking of z_k is not a design requirement and also, perhaps, the real-time measurement of the signal z may not be possible. It will be seen later in this chapter that the step can be implemented on-line in a natural way if $G_1 = G_e$.

The reader will note that, as expressed above, there are two control input sequences $\{u_k\}_{k\geq 0}$ and $\{\tilde{u}_k\}_{k\geq 0}$ and it is the sequence $\{\tilde{u}_k\}_{k\geq 0}$ that is applied to the plant in the first step. If $G_1 = 0$, then a simple calculation shows that the algorithm reduces to NOILC Algorithm 9.1 as $\tilde{u}_k = u_k$, $k \geq 1$.

11.3.2 Properties of the Switching Algorithm

The properties of this algorithm are obtained by producing the relationship between e_{k+1}^e and e_k^e. Firstly note that $z_k - z_{k+1} = G_1(u_k - u_{k+1})$, so that

$$u_{k+1} = \tilde{u}_{k+1} + G_1^* G_1 (u_k - u_{k+1}) = (I + G_1^* G_1)^{-1} \left(\tilde{u}_{k+1} + G_1^* G_1 u_k\right). \tag{11.18}$$

Writing $\tilde{u}_{k+1} = u_k + \varepsilon^{-2} G_e^*(I + \varepsilon^{-2} G_e G_e^*)^{-1} e_k^e$ then gives

$$u_{k+1} = u_k + (I + G_1^* G_1)^{-1} \varepsilon^{-2} G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1} e_k^e \tag{11.19}$$

and hence the error evolution

$$\begin{array}{rcl} e_{k+1}^e & = & L_1 e_k^e, \quad for \quad k \geq 0, \qquad\qquad where \quad the \quad operator \\ L_1 & = & \left(I - \varepsilon^{-2} G_e (I + G_1^* G_1)^{-1} G_e^* \Psi_1\right) \quad with \quad \Psi_1 = (I + \varepsilon^{-2} G_e G_e^*)^{-1}. \end{array} \tag{11.20}$$

The key to characterizing the behaviour of the algorithm is to examine L_1 as an operator on $\mathscr{Y}_e$ but in the alternative topology defined by the inner product

$$\langle y^e, w^e \rangle_0 = \langle y^e, \Psi_1 w^e \rangle_{\mathscr{Y}_e} \tag{11.21}$$

and associated norm $\|y^e\|_0 = \sqrt{\langle y^e, y^e \rangle_0}$.

Theorem 11.2 (Properties of L_1 in the Alternative Topology) *Using the above notation, $L_1 : \mathscr{Y}_e \to \mathscr{Y}_e$ is a positive, self-adjoint operator in the topology induced by $\langle \cdot, \cdot \rangle_0$. It has induced norm*

$$\|L_1\|_0 \leq 1 \tag{11.22}$$

with $ker[I - L_1] = ker[G_e^*]$ and $\overline{\mathscr{R}[I - L_1]} = \overline{\mathscr{R}[G_e]}$.

Proof Noting that $\Psi_1 L_1 = L_1^* \Psi_1$ (the adjoint being defined in the $\langle \cdot, \cdot \rangle_{\mathscr{Y}_e}$ topology),

$$\langle y^e, L_1 w^e \rangle_0 = \langle y^e, \Psi_1 L_1 w^e \rangle_{\mathscr{Y}_e} = \langle y^e, L_1^* \Psi_1 w^e \rangle_{\mathscr{Y}_e} = \langle L_1 y^e, \Psi_1 w^e \rangle_{\mathscr{Y}_e} = \langle L_1 y^e, w^e \rangle_0 \tag{11.23}$$

and hence L_1 is self-adjoint in the topology induced by the inner product $\langle \cdot, \cdot \rangle_0$ (see also Theorem 2.15). For the rest of the theorem, it is sufficient to prove that, in the original topology in $\mathscr{Y}_e$,

$$0 < \Psi_1 \left(I - \varepsilon^{-2} G_e (I + G_1^* G_1)^{-1} G_e^* \Psi_1\right) \leq \Psi_1. \tag{11.24}$$

The second inequality follows as the self-adjoint operator $\Psi_1 G_e (I + G_1^* G_1)^{-1} G_e^* \Psi_1 \geq 0$. The first inequality is a consequence of the identity

$$\begin{array}{rl} \Psi_1 L_1 = & \Psi_1 \left[I - \varepsilon^{-2} G_e G_e^* \Psi_1 + \varepsilon^{-2} G_e G_1^* (I + G_1 G_1^*)^{-1} G_1 G_e^* \Psi_1\right] \\ \geq & \Psi_1 \left[I - \varepsilon^{-2} G_e G_e^* \Psi_1\right] = \Psi_1^2 > 0. \end{array} \tag{11.25}$$

Finally, $(I - L_1)e = 0$ if, and only if, $G_e^* \Psi_1 e = 0$ i.e. $\Psi_1 e \in ker[G_e^*]$ which is equivalent to the condition $e \in ker[G_e^*]$. An immediate consequence of this result is that the closure of the range $\mathscr{R}[I - L_1]$ is just the orthogonal complement of $ker[G_e^*]$ in the alternative topology, a property shared by $\overline{\mathscr{R}[G_e]}$. That completes the proof. □

These properties immediately allow the application of the general convergence Theorems 5.4 and 5.9 to produce the result,

Theorem 11.3 (Error Convergence in Switching Algorithm 11.2) *Suppose that u_0 has been specified. Then the sequence of errors $\{e_k^e\}_{k\geq 0}$ satisfies the monotonicity condition and boundedness property*

$$\|e_{k+1}^e\|_0 \leq \|e_k^e\|_0 \quad \textit{for all} \quad k \geq 0. \tag{11.26}$$

In particular,

$$\lim_{k\to\infty} e_k^e = P_{ker[G^*]} e_0^e \tag{11.27}$$

where $P_{ker[G^*]}$ is the orthogonal projection operator onto $ker[G_e^*]$ in $\mathscr{Y}_e$ with respect to the inner product $\langle\cdot,\cdot\rangle_0$. In addition,

1. *The error sequence converges to zero if, and only if $e_0^e \in \overline{\mathscr{R}[G_e]}$.*
2. *Convergence to zero tracking error is assured for all $e_0^e \in \mathscr{Y}_e$ if and only if $\mathscr{Y}_e = \overline{\mathscr{R}[G_e]}$. This is always the case if $ker[G_e^*] = \{0\}$.*

Proof If $\|L_1\|_0 < 1$, then Theorem 5.4 implies that the error converges to zero monotonically for all initial errors e_0^e. The remaining case to consider is that when $\|L_1\|_0 = 1$. This case is covered by Theorem 5.9 noting that, from the previous result, $\overline{\mathscr{R}[I-L_1]} = \overline{\mathscr{R}[G_e]}$. □

Theorem 11.4 (Input Convergence Conditions for Switching Algorithm 11.2) *Using the ideas developed above, suppose that, in addition, the initial error $e_0^e \in \mathscr{R}[I-L_1]$ then both input signal sequences $\{u_k\}_{k\geq 0}$ and $\{\tilde{u}_k\}_{k\geq 0}$ converge in the norm topology of $\mathscr{U}$ to the same unique limit u_∞ (dependent on u_0 and r^e) defined by the relation*

$$\begin{gathered} u_\infty = u_0 + (I + G_1^* G_1)^{-1} \varepsilon^{-2} G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1} \hat{e}_0^e \\ \textit{where} \quad e_0^e = (I - L_1)\hat{e}_0 \quad \textit{and} \quad \hat{e}_0^e \in \mathscr{Y}_e. \end{gathered} \tag{11.28}$$

Proof Writing

$$u_{k+1} = u_k + (I + G_1^* G_1)^{-1} \varepsilon^{-2} G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1} e_k^e, \quad k \geq 0 \tag{11.29}$$

yields, using $e_{j+1}^e = L_1 e_j^e$ for $j \geq 0$,

$$u_{k+1} = u_k + (I + G_1^* G_1)^{-1} \varepsilon^{-2} G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1} L_1^k e_0^e, \quad k \geq 0 \tag{11.30}$$

so that

$$u_{k+1} = u_0 + (I + G_1^* G_1)^{-1} \varepsilon^{-2} G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1} \sum_{j=0}^{k} L_1^j e_0^e, \quad k \geq 0. \tag{11.31}$$

Let $\hat{e}_0^e$ be the unique vector in $\overline{\mathscr{R}[I - L_1]}$ satisfying $e_0^e = (I - L_1)\hat{e}_0^e$. It follows that

$$u_{k+1} = u_0 + (I + G_1^* G_1)^{-1} \varepsilon^{-2} G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1} (I - L_1^{k+1}) \hat{e}_0^e, \quad k \geq 0, \tag{11.32}$$

which converges to the defined u_∞ as $k \to \infty$ by Theorem 11.3. □

The results prove convergence of the switching Algorithm 11.2 but say nothing about the relationship between the limit and the auxiliary optimization problem. The key to describing the nature of the limit lies in input convergence Theorem 11.4 from which, if $e_0^e \in \mathscr{R}[I - L_1]$, the limit u_∞ exists and the error sequence $\{e_k^e\}_{k \geq 0}$ converges in norm to zero. As a consequence, the limit is described by the relations

$$\begin{gathered} r^e = G_e u_\infty + d^e, \quad u_\infty = u_0 + (I + G_1^* G_1)^{-1} G_e^* \gamma \\ \textit{and} \quad \gamma = \varepsilon^{-2} (I + \varepsilon^{-2} G_e G_e^*)^{-1} \hat{e}_0^e. \end{gathered} \tag{11.33}$$

Writing the auxiliary optimization problem as that of minimizing

$$J_{\mathscr{Z}} = \langle u - u_0, (I + G_1^* G_1)(u - u_0) \rangle_{\mathscr{U}} \quad \textit{subject to} \quad r^e = G_e u + d^e \tag{11.34}$$

then yields the reassuring result that

Theorem 11.5 (The Nature of the Limit of Algorithm 11.2) *If the assumptions of Theorem 11.4 hold, then Switching Algorithm 11.2 converges to a stationary point of the Lagrangian*

$$\mathscr{L}[u, \gamma] = \langle u - u_0, (I + G_1^* G_1)(u - u_0) \rangle_{\mathscr{U}} + 2 \langle \gamma, r^e - G_e u - d^e \rangle_{\mathscr{Y}_e} \tag{11.35}$$

where the Lagrange Multiplier $\gamma = \varepsilon^{-2} (I + \varepsilon^{-2} G_e G_e^*)^{-1} \hat{e}_0^e$ *. This stationary point input* u_∞ *is unique and corresponds to the minimum of* $\mathscr{L}$*.*

The proof is left as an exercise for the reader.

In summary, the switching algorithm uses familiar NOILC-like computations to construct a convergent solution to the tracking problem that also converges to a solution to the auxiliary optimization problem expressed in terms of a stationary point of an associated Lagrangian function. The proof and algorithm properties are expressed in terms of inputs $\{u_k\}_{k \geq 0}$ and their associated errors $\{e_k\}_{k \geq 0}$ although, as described in Algorithm 11.2, it is the input signals $\{\tilde{u}_k\}_{k \geq 0}$ that may be used on-line. For a feedforward implementation, both inputs are computed off-line so either could be used in principle.

11.3.3 Characterization of Convergence Rates

The detailed form of the convergence rate of Algorithm 11.2 depends on both G_1 and G_e. It is not, in general, possible to compute the spectrum of L_1. It is, however, possible to bound the eigenvalues of L_1 as follows,

Theorem 11.6 (Bounds on the Spectrum of L_1) *The spectrum (and hence all eigenvalues) of L_1 is (are) real. Points σ^2 in the spectrum spec$[L_1]$ can be bounded by the relations*

$$\frac{1}{1+\varepsilon^{-2}\|G_e^*\|^2} \leq \sigma^2 \leq \frac{\|\Psi_1\|+\|G_1\|^2}{1+\|G_1\|^2} \quad \textit{where} \quad \Psi_1 = (I+\varepsilon^{-2}G_eG_e^*)^{-1}. \tag{11.36}$$

Proof As L_1 is strictly positive and self adjoint in the topology defined by $\langle\cdot,\cdot\rangle_0$, its spectrum contains only real, positive values. They are bounded as above as

$$\begin{aligned}\Psi_1 L_1 &= \Psi_1 - \Psi_1\varepsilon^{-2}G_e(1+G_1^*G_1)^{-1}G_e^*\Psi_1 \leq \Psi_1 - \Psi_1\varepsilon^{-2}G_e(1+\|G_1\|^2)^{-1}G_e^*\Psi_1\\ &= \Psi_1\left[I-(1+\|G_1\|^2)^{-1}\varepsilon^{-2}G_eG_e^*\Psi_1\right] = \Psi_1\left[I-(1+\|G_1\|^2)^{-1}(I-\Psi_1)\right]\\ &= \Psi_1\left[\tfrac{\|G_1\|^2}{(1+\|G_1\|^2)}I + \tfrac{1}{(1+\|G_1\|^2)}\Psi_1\right] \leq \Psi_1\left(\tfrac{\|\Psi_1\|+\|G_1\|^2}{1+\|G_1\|^2}\right).\end{aligned} \tag{11.37}$$

Also, $\Psi_1 L_1 = \Psi_1 - \Psi_1\varepsilon^{-2}G_e(I+G_1^*G_1)^{-1}G_e^*\Psi_1$ is bounded from below by

$$\Psi_1\left[I-\varepsilon^{-2}G_eG_e^*\Psi_1\right] = \Psi_1^2 \geq \Psi_1(1+\varepsilon^{-2}\|G_e^*\|^2)^{-1}. \tag{11.38}$$

This completes the proof of the result. □

An immediate consequence of this is that, for all $k \geq 0$,

$$\|e_{k+1}^e\|_0 \leq \lambda_e\|e_k^e\|_0 \quad \textit{where} \quad \lambda_e = \left(\frac{\|\Psi_1\|+\|G_1\|^2}{1+\|G_1\|^2}\right). \tag{11.39}$$

It is deduced that

1. The norm sequence typically has a geometric bound $\lambda_e < 1$ only when $\|\Psi_1\| < 1$, a situation that is achieved only when $G_eG_e^*$ has a bounded inverse. It is necessary that the origin is not in the spectrum of $G_eG_e^*$, a condition that requires that $ker[G_e^*] \neq \{0\}$.
2. The remaining case when $\|\Psi_1\| = 1$ corresponds to situations where either $ker[G_e^*] \neq \{0\}$ and/or the tracking space $\mathscr{Y}_e$ is infinite dimensional and $0 \in spec[G_eG_e^*]$.

The first case covers applications including discrete state space systems, multi-rate systems and the intermediate point tracking problems of Algorithm 10.4 with the added dimension of auxiliary optimization. The second case includes application to continuous state space systems whether that application is the basic NOILC Algorithm 9.1 or, for example, its multi-task versions as in Algorithms 10.6 and 10.8.

Finally, the choice of weights in $J_{\mathscr{Z}}$ (or, equivalently, inner products in $\mathscr{Z}$ and $\mathscr{U}$) will influence convergence rates through the operator $G_1^*G_1$. To represent this replace $J_{\mathscr{Z}}$ by

$$J_{\mathscr{Z}} = \|z-z_0\|_{\mathscr{Z}}^2 + \varepsilon_1^2\|u-u_0\|_{\mathscr{U}}^2 \quad \textit{where} \quad \varepsilon_1^2 > 0 \tag{11.40}$$

is a parameter representing changes in the relative weights of the two terms in the objective function. The effect on the analysis is to replace G_1^* by $\varepsilon_1^{-2}G_1^*$. Consider the equation $w_{\varepsilon_1} = (I+\varepsilon_1^{-2}G_1^*G_1)^{-1}v$, with $v \neq 0$, as $\varepsilon_1 > 0$ varies. Note that $\|w_{\varepsilon_1}\|_{\mathscr{U}} \leq \|v\|_{\mathscr{U}}$ for all $\varepsilon_1 > 0$. Writing the equation in the form $\varepsilon_1^2 w_{\varepsilon_1} + G_1^*G_1 w_{\varepsilon_1} = \varepsilon_1^2 v$ and letting $\varepsilon_1 \to 0$ indicates that $\lim_{\varepsilon_1 \to 0} \|G_1 w_{\varepsilon_1}\|_{\mathscr{U}} = 0$. It follows that, using this notation,

Theorem 11.7 (Algorithm Behaviour as $\varepsilon_1 \to 0+$) *Suppose that* $\mathscr{U} = \overline{\mathscr{R}[G_1^*]}$. *Then there exists a non-zero initial error* e_0^e *such that the iteration error* $e_1^e = L_1 e_0^e$ *in iteration number one converges to* e_0^e *in the weak topology in* $\mathscr{Y}_e$ *as* $\varepsilon_1 \to 0+$.

Proof First note that, for any $f \in \mathscr{Z}$, $\lim_{\varepsilon_1 \to 0} \langle f, G_1 w_{\varepsilon_1} \rangle_{\mathscr{Z}} = \lim_{\varepsilon_1 \to 0} \langle G_1^* f, w_{\varepsilon_1} \rangle_{\mathscr{U}}$ $= 0$ which proves weak convergence as $\{w_{\varepsilon_1}\}_{\varepsilon_1 > 0}$ is bounded and $\mathscr{R}[G_1^*]$ is dense. Next choose any e_0^e such that $v = G_e^* \Psi_1 e_0^e \neq 0$ and note that, for any $f \in \mathscr{Y}_e$,

$$\begin{aligned} \langle f, e_0^e - e_1^e \rangle_{\mathscr{Y}_e} &= \langle f, (I - L_1) e_0^e \rangle_{\mathscr{Y}_e} \\ = \varepsilon^{-2} \langle f, G_e (I + \varepsilon_1^{-2} G_1^* G_1)^{-1} v \rangle_{\mathscr{Y}_e} &= \varepsilon^{-2} \langle G_e^* f, w_{\varepsilon_1} \rangle_{\mathscr{U}} \to 0 \end{aligned} \tag{11.41}$$

as $\varepsilon_1 \to 0$. This completes the proof. □

The value of this result lies in its interpretation in terms of the choice of $J_{\mathscr{Z}}$. More precisely, as the weighting on $(z - z_0)$ increases relative to the weight allocated to $(u - u_0)$, there will be a tendency for convergence rates to slow. That is, there is an implicit limitation in the switching algorithm that says, either accept slow convergence rates or accept that the priority in auxiliary optimization must reflect a priority to minimize the energy measure $\|u - u_0\|_{\mathscr{U}}^2$. This does not mean that control over the dynamics of z via the magnitude of $\|z - z_0\|_{\mathscr{Z}}^2$ cannot be usefully achieved.

The details of performance will depend on the details of the parameters and operators used and occurring in the application. The examples described in the next few sections are chosen for their relative simplicity, their potential usefulness in practice and their links to previous chapters.

11.3.4 Decoupling Minimum Energy Representations from NOILC

Consider the choice of $z = u$ (and hence $G_1 = I$ and $d_1 = 0$). This apparently trivial definition does have potential value as, if $J_{\mathscr{Z}} = \|u - u_0\|_{\mathscr{Z}}^2 + \|u - u_0\|_{\mathscr{U}}^2$ represents the control energy function to be minimized, then changes to the inner products in $\mathscr{Z}$ and $\mathscr{U}$ that leave this unchanged may allow the inner product in $\mathscr{U}$ to be chosen to improve conditioning or convergence of an algorithm. For linear, continuous state space systems, choose the inner products

$$\langle u, v\rangle_{\mathscr{U}} = \int_0^T u^T(t)Rv(t)dt, \quad \langle u, v\rangle_{\mathscr{Z}} = \int_0^T u^T(t)R_{\mathscr{Z}}v(t)dt,$$
$$so \quad that \quad J_{\mathscr{Z}}(u, z_0, u_0) = \int_0^T (u(t) - u_0(t))^T(R_{\mathscr{Z}} + R)(u(t) - u_0(t))dt. \tag{11.42}$$

If the weight defining the chosen control energy is R_E, then the minimum energy solution will be the same if R_E is replaced by $r_E^2 R_E$ where $r_E^2 > 0$ can be chosen arbitrarily. If the sum $r_E^2 R_E = R_{\mathscr{Z}} + R$ and is satisfied, then r_E^2 and the two symmetric and positive definite weights $R_{\mathscr{Z}}$ and R can be varied to influence algorithm performance in STEP ONE whilst avoiding the problem of slow convergence due to STEP TWO. In this case STEP TWO of Switching Algorithm 11.2 reduces to the minimization of

$$(u_k(t) - u(t))^T R_{\mathscr{Z}}(u_k(t) - u(t)) + (\tilde{u}_{k+1}(t) - u(t))^T R(\tilde{u}_{k+1}(t) - u(t)), \tag{11.43}$$

for every $t \in [0, T]$. That is, $u_{k+1}(t)$ is obtained from the formula

$$u_{k+1}(t) = (R_{\mathscr{Z}} + R)^{-1}(R_{\mathscr{Z}}u_k(t) + R\tilde{u}_{k+1}(t)) \quad for \quad t \in [0, T]. \tag{11.44}$$

The case of linear, discrete state space systems can be derived in a similar way.

11.3.5 Intermediate Point Tracking and the Choice $G_1 = G$

Suppose that the auxiliary dynamics is simply the underlying plant dynamics $y = Gu + d$ evolving in $\mathscr{Y}$. Then the choice of $G_1 = G$ and $d_1 = d$ is a statement that, although the tracking problem is the main objective of iteration, there is a secondary objective of influencing the nature of the resultant underlying output behaviour. In this case without loss of generality, choose $\mathscr{Z} = \mathscr{Y}$ and

$$J_{\mathscr{Z}}[u, y_0, u_0] = \|y - y_0\|_{\mathscr{Y}}^2 + \varepsilon_1^2\|u - u_0\|_{\mathscr{U}}^2 \tag{11.45}$$

where $\varepsilon_1^2 > 0$ provides a NOILC-like mechanism for influencing the relative weights in $J_{\mathscr{Z}}$. In this case STEP TWO in Algorithm 11.2 is precisely a NOILC iteration for the plant with ε replaced by ε_1, the reference signal replaced by y_k and u_k replaced by $\tilde{u}_{k+1}$. This step could be completed on-line for a feedback implementation, or off-line, using a plant model.

One interesting possibility for the above is the use of auxiliary optimization for influencing the form of the solution of the Intermediate Point Algorithm 10.4 for state space systems. The solution of the IPNOILC problem also minimizes $\|u - u_0\|_{\mathscr{U}}^2$, a fact that can lead to periods of relative input inactivity and decay of outputs. Conditioning of the solution could be undertaken using a mix of algorithms. One example is as follows:

1. Start the control design by choosing a reference signal $r(t)$ on $[0, T]$ that satisfies the intermediate point requirement at the intermediate times $\{t_j\}$ and represents what is seen as a "good" trajectory between these times. Note that, tracking of $r(t)$ is not the ultimate objective.
2. Next, undertake a number ($k_a \geq 1$ say) of iterations of NOILC Algorithm 9.1 to produce an output $y_{k_a}(t)$, $t \in [0, T]$ that is regarded as reasonable.
3. Finally, use the data $\mathscr{D} = (y_{k_a}, u_{k_a})$ to initialize the Switching Algorithm 11.2 with G_e and $\mathscr{Y}_e$ as defined for intermediate point problems, $J_{\mathscr{Z}}$ defined as above and with the data $y_0 = y_{k_a}$ and $u_0 = u_{k_a}$. Convergence to the solution of the associated auxiliary optimization problem will then provide a limit that solves the intermediate point tracking problem whilst keeping plant outputs and control inputs "close to" the data set $\mathscr{D}$.

The mixture of algorithms is, in effect, a new algorithm. It is not presented in this way as its properties are clearly those of Algorithm 11.2 used as an approach to solving/conditioning the Intermediate Point tracking problem. The first k_a "iterations" set up the data $\mathscr{D}$. The degree to which they represent the nature of the converged solution cannot be quantified at a general theoretical level. Intuitively, however, the approach has the potential to positively influence the achieved solution.

11.3.6 Restructuring the NOILC Spectrum by Choosing $G_1 = G_e$

This section considers the use of Auxiliary Optimization as a tool in the re-design of NOILC Algorithm 9.1. To do this consider the case when $G_1 = G_e$ and $d_1 = d^e$ and, in effect, $\mathscr{Z} = \mathscr{Y}_e$ but with

$$J_{\mathscr{Z}}[u, y_0^e, u_0] = \|y^e - y_0^e\|^2_{\mathscr{Y}_e} + \varepsilon_1^2 \|u - u_0\|^2_{\mathscr{U}} \tag{11.46}$$

which is the NOILC objective function of Algorithm 9.1 with ε replaced by a, possibly different, value ε_1. The corresponding operator L_1 has the form

$$\begin{aligned}
L_1 &= I - \varepsilon^{-2} G_e (I + \varepsilon_1^{-2} G_e^* G_e)^{-1} G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1} \\
&= (I + \varepsilon^{-2} G_e G_e^*)^{-1} + \left((I + \varepsilon^{-2} G_e G_e^*)^{-1} \varepsilon^{-2} G_e G_e^*\right) \left((I + \varepsilon_1^{-2} G_e G_e^*)^{-1} \varepsilon_1^{-2} G_e G_e^*\right) \\
&\geq (I + \varepsilon^{-2} G_e G_e^*)^{-1} \quad \textit{in} \quad \mathscr{Y}_e. \\
&\textit{or, alternatively,} \\
L_1 &= (I + \varepsilon_1^{-2} G_e G_e^*)^{-1} \varepsilon_1^{-2} G_e G_e^* + (I + \varepsilon^{-2} G_e G_e^*)^{-1} (I + \varepsilon_1^{-2} G_e G_e^*)^{-1} \\
&\geq (I + \varepsilon_1^{-2} G_e G_e^*)^{-1} \varepsilon_1^{-2} G_e G_e^*.
\end{aligned} \tag{11.47}$$

These expressions describe the effect of the auxiliary optimization and, in particular, the first suggests its tendency to slow down convergence rates when compared with NOILC Algorithm 9.1 whilst the second supports Theorem 11.7 by suggesting slow convergence if ε_1 is small. In the case when $G_1 = G_e$ however, the minimization of

$J_{\mathscr{Z}}$ is ultimately identical to the minimization of $\|u-u_0\|^2_{\mathscr{U}}$ whenever $y^e \to r^e$. The algorithm therefore converges to the same solution as Algorithm 9.1 but the dynamics of the iteration is changed. This change is implicit in the form of L_1 but is most easily characterized using spectral values or, more precisely, eigenvalue behaviour.

An Eigenvalue Interpretation: Suppose therefore that $G_eG_e^* : \mathscr{Y}_e \to \mathscr{Y}_e$ has zero eigenvalues with eigenvectors equal to any element of $ker[G_e^*]$ and that the remaining non-zero spectral values are eigenvalues $\|G_e^*\|^2 = \sigma_1^2 \geq \sigma_2^2 \geq \cdots$ which are all strictly positive and have associated orthonormal eigenvectors $\{v_j\}_{j\geq 1}$ spanning $\overline{\mathscr{R}[G_e]}$. The Spectral Mapping Theorem implies that the eigenvalues of L_1 are hence unity on $ker[G_e^*]$, and can be expressed, for $j \geq 1$, in two ways

$$\begin{aligned}
\sigma_j^2(L_1) &= \frac{1}{(1+\varepsilon^{-2}\sigma_j^2)} + \left(\frac{\varepsilon^{-2}\sigma_j^2}{(1+\varepsilon^{-2}\sigma_j^2)}\right)\left(\frac{\varepsilon_1^{-2}\sigma_j^2}{(1+\varepsilon_1^{-2}\sigma_j^2)}\right) > \frac{1}{(1+\varepsilon^{-2}\sigma_j^2)} \quad \textit{and} \\
\sigma_j^2(L_1) &= \frac{\varepsilon_1^{-2}\sigma_j^2}{(1+\varepsilon_1^{-2}\sigma_j^2)} + \left(\frac{1}{(1+\varepsilon^{-2}\sigma_j^2)}\right)\left(\frac{1}{(1+\varepsilon_1^{-2}\sigma_j^2)}\right) > \frac{\varepsilon_1^{-2}\sigma_j^2}{(1+\varepsilon_1^{-2}\sigma_j^2)}.
\end{aligned} \tag{11.48}$$

Introduction of the auxiliary optimization therefore implies that convergence of the Switching Algorithm 11.2 is slower than NOILC Algorithm 9.1. At first sight, this suggests that the algorithm has little value. However, the point to note is that the effect of switching is to change the evolution of the component of the output that lies in the subspace spanned by the eigenvectors corresponding to each new eigenvalue value $\sigma_j^2(L_1)$. Perhaps of more significance is the observation that the influence of the algorithm using the choice of ε^2 and ε_1^2 is to modify the distribution of these changes. Some of these modifications can be linked to potentially useful properties for practical applications. Some of these properties are discussed below:

1. For small values of σ_j^2, the presence of the term σ_j^4 in the second term suggests that these eigenvector components evolve as $(\sigma_j^2(L_1))^k \approx (1+\varepsilon^{-2}\sigma_j^2)^{-k}$ which is that of NOILC Algorithm 9.1. That is, auxiliary optimization techniques have little effect on these components.
   Conclusion: Iteration dynamics for small eigenvalues (high frequencies) is relatively unaffected by the addition of the auxiliary optimization criterion. This component of Algorithm performance is influenced solely by the choice of ε^2.
2. As $\varepsilon^2 \to \infty$, the non-unity eigenvalues $\sigma_j^2(L_1)$ approach the value unity. Eigencomponents are then almost unchanged from iteration to iteration.
3. As $\varepsilon^2 \to 0$, the non-unity eigenvalues $\sigma_j^2(L_1)$ approach $(1+\varepsilon_1^{-2}\sigma_j^2)^{-1}\varepsilon_1^{-2}\sigma_j^2$ from above. This eigenvalue distribution eliminates the low eigenvalue components rapidly with larger eigenvalues being eliminated more slowly. The slowest rate of elimination is described by the geometric sequence $\left((1+\lambda)^{-1}\lambda\right)^k$ where λ is defined by $\lambda = \varepsilon_1^{-2}\sigma_1^2 = \varepsilon_1^{-2}\|G_e^*\|^2$. For example, if $\lambda = 1$, the contribution of each eigenvalue reduces faster than $\left(\frac{1}{2}\right)^k$.

Conclusion: As $\varepsilon^2 \to 0$, the convergence rate is dominated by the behaviour and parameter choice in the auxiliary optimization step. In particular, it is dominated by the ratio λ between $\|G_e^*\|^2$ and ε_1^2.

4. The eigenvalue variation with ε can be bounded in terms of λ. That is,

$$\begin{aligned}\sigma_j^2(L_1) &\le \frac{1}{(1+\varepsilon^{-2}\sigma_j^2)} + \left(\frac{\varepsilon^{-2}\sigma_j^2}{(1+\varepsilon^{-2}\sigma_j^2)}\right)\left(\frac{\lambda}{(1+\lambda)}\right) \\ &= \frac{\lambda}{(1+\lambda)} + \frac{1}{(1+\varepsilon^{-2}\sigma_j^2)}\left(\frac{1}{(1+\lambda)}\right)\end{aligned} \tag{11.49}$$

This formula bounds the spectrum of L_1 in terms of the spectrum $(1+\varepsilon^{-2}\sigma_j^2)^{-1}$ of the NOILC operator $(I+\varepsilon^{-2}G_eG_e^*)^{-1}$ and λ. As λ gets smaller (as ε_1 increases), the weighting of the input in $J_{\mathscr{Z}}$ gets larger and the convergence rate approaches that of basic NOILC Algorithm 9.1. If, however, λ is fixed and ε is reduced substantially, then $\sigma_j(L_1) \approx \lambda/(1+\lambda)$ over a wide range of the larger eigenvalues. *Conclusion: Useful eigenvalue bounds can be expressed in terms of $\lambda = \varepsilon_1^{-2}\|G_e^*\|^2$ and ε. They provide a useful way of visualizing the effects of parameter choice on the spectrum.*

5. Relative convergence speeds can be investigated by considering

$$\begin{aligned}(1+\varepsilon_1^{-2}\sigma_j^2)^{-1} - \sigma_j^2(L_1) &= \frac{\varepsilon^{-2}\sigma_j^2}{(1+\varepsilon^{-2}\sigma_j^2)}\frac{1}{(1+\varepsilon_1^{-2}\sigma_j^2)} - \frac{\varepsilon_1^{-2}\sigma_j^2}{(1+\varepsilon_1^{-2}\sigma_j^2)} \\ &= \frac{\sigma_j^2}{(1+\varepsilon_1^{-2}\sigma_j^2)}\left(\frac{\varepsilon^{-2}}{(1+\varepsilon^{-2}\sigma_j^2)} - \varepsilon_1^{-2}\right).\end{aligned} \tag{11.50}$$

Conclusion: The auxiliary switching algorithm converges faster than Algorithm 9.1 with ε replaced by ε_1 if

$$\varepsilon_1^2 \ge \varepsilon^2 + \|G_e^*\|^2 \qquad (or, \quad equivalently, \quad 1 \ge \varepsilon_1^{-2}\varepsilon^2 + \lambda). \tag{11.51}$$

This follows by requiring the expression above to be positive for all $j \ge 1$.

6. In particular, the previous items suggest a "flattening" of the spectrum of L_1 relative to that of NOILC can be achieved. To illustrate this, set $\varepsilon_1^2 = \varepsilon^2 + \|G_e^*\|^2$, then eigen-components in the vicinity of $\sigma_1^2 = \|G_e^*\|^2$ converge at the same rate as those in NOILC with the weight ε_1^2 but eigen-components in some range below this converge faster than that achieved by NOILC.

Design Implications? The suggested interpretation of these observations is that, with $G_1 = G_e$, the values of ε^2 and ε_1^2 are the major influences on the nature of the convergence. In particular, the motivation for choosing $G_1 = G_e$ is found in the ability to influence both the convergence rate and the range of eigenvalues that have that convergence rate. The approach uses high gains (small values of ε) in STEP ONE whilst using lower gains (larger values of $\varepsilon_1 > \varepsilon$) in STEP TWO of Algorithm 11.2. As lower gains are normally preferred for physical plant operation, it is therefore expected that either a feedforward implementation is used or,

alternatively, STEP TWO will be better suited to on-line implementation with STEP ONE being computed off-line using a model. If this is done, then Inequality (11.51) is a sufficient condition to ensure that a real benefit is achieved relative to a NOILC algorithm with the weight value ε_1^2.

11.4 A Note on Robustness of Switching Algorithms

Suppose that a feedforward implementation of Algorithm 11.2 is used. Suppose also that the parameter ε_1 introduced into the auxiliary objective function to analyse the effect of relative control weightings is retained in what follows. In principle, modelling errors for both G_e and G_1 should be considered but, for our purposes here, only errors in G_e will be considered. In effect, G_1 is to be seen only as a mathematical model defining the auxiliary minimization objective in terms of the computed z and known input u. In more detail, if, on iteration k, the input u_k produced the measured output e_k^e, then

1. compute the input $\tilde{u}_{k+1}$ from STEP ONE of Switching Algorithm 11.2. Using the underlying representation,

$$\tilde{u}_{k+1} = u_k + \varepsilon^{-2} G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1} e_k^e = u_k + \Delta u_{k+1}, \tag{11.52}$$

   Δ_{k+1} is identical to the input generated in one step of NOILC Algorithm 9.1 with initial input $u_k = 0$, zero initial conditions and reference signal $r^e = e_k^e$.
2. Next undertake STEP TWO using the model G_1 to evaluate the input u_{k+1} for iteration $k+1$. This produces the input u_{k+1} which has an underlying representation,

$$\begin{array}{rcl} u_{k+1} & = & \tilde{u}_{k+1} + \varepsilon_1^{-2} G_1^* (z_k - z_{k+1}) \\ & = & u_k + (I + \varepsilon_1^{-2} G_1^* G_1)^{-1} \varepsilon^{-2} G_e^* (I + \varepsilon^{-2} G_e G_e^*)^{-1} e_k^e. \end{array} \tag{11.53}$$

If the plant can be described by a model UG_e with left multiplicative perturbation $U : \mathscr{Y}_e \to \mathscr{Y}_e$, then, applying this input to the plant indicates that the plant error evolution is described by

$$e_{k+1}^e = (I - U(I - L_1))\, e_k^e, \quad for \quad k \geq 0. \tag{11.54}$$

The algebraic similarities with the analysis of Sect. 10.7.6 are apparent. In particular, the subspace decomposition

$$\mathscr{Y}_e = ker[G_e^*] \oplus \overline{\mathscr{R}[UG_e]} \tag{11.55}$$

is seen to be desirable and can be guaranteed using Theorem 9.13 with G replaced by G_e. Also, as $ker[I - L_1] = ker[G_e^*]$, any component of e_0^e that lies in $ker[G_e^*]$

remains unchanged from iteration to iteration. The evolution of the error lies therefore in $\overline{\mathscr{R}[UG_e]}$.

The effect of the auxiliary optimization step is that *two additional alternative topologies* on $\overline{\mathscr{R}[UG_e]}$ are required for the following robustness analysis. These are defined by the inner products,

$$\langle y^e, w^e\rangle_0 = \langle y^e, \Psi_1 w^e\rangle_{\mathscr{Y}_e} \quad \textit{and} \quad \langle y^e, w^e\rangle_1 = \langle y^e, \Psi_1(I - L_1)w^e\rangle_{\mathscr{Y}_e}\,. \tag{11.56}$$

and associated norms $\|\cdot\|_0$ and $\|\cdot\|_1$. The norm $\|\cdot\|_0$ is topologically equivalent to $\|\cdot\|_{\mathscr{Y}_e}$. Using the inner product $\langle\cdot,\cdot\rangle_0$ in $\mathscr{Y}_e$, Theorem 9.13 indicates that the subspace decomposition is guaranteed if, for some $\varepsilon_0^2 > 0$,

$$\Psi_1 U + U^*\Psi_1 \geq \varepsilon_0^2 \Psi_1. \tag{11.57}$$

Robustness is then defined as the requirement that the error sequence has the property that, whenever $e_0^e \in \overline{\mathscr{R}[UG_e]}$, the error norm $\|e_k^e\|_1$ reduces from iteration to iteration despite the presence of the modelling error U. That is,

$$\textit{(Robust Monotonicity)} \qquad \|e_{k+1}\|_1 < \|e_k\|_1, \quad \textit{for all } k \geq 0. \tag{11.58}$$

Supposing that $e_0^e \in \overline{\mathscr{R}[UG_e]}$, the condition $\|e_1^e\|_1^2 < \|e_0^e\|_1^2$ reduces, after a little algebraic manipulation and using the identity $\Psi_1(I - L_1) = (I - L_1)^*\Psi_1$, to the inequality, with respect to the inner product $\langle\cdot,\cdot\rangle_{\mathscr{Y}_e}$,

$$\Psi_1 U + U^*\Psi_1 > U^*\Psi_1(I - L_1)U \quad \textit{on the subspace} \quad (I - L_1)\overline{\mathscr{R}[UG_e]}. \tag{11.59}$$

This condition can be simplified by noting that $(I - L_1)\overline{\mathscr{R}[UG_e]} \subset \mathscr{R}[G_e]$ and replacing it with the condition given in the following Theorem,

Theorem 11.8 (Robustness of the Auxiliary Optimization Switching Algorithm) *The subspace decomposition $\mathscr{Y}_e = ker[G_e^*] \oplus \overline{\mathscr{R}[UG_e]}$ holds true and, in the presence of the left multiplicative modelling error U, the robust monotonicity condition $\|e_{k+1}^e\|_1 < \|e_k^e\|_1$ for all $k \geq 0$ and all $e_0^e \in \overline{\mathscr{R}[UG_e]}$ is guaranteed if*

$$\Psi_1 U + U^*\Psi_1 > U^*\Psi_1(I - L_1)U + \varepsilon_0^2\Psi_1 \quad \textit{on} \quad \mathscr{R}[G_e] \quad \textit{for some} \quad \varepsilon_0^2 > 0. \tag{11.60}$$

Proof The proof of monotonicity follows from the previous discussion. The addition of the final term $\varepsilon_0^2\Psi_1 > 0$ ensures that the subspace decomposition holds as $U^*\Psi_1(I - L_1)U \geq 0$. □

As Ψ_1 occurs in all terms, ε^2 plays quite a complex role in assessing robustness. In contrast ε_1 occurs only in $\Psi_1(I - L_1) = \varepsilon^{-2}\Psi_1 G_e(I + \varepsilon_1^{-2}G_1^*G_1)^{-1}G_e^*\Psi_1$ which suggests that robustness increases as ε_1^2 gets smaller. A more conservative, but simpler, robustness condition is obtained by writing

$$\Psi_1(I - L_1) \leq \Psi_1 \varepsilon^{-2} G_e G_e^* \Psi_1 \leq \Psi_1 \frac{\varepsilon^{-2}\|G_e^*\|^2}{(1+\varepsilon^{-2}\|G_e^*\|^2)} \tag{11.61}$$

Using this bound, a sufficient condition for monotonic robustness is hence that

$$\begin{array}{ccc} \Psi_1 U + U^* \Psi_1 & > & \beta U^* \Psi_1 U + \varepsilon_0^2 \Psi_1 \quad on \quad \mathscr{R}[G_e] \\ with & & \beta = \frac{\varepsilon^{-2}\|G_e^*\|^2}{(1+\varepsilon^{-2}\|G_e^*\|^2)}. \end{array} \tag{11.62}$$

Regarding U as an operator on $\overline{\mathscr{R}[G_e]}$ into $\mathscr{Y}_e$, this is just $\|I - \beta U\|_0 \leq 1 - \beta\varepsilon_0^2$ or, more precisely,

Theorem 11.9 (A Norm-based Robustness Condition) *Using the assumptions of Theorem 11.8, robust monotonicity is guaranteed in the presence of the left multiplicative modelling error U if*

$$\|I - \beta U\|_0 < 1 \quad on \quad \overline{\mathscr{R}[G_e]} \qquad (a\ Robust\ Monotonicity\ Condition), \tag{11.63}$$

the induced operator norm being computed in the topology defined by the inner product $\langle \cdot, \cdot \rangle_0$ on $\overline{\mathscr{R}[G_e]}$. A sufficient condition for this to be true is that

$$\|I - \beta U\|_0 < 1 \quad on \quad \mathscr{Y}_e \qquad (another\ Robust\ Monotonicity\ Condition), \tag{11.64}$$

the induced operator norm being computed in the topology defined by the inner product $\langle \cdot, \cdot \rangle_0$ on $\mathscr{Y}_e$. These two conditions coincide if $\mathscr{R}[G_e]$ is dense in $\mathscr{Y}_e$.

In conclusion, the switching algorithm has a degree of robustness as expressed by the inequality of equation (11.63) which states, in simple terms, that U should not differ too much from the identity. This expression is expressed in the alternative topology and has a formal similarity to the robustness conditions for the Inverse Model Algorithm of Chap. 6 and for the Inverse Auxiliary Optimization Algorithm of Sect. 11.2. The norm of $I - \beta U$ is however computed in the topology induced by $\langle \cdot, \cdot \rangle_0$ for the restriction of $I - \beta U$ to $\overline{\mathscr{R}[G_e]}$, the computation of which is a non-trivial task even when $\overline{\mathscr{R}[G_e]} = \mathscr{Y}_e$. Note that its value depends, via Ψ_1, not only on the plant model G_e but also on the value of the weight parameter ε^2. This is most easily expressed in the case when $\mathscr{Y}_e = \overline{\mathscr{R}[G_e]}$ by noting that,

$$\begin{array}{l} \sup_{y^e \neq 0} \left(\frac{\langle (I-\beta U)y^e, \Psi_1 (I-\beta U) y^e \rangle_{\mathscr{Y}_e}}{\langle y^e, \Psi_1 y^e \rangle_{\mathscr{Y}_e}} \right) \\ \quad \leq (1 + \varepsilon^{-2}\|G_e^*\|^2) \sup_{y^e \neq 0} \left(\frac{\langle (I-\beta U)y^e, (I-\beta U) y^e \rangle_{\mathscr{Y}_e}}{\langle y^e, y^e \rangle_{\mathscr{Y}_e}} \right). \end{array} \tag{11.65}$$

Hence, expressed in terms of the original topology induced by $\langle \cdot, \cdot \rangle_{\mathscr{Y}_e}$, a sufficient condition for Eq. (11.63) to hold is that

$$\|I - \beta U\| < \left(\frac{1}{(1 + \varepsilon^{-2}\|G_e^*\|^2)} \right)^{0.5}. \tag{11.66}$$

This expression is likely to be highly conservative but it does suggest that robustness reduces substantially as ε becomes small as robustness depends on the proximity of U to the identity I. For example, if $G_e = G$, $\mathscr{Y}_e = \mathscr{Y}$ (with inherited inner products) and the plant is a linear, discrete time, multi-input, single output state space model $S(A, B, C, D)$, then U can be characterized by a discrete transfer function $U(z)$. In this case, the condition becomes, assuming that $U(z)$ is stable,

$$|\frac{1}{\beta} - U(z)| < \left(\frac{1}{\beta}\right)\left(\frac{1}{(1+\varepsilon^{-2}\|G_e^*\|^2)}\right)^{0.5} \quad for \quad |z| = 1, \tag{11.67}$$

where $\|G_e^*\| = \|G_e\|$ can be replaced by the H_∞ norm of the transfer function matrix of G, namely

$$\|G(z)\|_\infty = \sup_{|z|=1} |G(z)R^{-1}G^T(z^{-1})Q|. \tag{11.68}$$

where Q and R are the weights in the inner products in $\mathscr{Y}$ and $\mathscr{U}$ respectively. The graphical interpretation of this condition is that $U(z)$ lies, when $|z| = 1$, in the interior of a circle of centre β^{-1} and radius $\beta^{-1}\left(\frac{1}{(1+\varepsilon^{-2}\|G_e^*\|^2)}\right)^{0.5} < \beta^{-1}$. The reader should verify that this circle contains the point $(1, 0)$ representing the zero error case $U = I$.

11.5 The Switching Algorithm When $G_eG_e^*$ Is Invertible

The advantages of Switching Algorithm 11.2 include the facts that $\mathscr{Y}_e$ can be finite or infinite dimensional and few assumptions are made about properties of G_e. The algorithm requires two applications of NOILC algorithms on each iteration. This short section explores the idea that this requirement can be removed if $G_eG_e^*$ has a computable bounded inverse. This assumption includes some infinite dimensional cases and all finite dimensional cases and immediately yields the consequence that $ker[G_e^*] = \{0\}$ and $\mathscr{Y}_e = \overline{\mathscr{R}[G_e]}$. The computation of the inverse is simpler and more reliable when $\mathscr{Y}_e$ is finite dimensional and of "small" dimension. Such cases include, for example, the Intermediate Point NOILC Algorithm 10.4 for state space systems and its discrete time equivalent. The following algorithm definition and discussion follows a similar structure to that of previous sections. Indeed, the reader will note that the analysis is almost identical in many places, differing mainly in some of the operator and parameter bounds.

Algorithm 11.3 (A Simplified Switching Algorithm when $G_eG_e^*$ is Invertible) With the notation of previous sections, suppose that $G_eG_e^*$ has a bounded inverse and that it has been computed. Suppose also that the initial input $u_0 \in \mathscr{U}$ is given and that the associated z_0 and error $e_0^e = r^e - y_0^e$ has been computed/measured. Each iteration with index $k + 1$ then consists of TWO steps:

STEP ONE: Given the input u_k, compute (off-line) the input $\tilde{u}_{k+1}$ as as an input that tracks r^e precisely and solves the problem

$$\tilde{u}_{k+1} = \arg\min_{u\in\mathscr{U}}\{\ \|u-u_k\|^2_{\mathscr{U}}\ :\quad \textit{subject}\quad \textit{to}\quad r^e = G_e u + d^e\ \}. \tag{11.69}$$

That is, use the data (u_k, e^e_k) to compute the "intermediate" input signal

$$\tilde{u}_{k+1} = u_k + G^*_e(G_eG^*_e)^{-1}e^e_k \quad \textit{where}\quad e^e_k = r^e - y^e_k. \tag{11.70}$$

STEP TWO: Compute the signal z_k associated with u_k and use (off-line or on-line) NOILC techniques to compute the control signal u_{k+1} as the control signal that minimizes

$$J_{\mathscr{Z}}(u, z_k, \tilde{u}_{k+1}) = \|z_k - z\|^2_{\mathscr{Z}} + \|u-\tilde{u}_{k+1}\|^2_{\mathscr{U}} \tag{11.71}$$

subject to the auxiliary dynamical constraint $z = G_1u + d_1$. That is,

$$u_{k+1} = \tilde{u}_{k+1} + G^*_1\,(z_k - z_{k+1})\,. \tag{11.72}$$

This algorithm has a very similar interpretation to that of Algorithm 11.2. It is linked to that algorithm in a formal mathematical sense by the replacement of the factor $\varepsilon^{-2}\Psi_1 = \varepsilon^{-2}(I+\varepsilon^{-2}G_eG^*_e)^{-1}$ in the expression for $\tilde{u}_{k+1}$ by its limit $(G_eG^*_e)^{-1}$ as $\varepsilon\to 0$. Alternatively, it can be seen as the construction of $\tilde{u}_{k+1}$ by setting $\varepsilon = 1$ and defining $\Psi_1 = (G_eG^*_e)^{-1}$. Using this simple substitution, the algorithm evolution is described by the familiar formula

$$\begin{array}{l} u_{k+1} = u_k + (I+G^*_1G_1)^{-1}G^*_e\Psi_1 e^e_k,\quad \textit{and}\quad e^e_{k+1} = L_1e^e_k,\quad k\geq 0,\\ \textit{with}\quad L_1 = I - G_e(I+G^*_1G_1)^{-1}G^*_e\Psi_1\ \textit{and}\quad \Psi_1 = (G_eG^*_e)^{-1}. \end{array} \tag{11.73}$$

The operator L_1 is self-adjoint in the alternative topology defined by the inner product $\langle y^e, w^e\rangle_0 = \langle y^e, (G_eG^*_e)^{-1}w^e\rangle_{\mathscr{Y}_e}$ and, with respect to this inner product,

$$0 \leq L_1 \leq \frac{\|G_1\|^2}{1+\|G_1\|^2}I < I, \tag{11.74}$$

as $\Psi_1\,(L_1 - I) = -\Psi_1G_e(I+G^*_1G_1)^{-1}G^*_e\Psi_1$ gives

$$\Psi_1\,(L_1-I) \leq -\frac{1}{1+\|G_1\|^2}\Psi_1G_eG^*_e\Psi_1 = -\frac{1}{1+\|G_1\|^2}\Psi_1. \tag{11.75}$$

As a consequence, its induced norm $\|L_1\|_0 < 1$ and its spectral radius $r(L_1) < 1$. The convergence of the algorithm is then guaranteed with the monotonicity condition

$$\|e^e_{k+1}\|_0 < \|e^e_k\|_0,\quad \textit{for}\quad \textit{all}\quad k\geq 0,\quad \textit{and}\quad \lim_{k\to\infty} e^e_k = \lim_{k\to\infty} L^k_1e^e_0 = 0. \tag{11.76}$$

The input sequence satisfies the condition

$$\begin{aligned} u_{k+1} - u_0 &= (I + G_1^* G_1)^{-1} G_e^* \Psi_1 \sum_{j=0}^{k} e_j^e = (I + G_1^* G_1)^{-1} G_e^* \Psi_1 \sum_{j=0}^{k} L_1^j e_0^e \\ &= (I + G_1^* G_1)^{-1} G_e^* \Psi_1 (I - L_1)^{-1} \left(I - L_1^{k+1} \right) e_0^e \\ &= (I + G_1^* G_1)^{-1} G_e^* (G_e (I + G_1^* G_1)^{-1} G_e^*)^{-1} \left(I - L_1^{k+1} \right) e_0^e \end{aligned} \tag{11.77}$$

where the inverse exists and is bounded as $G_e(I + G_1^* G_1)^{-1} G_e^* \geq (1 + \|G_1\|^2)^{-1} G_e G_e^*$. As a consequence, the following limits exist

$$\begin{aligned} &\lim_{k \to \infty} u_k = u_\infty = u_0 + (I + G_1^* G_1)^{-1} G_e^* (G_e (I + G_1^* G_1)^{-1} G_e^*)^{-1} e_0^e \\ &\lim_{k \to \infty} z_k = z_\infty = G_1 u_\infty + d_1 \end{aligned} \tag{11.78}$$

Application of Theorem 11.1 can now be used to prove that Algorithm 11.3 hence has the desired properties of convergence to a solution of the auxiliary optimization problem. The computations are significantly simplified in STEP ONE and, in terms of analysis, the techniques of Sect. 11.3 can be used by the simple substitution of $\varepsilon = 1$ and replacing Ψ_1 by $(G_e G_e^*)^{-1}$. However, the error norm sequence is not necessarily monotonic in the norm $\|\cdot\|_{\mathscr{Y}_e}$ but is monotonic in the norm $\|\cdot\|_0$.

A consideration of the robustness of the algorithm (in the presence of a left multiplicative modelling error U) has the same general form as that in Sect. 11.3 but there are some changes. The general condition for monotonic robustness in the topology defined by the inner product $\langle y^e, w^e \rangle_1 = \langle y^e, \Psi_1 (I - L_1) w^e \rangle_{\mathscr{Y}_e}$ is that, for some $\varepsilon_0^2 > 0$,

$$\Psi_1 U + U^* \Psi > U^* \Psi_1 (I - L_1) U + \varepsilon_0^2 I \quad on \quad \mathscr{Y}_e \tag{11.79}$$

but the change in the form of Ψ_1 and the inequality $\Psi_1 (I - L_1) \leq \Psi_1$ obtained by removing G_1 from the expression implies that the condition reduces to

$$\|I - U\|_0 < 1 \quad and \quad hence \quad r\left(\Psi_1^{-1} (I - U^*) \Psi_1 (I - U) \right) < 1 \tag{11.80}$$

which are similar formulae to those obtained for the Switching Algorithm 11.2. Two changes of note are, in particular, that

1. the parameter $\beta = 1$ in this case and,
2. in terms of the norms in the original topology in $\mathscr{Y}_e$, the robustness condition is satisfied if the induced operator norm $\|I - U\|$ is strictly less than the square root of the ratio of the minimum and maximum eigenvalues of $G_e G_e^*$.

11.6 Discussion and Further Reading

The idea of adding auxiliary optimization objectives was introduced by the author and his co-workers [41] and an extensive theoretical analysis is presented in [90, 91] with the results verified by laboratory experimentation. In these references, it was assumed that $\mathscr{Y}_e$ is finite dimensional. The presentation in this chapter has removed this assumption and hence extended the range of applications considerably. The robustness analysis provided for switching algorithms is also new to the literature.

For presentational purposes, some topics have been omitted and the focus has concentrated on the demonstration of the relevant properties of what is seen as the case with the greatest relevance to practice. The additional properties discussed in the references include

1. The possibility [90] of enhancing STEP ONE of Switching Algorithm 11.2 to include $k_0 \geq 1$ "inner iterations" of NOILC Algorithm 9.1 from the starting data u_k. The benefits of this increase in algorithm complexity have not yet been demonstrated. Note, however, that as $k_0 \to \infty$ Theorem 9.3 and its consequences indicate that this step computes an input that tracks r^e exactly and minimizes $\|u - u_k\|^2_{\mathscr{U}}$ which is exactly STEP ONE of Algorithm 11.3. It follows that the algorithm of Sect. 11.5 can be regarded as the special case of this more general algorithm with the choice of $k_0 = \infty$.
2. The computation [91] of an appropriate right inverse for the Inverse Algorithm 11.1 when the auxiliary system G_1 is generated by a linear, discrete time, state space model $S(A_1, B_1, C_1)$.
3. For the Inverse Model Algorithm 11.1, two additional design concerns are addressed in [91], namely

   a. The effects of zero-mean, additive, output noise. Noise prevents convergence to zero in practice but, in a statistical sense, tends to drive initial errors systematically to smaller values. The asymptotic magnitudes of the norms is related to the covariance matrix of the noise.
   b. Similar techniques can be used to produce a statistical analysis of the effects of random variations in the initial condition $x_k(0)$ on each iteration. Again, this implementation issue does not prevent useful convergence to errors with magnitude related to the co-variance matrix describing these variations.

A common feature of the approaches to analysis is the use of modified inner product topologies in $\mathscr{Y}_e$ to study monotonic convergence of errors.

1. The Hilbert space structure is retained but the inner product $\langle \cdot, \cdot \rangle_{\mathscr{Y}_e}$ is replaced by a topologically equivalent description using the inner product $\langle y^e, w^e \rangle_0 = \langle y^e, \Psi_1 w^e \rangle_{\mathscr{Y}_e}$ and the associated induced norm. The form of Ψ_1 depends on the algorithm considered but its effect is to ensure that L_1 is self adjoint, positive and $< I$ in the new topology. This proves monotonicity of $\{\|e_k\|_0\}_{k \geq 0}$ but does not, in general, imply that the norm sequence $\{\|e_k\|_{\mathscr{Y}_e}\}_{k \geq 0}$ is also monotonic. Boundedness is however proved as $e^e_k \to 0$ as $k \to \infty$.

2. For robustness studies, yet another modified topology has been introduced using the inner product $\langle y^e, w^e\rangle_1 = \langle y^e, \Psi_1(I-L_1)w^e\rangle_{\mathscr{Y}_e}$ on the subspace $\overline{\mathscr{R}[G_e]}$ of $\mathscr{Y}_e$. The conditions for robust monotonicity are derived in this topology. The approach makes it possible to prove robust monotonicity of the norm sequence $\{\|e_k\|_1\}_{k\geq 0}$ but, again, this does not, in general, imply that the norm sequence $\{\|e_k\|_{\mathscr{Y}_e}\}_{k\geq 0}$ is also monotonic.

The reader should note that the ideas of incorporating auxiliary optimization objectives into NOILC relies only on NOILC concepts and hence applies quite generally to, for example, Intermediate Point, Multi-rate, Predictive and Multi-task Algorithms with or without initial condition variation. The realizations of the algorithm in each area of application depends solely on the form of G_e and its adjoint operator G_e^*. The choice of inner products in $\mathscr{U}$ and $\mathscr{Y}_e$ does not affect the reality of convergence but does influence the nature of the convergence including the rate of convergence. For state space systems, this issue is represented by the choice of Q, R and $Q_{\mathscr{Z}}$. The chapter has indicated the trends as their "values" increase or decrease and has linked the effects to the consequent operator norms $\|G_e^*\|^2$. This still needs decisions to be made at the algorithm design stage and may be problematic, particularly in the case of MIMO systems when these weights are matrix valued.

Although possessing considerable generality, the material in the chapter has not addressed fully many questions of interest including

1. The issue of robustness in the presence of right multiplicative modelling errors.
2. The further practical development of the links to the Intermediate Point NOILC Problem (see Sects. 11.3.5 and 10.5 and the following material) obtained by taking $G_1 = G$ and choosing u_0 and y_0 to suit the objectives of the application.
3. The real benefits of choosing $G_1 = G_e$ in practice. This intriguing possibility provides a mechanism for shaping of the eigenvalues of the algorithm operator L_1 and hence influencing the nature and form of the convergence using NOILC computations alone.
4. The choice of auxiliary variable and $J_{\mathscr{Z}}$ may arise from physical considerations but using spectral factorization methods to write

$$G_1^T(z^{-1})Q_{\mathscr{Z}}G_1(z) + \varepsilon_1^2 R = F^T(z^{-1})RF(z), \tag{11.81}$$

   where $F(z)$ is $\ell \times \ell$, asymptotically stable and minimum-phase, links the criterion $J_{\mathscr{Z}}$ to the squared norm of the filtered input $F(u-u_0)$ in $\mathscr{U}$. Conversely, choosing $F(z)$ and using it to construct G_1 provides a useful way of influencing frequency content in the converged input.
5. The use of the feedback form of algorithm implementation requires the measurement of the state vector $x(t)$ and/or $z(t)$. In practice, this may be done using a state observer based on a model. Issues of estimation errors and robustness naturally arise in such realizations and need further research.

This chapter concludes the discussion of the optimization paradigm for linear systems based on norm-optimal computations. In many ways, it represents a robust, benchmark solution to the problem of Iterative Learning Control as it has considerable generality and, by careful problem specification, can address many issues that relate to tasks seen in many applications.

Chapter 12
Iteration as Successive Projection

The material of the preceding chapters has provided a benchmark set of algorithms for Iterative Control and supported those algorithms using convergence analysis and robustness criteria together with insights into the choice and influence of parameters available to the design engineer. The use of an optimization paradigm has permitted a quite general analysis based on an operator description of systems dynamics and uses the geometry induced in Hilbert space by inner product and norm descriptions. As presented, it has a degree of completeness but it does leave open questions that need further research. Some of these include issues of successful Iterative Control in the presence of input and output constraints, situations that can be approached using the material and algorithms in this chapter.

12.1 Convergence Versus Proximity

This chapter modifies the NOILC methodology using a change in the interpretation of the objectives of Iterative Control. To illustrate the ideas, NOILC Algorithm 9.1 is considered initially. More precisely, the operator descriptions and Hilbert space structures of the NOILC algorithm are retained and the objective of Iterative Control is seen as finding an input u associated with the intersection of TWO sets in the product space $H = \mathscr{Y} \times \mathscr{U}$, namely

$$\begin{aligned} \textit{System Dynamic Relationships} - \quad & S_D = \{\, (e,u) \;:\; e = r - y\,,\; y = Gu + d \,\} \\ \textit{Tracking Objective Solutions} - \quad & S_T = \{\, (e,u) \;:\; e = 0 \,\} \end{aligned} \tag{12.1}$$

Successful Iterative Algorithms can be seen as algorithms with both off-line and experimental components that generate sequences $\{(e_k, u_k)\}_{k\geq 0} \in \mathscr{Y} \times \mathscr{U}$ that, ideally, converge to a point in the intersection $S_D \cap S_T$.

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_12

To use this idea in algorithm development, it is necessary to assume that $S_D \cap S_T$ is non-empty and hence that there exists at least one point $(0, u)$ that solves the tracking control problem for the system considered. The topology in $\mathscr{Y} \times \mathscr{U}$ is defined by the inner product and norm(squared)

$$\begin{array}{rcl} \langle (e,u),(w,v)\rangle_{\mathscr{Y}\times\mathscr{U}} & = & \langle e,w\rangle_{\mathscr{Y}} + \varepsilon^2\langle u,v\rangle_{\mathscr{U}} \\ \|(e,u)\|^2_{\mathscr{Y}\times\mathscr{U}} & = & \|e\|^2_{\mathscr{Y}} + \varepsilon^2\|u\|^2_{\mathscr{U}} \end{array} \tag{12.2}$$

where the weight parameter $\varepsilon^2 > 0$ is included to link the presentation to that of NOILC in Chap. 9. Note that convergence of an algorithm can be associated with the distance between iterates (e_k, u_k) and the sets S_D and/or S_T. More precisely,

1. *any sequence* $\{e_k, u_k)\}_{k\geq 0}$ *in* S_D *that satisfies the system dynamic equations and also has the property that it is approaches* S_T *as* $k \to \infty$, *that is*

$$\lim_{k\to\infty} \inf_{(0,u)\in S_T} \|(0,u) - (e_k,u_k)\|_{\mathscr{Y}\times\mathscr{U}} = 0, \tag{12.3}$$

will satisfy the tracking objective $e = 0$ *arbitrarily closely and* $\lim_{k\to\infty} e_k = 0$.

Proof Choosing a sequence $\{(0,\tilde{u}_k)\}_{k\geq 0}$ in S_T that is sufficiently close to (e_k, u_k) to ensure that that $\lim_{k\to\infty} \|(0,\tilde{u}_k) - (e_k,u_k)\|_{\mathscr{Y}\times\mathscr{U}} = 0$ leads to the consequence that $\|(e_k,u_k) - (0,\tilde{u}_k)\|^2 = \|e_k\|^2 + \|u_k - \tilde{u}_k\|^2 \geq \|e_k\|^2$ becomes arbitrarily small. □

2. *Similarly, consider any sequence* $\{0, u_k)\}_{k\geq 0}$ *in* S_T *that has the property that it is approaches* S_D *as* $k \to \infty$, *that is*

$$\lim_{k\to\infty} \inf_{(e,u)\in S_D} \|(0,u_k) - (e,u)\|_{\mathscr{Y}\times\mathscr{U}} = 0. \tag{12.4}$$

Then any convergent subsequence has a limit input u_∞ *satisfying the tracking objective* $e = 0$ *exactly*.

Convergence is therefore described in terms of the "proximity" of iterates (e_k, u_k) to the sets S_D and S_T. Iterative Control cannot, in practice, realize the infinite number of iterations needed for exact tracking so the best that can be expected is arbitrarily accurate tracking followed by algorithm termination when the desired accuracy is achieved. In this sense, ILC algorithms can be seen as being *Proximity Algorithms*, a concept that is illustrated by the following definition in the context of NOILC.

Definition 12.1 (*Proximity Algorithms in Iterative Control*) Suppose that $S_D \cap S_T$ is non-empty. Suppose that the solution of an Iterative Control problem is defined by any point in $S_D \cap S_T$ in the product Hilbert space $H = \mathscr{Y} \times \mathscr{U}$. Then an iterative algorithm generating a sequence of pairs $\{(e_k, u_k)\}_{k\geq 0}$ is said to be a *proximity algorithm* if it has the property that

$$\lim_{k\to\infty} \inf_{(e,u)\in S_D} \|(e,u)-(0,u_k)\|_{\mathscr{Y}\times\mathscr{U}} = \lim_{k\to\infty} \inf_{(0,u)\in S_T} \|(0,u)-(e_k,u_k)\|_{\mathscr{Y}\times\mathscr{U}} = 0. \tag{12.5}$$

Theorem 12.1 (NOILC is a Convergent Proximity Algorithm) *The Norm Optimal Iterative Learning Control Algorithm 9.1 is a Proximity Algorithm if* $e_0 \in \mathscr{R}[G]$. *In particular, the algorithm can be described by the two steps, for all* $k \geq 0$,

$$\begin{array}{rrcl} \textit{STEP ONE}: & (0,u_k) & = & \arg\min_{(0,u)\in S_T} \|(0,u)-(e_k,u_k)\|_{\mathscr{Y}\times\mathscr{U}} \\ \textit{STEP TWO}: & (e_{k+1},u_{k+1}) & = & \arg\min_{(e,u)\in S_D} \|(e,u)-(0,u_k)\|_{\mathscr{Y}\times\mathscr{U}} \end{array} \tag{12.6}$$

Proof The assumption that $e_0 \in \mathscr{R}[G]$ implies that $r - d \in \mathscr{R}[G]$. It therefore ensures (Theorems 9.2 and 9.3) that the error converges to zero, the input signal sequence converges and $S_D \cap S_T$ is non-empty. The remainder of the proof is left as an exercise for the reader who will note that STEP ONE follows from the definitions of norms and needs no computation. STEP TWO is the NOILC optimization problem defining input updates as $\|(e,u)-(0,u_k)\|^2_{\mathscr{Y}\times\mathscr{U}} = J(u,u_k)$. □

These observations naturally lead to a questioning of the focus on convergence in algorithm construction and replaces it with the objective of ensuring *proximity* properties of the sequence $\{(e_k,u_k)\}_{k\geq 0}$. In what follows, these ideas are generalized to create classes of algorithms that have similar proximity properties. The approach generates new algorithms and make it possible to include input and/or output/error signal constraints in the iteration process. It also suggests mechanisms for introducing acceleration of convergence (Chap. 13).

12.2 Successive Projection and Proximity Algorithms

This section uses the two step structure of Theorem 12.1 to motivate a *Successive Projection* approach relevant to Iterative Learning Control. More precisely, Iterative Algorithms in a real Hilbert space H are regarded as algorithms that aim to find any point $x \in \bigcap_{j=1}^{N_S} S_j$ where each set $S_j \subset H$, $1 \leq j \leq N_S$ is both closed and convex. The number of sets N_S is assumed to be finite. In terms of the discussion and notation of Sect. 12.1, $N_S = 2$ and $S_1 = S_D$ with $S_2 = S_T$.

Algorithm 12.1 (*Successive Projection in Hilbert Space*) Suppose that the sets $\{S_j\}_{1\leq j\leq N_S}$ are arranged to form an infinite sequence $\{\tilde{S}_p\}_{p\geq 1}$ within which each and every set S_j occurs an infinite number of times. Then a successive projection algorithm for the construction of a point $x \in \bigcap_{j=1}^{N_S}$ is defined by choosing a starting point $s_0 \in H$ and constructing the sequence $\{s_j\}_{j\geq 0}$ with $s_j \in \tilde{S}_j$, $j \geq 1$ by solving, recursively, the infinite number of (minimum distance or minimum norm) optimization problems

$$s_{j+1} = \arg \min_{s \in \tilde{S}_{j+1}} \|s - s_j\|_H \,, \qquad for \quad j \geq 0. \tag{12.7}$$

For example, for NOILC Theorem 12.1, given a starting condition $s_0 = (0, u_0)$, the sequence $\{\tilde{S}_p\}_{p \geq 1}$ is simply the alternating sequence $S_D, S_T, S_D, S_T, S_D, S_T \ldots$ which generates the iterates $s_1 = (e_1, u_1)$, $s_2 = (0, u_1)$, $s_3 = (e_2, u_2)$, $s_4 = (0, u_2)$, ... or, more generally

$$\begin{array}{c} s_{2k+1} = (e_{k+1}, u_{k+1}) \quad and \quad s_{2k} = (0, u_k) \,, \quad k \geq 0 \\ (Successive \quad Projection \quad and \quad NOILC \quad Iterates). \end{array} \tag{12.8}$$

The properties of Algorithm 12.1 are derived as follows,

Theorem 12.2 (Asymptotic Properties of Algorithm 12.1) *Using the notation given above, suppose that* $\bigcap_{j=1}^{N_S} S_j$ *is non-empty. Then, for any point* $x \in \bigcap_{j=1}^{N_S} S_j$ *, Algorithm 12.1 has the properties that*

$$\begin{array}{rcl} \|x - s_{j-1}\|_H^2 & \geq & \|x - s_j\|_H^2 + \|s_j - s_{j-1}\|_H^2 \geq \|x - s_j\|_H^2 \,, \quad for \quad all \quad j \geq 1, \\ and \quad \|x - s_0\|_H^2 & \geq & \sum_{p=1}^{\infty} \|s_p - s_{p-1}\|_H^2. \end{array} \tag{12.9}$$

As a consequence,

$$\lim_{j \to \infty} \|s_{j+1} - s_j\|_H^2 = 0. \tag{12.10}$$

Proof The proof uses Theorem 2.16 which implies that, for all $j \geq 1$, $\langle s - s_j, s_j - s_{j-1} \rangle_H \geq 0$ for all $s \in \tilde{S}_j$. Choosing $s = x$ and writing $x - s_{j-1} = (x - s_j) + (s_j - s_{j-1})$,

$$\begin{array}{rcl} \|x - s_{j-1}\|_H^2 & = & \|x - s_j\|_H^2 + \|s_j - s_{j-1}\|_H^2 + 2\langle x - s_j, s_j - s_{j-1} \rangle_H \\ & \geq & \|x - s_j\|_H^2 + \|s_j - s_{j-1}\|_H^2 \;\geq\; \|x - s_j\|_H^2. \end{array} \tag{12.11}$$

Applying induction then gives

$$\|x - s_0\|_H^2 \;\geq\; \|x - s_j\|_H^2 + \sum_{p=1}^{j} \|s_p - s_{p-1}\|_H^2 \quad for \quad all \quad j \geq 1. \tag{12.12}$$

Letting $j \to \infty$, noting that the right hand side is positive and monotonic in "j", proves convergence of the series and the fact that $\|s_j - s_{j-1}\|_H$ becomes arbitrarily small. The result is now proved. □

The result can be interpreted as stating that the sequence $\{s_j\}_{j \geq 0}$ gets closer to each and every point of $x \in \bigcap_{j=1}^{N_S} S_j$ as the iterations evolve. In addition, the iteration difference $s_j - s_{j-1}$ ultimately becomes infinitesimally small. In effect, the iterates become better and better approximations to a solution to the problem but, ultimately, convergence rates slow.

In the special case of $N_S = 2$ and choosing the sequence $\{\tilde{S}_j\}_{j\geq 1}$ to be of the alternating form of either $S_1, S_2, S_1, S_2, \ldots$ or $S_2, S_1, S_2, S_1, \ldots$ immediately gives

$$\lim_{j\to\infty} \min_{s\in S_1} \|s - s_j\|_H = \lim_{j\to\infty} \min_{s\in S_2} \|s - s_j\|_H = 0 \tag{12.13}$$

and hence the algorithm is a proximity algorithm in a similar sense to that stated in Definition 12.1. That is, the iterates become arbitrarily close to both sets S_1 and S_2. The case of $N_S = 2$ has an additional property, namely,

Theorem 12.3 (Successive Projection: monotonicity properties when $N_S = 2$) *With the assumptions of Theorem 12.2, suppose that $N_S = 2$ and that $\{\tilde{S}_j\}_{j\geq 1}$ is the alternating sequence $S_1, S_2, S_1, S_2, \ldots$. Then, choosing $s_0 \in H$, the resultant iterates $\{s_j\}_{j\geq 1}$ satisfy the additional monotonicity property,*

$$\|s_{j+2} - s_{j+1}\|_H \leq \|s_{j+1} - s_j\|_H \,, \quad \textit{for all} \quad j \geq 1. \tag{12.14}$$

Proof Suppose that $s_{j+1} \in S_1$ then s_{j+2} is the nearest point in S_2. Clearly $s_j \in S_2$ is more distant which is precisely the statement to be proved. As the same argument applies when $s_{j+1} \in S_2$, the result is proved. □

An immediate application of this result is a derivation of the properties of NOILC as a successive projection algorithm.

Theorem 12.4 (NOILC Properties Revisited) *Using the notation of Theorem 12.1 to describe NOILC Algorithm 9.1 and setting $S_D = S_1$, $S_T = S_2$ with a starting point $s_0 = (0, u_0) \in S_2 \subset H = \mathscr{Y} \times \mathscr{U}$, suppose that a solution to the tracking problem exists and hence that $S_1 \cap S_2$ is non-empty. Under these conditions, for all $k \geq 0$,*

$$\|e_{k+1}\|^2_{\mathscr{Y}} \leq J(u_{k+1}, u_k) = \|e_{k+1}\|^2_{\mathscr{Y}} + \varepsilon^2 \|u_{k+1} - u_k\|^2_{\mathscr{U}} \leq \|e_k\|^2_{\mathscr{Y}} \tag{12.15}$$

In addition, the infinite summation $\sum_{k=0}^{\infty} \left(J(u_{k+1}, u_k) + \|e_{k+1}\|^2_{\mathscr{Y}}\right)$ converges so that, both

$$\sum_{k=0}^{\infty} \|e_k\|^2_{\mathscr{Y}} < \infty \quad \textit{and} \quad \sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2_{\mathscr{U}} < \infty. \tag{12.16}$$

Proof The form of each s_j follows from the NOILC Algorithm 9.1 as interpreted in Theorem 12.1. The inequalities follow from the monotonicity properties in Theorem 12.3 as $s_{2k} - s_{2k-1} = (-e_k, 0)$, $s_{2k+1} - s_{2k} = (e_{k+1}, u_{k+1} - u_k)$ and $s_{2k+2} - s_{2k+1} = (-e_{k+1}, 0)$, noting that $\|s_{2k+1} - s_{2k}\|^2_H = J(u_{k+1}, u_k)$. Finally, Theorem 12.2 gives

$$\sum_{j=0}^{\infty} \|s_{j+1} - s_j\|^2_H = \sum_{k=0}^{\infty} \left(J(u_{k+1}, u_k) + \|e_{k+1}\|^2_{\mathscr{Y}}\right) < \infty \tag{12.17}$$

The result now follows from the definitions. □

Note: The result re-affirms the properties of NOILC Algorithm 9.1 with a geometric proof of properties of the error sequence, namely that the summation $\sum_{k=0}^{\infty} \|e_k\|_{\mathscr{Y}}^2$ is finite and hence that $e_k \to 0$ (in norm) as $k \to \infty$.

The link between proximity, successive projection and NOILC iteration is established above for the case of $N_S = 2$. More generally, proximity properties are guaranteed in, for example, the following case, which builds on the assumptions and outcomes of Theorem 12.2,

Theorem 12.5 (Multi-set Proximity Properties and $\{\tilde{S}_j\}_{j\geq 1}$) *Let $N_D \geq N_S$ be given. Suppose also that, for each index "p" in the sequence $\{\tilde{S}_j\}_{j\geq 1}$, the subset defined by values $p - N_D \leq j \leq p + N_D$ contains every set in $\{S_j\}_{1\leq j\leq N_S}$. Then Algorithm 12.1 is a proximity algorithm in the sense that*

$$\begin{array}{c} \lim_{j\to\infty} \min_{s\in S_p} \|s - s_j\|_H \;=\; 0, \quad \textit{for} \quad 1 \leq p \leq N_S \\ (\,A \quad \textit{Multi} - \textit{set} \quad \textit{Proximity} \quad \textit{Property}). \end{array} \tag{12.18}$$

That is, the points in the sequence $\{s_j\}_{j\geq 0}$ asymptotically approach every set S_j, $1 \leq j \leq N_S$.

Proof Convergence of $\|s_j - s_{j-1}\|_H$ to zero implies that, for all $\delta > 0$ (no matter how small), $\|s_{j+1} - s_j\|_H < \delta$ for all large enough indices "j". It follows that $\|s_j - s_p\|_H < N_D\delta$ for all large enough "p" and all "j" in the range $p - N_D \leq j \leq p + N_D$. This proves that, for all large enough values of "p", s_p is distant from each S_j by less than $N_D\delta$. The result follows as δ was chosen arbitrarily. □

The multi-set proximity property, in principle opens up an infinite number of new algorithms simply by changing the sequencing and repetition of the $\{S_j\}_{1\leq j\leq N_S}$ within $\{\tilde{S}_j\}_{j\geq 1}$. All such algorithms will be proximity algorithms and, in principle, the issues of convergence to a limit, the rate of convergence and the nature of that limit will require investigation. It is useful to note that the Switching Algorithm 11.2 is one such proximity algorithm.

Theorem 12.6 (Switching Algorithm for Auxiliary Optimization Revisited) *Using the notation of Chap. 11, let $H = \mathscr{Y}_e \times \mathscr{U} \times \mathscr{Z}$ with inner product*

$$\langle (e^e, u, z), (h, v, w)\rangle_H \;=\; \langle e^e, h\rangle_{\mathscr{Y}_e} + \varepsilon^2 \left(\langle u, v\rangle_{\mathscr{U}} + \langle z, w\rangle_{\mathscr{Z}}\right). \tag{12.19}$$

and associated induced norm $\|\cdot\|_H$. Under these conditions, suppose that the reference signal $r^e \in \mathscr{Y}_e$ can be created exactly by an input $u \in \mathscr{U}$, then, defining closed convex sets in H by

$$\begin{array}{rcl} S_1 & = & \{\, (e^e, u, z) \;:\; e^e = 0 \,\} \\ S_2 & = & \{\, (e^e, u, z) \;:\; e^e = r^e - G_e u - d^e \,\}, \\ S_3 & = & \{\, (e^e, u, z) \;:\; z = G_1 u + d_1 \,\} \\ \textit{and} \quad \textit{noting} \quad \textit{that} & & S_1 \cap S_2 \cap S_3 \quad \textit{is} \quad \textit{non} - \textit{empty}, \end{array} \tag{12.20}$$

Switching Algorithm 11.2 is recreated from a starting point $s_0 = (e_0^e, u_0, z_0) \in S_2 \cap S_3$ *by computing* $s_j \in \tilde{S}_j$, $j \geq 1$ *using the sequence*

$$\{\tilde{S}_j\}_{j\geq 1} = \{S_1, S_2, S_3, S_1, S_2, S_3, S_1, \ldots\}. \tag{12.21}$$

Proof First note that the projection of s_0 onto S_1 generates the iterate $s_1 = (0, u_0, z_0)$ by minimizing

$$\|(e^e, u, z) - (e_0^e, u_0, z_0)\|^2 = \|e^e - e_0^e\|_{\mathscr{Y}}^2 + \varepsilon^2(\|u - u_0\|_{\mathscr{Y}}^2 + \|z - z_0\|_{\mathscr{Y}}^2), \tag{12.22}$$

noting that $(e^e, u, z) \in S_1$ requires that $e^e = 0$ and that the minimizing values are $u = u_0$ and $z = z_0$.
Next, for any $k \geq 0$, STEP ONE of Algorithm 11.2 is just the projection $S_1 \to S_2$

$$(\tilde{e}_{k+1}^e, \tilde{u}_{k+1}, z_k) \;=\; \arg \min_{(e^e,u,z)\in S_2} \|(e^e, u, z) - (0, u_k, z_k)\|_H^2. \tag{12.23}$$

As the values of z are unconstrained in S_2, this norm minimization reduces to the choice of $z = z_k$ followed by solving the optimization problem

$$(\tilde{e}_{k+1}^e, \tilde{u}_{k+1}) \;=\; \arg \min_{(e^e,u)\in \mathscr{Y}_e\times\mathscr{U}} \left(\|e^e\|_{\mathscr{Y}_e}^2 + \varepsilon^2\|u - u_k\|_{\mathscr{U}}^2\right), \tag{12.24}$$

subject to the dynamic constraint $e^e = r^e - G_e u - d^e$. This is then followed by the projection $S_2 \to S_3$, to give

$$(\tilde{e}_{k+1}^e, u_{k+1}, z_{k+1}) \;=\; \arg \min_{(e^e,u,z)\in S_3} \|(e^e, u, z) - (\tilde{e}_{k+1}^e, \tilde{u}_{k+1}, z_k)\|_H^2. \tag{12.25}$$

As the values of e^e are unconstrained in S_3, this norm minimization reduces to the choice of $e^e = \tilde{e}_{k+1}^e$ followed by solving the optimization problem

$$(u_{k+1}, z_{k+1}) \;=\; \arg \min_{(u,z)\in \mathscr{U}\times\mathscr{Z}} \left(\|z - z_k\|_{\mathscr{Z}}^2 + \|u - \tilde{u}_{k+1}\|_{\mathscr{U}}^2\right), \tag{12.26}$$

subject to the dynamic constraint $z = G_1 u + d_1$. This is just STEP TWO of Algorithm 11.2. Finally, projecting this iterate back onto S_1 is the computation

$$(0, u_{k+1}, z_{k+1}) \;=\; \arg \min_{(e^e,u,z)\in S_3} \|(e^e, u, z) - (\tilde{e}_{k+1}^e, u_{k+1}, z_{k+1})\|_H^2. \tag{12.27}$$

This forms the starting condition for the next iteration. □

The result demonstrates existence of $N_S > 2$ sets defining some Iterative Control problems. The two sets S_2 and S_3 are created from the models of input-output behaviour and auxiliary variable dynamics. Neither set (or their intersection) requires that the tracking error is zero. The set S_1 is therefore introduced to ensure that the

intersection $S_1 \cap S_2 \cap S_3$ requires that $e^e = 0$. The combined effect is to "pull" the iterates into the position where useful control of the auxiliary variable z can be achieved (in this case, minimization of $J_{\mathscr{Z}}(u, u_0, z_0)$).
Note: Using the notation $(S_1, S_1)^{k_0}$ to denote the repetition of the pair (S_1, S_2) k_0 times or, equivalently, completing k_0 iterations of NOILC Algorithm 9.1, variations of the algorithm are easily created. They suggest the possibility of great flexibility in applications. For example, the algorithm resulting from the sequence

$$\{\tilde{S}_j\}_{j\geq 0} = \{ (S_1, S_2)^{k_0}, S_3, (S_1, S_2)^{k_1}, S_3, (S_1, S_2)^{k_3}, S_3, \dots \} \quad with \quad k_j \geq 1\,, \quad j \geq 0, \tag{12.28}$$

is also a proximity algorithm if the integer sequence $\{k_j\}_{j\geq 0}$ is bounded.

In summary, the Successive Projection Algorithm 12.1 provides an alternative viewpoint on the optimization-based algorithms described in previous chapters. It arises from their common Hilbert space context and the interpretation of NOILC computations as projections. The advantage of the operator-based methods used previously is that they provide a clear link between the dynamics, the observed properties of monotonicity, robustness and the nature of the limit. The advantage of successive projection and the idea of proximity is that it shows the great richness of the set of potential "new" algorithms. This is useful but, perhaps, a better proof of the benefits of successive projection approaches lies in

1. the handling of constraints on inputs and outputs/errors,
2. the management of signal magnitudes during the iteration process,
3. the introduction of new sets to NOILC Algorithm 9.1 to add projections that can improve or condition algorithm performance and/or
4. modifications to Algorithm 12.1 for the case of $N_S = 2$ that provide acceleration mechanisms.

12.3 Iterative Control with Constraints

Constraints on signal magnitude arise in many situations. Two forms of constraint can be envisaged, namely, those that must be satisfied at all times, and those that are desirable but not essential. For example, for input constrained problems there is

1. CASE ONE: the situation where the input signal constraints cannot be violated for physical or safety reasons or
2. CASE TWO: situations where input signals have no formal constraints but, as the choice of input signal producing an output exactly equal to a reference r is non-unique, it is desirable that the iteration process *asymptotically* selects one such input satisfying the constraints.

Both types of constraint can take many forms. For simplicity, the model of the process and the notation used in the Auxiliary Optimization problem (Chap. 11) is used to provide a framework for discussion. It is assumed that all or some of the

signals (u, y, z) (and, perhaps (y^e, e^e)) are required to satisfy constraints defined by set inclusion conditions written as

$$u \in \Omega_u \subset \mathscr{U}, \qquad y \in \Omega_y \subset \mathscr{Y}, \qquad z \in \Omega_z \subset \mathscr{Z}, \tag{12.29}$$

where each of the sets $\Omega_{(\cdot)}$ is closed and convex in the relevant Hilbert space. Examples of such sets include,

$$\begin{array}{rcl} \Omega_u & = & \{\, u \,:\, \|u\|^2_{\mathscr{U}} \leq M_u \,\} \subset \mathscr{U} \;-\; \textit{Limited Energy Control Requirements} \\ \Omega_y & = & \{\, y \,:\, \|y\|^2_{\mathscr{Y}} \leq M_y \,\} \subset \mathscr{Y} \;-\; \textit{Limiting Output Energy} \\ \Omega_z & = & \{\, z \,:\, \|z\|^2_{\mathscr{Z}} \leq M_z \,\} \subset \mathscr{Z} \;-\; \textit{Limited Auxiliary Variable Magnitudes} \end{array} \tag{12.30}$$

where M_u, M_y and M_z represent permissible, desired or specified norm magnitudes. For situations where G represents a single-input, single-output, discrete state space model on an interval $0 \leq t \leq N$, other examples include

$$\begin{array}{rcl} \Omega_u & = & \{\, u \,:\, |u(t)| \leq M_u(t) \,,\; 0 \leq t \leq N \,\} \;-\; \textit{Limited Input Magnitudes} \\ \Omega_y & = & \{\, y \,:\, |y(t)| \leq M_y(t) \,,\; 0 \leq t \leq N \,\} \;-\; \textit{Limiting Output Magnitudes} \end{array} \tag{12.31}$$

The absence of constraints is simply the case when $\Omega_{(\cdot)}$ is the whole space. For example, taking $\Omega_u = \mathscr{U}$ removes any constraint on the input signal.

A suitable Iterative Algorithm for a given problem will depend on the the details and complexities of the constraints sets. Some of these possibilities are discussed in the following sections beginning with the simplest case of NOILC Algorithm 9.1 with added input constraints.

12.3.1 NOILC with Input Constraints

Consider the Norm Optimal Algorithm 9.1 but with the addition of input constraints $u \in \Omega_u$. The following conceptual algorithm covers the case (CASE ONE) where input constraints are due to hardware limitations and cannot be violated,

Algorithm 12.2 (*NOILC with Input Constraints that must not be violated*) Suppose that the input constraint $u \in \Omega_u$ is added to the Norm Optimal Iterative Learning Control problem and that there exists an input $u_\infty \in \Omega_u$ such that perfect tracking of a reference signal r is achieved. Then, a proximity algorithm for computing an arbitrarily accurate solution of the problem starting from an initial input $u_0 \in \Omega_u$ is, for $k \geq 0$, the feedforward computation of the input to be used on iteration $k+1$ by solving the optimization problem,

$$u_{k+1} \;=\; \arg\min_{u \in \Omega_u} J(u, u_k) \qquad \textit{subject to} \quad e = r - Gu - d. \tag{12.32}$$

The algorithm has the property that the tracking error $e_k = r - y_k$ converges to zero as $k \to \infty$.

Proof of Algorithm Properties The analysis of the behaviour of this algorithm uses successive projection with the sets and the alternating sequence $\{\tilde{S}_j\}_{j\geq 1} = \{S_1, S_2, S_1, S_2, S_1, \ldots\}$ in $H = \mathscr{Y} \times \mathscr{U}$ defined by

$$\begin{array}{rcl} S_1 & = & \{\, (e,u) \;:\; e = r - Gu - d \;,\; u \in \Omega_u \,\} \quad \textit{and} \\ S_2 & = & \{\, (e,u) \;:\; e = 0 \,\} \end{array} \tag{12.33}$$

starting with the point $(0, u_0)$ in S_2. The inner product in H is

$$\langle (e,u), (w,v) \rangle_H = \langle e, w \rangle_{\mathscr{Y}} + \varepsilon^2 \langle u, v \rangle_{\mathscr{U}} . \tag{12.34}$$

The associated norm is defined by $\|(e,u)\|_H^2 = \|e\|_{\mathscr{Y}}^2 + \varepsilon^2 \|u\|_{\mathscr{U}}^2$. In NOILC terms, $\|(e,u) - (0,u_k)\|_H^2 = J(u,u_k)$. The projection of $(0,u_k) \in S_2$ onto S_1 is precisely the minimization specified in Algorithm 12.2 and leads to the pair (e_{k+1}, u_{k+1}). Projecting this back onto S_2 gives $(0, u_{k+1})$ which starts the next iteration with k replaced by $k+1$. Finally, $\lim_{k\to\infty} e_k = 0$ as, from Theorem 12.2, the algorithm is a proximity algorithm with $s_{2j-1} = (e_j, u_j)$ and $s_{2j} = (0, u_j)$, $j \geq 1$. Theorem 12.2, then gives $\|s_{2j} - s_{2j-1}\|_H = \|(0,u_j) - (e_j,u_j)\|_H = \|e_j\|_{\mathscr{Y}} \to 0$ as $k \to \infty$. □

Algorithm 12.2 is identical to the unconstrained version (Algorithm 9.1) but the minimization searches only over inputs within Ω_u. This normally means that, for example, the Riccati representations and solutions for linear state space systems cannot be used. A feedback implementation will, therefore, not be feasible, an observation that leaves only a feedforward implementation to be considered. However, a feed forward implementations is off-line and model-based and the reader will note that the constrained minimization of $J(u,u_k)$ then has no input from the observed data e_k generated on iteration k. The algorithm, in the form presented therefore is "blind" to actual control performance—a situation that is unacceptable, particularly in the presence of modelling errors.

This problem does not arise however in the case (CASE TWO) where the input is not constrained during the iterative process but where it is required that the input asymptotically satisfies a constraint $u \in \Omega_u$ to an arbitrary accuracy. The following algorithm provides an approach to a solution and, as a by-product, provides the option of a feedforward algorithm with inputs satisfying the constraint $u \in \Omega_u$, $k \geq 0$.

Algorithm 12.3 (*NOILC with Asymptotically Desirable Input Constraints*) Using the notation of Algorithm 12.2, suppose that a solution to the tracking problem exists that satisfies the input constraint $u \in \Omega_u$ but that the input signal is unconstrained during the iteration process. Then, a proximity algorithm for computing an arbitrarily accurate solution of the tracking problem using an input in Ω_u starting from an initial input $u_0 \in \Omega_u$ is, for $k \geq 0$, the repeated application of a two step process. The first step is the computation of an input $u_{k+1}^{(1)}$ by solving the unconstrained optimization problem,

$$u_{k+1}^{(1)} \;=\; \arg\min_{u\in\mathscr{U}} J(u,u_k) \qquad \textit{subject to} \quad e = r - Gu - d. \tag{12.35}$$

The second step replaces $u_{k+1}^{(1)}$ by the input u_{k+1} obtained by solving

$$u_{k+1} \;=\; \arg\min_{u\in\Omega_u} \|u - u_{k+1}^{(1)}\|_{\mathscr{U}}. \tag{12.36}$$

In particular,

1. $\lim_{k\to\infty} \|u_{k+1}^{(1)} - u_{k+1}\|_H = 0$ and hence, for large enough values of interation index k, $u_k^{(1)}$ satisfies the constraint $u \in \Omega_u$ arbitrarily accurately.
2. The algorithm has the property that the tracking errors, resulting from the use of either $u_{k+1}^{(1)}$ or u_{k+1}, converges to zero as $k \to \infty$. This gives the user a choice that can be made to suit the needs of the application.
3. For a feedforward implementation, off-line computation of both $u_{k+1}^{(1)}$ and u_{k+1} allows either to be used on the physical plant itself. For practical purposes this permits the user to satisfy the input constraints on every iteration. More precisely, using the input $u_k \in \Omega_u$ on each and every iteration, the constraints are satisfied and the full data set (e_k, u_k) can be used to compute $u_{k+1}^{(1)}$ using the normal feedforward computations for NOILC Algorithm 9.1. This provides the real link to plant behaviour that is necessary to ensure that the algorithm responds to the consequences of previous decisions.
4. For state space systems, the construction of $u_{k+1}^{(1)}$ can be done either off-line (a feedforward implementation) or as an on-line feedback implementation using Riccati-based state feedback. The construction of u_{k+1} will normally be undertaken off-line as it is typically independent of plant dynamics.

Proof of Algorithm Properties The relevant sets for the underlying successive projection algorithm are

$$\begin{aligned} S_1 \;&=\; \{\,(e,u) \;:\; e = r - Gu - d\,\} \quad \textit{and} \\ S_2 \;&=\; \{\,(e,u) \;:\; e = 0 \quad \textit{with} \quad u \in \Omega_u\,\} \;=\; \{0\}\times\Omega_u. \end{aligned} \tag{12.37}$$

The reader can verify that the algorithm is then a successive projection algorithm starting from $s_0 = (0, u_0) \in S_2$ and using the sequence $\{S_1, S_2, S_1, S_2, S_1, \ldots\}$. The existence of a solution ensures that $S_1 \cap S_2$ is nonempty so it is a proximity algorithm with $s_{2j} = (0, u_j)$ and $s_{2j-1} = (e_j^{(1)}, u_j^{(1)})$ for $j \geq 0$. The proof of convergence of the error then follows using an analysis similar to that used above for Algorithm 12.2. Noting that $\varepsilon^2\|u_j - u_j^{(1)}\|_{\mathscr{U}}^2 \leq \|s_{2j} - s_{2j-1}\|_H^2 \to 0$ and that each $u_j \in \Omega_u$, it follows that the input sequence $\{u_k^{(1)}\}_{k\geq 0}$ satisfies the constraints arbitrarily accurately if k is large. □

The set Ω_u is typically independent of the dynamics and can be computed off-line. For SISO state space application, examples include the formula-based solutions

$$\Omega_u = \{\, u \;:\; |u(t)| \le M_u \} \;\; gives \quad u_{k+1}(t) = \begin{pmatrix} M_u & if\ u^{(1)}(t) > M_u, \\ u^{(1)}(t) & if\ |u^{(1)}(t)| \le M_u \\ -M_u & if\ u^{(1)}(t) < -M_u, \end{pmatrix}$$
$$\Omega_u = \{\, u \;:\; \|u\|_{\mathscr{U}} \le M_u \,\} \; gives \quad u_{k+1}(t) = M_u \frac{u^{(1)}_{k+1}(t)}{\|u^{(1)}_{k+1}\|_{\mathscr{U}}}. \tag{12.38}$$

Finally, note that, as the operator G can take many forms, the discussion above also applies to other algorithms that have the original NOILC structure including Multi-task Algorithms and Intermediate Point Algorithms. In addition, the addition of initial conditions as "control" variables is included by a suitable definition of u and Ω_u.

12.3.2 General Analysis

The example of the previous section illustrates the possibilities implicit in successive projection. In what follows, it is shown that NOILC concepts are fundamental to the solution of a variety of constrained tracking problems in a way suitable for Iterative Learning Control. The model of Sect. 11.1 is used with dynamics represented by

$$\begin{array}{ccccc} y = Gu + d, & y^e = G_e u + d^e, & z = G_1 u + d_1 & with & \\ y \in \mathscr{Y}, & y^e \in \mathscr{Y}_e\,, & z \in \mathscr{Z} & and & u \in \mathscr{U}\,. \end{array} \tag{12.39}$$

The approach places the points (y^e, y, u, z) in the real Hilbert space $H = \mathscr{Y}_e \times \mathscr{Y} \times \mathscr{U} \times \mathscr{Z}$. The tracking objective is to track a a specified signal $r^e \in \mathscr{Y}_e$ using an input $u \in \Omega_u$ that generates a trajectory with zero tracking error

$$e^e = r^e - y^e = 0\,, \quad with \quad associated \quad signals \quad y \in \Omega_y \quad and \quad z \in \Omega_z. \tag{12.40}$$

Iterative algorithms therefore generate a sequence of signals $\{(e^e_k, y_k, u_k, z_k)\}_{k\ge 0}$ with the objective of asymptotically ensuring that (i) the tracking error $e^e = r^e - y^e$ approaches zero and that (ii) the signals (y, u, z) approach $\Omega_y \times \Omega_u \times \Omega_z$. As in the previous section, two cases are considered,

1. CASE ONE: the situation where the input signal constraints cannot be violated for physical or safety reasons or
2. CASE TWO: situations where the choice of input signal producing an output exactly equal to a reference r is non-unique. The role of constraints is purely to reflect a desire to converge to one such input that satisfies the constraints.

The inner product in H is given, with $\varepsilon^2 > 0$ and $\varepsilon_1^2 > 0$, by

$$\langle (y^e, y, u, z), (h^e, w, v, \psi)\rangle_H = \langle y^e, h^e\rangle_{\mathscr{Y}_e} + \varepsilon^2\varepsilon_1^{-2}\langle y, w\rangle_{\mathscr{Y}} + \varepsilon^2\left[\langle u, v\rangle_{\mathscr{U}} + \langle z, \psi\rangle_{\mathscr{Z}}\right] \tag{12.41}$$

with the associated norm $\|(y^e, y, u, z)\|_H = \sqrt{\langle (y^e, y, u, z), (y^e, y, u, z)\rangle_H}$.

The following conceptual algorithm is an approach to case one. It is a successive projection algorithm based on the following closed, convex sets in H,

$$\begin{array}{rcl} S_1 & = & \{\, (e^e, y, u, z) \;:\; e^e = r^e - G_e u - d^e \;,\; u \in \Omega_u \,\} \\ S_2 & = & \{\, (e^e, y, u, z) \;:\; y = Gu + d \;,\; y \in \Omega_y \;,\; u \in \Omega_u \,\} \\ S_3 & = & \{\, (e^e, y, u, z) \;:\; z = G_1 u + d_1 \;,\; z \in \Omega_z \;,\; u \in \Omega_u \,\} \\ S_4 & = & \{\, (e^e, y, u, z) \;:\; e^e = 0 \,\} \end{array} \tag{12.42}$$

Sets S_1, S_2 and S_3 describe the three sets of dynamics and associated constraints whilst S_4 "pulls" them together by requiring that the tracking error is zero. The sets intersect if, and only if, a solution to the problem exists.

Algorithm 12.4 (*A Conceptual Algorithm with Inviolable Constraints*) Suppose that $S_1 \cap S_2 \cap S_3 \cap S_4$ is non-empty. Let the data initializing the algorithm be $s_0 = (0, y_0, u_0, z_0) \in S_4$ and, for each $k \geq 0$, follow the four step process

1. Given the data $(0, y_k, u_k, z_k) \in S_4$, construct an input $u^{(1)}_{k+1}$ and associated tracking error e^e_{k+1} by solving (on-line or off-line) the constrained optimization problem,

$$\begin{array}{rcl} (e^e_{k+1}, u^{(1)}_{k+1}) & = & \arg\min_{u \in \Omega_u} \left(\|e^e\|^2_{\mathscr{Y}_e} + \varepsilon^2 \|u - u_k\|^2_{\mathscr{U}} \right) \\ & & \textit{subject to} \quad e^e = r^e - G_e u - d^e. \end{array} \tag{12.43}$$

   This step is exactly the projection of $(0, y_k, u_k, z_k) \in S_4$ onto S_1 to produce the data $(e^e_{k+1}, y_k, u^{(1)}_{k+1}, z_k)$.
2. Given the data $(e^e_{k+1}, y_k, u^{(1)}_{k+1}, z_k) \in S_1$, construct an input $u^{(2)}_{k+1}$ and associated output y_{k+1} by solving (on-line or off-line) the constrained optimization problem,

$$\begin{array}{rcl} (y_{k+1}, u^{(2)}_{k+1}) & = & \arg\min_{u \in \Omega_u \,,\, y \in \Omega_y} \left(\|y - y_k\|^2_{\mathscr{Y}_e} + \varepsilon_1^2 \|u - u^{(1)}_{k+1}\|^2_{\mathscr{U}} \right) \\ & & \textit{subject to} \quad y = Gu + d. \end{array} \tag{12.44}$$

   This step is exactly the projection of $(e^e_{k+1}, y_k, u^{(1)}_{k+1}, z_k) \in S_1$ onto S_2 to produce the data $(e^e_{k+1}, y_{k+1}, u^{(2)}_{k+1}, z_k)$.
3. Next compute the input u_{k+1} and associated auxiliary variable z_{k+1} by solving the constrained optimization problem

$$\begin{array}{rcl} (z_{k+1}, u_{k+1}) & = & \arg\min_{u \in \Omega_u \,,\, z \in \Omega_z} \left(\|z_k - z\|^2_{\mathscr{Z}} + \|u - u^{(2)}_{k+1}\|^2_{\mathscr{U}} \right) \\ & & \textit{subject to} \quad z = G_1 u + d_1. \end{array} \tag{12.45}$$

   This step is exactly the projection of $(e^e_{k+1}, y_{k+1}, u^{(2)}_{k+1}, z_k) \in S_2$ onto S_3 to produce the data $(e^e_{k+1}, y_{k+1}, u_{k+1}, z_{k+1})$.
4. Finally, the initial data for the next iteration is the point $(0, y_{k+1}, u_{k+1}, z_{k+1})$ which is just the projection of $(e^e_{k+1}, y_{k+1}, u_{k+1}, z_{k+1}) \in S_3$ onto S_4.

In particular,

1. by construction, each triple (y_k, u_k, z_k) satisfies the constraints and
2. the tracking errors $\{e_k^e\}_{k\geq 0}$ converge to zero as $k \to \infty$.

Proof of Algorithm Properties In terms of the notation of successive projection, $s_{4j} = (0, y_j, u_j, z_j)$, $s_{4j+1} = (e_{j+1}^e, y_j, u_{j+1}^{(1)}, z_j)$, $s_{4j+2} = (e_{j+1}^e, y_{j+1}, u_{j+1}^{(2)}, z_j)$, $s_{4j+3} = (e_{j+1}^e, y_{j+1}, u_{j+1}, z_{j+1})$. In particular $\|e_{j+1}^e\|_{\mathscr{Y}_e}^2 = \|s_{4(j+1)} - s_{4j+3}\|_H^2$ and convergence of the tracking error follows as Theorem 12.2 shows that $\lim_{j\to\infty} \|s_{j+1} - s_j\|_H = 0$. □

Clearly, because the constraints are embedded within each successive projection, the computational problems met in solving constrained optimization problems off-line are typically very challenging. In addition, the algorithm suffers from the problem noted in Sect. 12.3.1, namely that, using state space systems as a basis for the discussion,

1. the solution typically cannot be implemented using on-line feedback mechanisms.
2. In addition, feedforward implementations are also problematic as there is no obvious mechanism for embedding data from iteration k into the computations for iteration $k+1$. The algorithm is essentially off-line, model-based and "blind" to plant behaviours.

One way to achieve a resolution of the problem is to note that the sets of acceptable solutions is precisely $S_1 \cap S_2 \cap S_3 \cap S_4$ and that any redefinition of the component sets that leaves this intersection unchanged can also be used as the basis of an algorithm. This opportunity offers many choices, particularly for the case where, for example, the input is unconstrained during the iterations but the desirable outcome is that u approaches Ω_u arbitrarily closely in the limit. Some examples of this idea can be stated as modifications to Algorithm 12.4. For example,

Algorithm 12.5 (*General Tracking with Asymptotic Input Constraints*) Using the notation of Algorithm 12.4, suppose that the input constraints in S_1 are removed and transferred to S_4, and S_1 and S_4 are re-defined to become

$$\begin{aligned} S_1 &= \{\, (e^e, y, u, z) \,:\, e^e = r^e - G_e u - d^e \,\} \quad (\textit{Unconstrained Dynamics}) \\ S_4 &= \{\, (e^e, y, u, z) \,:\, e^e = 0\,,\ u \in \Omega_u \}. \end{aligned} \tag{12.46}$$

Then $S_1 \cap S_2 \cap S_3 \cap S_4$ is unchanged and Algorithm 12.4, with STEP ONE replaced by the unconstrained optimization

$$\begin{aligned} \textit{STEP ONE} \;:\quad (e_{k+1}^e, u_{k+1}^{(1)}) = \arg\min_{u\in\mathscr{U}} & \left(\|e^e\|_{\mathscr{Y}_e}^2 + \varepsilon^2 \|u - u_k\|_{\mathscr{U}}^2 \right) \\ & \textit{subject to} \quad e^e = r^e - G_e u - d^e, \end{aligned} \tag{12.47}$$

is still a proximity algorithm that retains the properties of Algorithm 12.4.

Note: The new version of STEP ONE is simply one iteration of NOILC Algorithm 9.1. The absence of input constraints in S_1 permits, using the state space system as an example for discussion, the implementation of STEP ONE either in a feedback form (using the data u_k and state feedback) or in a feedforward form (using the observed plant data (e_k^e, u_k)).

The modification described provides a link between the iteration process and actual, observed plant data but still requires the solution of constrained optimization problems in STEPS 2 and 3 (presumed to be off-line). This may be acceptable for a given application but the reader will note that the constraint transfer process used in Algorithm 12.5 could also be applied to one or both of these steps. For example, if it is decided that all dynamic optimizations are to be unconstrained, the algorithm becomes

Algorithm 12.6 (*Decoupled Constraints and Dynamic Optimization*) Using the notation of Algorithm 12.4, suppose that the constraints on (y, u, z) are removed from the sets S_1, S_2 and S_3 and transferred to S_4, to give

$$S_4 \;=\; \{\; (e^e, y, u, z) \;:\; e^e = 0\;,\; y \in \Omega_y\;,\; u \in \Omega_u\;,\; z \in \Omega_z\;\} \tag{12.48}$$

Then $S_1 \cap S_2 \cap S_3 \cap S_4$ is unchanged and Algorithm 12.4, with STEPS ONE, TWO and THREE replaced by unconstrained versions of the optimization problems mapping data as follows

$$\begin{array}{rcl} \mathit{STEP\ \ ONE} & : & (0, y_k, u_k, z_k) \to (e^e_{k+1}, y_k, u^{(1)}_{k+1}, z_k) \\ \mathit{STEP\ \ TWO} & : & (e^e_{k+1}, y_k, u^{(1)}_{k+1}, z_k) \to (e^e_{k+1}, y^{(1)}_{k+1}, u^{(2)}_{k+1}, z_k) \\ \mathit{STEP\ \ THREE} & : & (e^e_{k+1}, y^{(1)}_{k+1}, u^{(2)}_{k+1}, z_k) \to (e^e_{k+1}, y^{(1)}_{k+1}, u^{(3)}_{k+1}, z^{(1)}_{k+1}) \end{array} \tag{12.49}$$

whilst STEP FOUR becomes the generation of $(0, y_{k+1}, u_{k+1}, z_{k+1})$ from the optimizations

$$\begin{array}{c} y_{k+1} = \arg\min_{y \in \Omega_y} \|y - y^{(1)}_{k+1}\|_{\mathscr{Y}}\;, \quad u_{k+1} = \arg\min_{u \in \Omega_u} \|u - u^{(3)}_{k+1}\|_{\mathscr{U}} \\ \mathit{and} \qquad z_{k+1} = \arg\min_{z \in \Omega_z} \|z - z^{(1)}_{k+1}\|_{\mathscr{Z}}, \end{array} \tag{12.50}$$

yields a proximity algorithm that retains the properties of Algorithm 12.4.

The reader may note that the use of successive projection opens up a wide range of possibilities for algorithm development based on choice of the $\{S_j\}$ and their sequencing in $\{\tilde{S}_j\}$. For example, the reader can confirm that a proximity algorithm that retains the unconstrained optimization associated with S_1, S_2 and S_3 but places greater emphasis on ensuring that the input is constrained is created using the sequence

$$S_1, S_4, S_2, S_4, S_3, S_4, S_1, S_4, S_2, S_4, S_3, S_4, S_1, \ldots \tag{12.51}$$

in which the iterations are interlaced by projections onto the new constraint set S_4.

The structuring of the problem and its link to its implementation will therefore, in general, need careful consideration, particularly when constraints are introduced to shape the form of the desired solution. In the next few sections, a number of problems are considered "from first principles" to show how the ideas could apply in practice. For simplicity, this is done by extending or modifying algorithms derived earlier in this text to cope with constraints.

12.3.3 Intermediate Point Control with Input and Output Constraints

It was observed in Chap. 10 that the Intermediate Point Algorithm 10.4, in the form described, provides no influence or control over the form of the input signal or the behaviour of the plant output between intermediate points. A special demonstration of the power of successive projection can be obtained by generalizing the Intermediate Point Algorithm 10.4 to include the convergence to a solution that satisfies both input constraints $u \in \Omega_u \subset \mathscr{U}$ and specified output constraints $y \in \Omega_y \subset \mathscr{Y}$. In terms of the problem of Sect. 12.3.2, the auxiliary variable z is not defined and is removed from the problem. The motivation for algorithm development is to control the behaviour of the output $y(t)$ between the intermediate points $\{t_j\}$ by using desirable ranges of values to define the closed convex set Ω_y and to achieve this controlled behaviour without using excessively large input signals. The set Ω_y can be constructed in many ways. For example, an "additional reference signal" can be added to the problem. More precisely, r^e remains the prime reference signal for tracking purposes but an additional "reference" signal $r(t), t \in [0, T]$ is introduced. This signal is chosen by the design engineer to represent a desirable but, not essential, output on $[0, T]$. This reference is then used to construct the constraint set. An example of a reference signal $r(t)$ could be the straight line segments jointing points that satisfy the tracking condition $y^e = r^e$. Examples of constraints might take the form (where m denotes the number of outputs and $y_j(t)$ denotes, temporarily, the element of the $m \times 1$ vector $y(t)$ in the jth position/row)

$$|r_j(t) - y_j(t)| \leq M_{y,j}(t), \quad for \quad t \in [0, T] \quad - \quad (Point - wise \quad Constraints). \tag{12.52}$$

That is, the upper and lower bounds $r_j(t) \pm M_{y,j}(t)$ generate an "envelope" within which desirable values of the output signal $y_j(t)$ are located. For consistency purposes, the values of $r(t)$ at the intermediate points should be consistent with the choice of r^e.

A suitable successive projection-based Iterative Algorithm is set in the product space $H = \mathscr{Y}_e \times \mathscr{Y} \times \mathscr{U}$ of triples (e^e, y, u) with inner product

$$\langle (e^e, y, u), (h^e, w, v) \rangle_H = \langle e^e, h^e \rangle_{\mathscr{Y}_e} + \varepsilon^2 \varepsilon_1^{-2} \left(\langle y, w \rangle_{\mathscr{Y}} + \varepsilon_1^2 \langle u, v \rangle_{\mathscr{U}} \right). \tag{12.53}$$

The sets can be defined as follows,

$$\begin{array}{rcl} S_1 & = & \{\, (e^e, y, u) : \; e^e = r^e - G_e u - d^e \,\}\,, \\ S_2 & = & \{\, (e^e, y, u) : \; y = Gu + d \,\} \;\; \textit{and} \\ S_3 & = & \{\, (e^e, y, u) : \; e^e = 0\,, \;\; y \in \Omega_y \quad \textit{and} \quad u \in \Omega_u \quad \}. \end{array} \tag{12.54}$$

Note that $S_1 \cap S_2 \cap S_3$ defines triples $(0, y, u) \in \{0\} \times \Omega_y \times \Omega_u$ that also satisfy the dynamic equations and hence is the set of solutions to the constrained tracking problem. The sequence $\{\tilde{S}_j\}_{j\geq 0}$ defining the projections is the cyclic sequence

$$\{\tilde{S}_j\}_{j\geq 1} \;=\; \{S_1, S_2, S_3, S_1, S_2, S_3, S_1, \ldots\}. \tag{12.55}$$

The starting condition is non-unique. One choice of the initial point $s_0 = (0, r_0, u_0) \in S_3$ is obtained by choosing $u_0 \in \Omega_u$ where $r_0 = r \in \Omega_y$. The reader can now verify that the resultant proximity algorithm takes the form,

Algorithm 12.7 (*Tracking with Output Variable Constraints*) Using the notation of Sect. 10.5 suppose that a solution to the intermediate point tracking problem exists using an input $u \in \Omega_u$ with output trajectory $y \in \Omega_y$ and that the input signal is unconstrained during the iteration process but, ideally, becomes arbitrarily close to the constraint set Ω_u. Then, a proximity algorithm for computing an arbitrarily accurate solution of the tracking problem starting from an initial data set $s_0 = (0, r_0, u_0) \in S_3$ is, for $k \geq 0$, the repeated application of the following three step process:

1. STEP ONE: Given the data $(0, r_k, u_k) \in S_3$, find the input $u^{(1)}_{k+1}$ and associated tracking error e^e_{k+1} by solving the unconstrained intermediate point optimization problem,

$$\begin{array}{rcl} (e^e_{k+1}, u^{(1)}_{k+1}) & = & \arg\min_{u\in\mathscr{U}} \left(\|e^e\|^2_{\mathscr{Y}_e} + \varepsilon^2 \|u - u_k\|^2_{\mathscr{U}} \right) \\ & & \textit{subject} \quad \textit{to} \quad e^e = r^e - G_e u - d^e. \end{array} \tag{12.56}$$

   This step is exactly the projection of $(0, r_k, u_k) \in S_3$ onto S_1 to produce the data $(e^e_{k+1}, r_k, u^{(1)}_{k+1})$ using the techniques of Sect. 10.5 in either a feedback or feedforward form.
2. STEP TWO: Next compute the input $u^{(2)}_{k+1}$ and associated output y_{k+1} by solving the unconstrained optimization problem

$$\begin{array}{rcl} (y_{k+1}, u^{(2)}_{k+1}) & = & \arg\min_{u\in\mathscr{U}} \left(\|r_k - y\|^2_{\mathscr{Y}_e} + \varepsilon_1^2 \|u - u^{(1)}_{k+1}\|^2_{\mathscr{U}} \right) \\ & & \textit{subject} \quad \textit{to} \quad y = Gu + d. \end{array} \tag{12.57}$$

   This step is exactly the projection of $(e^e_{k+1}, r_k, u^{(1)}_k) \in S_1$ onto S_2 to produce $(e^e_{k+1}, y_{k+1}, u^{(2)}_{k+1})$. It is computed by one iteration of the NOILC Algorithm 9.1 using initial input $u^{(1)}_k$ and iteration dependent "reference" signal r_k and with ε replaced by ε_1.

3. STEP THREE: Finally, set up the initial data for the next iteration by setting

$$r_{k+1} = \arg\min_{r\in\Omega_y} \|r - y_{k+1}\|^2_{\mathscr{Y}} \quad \textit{and} \quad u_{k+1} = \arg\min_{u\in\Omega_u} \|u - u^{(2)}_{k+1}\|^2_{\mathscr{U}}. \tag{12.58}$$

This is the projection of $(e^e_{k+1}, y_{k+1}, u^{(2)}_{k+1}) \in S_2$ onto S_3 to produce $(0, r_{k+1}, u_{k+1})$ which is the triple that starts the next iteration. u_{k+1} will normally be computed off-line.

Notes: Links to plant performance can be included by implementing STEPS ONE and/or TWO on-line or off-line in feedforward or feedback form. In practice, each iteration may require several plant experiments to obtain response data but this price seems to be unavoidable if the additional constraints are to be satisfied. In addition,

1. $\lim_{k\to\infty} \|u^{(2)}_{k+1} - u^{(1)}_{k+1}\|_H = \lim_{k\to\infty} \|u^{(1)}_{k+1} - u_{k+1}\|_H = 0$ and hence, for large enough values of interation index k, both $u^{(1)}_k$ and $u^{(2)}_k$ satisfy the constraint $u \in \Omega_u$ arbitrarily accurately.
2. In a similar manner, the output sequence $\{y_k\}_{k\geq 0}$ satisfies $\lim_{k\to\infty} \|r_k - y_k\|_{\mathscr{Y}} = 0$ and hence the outputs increasingly satisfy the required constraints $y \in \Omega_y$ to an arbitrary accuracy.
3. The algorithm has the property that the tracking errors $\{e^e_k\}_{k\geq 0}$ converge to zero as $k \to \infty$.

The algorithm is a proximity algorithm by construction. Its convergence properties follow from the definitions and the properties of such algorithms as expressed in Theorem 12.2. The details are left to the reader as part of his/her study during which he or she may wish to investigate the proximity properties of and motivations for other algorithms induced by changes to the set sequencing. For example, the sequence

$$\{\tilde{S}_j\}_{j\geq 1} = \{(S_1, S_2)^{k_1}, S_3, (S_1, S_2)^{k_2}, S_3, \ldots\} \quad (\textit{Repetition}), \tag{12.59}$$

defines a new algorithm with the desired proximity properties.

12.3.4 Iterative Control to Satisfy Auxiliary Variable Bounds

Successive projection can also be used to reformulate Algorithm 11.2 (Sects. 11.1 and 11.3) by modifying the objectives of the control. More precisely, this section uses the model and notation of those sections but replaces the objective of successfully achieving tracking of a reference signal r^e whilst simultaneously minimizing an auxiliary optimization criterion $J_{\mathscr{Z}}$ by that of

1. successful tracking of r^e using a control signal $u \in \Omega_u$ and
2. simultaneously ensuring that the auxiliary variable z satisfies a specified closed and convex constraint $z \in \Omega_z \subset \mathscr{Z}$.

The space $H = \mathscr{Y}_e \times \mathscr{U} \times \mathscr{Z}$ of triples (e^e, u, z) is used with the inner product

$$\langle (e^e, u, z), (h^e, v, w) \rangle_H = \langle e^e, h^e \rangle_{\mathscr{Y}_e} + \varepsilon^2 \left(\langle u, v \rangle_{\mathscr{U}} + \langle z, w \rangle_{\mathscr{Z}} \right). \tag{12.60}$$

The iteration sequence $\{\tilde{S}_j\}_{j \geq 0}$ cycles around the sets

$$\begin{array}{rcl} S_1 & = & \{ (e^e, u, z) : \ e^e = r^e - G_e u - d^e \} \\ S_2 & = & \{ (e^e, u, z) : \ z = G_1 u + d_1 \} \\ S_3 & = & \{ (e^e, u, z) : \ e^e = 0 \ , \ u \in \Omega_u \quad \textit{and} \quad z \in \Omega_z \quad \}. \end{array} \tag{12.61}$$

It is a simple matter to prove that, if a solution exists, the following algorithm is a proximity algorithm and hence solves the control problem to increasing and arbitrary accuracy as iterations progress.

Algorithm 12.8 (*Tracking with Auxiliary Variable Constraints*) Let the initial data be $s_0 = (0, u_0, z_0) \in S_3$ and for each $k \geq 0$ follow the three step process

1. STEP ONE: Given the data $(0, u_k, z_k) \in S_3$, construct an input $u^{(1)}_{k+1}$ and associated tracking error e^e_{k+1} by solving (on-line or off-line) the unconstrained optimization problem,

$$\begin{array}{rcl} (e^e_{k+1}, u^{(1)}_{k+1}) & = & \arg\min_{u \in \mathscr{U}} \left(\|e^e\|^2_{\mathscr{Y}_e} + \varepsilon^2 \|u - u_k\|^2_{\mathscr{U}} \right) \\ & & \textit{subject to} \quad e^e = r^e - G_e u - d^e. \end{array} \tag{12.62}$$

   This step is exactly the projection of $(0, u_k, z_k) \in S_3$ onto S_1 to produce the data $(e^e_{k+1}, u^{(1)}_{k+1}, z_k)$.

2. STEP TWO: Next compute the input $u^{(2)}_{k+1}$ and associated auxiliary variable z_{k+1} by using NOILC techniques to solve the unconstrained optimization problem

$$\begin{array}{rcl} (z^{(1)}_{k+1}, u^{(2)}_{k+1}) & = & \arg\min_{u \in \mathscr{U}} \left(\|z_k - z\|^2_{\mathscr{Z}} + \|u - u^{(1)}_{k+1}\|^2_{\mathscr{U}} \right) \\ & & \textit{subject to} \quad z = G_1 u + d_1. \end{array} \tag{12.63}$$

   This step is exactly the projection of $(e^e_{k+1}, u^{(1)}_{k+1}, z_k) \in S_1$ onto S_2 to produce $(e^e_{k+1}, u^{(2)}_{k+1}, z^{(1)}_{k+1})$.

3. STEP THREE: Finally, set up the initial data for the next iteration by setting

$$z_{k+1} = \arg\min_{z \in \Omega_z} \|z - z^{(1)}_{k+1}\|^2_{\mathscr{Z}} \quad \textit{and} \quad u_{k+1} = \arg\min_{u \in \Omega_u} \|u - u^{(2)}_{k+1}\|^2_{\mathscr{U}}. \tag{12.64}$$

   This is just the projection of the triple $(e^e_{k+1}, u^{(2)}_{k+1}, z^{(1)}_{k+1}) \in S_2$ onto S_3 to produce $(0, u_{k+1}, z_{k+1})$.

In particular, using the properties of successive projection,

1. for large enough values of interation index k, both $u_k^{(1)}$ and $u_k^{(2)}$ satisfy the constraint $u \in \Omega_u$ arbitrarily accurately.
2. In a similar manner, the auxiliary variables satisfy $\lim_{k\to\infty} \|z_k - z_k^{(1)}\|_{\mathscr{Z}} = 0$ and hence they increasingly satisfy the required constraints $z \in \Omega_z$ to an arbitrary accuracy.
3. The tracking errors $\{e_k^e\}_{k\geq 0}$ converges to zero as $k \to \infty$.

The typical simplicity of the computations of STEP THREE are illustrated as follows:

$$\begin{aligned} &\textit{If} \quad \Omega_z = \{\, z \,:\, \|z\|_{\mathscr{Z}} \leq M_z \,\} \quad \textit{then} \quad z_{k+1} = z_{k+1}^{(1)} \quad \textit{if} \quad \|z_{k+1}^{(1)}\|_{\mathscr{Z}} \leq M_z. \\ &\qquad\qquad \textit{Otherwise it takes the value} \quad z_{k+1} = M_z \frac{z_{k+1}^{(1)}}{\|z_{k+1}^{(1)}\|_{\mathscr{Z}}}. \end{aligned} \tag{12.65}$$

12.3.5 An Overview and Summary

The previous section has demonstrated the use of successive projection as a tool in solving Iterative Control problems (regarded as proximity problems). The approach has the benefits of having a geometric interpretation and having natural links to Norm Optimal Iterative Learning Control Algorithms exemplified by Sect. 9.1 and its many realizations in, for example, the Intermediate Point Algorithm 10.4 and the Auxiliary Optimization Algorithms in Chap. 11. The approaches apply to other algorithms including Multi-task Algorithms 10.5, 10.6 and 10.8. The last example is interesting as it opens up the possibility, by regarding initial conditions as part of the control input, of selecting an input from a defined closed convex set.

Algorithm development can be summarized as the definition of a suitable "set of sets" $\{S_j\}_{1\leq j\leq N_S}$ in a real Hilbert space H with the properties that any point in the intersection $\bigcap_{j=1}^{N_S} S_j$ solves the combined tracking problem with any additional objectives. These sets are then put in a sequence $\{\tilde{S}_1, \tilde{S}_2, \tilde{S}_3, \ldots\}$ typically satisfying a condition such as that given in Theorem 12.5. An algorithm is then obtained by solving the sequence of optimization problems generated by successive projection. Assuming that at least one solution to the problem exists, the algorithm is then a proximity algorithm that creates iterates that become arbitrarily close to every set S_j , $1 \leq j \leq N_S$.

In the real world, it is important to remember that part of this process involves experimental work on the plant to ensure that the algorithm is informed by the actual response of the physical system.

The structure of algorithm development is clear and relatively simple. The challenge in practice will be the choice of sets and inner products and norms that also satisfy the need for good convergence rates combined with mechanisms for using measured plant data in the iteration process. This is essential to ensure that the

algorithm responds to the consequences of its previous inputs. In practice this is necessary for situations where plant modelling errors are present. Some of the associated robustness questions have been addressed in previous chapters. Unfortunately, at this time, there is no robustness theory for more general cases, particularly those where signals are subject to constraints and/or complex sequencing in $\{\tilde{S}_j\}_{j\geq 1}$.

12.4 "Iteration Management" by Operator Intervention

In this section successive projection is moved from its position as an approach that forms an Iterative Algorithm in its own right. Its new role reflects the possibility of including the presence of a human operator in the decision-making process prior to each iteration. In previous algorithms, the presence of the operator played no role in algorithm development and performance was analysed using theoretical tools and the mathematical relationships between signals and iterates. In effect the operator was a passive presence at the real-world implementation whose main roles were limited to parameter choice (or variation), algorithm initialization, algorithm termination and/or coping with fault situations. In contrast, this section regards *Iterative Algorithms as human operator driven processes* defined by

1. **Tracking**: the need to satisfy tracking objectives plus any associated performance objectives and constraints and
2. **Intervention**: providing the option for the human operator to specify or modify the aims of each iteration using appropriate interventions.

It is the second point that guides the material that follows. It is assumed that an operator, having seen the outcomes of previous iterations, has a view of what is acceptable or desirable behaviour in the next iteration. The expression of these views is formed by choosing closed convex sets to shape or constrain the form of the next error signal and/or constrain input values. These sets, together with a plant model, form the data for selecting the next input signal using a supporting computational toolbox. The ideas are suitable for a feedforward implementation.

For simplicity of presentation, the following discussion considers operator driven iteration as a modification to the NOILC Algorithm 9.1. The plant is modelled by $y = Gu+d,\ u \in \mathscr{U},\ y \in \mathscr{Y}$ and a reference signal $r \in \mathscr{Y}$ is specified. A conceptual algorithm that places the operator at the centre of the iteration process is as follows:

Algorithm 12.9 (*Iteration Management using Constraints*) Suppose that error and input data (e_k, u_k) have been obtained from iteration k. Suppose also that the operator wishes to choose an input u_{k+1} for implementation on iteration $k+1$ but has views/preferences on what the preferred characteristics of the signals (e_{k+1}, u_{k+1}) should be. The procedure is as follows,

1. STEP ONE: Compute the data $(e_{k+1}^{(0)}, u_{k+1}^{(0)})$ to be the off-line, feedforward prediction of input and error obtained from the data (e_k, u_k), namely

$$u_{k+1}^{(0)} = u_k + \Delta u_k \quad \textit{and} \quad e_{k+1}^{(0)} = r - G u_{k+1}^{(0)} - d, \tag{12.66}$$

   where Δu_k is the control change computed using one iteration of NOILC Algorithm 9.1 from the initial input $u = 0$ using the model $y = Gu$ with zero initial conditions and a reference signal equal to e_k.
2. STEP TWO: The data $(e_{k+1}^{(0)}, u_{k+1}^{(0)})$ has a special role in this computation as it is the outcome of the normal NOILC iteration and hence is unaffected by operator intervention. However,
   a. if the data has acceptable characteristics, then set $p = 0$ and go to STEP FOUR. Otherwise,
   b. the data can be used to inform the operator's choice of closed, convex constraint sets $\Omega_e(k+1) \subset \mathscr{Y}$ and $\Omega_u(k+1) \subset \mathscr{U}$ representing his or her preferences for the characteristics of the predicted errors and input signals on iteration $k+1$.
3. STEP THREE: Define the following sets in $\mathscr{Y} \times \mathscr{U}$,

$$\begin{array}{rcll} S_1 & = & \{\, (e,u) \,:\, e = r - Gu - d \,\} & (\textit{Plant Dynamic Model}) \\ S_2 & = & \{\, (e,u) \,:\, e \in \Omega_e(k+1)\,,\ u \in \Omega_u(k+1) \,\} & (\textit{Iteration Objectives}). \end{array} \tag{12.67}$$

   The data $s_0 = (e_{k+1}^{(0)}, u_{k+1}^{(0)})$ is then used in an off-line, model-based, successive projection algorithm with alternating sets $\{S_2, S_1, S_2, S_1, \ldots\}_{j \geq 1}$ to generate iterates $(e_{k+1}^{(j)}, u_{k+1}^{(j)}) \in S_1,\ j = 1, 2, \ldots$ and $(\tilde{e}_{k+1}^{(j)}, \tilde{u}_{k+1}^{(j)}) \in S_2,\ j = 1, 2, \ldots$.
   a. Projection of $(e_{k+1}^{(j-1)}, u_{k+1}^{(j-1)}) \in S_1$ onto S_2 to produce $(\tilde{e}_{k+1}^{(j)}, \tilde{u}_{k+1}^{(j)}) \in S_2$ is normally a simple process expressed in the form

$$\begin{array}{rcl} \tilde{e}_{k+1}^{(j)} & = & \arg\min_{e \in \Omega_e(k+1)} \| e - e_{k+1}^{(j-1)} \|_{\mathscr{Y}}^2 \quad \textit{and} \\ \tilde{u}_{k+1}^{(j)} & = & \arg\min_{u \in \Omega_u(k+1)} \| u - u_{k+1}^{(j-1)} \|_{\mathscr{U}}^2 \end{array} \tag{12.68}$$

   b. Projection of $(\tilde{e}_{k+1}^{(j)}, \tilde{u}_{k+1}^{(j)}) \in S_2$ onto S_1 to produce $(e_{k+1}^{(j)}, u_{k+1}^{(j)}) \in S_1$ is the off-line NOILC computation of the signals minimizing

$$\| \tilde{e}_{k+1}^{(j)} - e \|_{\mathscr{Y}}^2 + \| u - \tilde{u}_{k+1}^{(j)} \|_{\mathscr{U}}^2 \tag{12.69}$$

   subject to the plant dynamic model $e = r - Gu - d$.
   Note: A moment's consideration indicates that this is precisely the NOILC objective function that appears in in Algorithm 9.1 with u_k replaced by $\tilde{u}_{k+1}^{(j)}$ and the reference signal r replaced by $r - \tilde{e}_{k+1}^{(j)}$.

This iterative process is terminated at a point $(e_{k+1}^{(p)}, u_{k+1}^{(p)}) \in S_1$ which satisfies the constraints $e_{k+1}^{(p)} \in \Omega_e(k+1)$ and $u_{k+1}^{(p)} \in \Omega_u(k+1)$ to an acceptable accuracy. The number of iterations p required is not fixed and will typically vary with index k and the desired accuracy.

4. STEP FOUR: The input signal u_{k+1} is then set equal to $u_{k+1}^{(p)}$ and applied to the plant to generate a new error signal e_{k+1}. Then, *either* a new iteration begins by returning to STEP ONE with the data (e_{k+1}, u_{k+1}) and k increased by unity, *or* the iterative process is terminated as acceptable tracking has been achieved.

In practice, the choice of the sequences $\{\Omega_e(j)\}_{j\geq 1}$ and $\{\Omega_u(j)\}_{j\geq 1}$ are crucial to algorithm performance. Convergence to acceptable tracking errors simply requires that each iteration successfully achieves the operator's requirements and that, for some iteration index $k = N_F$, the set $\Omega_e(N_F)$ only contains signals that satisfy the ultimate tracking accuracy requirements of the application.

The overall process consists of operator-guided "outer iterations" indexed by k each of which consists of a number of (off-line) "inner iterations" of the successive projection algorithm. In the absence of modelling errors, it results in a new iteration input u_{k+1} that is predicted to ensure that the error satisfies the iteration constraint to a good accuracy. If modelling errors are present and substantial, this objective may be achieved but only approximately.

For the purposes of discussion, consider an ℓ-input, m-output state space system $S(A, B, C)$ with inner products defined by $m \times m$ and $\ell \times \ell$ matrices Q and R respectively. Given a reference signal $r(t)$, the performance of NOILC Algorithm 9.1 in the form of Algorithm 9.5 will depend on the choices of Q and R in the sense that the relationship between these choices and the behaviour and magnitude of outputs in each loop/channel as iterations progress cannot be predicted. If the operator has views or concerns about the dynamics of particular channel outputs following iteration k, the set $\Omega_e(k+1)$ can be defined to reflect those concerns. For example,

1. in an attempt to control convergence rates, the operator may choose to define

$$\Omega_e(k+1) \;=\; \{\; (e,u) \;:\; \|e\|_{\mathscr{Y}} \leq \gamma_{k+1}\|e_k\|_{\mathscr{Y}} \;\} \tag{12.70}$$

where γ_{k+1} represents the desired fractional reduction of the error. For example, choosing $\gamma_{k+1} = 0.5$ expresses the desire that the error norm will (more than) halve in the next iteration.

2. Alternatively, although the error norm may be reducing, the distribution of this reduction across the channels may not be what is preferred. In this case $\Omega_e(k+1)$ could be a set defining desired norm bounds on individual channel errors or pointwise bounds on individual errors. In this way, the operator can reflect preferences and also manage situations arising from previous iterations.

Note that the role of Q and R in this approach is no longer that of ensuring good convergence rates. Rather, it is that of ensuring a good performance from the inner iteration processes.

This section concludes with some examples describing how constraint sets can be constructed. For simplicity of presentation, the state space model is again used and subscripts on errors will, for this discussion, describe the row position of the error in its vector format $[e_1(t), \ldots, e_m(t)]^T$. If, in addition, Q is diagonal of the form $Q = diag[Q_1, Q_2, \ldots, Q_m]$,

1. the definition of the constraint set

$$\Omega_e(k+1) \;=\; \{\, e \;:\; \int_0^T e_j^2(t)dt \;\leq\; M_j\,, \quad 1 \leq j \leq m \,\} \tag{12.71}$$

   places desired bounds on the mean-square errors in each error channel. These bounds can be an expression of a requirement that error norms in each channel are reduced at specified rates. Alternatively, they could reflect a desire to concentrate on reductions in some loops whilst preventing excessive increases in the remaining loops.
2. Another possible constraint description takes the form

$$\Omega_e(k+1) \;=\; \{\, e \;:\; |e_j(t)| \;\leq\; M_j(t)\,, \quad 1 \leq j \leq m, \quad t \in [0,T] \,\} \tag{12.72}$$

   where now $M_j(t)$ describes a time-varying bound on the error. Such a description could be used to concentrate on errors in certain subintervals if this is deemed to be important for system operation.

A mix of these two constraint definitions could also be used in an obvious way.

12.5 What Happens If S_1 and S_2 Do Not Intersect?

The purpose of this shorts section is to consider possible algorithm behaviours in the case when the defining sets $\{S_j\}_{1 \leq j \leq N_S}$ do not intersect. That is, a tracking problem has been defined that has no solution. This has already been seen to occur, for example, in NOILC Algorithm 9.1 when the range of G, $\mathscr{R}[G]$, is a proper subspace of $\mathscr{Y}$ and the user specifies a reference signal such that $r - d$ does not lie in that subspace. For NOILC and related gradient algorithms, linear analysis shows, typically, that monotonic convergence is achieved to a limit error equal to the component of e_0 that lies in $ker[G^*]$. Such analyses are not appropriate for more general, constrained, tracking problems. For example, using Algorithm 12.3, perfect tracking may be possible if the input is unconstrained by not possible if input constraints are too tight.

To illustrate the sort of behaviours that might be observed in practice, successive projection between $N_S = 2$, closed, convex sets S_1 and S_2 in a real Hilbert space H is considered in the case where the intersection $S_1 \cap S_2$ is empty. The projection process is still well-defined for such a case but convergence to a point in the intersection is clearly not possible. The following result provides some insight into situations where H is finite dimensional and S_2 is also bounded.

Theorem 12.7 (Asymptotic Behaviour when $S_1 \cap S_2$ is Empty) *Suppose that two closed,convex sets S_1 and S_2 are given in a real, finite-dimensional Hilbert space H and that their intersection is empty. Suppose also that S_2 is bounded. Then, defining the ordered set of sets $\{\tilde{S}_j\}_{j\geq 1}$ by the alternating sequence $\{S_1, S_2, S_1, S_2, S_1, \ldots\}$, the successive projection algorithm beginning with a point $s_0 \in S_2$ and subsequently evaluating the iterates $\{s_j\}_{j\geq 1}$ by the optimization problem*

$$s_{j+1} = \arg \min_{s\in\tilde{S}_{j+1}} \|s - s_j\|_H^2 , \quad j \geq 0, \tag{12.73}$$

has the properties that

$$\begin{array}{c} \|s_{j+2} - s_{j+1}\|_H \leq \|s_{j+1} - s_j\|_H \quad (\textit{Monotonicity}), \\ \lim_{j\to\infty} \|s_{j+2} - s_j\|_H = 0 \quad \textit{and} \quad \lim_{j\to\infty} \|s_{j+1} - s_j\|_H = \delta \\ \textit{where} \quad \delta = \inf\{ \|v_1 - v_2\| \ : \ \textit{with } v_1 \in S_1 \textit{ and } v_2 \in S_2 \} \end{array} \tag{12.74}$$

is the minimum distance between the two sets.

The theorem has a simple interpretation, simply stated as a form of convergence. It does not say that the iterates converge but, rather, says that the sequence alternates (in a form of "limit cycle") between the two sets, getting closer together all the time. Critically, it demonstrates that, ultimately, it is not possible to create iterates in S_1 (respectively, S_2) that are any closer to S_2 (respectively, S_1). In Iterative Control terms, the system dynamics can be satisfied with signals that best approximate the constraints represented by their companion set. Best is defined by the minimum distance property.

Proof of Theorem 12.7 The monotonicity property follows from the definitions of the iterates as s_{j+2} is closer than any other point to s_{j+1}. Monotonicity guarantees the existence of the real number δ which is strictly positive as the intersection is empty and S_2 is closed and bounded (and hence compact). If $\delta = 0$, then an argument based on selection of appropriate subsequences creates limit points $\hat{s}_1 \in S_1$ and $\hat{s}_2 \in S_2$ satisfying $\|\hat{s}_1 - \hat{s}_2\|_H = 0$. That is, $\hat{s}_1 = \hat{s}_2$ which contradicts the non-intersection assumption. Hence $\delta > 0$.
Next, using the property $\langle s_j - s_{j+2}, s_{j+2} - s_{j+1}\rangle_H \geq 0$, write

$$\begin{array}{rcl} \|s_{j+2} - s_j\|_H^2 & = & \|s_{j+2} - s_{j+1}\|_H^2 + \|s_{j+1} - s_j\|_H^2 + 2\langle s_{j+2} - s_{j+1}, s_{j+1} - s_j\rangle_H \\ \textit{so that} & & \\ \|s_{j+2} - s_j\|_H^2 & = & \|s_{j+2} - s_{j+1}\|_H^2 + \|s_{j+1} - s_j\|_H^2 \\ & & +2\langle s_{j+2} - s_{j+1}, (s_{j+1} - s_{j+2}) + (s_{j+2} - s_j)\rangle_H \\ & \leq & \|s_{j+1} - s_j\|_H^2 - \|s_{j+2} - s_{j+1}\|_H^2 \to 0+ \end{array} \tag{12.75}$$

as $k \to \infty$ as required. The compactness of S_2 means that a subsequence of iterates $\{s_{j_p}\}_{p\geq 1}$ in S_2 converges to a point $\hat{s}_2 \in S_2$. Set $\delta = \inf_{p\geq 1} \|s_{j_p+1} - \hat{s}_2\|_H$ and, using the optimization properties, that this implies that $\delta = \inf_{s\in S_1} \|s - \hat{s}_2\|_H$. The shifted sequence $\{s_{j_p+1}\}_{p\geq 1}$ in S_1 is bounded and hence, without loss of generality, it is

possible to assume that it converges to a point $\hat{s}_1 \in S_1$ with $\delta = \inf_{s\in S_1} \|s_{j_p+1} - \hat{s}_1\|_H$ and hence $\delta = \inf_{s\in S_2} \|s - \hat{s}_1\|_H$. In particular, $\delta = \|\hat{s}_2 - \hat{s}_1\|_H$. As a consequence,

$$\begin{array}{l} \langle s - \hat{s}_1, \hat{s}_1 - \hat{s}_2\rangle_H \geq 0 \quad \textit{for all} \quad s \in S_1 \quad \textit{and} \\ \langle s - \hat{s}_2, \hat{s}_2 - \hat{s}_1\rangle_H \geq 0 \quad \textit{for all} \quad s \in S_2. \end{array} \tag{12.76}$$

Now let P_s be the orthogonal projection onto the subspace spanned by $\hat{s}_1 - \hat{s}_2$. Then, for any vector $s \in H$

$$P_s s \;=\; \frac{\langle \hat{s}_1 - \hat{s}_2, s\rangle_H}{\|\hat{s}_1 - \hat{s}_2\|_H^2}(\hat{s}_1 - \hat{s}_2) \quad \textit{and} \quad P_s(\hat{s}_1 - \hat{s}_2) = \hat{s}_1 - \hat{s}_2. \tag{12.77}$$

It follows that, for any $s_1 \in S_1$ and $s_2 \in S_2$,

$$\begin{aligned} \|s_1 - s_2\|_H^2 &\geq \|P_s(s_1 - s_2)\|_H^2 = \|P_s\left(s_1 - \hat{s}_1\right) + (\hat{s}_1 - \hat{s}_2) + P_s(\hat{s}_2 - s_2)\right)\|_H^2 \\ &= \|\hat{s}_1 - \hat{s}_2\|_H^2 \left(1 + \tfrac{\langle s_1 - \hat{s}_1, \hat{s}_1 - \hat{s}_2\rangle_H}{\|\hat{s}_1 - \hat{s}_2\|_H^2} + \tfrac{\langle s_2 - \hat{s}_2, \hat{s}_2 - \hat{s}_1\rangle_H}{\|\hat{s}_1 - \hat{s}_2\|_H^2}\right)^2 \\ &\geq \|\hat{s}_1 - \hat{s}_2\|_H^2. \end{aligned} \tag{12.78}$$

This completes the proof of the theorem. □

In practice, the calculation of $\hat{s}_1$ and $\hat{s}_2$ is not possible prior to the iterative process. In certain cases, it is easy, however, to demonstrate that they are uniquely defined and have properties that can have design interpretations. For example, suppose that H is finite dimensional and consider a variation of Algorithm 12.3 obtained by setting $H = \mathscr{U}$ and using the *two* sets

$$\begin{array}{ll} S_1 = \{\, u \;:\; r = Gu + d \,\} & (\textit{Exact Tracking Achieved}) \\ S_2 = \{\, u \;:\; \|u - u_0\|_{\mathscr{U}} \leq M_u \,\} & (\textit{Bounded Input Energy}) \end{array} \tag{12.79}$$

Algorithm 12.10 (*NOILC Algorithm with Bounded Input Energy*) Suppose that GG^* has a bounded inverse in $\mathscr{Y}$ and that $H = \mathscr{U}$ is finite dimensional. Using the notation defined above, let $u_0 \in S_2$ initiate the following iterative algorithm for $k \geq 0$,

STEP ONE: Given the input $u_k \in S_2$ and associated tracking error $e_k = r - y_k$, compute the input

$$u_{k+1}^{(1)} = u_k + G^*(GG^*)^{-1}e_k \tag{12.80}$$

That is, project the point u_k onto the tracking requirement S_1. Equivalently, the input $u_{k+1}^{(1)}$ satisfies the tracking requirement and minimizes $\|u - u_k\|_{\mathscr{U}}^2$.
STEP TWO: Replace $u_{k+1}^{(1)}$ by the input signal for the next iteration

$$u_{k+1} = \begin{pmatrix} u_{k+1}^{(1)} & \text{if} \quad \|u_{k+1}^{(1)} - u_0\| \leq M_u \\ u_0 + \frac{M_u}{\|u_{k+1}^{(1)} - u_0\|_{\mathscr{U}}}(u_{k+1}^{(1)} - u_0) & \text{if} \quad \|u_{k+1}^{(1)} - u_0\| > M_u \end{pmatrix} \tag{12.81}$$

obtained by projecting $u_{k+1}^{(1)}$ onto S_2.

The reader should be able to verify that the algorithm has the following properties

1. If $S_1 \cap S_2$ is non-empty, then the algorithm is a proximity algorithm with

$$\lim_{k\to\infty} \|u_{k+1} - u_{k+1}^{(1)}\|_{\mathscr{U}} = 0 \tag{12.82}$$

   As iterations progress,

   a. inputs $u_k^{(1)}$ satisfy the tracking requirements and also ultimately satisfy the norm constraint requirement to an arbitrary accuracy. In addition,
   b. the inputs u_k satisfy the constraint and the tracking requirement to an arbitrary accuracy.

2. If $S_1 \cap S_2$ is empty, then, noting that the form of S_1 and S_2 indicates that the points $u_\infty^{(1)} \in S_1$ and $u_\infty \in S_2$ satisfying $\delta = \|u_\infty^{(1)} - u_\infty\|$ are uniquely defined and that the following limits

$$\lim_{k\to\infty} u_k^{(1)} = u_\infty^{(1)} \quad \textit{and} \quad \lim_{k\to\infty} u_k = u_\infty \quad \textit{exist}, \tag{12.83}$$

   it follows that the smallest value of M_u for which tracking is possible is obtained by the map

$$M_u \quad \mapsto \quad M_u + \|u_\infty - u_\infty^{(1)}\|_{\mathscr{U}}. \tag{12.84}$$

12.6 Discussion and Further Reading

Successive projection has been proposed in several contexts [14, 15, 18, 36] and was proposed for solution of differential algebraic equations in [95]. It has a conceptual simplicity that fits very well into the Hilbert space setting and the sequential use of familiar optimization (minimum norm) algorithms in a process that finds "feasible points" in the intersection of closed convex sets. Perfect tracking problems can be seen as feasible point problems and hence successive projection is very consistent with typical Iterative Learning Control tasks for linear systems. The choice of the term *"Proximity Algorithm"* used in this text reflects the practical realities that Iterative Control is only ever applied for a finite number of iterations and that *exact* tracking is un-achievable and, in reality, not explicitly required. If this is accepted, the asymptotic property of successive projection iterates $\{s_j\}_{j\geq 1}$ of being arbitrarily close to all sets

defining the Iterative Control problem can be a good mathematical representation of what is needed in practice.

One major structural assumption is that all sets $\{S_j\}_{1\leq j\leq N_S}$ are closed and convex. The second is that the intersection $\bigcap_{j=1}^{N_S} S_j$ is non-empty (that is, a solution to the tracking problem exists). With these assumptions, successive projection provides an alternative to operator theory and can be used to analyse algorithms such as the many forms of NOILC Algorithm 9.1, Auxiliary Optimization Algorithm 12.4 and Predictive NOILC Algorithm 10.9. The operator approach has the advantages of being a familiar parallel to transfer function analysis and providing more details on the form of the convergence, particularly when an assumed eigenstructure is present. It is, however, a linear analysis and, typically, neither allows constraints to be included nor does it easily reveal the many other algorithms that can be generated. In contrast, successive projection can be applied to these problems to reveal a wide variety of new results as,

1. The sequence of projections is chosen by iteratively projecting onto a sequence of sets $\{\tilde{S}_j\}_{j\geq 1}$ taken from the $\{S_j\}_{1\leq j\leq N_S}$. Different sequences will produce different algorithms with different convergence properties.
2. The definitions of the sets $\{S_j\}_{1\leq j\leq N_S}$ also affects their computational form and the complexity of the projection evaluations. In Iterative Control, the choices are often non-unique. This is true when input, output and auxiliary variable constraints play a role in defining the tracking task. The algorithm designer has a choice of associating the constraints with the dynamics such as in Conceptual Algorithm 12.4 (when projections are constrained dynamic optimization problems) or separating the constraints into another set (when the dynamical optimization calculations are unconstrained) to reveal algorithms such as Algorithms 12.5 and 12.6.

In their simplest form, the ideas allow useful extensions of NOILC to include constraints [26]. In this text, the ideas are taken further as in Algorithms 12.2 and 12.3. They also permit extensions of Intermediate Point control to include the shaping of output trajectories between intermediate points in the form of Algorithm 12.7 and extensions of auxiliary variable control algorithms by replacing optimization by a need to satisfy auxiliary variable bounds as in Algorithm 12.8. The reader will note that, by an appropriate definition of operators, sets and signals, other algorithms such as Multi-task Algorithms and variations in initial conditions (see Chap. 10) can be included.

The theory is based on the assumption that the tracking problems defined have at least one solution. Section 12.5 offers some insight into what could happen in such circumstances. The analysis of the case of $N_S > 2$ has infinite variability in the sequencing of the sets $\{S_j\}_{1\leq N_S}$ in $\{\tilde{S}_j\}_{j\geq 1}$ and an analysis has not been attempted here. The case of $N_S = 2$ is, however, simple enough to allow some analysis and suggests that, as a rule, iterations in multi-set analysis will remain bounded and oscillate between the sets in a way that can be quite complex.

Do not get carried away by the mathematical formulation however as, ultimately, the algorithms should be designed to be applied with at least some part of the computations being represented by on-line experiments and the use of measured data in

the next computational stage. That is, whatever the chosen algorithm structure, care should be taken to ensure that measurement data from iteration k does influence the choice of input for iteration $k+1$ for, without this property, the actual response of the plant to an applied input will, effectively be ignored. This is clearly undesirable, particularly in situations where modelling errors exist. The analysis of unconstrained algorithms in previous chapters suggests that successive projection can be a very robust approach to practice but, at this time, there is no formal theory available to describe this robustness or provide tests for its presence. A moment's thought by the reader will reveal the fact that modelling errors change the plant model G to, say, UG so that at least one of the sets is changed. That is, off-line computations are projecting onto the wrong set!

Finally, Sect. 12.4 introduces the idea of *"iteration management"* which has the potential, in the author's view, to transform Iterative Learning Control paradigms from automated, autonomous algorithms into decision support tools for operators. It introduces additional degrees of freedom into Iterative Control by adding, as iterations progress, the knowledge and experience of the user and his or her ability to see patterns in the data. The advantages and cost-effectiveness of such *intelligent intervention* should not be underestimated but, one supposes, it must be balanced against the consequent training requirements for operators.

Chapter 13
Acceleration and Successive Projection

The previous chapters have described and analysed many Iterative Control algorithms based on optimization principles. In the main, convergence was assured by monotonic reductions in the error norm combined with semi-quantitative rules for convergence rates based on experience of optimal control theory, the influence and choice of weight parameters and matrices (in norm definitions) and insights obtained using eigenvalue and/or frequency domain analysis. These rules introduce their own problems. Using, for example, NOILC Algorithm 9.1 as a model, rapid convergence is normally associated with low control weighting in the objective function. For state space systems, this leads to high gain state variable feedback and high gain feedback may not be advisable for the application considered. Therefore, despite the persuasive content of the rules, they provide trends rather than detail of likely outcomes and, following these trends to their natural conclusion can produce unacceptable control system characteristics.

The idea of creating rapidly convergent algorithms without undue need for undesirable control system characteristics (such as high gain controls), therefore, merits further research. Three approaches to the problem are presented, namely,

1. Saving on-plant-time and cost by off-line iterations using a plant model (Sect. 13.1).
2. Section 13.2, extends successive projection to improve convergence rates by adding in extrapolation (sometimes called over-relaxation) factors.
3. In Sect. 13.3, acceleration is achieved using successive projection and iteration dependent choice of parameterized sets $S_2(\sigma^2)$.

Both sections rely heavily on the material in Chaps. 9 and 12 as the underlying computations retain the NOILC computational framework at their core.

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_13

13.1 Replacing Plant Iterations by Off-Line Iterations

One simple acceleration mechanism is obtained by noting that *there is no need for every mathematical iteration to include experimental data collection*. In reality, only iterations that are implemented on the plant are normally counted by the user when costing the process in terms of time undertaking plant experimental work. Model-based, off-line iterations simply add to the computational burden but do not add to the cost. This idea applies to any iterative algorithm but, for NOILC Algorithm 9.1, details could be as follows,

1. implementation can take the form of partition of the iteration indices into two disjoint sets I_1 and I_2 and implementing the algorithm on the plant when $k \in I_1$ and using model-based computation when $k \in I_2$. The choices $I_1 = \{0, 2, 4, 6, 8, \ldots\}$ and $I_2 = \{1, 3, 5, 7, \ldots\}$ describes a situation where, following initialization with $k = 0$, the iteration is applied to the plant when k is even but is model-based (and off-line) when k is odd. Different partitions can be used if regarded as useful but, intuitively, the ratio of on-line to off-line iterations should not be too small if the plant is to have data input to the process.
2. Using the same notation, the off-line iterations could be based on different algorithms or simply use different parameters. For NOILC and state space systems, the control weighting ε^2 used when $k \in I_1$ could represent the need to have monotonic behaviours but avoid the use of on-line, high gain state feedback controls. For off-line iterations, the weight can be reduced to achieve faster error norm reduction. The reader will note the connection between this idea and the material in Sect. 11.4.
3. In both cases, the number of plant iterations required is reduced and acceleration is achieved. This is obvious for the first suggestion as, in the absence of modelling errors, the error update relationship $e_{k+1} = Le_k$ is unchanged. For the second, the change in on-line value ε^2 to a smaller, off-line value ε^2_{off} implies that errors are updated, alternately, as $e_{k+1} = L_1 e_k$ (off-line) and $e_{k+1} = L_0 e_k$ (on-line). The changing parameter is represented by the inequality $L_1 \leq L_0 \leq I$ and it follows that acceleration is achieved as $e_{k+2} = L_1 L_0 e_k$ for all $k \geq 0$ and $L_1 L_0 \leq L_0^2$.

There are many variations on the ideas expressed above, not all of which link easily to successive projection. The interested reader is invited to explore the possibilities.

13.2 Accelerating Algorithms Using Extrapolation

In this section, attention is focussed on successive projection algorithms with just $N_S = 2$ closed, convex sets $S_1 \subset H$ and $S_2 \subset H$ in a real Hilbert space H. In particular, it is assumed that S_1 is a linear variety. This assumption is natural as, typically, S_1 describes the dynamics of the plant in the absence of constraints.

13.2.1 Successive Projection and Extrapolation Algorithms

The following Algorithm defines the proposed extrapolation algorithm using the notation of Chap. 12. It covers the case of two closed, convex sets S_1 and S_2 where the set S_1 is a linear variety in a real Hilbert space H. The starting point is a point $s_0 \in S_2$ and successive projections are onto the sequence of sets $\{\tilde{S}_j\}_{j\geq 1}$ defined by the alternating sequence $\{S_1, S_2, S_1, S_2, S_1, \ldots\}$.

Algorithm 13.1 (*Successive Projection with Extrapolation Factors*) Suppose that $S_1 \cap S_2$ is non-empty. Then an accelerated successive projection algorithm for the construction of iterates that approach $S_1 \cap S_2$ arbitrarily closely is defined by choosing a starting point $s_0 \in S_2$ and constructing the sequence $\{s_j\}_{j\geq 0}$ with $s_j \in \tilde{S}_j, j \geq 1$, by solving, recursively, the optimization problems

$$\tilde{s}_{j+1} = \arg \min_{s\in\tilde{S}_{j+1}} \|s - s_j\|_H, \quad \textit{for } j \geq 0 \tag{13.1}$$

and defining new iterates s_{j+1} by the relations $s_1 = \tilde{s}_1$ and, more generally, for $j \geq 1$,

$$s_{j+1} = \begin{pmatrix} \tilde{s}_{j+1}, & \textit{if } \tilde{S}_{j+1} = S_2 \quad (j \textit{ odd}) \\ s_{j-1} + \lambda_{j+1}(\tilde{s}_{j+1} - s_{j-1}), & \textit{if } \tilde{S}_{j+1} = S_1 \quad (j \textit{ even}) \end{pmatrix} \tag{13.2}$$

where λ_{j+1} can be chosen to be any value in the range

$$1 \leq \lambda_{j+1} \leq \frac{\|s_j - s_{j-1}\|_H^2}{\|\tilde{s}_{j+1} - s_{j-1}\|_H^2}. \tag{13.3}$$

For all such sequences and for all choices of point $x \in S_1 \cap S_2$,

$$\begin{gathered} \|x - s_{2j-1}\|^2 \geq \|x - s_{2j+1}\|^2 + \underbrace{\lambda_{2j+1}}\, \|s_{2j} - s_{2j-1}\|_H^2, \quad \textit{for } j \geq 1, \\ (\textit{The Effect of the Extrapolation Factor } \lambda_{2j+1}) \end{gathered} \tag{13.4}$$

and

$$\|x - s_1\|_H^2 \geq \sum_{j=1}^{\infty} \lambda_{2j+1} \|s_{2j} - s_{2j-1}\|_H^2 \geq \sum_{j=1}^{\infty} \|s_{2j} - s_{2j-1}\|_H^2. \tag{13.5}$$

As a consequence,

$$\begin{gathered} \lim_{j\to\infty} \|s_{2j} - s_{2j-1}\|_H = 0, \quad \textit{so that} \quad \lim_{j\to\infty} \|s_{2j} - \tilde{s}_{2j+1}\|_H = 0 \\ \textit{and} \qquad \lim_{j\to\infty} \inf_{s\in S_2} \|s - s_{2j-1}\|_H = 0. \end{gathered} \tag{13.6}$$

That is,

1. the algorithm generates a sequence of iterates that, from Eq. (13.4), get closer to $S_1 \cap S_2$ and ultimately, Eq. (13.6), lie arbitrarily close to both S_1 and S_2. It is therefore a proximity algorithm.
2. If the user chooses the value $\lambda_{2j+1} = 1$ for all indices j, the algorithm is precisely NOILC Algorithm 9.1 as expressed in the form of Theorems 12.1 and 12.4.
3. Using a value $\lambda_{2j+1} > 1$ appears, from Eq. (13.4), to reduce the range within which $\|x - s_{2j+1}\|_H^2$ can sit as compared with that achieved by NOILC. This is the fact that supports the interpretation that using larger values of the extrapolation factor will tend to accelerate the iteration process. However,
4. in practice, small errors in $\tilde{s}_{2j+1}$ and s_{2j-1} could lead to larger errors in s_{2j+1}. That is, the algorithm is likely to be less robust than NOILC, particularly if λ_{2j+1} is large. In practice, therefore, it is advisable to limit it to a chosen maximum value $\lambda_{max} \geq 1$ even if larger values are allowed by the (error free) theory. That is, the range defined by Eq. (13.17) is replaced by the range

$$1 \leq \lambda_{j+1} \leq \min\{\frac{\|s_j - s_{j-1}\|_H^2}{\|\tilde{s}_{j+1} - s_{j-1}\|_H^2}, \lambda_{max}\}. \tag{13.7}$$

Note: The algorithm has very precise properties that are revealed very well by the general approach used. In the form described, the extrapolation is placed in the set S_1. The reader will note that, if both S_1 and S_2 are closed linear varieties, there will be two choices of set that can be used.

Proof of Algorithm Properties A demonstration that the algorithm is well-defined and has the stated properties now follows. The first step is to show that the value of λ_{j+1} is well defined. First assume that j is even so that $s_{j+1} \in \tilde{S}_{j+1} = S_1$. Next observe that, if the algorithm has not converged, $\|s_j - s_{j-1}\|_H \neq 0$ and it is then only necessary to show that $0 < \|\tilde{s}_{j+1} - s_{j-1}\|_H^2 \leq \|s_j - s_{j-1}\|_H^2$. If $\|\tilde{s}_{j+1} - s_{j-1}\|_H^2 = 0$ then $\tilde{s}_{j+1} = s_{j-1}$. Using the linear variety assumption for S_1, it follows that $(\tilde{s}_{j+1} - s_j) \perp (x - \tilde{s}_{j+1})$ for all $x \in S_1$. Using the fact that $\langle x - s_j, s_j - s_{j-1}\rangle_H \geq 0$ for all $x \in S_2$ then gives two expressions, satisfied for all $x \in S_1 \cap S_2$,

$$\langle x - s_j, s_j - s_{j-1}\rangle_H \geq 0 \quad \textit{and} \quad \langle x - s_{j-1}, s_{j-1} - s_j\rangle_H = 0 \tag{13.8}$$

so that $\|s_j - s_{j-1}\|_H^2 \leq 0$ which is impossible as $s_j \neq s_{j-1}$ by assumption. Therefore $\tilde{s}_{j+1} \neq s_{j-1}$ and the fact that $\lambda_{j+1} > 1$ then follows from the identity $\|s_j - s_{j-1}\|_H^2 = \|s_j - \tilde{s}_{j+1}\|_H^2 + \|\tilde{s}_{j+1} - s_{j-1}\|_H^2 > \|\tilde{s}_{j+1} - s_{j-1}\|_H^2$.
Next, let $x \in S_1 \cap S_2$ and consider, for $j \geq 1$,

$$\begin{aligned}\langle s_{2j+1} - s_{2j-1}, s_{2j-1} - x\rangle_H &= \lambda_{2j+1}\langle \tilde{s}_{2j+1} - s_{2j-1}, s_{2j-1} - x\rangle_H \\ &= \lambda_{2j+1}\langle \left(\tilde{s}_{2j+1} - s_{2j}\right) + s_{2j} - s_{2j-1}, s_{2j-1} - x\rangle_H \\ &= \lambda_{2j+1}\langle s_{2j} - s_{2j-1}, s_{2j-1} - x\rangle_H\end{aligned} \tag{13.9}$$

where the orthogonality of $\tilde{s}_{2j+1} - s_{2j}$ and $s_{2j-1} - x$ has been used. It follows that

$$\begin{aligned}\langle s_{2j+1} - s_{2j-1}, s_{2j-1} - x\rangle_H &= \lambda_{2j+1}\langle s_{2j} - s_{2j-1}, s_{2j-1} - s_{2j} + s_{2j} - x\rangle_H \\ &= -\lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2 \\ &\quad + \lambda_{2j+1}\langle s_{2j} - s_{2j-1}, s_{2j} - x\rangle_H \\ &\leq -\lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2.\end{aligned} \tag{13.10}$$

Writing

$$\begin{aligned}\lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2 &= \lambda_{2j+1}\left(\frac{\|s_{2j}-s_{2j-1}\|_H^2}{\|\tilde{s}_{2j+1}-s_{2j-1}\|_H^2}\right)\|\tilde{s}_{2j+1} - s_{2j-1}\|_H^2 \\ &\geq \lambda_{2j+1}^2\|\tilde{s}_{2j+1} - s_{2j-1}\|_H^2 = \|s_{2j+1} - s_{2j-1}\|_H^2\end{aligned} \tag{13.11}$$

then gives

$$\begin{aligned}\|s_{2j+1} - x\|_H^2 &= \|s_{2j-1} - x\|_H^2 + \|s_{2j+1} - s_{2j-1}\|_H^2 + 2\langle s_{2j+1} - s_{2j-1}, s_{2j-1} - x\rangle_H \\ &\leq \|s_{2j-1} - x\|_H^2 + \|s_{2j+1} - s_{2j-1}\|_H^2 - 2\lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2\end{aligned} \tag{13.12}$$

Re-arranging yields the required inequality (13.4) as

$$\begin{aligned}\|s_{2j-1} - x\|_H^2 &\geq \|s_{2j+1} - x\|_H^2 + \left(\lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2 - \|s_{2j+1} - s_{2j-1}\|_H^2\right) \\ &\quad + \lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2 \\ &\geq \|s_{2j+1} - x\|_H^2 + \lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2\end{aligned} \tag{13.13}$$

The remainder of the proof is an application of an induction argument and the condition $\lambda_{2j+1} \geq 1$. This yields Eq. (13.5) and hence Eq. (13.6). □

13.2.2 NOILC: Acceleration Using Extrapolation

Algorithm 13.1 combines the computations of NOILC Algorithm 9.1 (and all its subsequent variations) with a simple linear combination in S_1 parameterized by λ_{2j+1}. In structure it has some similarity to the use of relaxation methods but the fact that $\lambda_{2j+1} \geq 1$ links it more to extrapolation methodologies. The precise link with NOILC Algorithm 9.1 is obtained by generalizing Algorithm 12.3 to include extrapolation. The sets used are

$$S_1 = \{(e, u) : e = r - Gu - d\} \quad \textit{and} \quad S_2 = \{(e, u) : e = 0, u \in \Omega_u\}. \tag{13.14}$$

Projection of (e, u) onto S_2 is typically an off-line computation which, in the unconstrained case, yields $(0, u)$ so no computation is needed. The algorithm statement therefore can concentrate on the projection onto S_1 and takes the form described

below. Note that there is some change in the notation used to make Algorithm 13.1 match that used in NOILC studies in previous chapters.

Algorithm 13.2 (*NOILC with Extrapolation and Input Constraints*) Suppose that there exists an input $u \in \Omega_u$ that generates a zero tracking error. Then, using the notation and terminology of Algorithms 9.1 and 12.3, the Norm Optimal Iterative Learning Control algorithm (with extrapolation) generates a sequence of inputs $\{u_k\}_{k\geq 0}$ (and associated errors $\{e_k\}_{k\geq 0}$) by using the process,

STEP ONE (**Initialization**): Choose $u_0 \in \mathscr{U}$ and find the error response e_0 to generate the iterate $(e_0, u_0) \in S_1$. Then, for $k \geq 0$, undertake steps 2, 3, 4, 5 iteratively until the desired accuracy has been achieved.
STEP TWO (**Projection onto** S_2): Given the data $(e_k, u_k) \in S_1$, find the constrained input $u_k^{(1)}$ and hence the point $(0, u_k^{(1)}) \in S_2$ solving

$$u_k^{(1)} = \arg\min_{u\in\Omega_u} \|u - u_k\|^2_{\mathscr{U}}. \tag{13.15}$$

Note: For application to physical systems, it is important to ensure that plant response data is included in the next step. In the above, this requirement is included the construction of the plant error response $e_k^{(1)}$ to the input $u_k^{(1)}$.
STEP THREE: (**Projection onto** S_1) Project the data $(0, u_k^{(1)}) \in S_2$ onto S_1, using off-line or on-line calculations to find the minimum distance to S_1 expressed as the problem of finding the solution of the NOILC optimization problem

$$u_{k+1}^{(2)} = \arg\min_{u\in\mathscr{U}}\{J(u, u_k^{(1)}) : e = r - y, y = Gu + d\} \tag{13.16}$$

and the associated tracking error $e_{k+1}^{(2)}$. To ensure a link to plant data, the process should be driven by $e_k^{(1)}$ rather than $u_k^{(1)}$.
STEP FOUR (**Evaluation of an Extrapolation Factor**): Choose a value of extrapolation factor $\lambda(k+1) \geq 1$ in the range

$$1 \leq \lambda(k+1) \leq \frac{\|e_k\|^2_{\mathscr{Y}} + \varepsilon^2\|u_k^{(1)} - u_k\|^2_{\mathscr{U}}}{\|e_{k+1}^{(2)} - e_k\|^2_{\mathscr{Y}} + \varepsilon^2\|u_{k+1}^{(2)} - u_k\|^2_{\mathscr{U}}} \tag{13.17}$$

to produce the new control input

$$u_{k+1} = u_k + \lambda(k+1)\left(u_{k+1}^{(2)} - u_k\right). \tag{13.18}$$

STEP FIVE (**Error Measurement**): Compute the response e_{k+1} to u_{k+1} by, either using the off-line formula

$$e_{k+1} = e_k + \lambda(k+1)\left(e_{k+1}^{(2)} - e_k\right), \tag{13.19}$$

or using u_{k+1} on-line to find the measured tracking error. This step has then generated the data (e_{k+1}, u_{k+1}) and the procedure returns to STEP TWO.
The statement of and analysis following Algorithm 13.1 then proves that the tracking error converges to zero and the input signals becomes arbitrarily close to the constraint set Ω_u.

Algorithm 13.1 in the form of Algorithm 13.2 has potential value in accelerating NOILC algorithms without the need to use low input signal weighting in the objective function. This is true for linear, discrete or continuous, state space systems $S(A, B, C)$ where low control weights leads to high gain state feedback through the Riccati matrix solution of the optimization problem. For example, the inclusion of extrapolation in Algorithm reference 9.4 is a simple modification with the added calculation of $\lambda(k+1)$ and the extrapolation formulae for u_{k+1} and e_{k+1}. The signal $u_{k+1}^{(2)}$ is just, with a change in notation, the iterate u_{k+1} computed in Algorithm 9.4.

13.3 A Notch Algorithm Using Parameterized Sets

With the exception of the ideas of Iteration Management in Sect. 12.4, the discussion has, so far, assumed that the sets $S_1, S_2, S_3, \ldots, S_{N_S}$ are iteration independent. There is no mathematical reason why this should be the case but the choice of any variation in the definition of the sets will need detailed consideration and analysis. One set almost always describes the system dynamics. This section considers the choice of a companion set that is based on the idea of accelerating algorithm convergence by "annihilation" of part of the spectrum of GG^*. The notation of the NOILC Algorithm 9.1 is used for a system with dynamics $y = Gu + d$ and underlying spaces $\mathscr{Y}$ and $\mathscr{U}$, although the ideas apply more generally by suitable choice of G.

The system is required to track a reference $r \in \mathscr{Y}$ with tracking error $e = r - y$ equal to zero. Plant behaviours are associated with data points (e, u) in the product space $H = \mathscr{Y} \times \mathscr{U}$. H is a real Hilbert space with inner product

$$\langle (e, u), (w, v) \rangle_H = \langle e, w \rangle_{\mathscr{Y}} + \varepsilon^2 \langle u, v \rangle_{\mathscr{U}}, \quad \textit{where } \varepsilon^2 > 0. \tag{13.20}$$

13.3.1 Creating a Spectral Notch: Computation and Properties

The plant is identified with the set

$$S_1 = \{(e, u) : e = r - Gu - d\} \subset H, \quad (\textit{Plant Dynamics}). \tag{13.21}$$

The process considered here is that of creating a set $S_2 \subset H$, projecting a point $(e_0, u_0) \in S_1$ onto S_2 to create a point $(e_1^{(1)}, u_1^{(1)}) \in S_2$. This is then followed by

the projection of this point back onto S_1 to give a point $(e_1, u_1) \in S_1$. To be useful in practice, it is essential that e_1 has useful properties of monotonic norm reduction $\|e_1\|_{\mathscr{Y}} \leq \|e_0\|_{\mathscr{Y}}$ for all e_0. A stronger version of this requirement is that the norm reduction has the potential to be considerably greater than that achieved by NOILC Algorithm 9.1 for the given norm(s) and weight. The presentation takes the form of a constructive argument.

The set S_2 is parameterized by a single parameter $\sigma^2 > 0$ and denoted by

$$S_2(\sigma^2) = \{(\tilde{e}, u) : -\sigma^2\tilde{e} = r - Gu - d\} \subset H, \quad (\textit{Modified Dynamics}). \quad (13.22)$$

Writing $\tilde{e} = -\sigma^{-2}(r - Gu - d) = -\sigma^{-2}e$ identifies the signal $\tilde{e}$ as the error in tracking a signal $-\sigma^{-2}r$ with output defined by $-\sigma^{-2}Gu - \sigma^{-2}d$. That is, G is replaced by $-\sigma^{-2}G$ and d is replaced by $-\sigma^{-2}d$. If G is a state space model $S(A, B, C, D)$ with initial state $x(0) = x_0$, this change has two interpretations, namely that, either

1. the model $S(A, B, C, D)$ can be replaced by $S(A, B, -\sigma^{-2}C, -\sigma^{-2}D)$ with no change in x_0
2. or replaced by $S(A, -\sigma^{-2}B, C, D))$ if x_0 is replaced by $-\sigma^{-2}x_0$.

The first step is to suppose that $(e_0, u_0) \in S_1$ is given. The projection onto $S_2(\sigma^2)$ is then

$$\begin{array}{c}(e_1^{(1)}, u_1^{(1)}) = \arg\min_{(\tilde{e},\tilde{u})\in S_2(\sigma^2)} \left(\|\tilde{e} - e_0\|_{\mathscr{Y}}^2 + \varepsilon^2\|\tilde{u} - u_0\|_{\mathscr{U}}^2\right) \\ \textit{subject to the constraints} \quad \tilde{e} = -\sigma^{-2}(r - G\tilde{u} - d).\end{array} \quad (13.23)$$

This is simply a NOILC problem but it has two important interpretations, the first of which is

1. application of NOILC Algorithm 9.1 for the modified dynamics with the reference signal replaced by $-\sigma^{-2}r - e_0$.
2. The second interpretation is, again, that of NOILC Algorithm 9.1. Using a scaling factor of $(-\sigma^2)^2$ on the objective function, the optimization can be written as

$$\begin{array}{c}((-\sigma^2)e_1^{(1)}, u_1^{(1)}) = \arg\min_{(e,u)\in S_2(\sigma^2)} \left(\|e + \sigma^2 e_0\|_{\mathscr{Y}}^2 + \sigma^4\varepsilon^2\|u - u_0\|_{\mathscr{U}}^2\right) \\ \textit{subject to the constraints} \quad e = r - y \textit{ and } \quad y = Gu + d.\end{array} \quad (13.24)$$

This problem is precisely that of the application of NOILC to the original dynamics with reference replaced by $r + \sigma^2 e_0$ and weight parameter ε^2 replaced by $\sigma^4\varepsilon^2$. That is, the computation can be undertaken using the substitutions

$$\begin{array}{ll}\varepsilon^2 \mapsto \sigma^4\varepsilon^2 & (\textit{weight change}) \quad \textit{and} \\ r \mapsto r + \sigma^2 e_0 & (\textit{error adjusted reference})\end{array} \quad (13.25)$$

followed by scaling of the error resulting from $u_1^{(1)}$ by $-\sigma^{-2}$.

The two interpretations offer two alternative approaches to evaluation. The first is, in the form presented, off-line whilst the second could also be off-line but, in principle, could be on-line. This advantage in this case would be that the simple parameter change and modification to the reference is easily incorporated into the NOILC implementation. The first interpretation is used in the following analysis.

Projection onto S_2: The analysis of Chap. 9 indicates that

$$u_1^{(1)} = u_0 + \varepsilon^{-2}(-\sigma^{-2}G)^* \left(e_1^{(1)} - e_0\right) = u_0 - \sigma^{-2}\varepsilon^{-2}G^* \left(e_1^{(1)} - e_0\right). \quad (13.26)$$

The consequent "error" for the modified dynamics is then given by

$$\begin{aligned} e_1^{(1)} &= -\sigma^{-2}\left(r - Gu_1^{(1)} - d\right) \\ &= -\sigma^{-2}\left(e_0 + \sigma^{-2}\varepsilon^{-2}GG^*e_1^{(1)} - \sigma^{-2}\varepsilon^{-2}GG^*e_0\right) \\ \textit{so that} \quad e_1^{(1)} &= (\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)e_0. \end{aligned} \quad (13.27)$$

Projection onto S_1: Next compute the projection of $(e_1^{(1)}, u_1^{(1)}) \in S_2(\sigma^2)$ onto S_1 to give

$$\begin{aligned} (e_1, u_1) = \arg\min_{(e,u)\in S_1} &\left(\|e - e_1^{(1)}\|_{\mathscr{Y}}^2 + \varepsilon^2\|u - u_1^{(1)}\|_{\mathscr{U}}^2\right) \\ \textit{subject to the constraints} \quad & e = r - Gu - d. \end{aligned} \quad (13.28)$$

This is a NOILC problem with r replaced by $r - e_1^{(1)}$ and hence $u_1 = u_1^{(1)} + \varepsilon^{-2}G^*\left(e_1 - e_1^{(1)}\right)$. Substituting as required then gives

$$\begin{aligned} e_1 = r - Gu_1 - d &= \left(r - Gu_1^{(1)} - d\right) - \varepsilon^{-2}GG^*\left(e_1 - e_1^{(1)}\right) \\ &= -\sigma^2 e_1^{(1)} - \varepsilon^{-2}GG^*\left(e_1 - e_1^{(1)}\right) \\ \textit{so that} \quad e_1 &= \left(I + \varepsilon^{-2}GG^*\right)^{-1}\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)e_1^{(1)}. \end{aligned} \quad (13.29)$$

Combining the two projections hence states that the resultant errors e_1 and e_0 in S_1 are related by the linear mapping $e_1 = L(\sigma^2)e_0$ where the operator $L(\sigma^2) : \mathscr{Y} \to \mathscr{Y}$ is bounded and has the form

$$L(\sigma^2) = \left(I + \varepsilon^{-2}GG^*\right)^{-1}(\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^2 \quad (13.30)$$

where the fact that the terms commute has been used to simplify the expression.

Interesting observations about the form of $L(\sigma^2)$ include the following,

1. the operator has the form of the familiar operator $(I + \varepsilon^{-2}GG^*)^{-1}$ that describes the error evolution for NOILC Algorithm 9.1 modified by the multiplicative factor $\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^2(\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}$.

2. The factor $(\varepsilon^{-2}GG^* - \sigma^2 I)$ indicates that $e_1 = 0$ if $\varepsilon^{-2}GG^* e_0 = \sigma^2 e_0$. That is, if $\varepsilon^2\sigma^2$ is an eigenvalue of GG^* and e_0 is an associated eigenvector.

The second observation indicates one possible effect of the combined projection, namely that for some initial errors e_0, the algorithm "annihilates" the signal to produce a consequent error $e_1 = 0$. A more general statement of this property is as follows,

Theorem 13.1 (Approximate Annihilation Properties of $L(\sigma^2)$) *Using the notation and definitions of the preceding discussion, the operator $L(\sigma^2) : \mathscr{Y} \to \mathscr{Y}$ is self-adjoint and has the property that*

$$\begin{gathered} \gamma_1^2 \left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^2 \le L(\sigma^2) \le \gamma_2^2 \left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^2 \\ \text{with} \\ 0 < \gamma_1^2 = \frac{1}{(1+\varepsilon^{-2}\|G^*\|^2)(\sigma^4+\varepsilon^{-2}\|G^*\|^2)} \le \gamma_2^2 = \sigma^{-4}. \end{gathered} \tag{13.31}$$

As a consequence, $e_1 = 0$ if e_0 is an eigenvector of GG^* with eigenvalue $\varepsilon^2\sigma^2$ and, more generally, e_1 is arbitrarily small (relative to e_0) if e_0 has the property that $\|(\varepsilon^{-2}GG^* - \sigma^2 I)e_0\|_{\mathscr{Y}} \ll \|e_0\|_{\mathscr{Y}}$.

Proof $L(\sigma^2)$ is self-adjoint as it is a function of the self-adjoint operator GG^*. The inequality for $L(\sigma^2)$ follows from the properties seen in Theorem 9.1 applied to both $(I + \varepsilon^{-2}GG^*)^{-1}$ and $(\sigma^4 I + \varepsilon^{-2}GG^*)^{-1} = \sigma^{-4}(I + \sigma^{-4}\varepsilon^{-2}GG^*)^{-1}$. The eigenvector property then follows easily by writing the inequality in the form

$$\gamma_1^4 \left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^4 \le L^2(\sigma^2) \le \gamma_2^4 \left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^4 \tag{13.32}$$

so that $\|e_1\|^2 = \|L(\sigma^2)e_0\|^2 \le \gamma_2^4 \|(\varepsilon^{-2}GG^* - \sigma^2 I)^2 e_0\|^2$. The result now follows easily. □

The operator also has more detailed properties expressed as follows,

Theorem 13.2 (Bounds on $L(\sigma^2)$ and Monotonicity)

$$\begin{gathered} 0 \le L(\sigma^2) \le I, \quad ker[I - L(\sigma^2)] = ker[G^*] \\ \text{and} \\ \overline{\mathscr{R}[I - L(\sigma^2)]} = \overline{\mathscr{R}[G]} = ker[G^*]^{\perp} \end{gathered} \tag{13.33}$$

so that, if $e_1 = L(\sigma^2)e_0$, then $\|e_1\|_{\mathscr{Y}} \le \|e_0\|_{\mathscr{Y}}$ for all $e_0 \in \mathscr{Y}$. In particular, $\|e_1\|_{\mathscr{Y}} < \|e_0\|_{\mathscr{Y}}$ for all e_0 that do not lie in $ker[G^*]$.

Proof The proof that $L(\sigma^2) \ge 0$ follows from Theorem 13.1 as $(\sigma^2 I - \varepsilon^{-2}GG^*)^2 \ge 0$. Next, for simplicity, write $X = \varepsilon^{-2}GG^* \ge 0$ so that, as required,

$$L(\sigma^2) = I - (I + X)^{-1}(\sigma^4 I + X)^{-1}\left(1 + \sigma^2\right)^2 X \le I. \tag{13.34}$$

The same formula shows that $ker[I - L(\sigma^2)] = ker[X] = ker[G^*]$, a fact that, together with $ker[G^*]^{\perp} = \overline{\mathscr{R}[G]}$, also proves the correspondence of the closure of the ranges. It follows that $\|e_1\| \leq \|e_0\|$ in all cases. Finally, noting that $I - L(\sigma^2)^2 = X_0(I + L(\sigma^2))X_0$ where X_0 is a positive definite, self-adjoint, square root of $I - L(\sigma^2) \geq 0$, any situation where $\|e_1\|^2 = \|e_0\|^2$ corresponds to the case where $0 = \langle e_0, (I - L^2(\sigma^2))e_0\rangle_{\mathscr{Y}} \geq \|X_0 e_0\|^2_{\mathscr{Y}}$. This implies that $e_0 \in ker[X_0]$. The proof is now complete as $ker[X_0] = ker[I - L(\sigma^2)] = ker[G^*]$. □

In general therefore, *the two step projection process has properties of both error norm reduction and annihilation*. The detailed form of the annihilation property in a particular situation depends on the choice of $\sigma^2 > 0$. At first sight, this appears to improve on the properties of NOILC Algorithm 9.1 but there is no guarantee, for an arbitrary choice of e_0 that the error norm reduction achieved is greater than that achieved using NOILC. A guarantee can be provided as follows,

Theorem 13.3 (Guaranteeing Improvements on NOILC) *With the assumptions of Theorem 13.2,*

$$L(\sigma^2) \leq (I + \varepsilon^{-2}GG^*)^{-1} \quad \text{if} \quad \varepsilon^{-2}\|G^*\|^2 \leq 1 + 2\sigma^2. \tag{13.35}$$

This condition is satisfied for all $\sigma^2 > 0$ if $\varepsilon^{-2}\|G^*\|^2 \leq 1$, a condition that requires the weight ε^2 to be sufficiently large.

Note: If applied, the stated limitation imposed on the value of σ^2 ensures that the error norm resulting from the use of one iteration of NOILC from any initial error e_0 is greater than that achieved using the two step projection process considered in this section. If it is violated, the reader should be able to use an eigenvalue/eigenvector methodology to show that reductions will be achieved in all eigen-subspaces corresponding to eigenvalues of GG^* strictly less than $\varepsilon^2\left(1 + 2\sigma^2\right)$.

Proof Examination of $L(\sigma^2)$ indicates that it is only necessary to ensure that, with $X = \varepsilon^{-2}GG^*$, the operator $(\sigma^4 I + X)^{-1}(\sigma^2 I - X)^2 \leq I$. This is simply the condition $(\sigma^4 I + X) - (\sigma^2 I - X)^2 = X((1 + 2\sigma^2)I - X) \geq 0$ which is satisfied if $((1 + 2\sigma^2)I - X) \geq 0$. The result follows from the fact that $X \leq \|X\|I$ and the relation $\|X\| = \varepsilon^{-2}\|G^*\|^2$. □

An easily obtained insight into the underlying effects on the error is to note, using Theorem 13.2, that $L(\sigma^2)$ leaves errors in $ker[G^*]$ unchanged. In addition, it maps $\overline{\mathscr{R}[G]}$ into itself. Suppose, therefore that GG^* has strictly positive eigenvalues $\{\sigma_j^2\}_{j\geq 1}$ satisfying the order property $\|G^*\|^2 = \sigma_1^2 \geq \sigma_2^2 \geq \sigma_1^3 \geq \cdots$ and generating a complete set of orthonormal eigenvectors $\{v_j\}_{j\geq 1}$ spanning $\overline{\mathscr{R}[G]}$. Write $e_0 = \sum_{j\geq 1} \gamma_j v_j + e_0^{(2)}$ with $e_0^{(2)} \in ker[G^*]$ and suitable scalars $\{\gamma_j\}_{j\geq 1}$ satisfying $\sum_{j\geq 1} \gamma_j^2 < \infty$. Using the Spectral Mapping Theorem, the eigenvalues of $L(\sigma^2)$ take the values $\{f(\sigma_j^2, \sigma^2)\}_{j\geq 0}$ where

$$f(\mu, \sigma^2) = \frac{(\sigma^2 - \varepsilon^{-2}\mu)^2}{(1 + \varepsilon^{-2}\mu)(\sigma^4 + \varepsilon^{-2}\mu)} \tag{13.36}$$

and $0 \leq f(\mu, \sigma^2) \leq 1$ for all $\mu > 0$. In addition, it follows that

$$L(\sigma^2)e_0 = \sum_{j \geq 1} \underbrace{\gamma_j f(\sigma_j^2, \sigma^2)}\, v_j + e_0^{(2)}. \tag{13.37}$$

Note that,

1. each eigenvector v_j has its contribution to e_0 reduced by a factor of $f(\sigma_j^2, \sigma^2)$.
2. Choosing $\sigma^2 = \varepsilon^{-2}\mu$ and $\mu = \sigma_p^2$ then gives $f(\sigma_p^2, \sigma^2) = 0$ and the eigenvector v_p is eliminated from the resultant error e_1.
3. In a similar manner, all eigenvalues μ of GG^* that are close to $\varepsilon^2\sigma^2$ are "almost" eliminated from the resultant error e_1.

These properties are the motivation for the use of the word *"notch"* to describe the outcome of the projection process. Clearly the choice of σ^2 can have a benefit by eliminating, or almost eliminating, specific eigenvalue components of the error. This property is used in the next section as the basis of algorithm development and also used in Sect. 13.3.3 where the choice of σ^2 is related to approximate elimination of frequency components of the error.

As a final point in this section, the function $f(\mu, \sigma^2)$, on the interval $0 < \mu \leq \|G^*\|^2$ can be interpreted as a function shaping the spectrum of $L(\sigma^2)$ as a function of the spectrum of GG^*. It is useful to note that, even if the eigenvalues of GG^* are not known, $f(\mu, \sigma^2)$ offers the opportunity of assessing the effect on any eigenvalues at, or in the vicinity of, a chosen point μ. It is the factor by which the contributions of eigenvectors to the representation of e_0 are reduced by the two step "notch" process. The reader will be able to prove the following result using elementary algebra and calculus,

Theorem 13.4 (Properties of $f(\mu, \sigma^2)$ for $\mu \in [0, \infty)$) *Suppose that $\sigma^2 > 0$ is fixed. Then the continuous, differentiable function $f(\mu, \sigma^2)$ is positive for $\mu \in [0, \infty)$ and has the properties of being monotonically decreasing on the interval $[0, \varepsilon^2\sigma^2]$, monotonically increasing on $[\varepsilon^2\sigma^2, \infty)$ with a unique minimum at the point $\mu_0 = \varepsilon^2\sigma^2$. In addition,*

$$f(0, \sigma^2) = 1, \quad f(\varepsilon^2\sigma^2, \sigma^2) = 0, \quad \textit{and} \quad \lim_{\mu \to +\infty} f(\mu, \sigma^2) = 1. \tag{13.38}$$

The shaping of the function $f(\mu, \sigma^2)$ and its effect on error norm reduction is a design option. A possible approach to influencing the shape is

1. to choose the point μ_0 where f takes its minimum value of zero and hence to approximately annihilate the contribution of the spectrum of GG^* in the vicinity of this value.
2. As $\mu = \varepsilon^2\sigma^2$, choose ε^2 and σ^2 to satisfy the conditions of Theorem 13.3. This condition has the alternative form

$$\|G^*\|^2 \leq \varepsilon^2 + 2\mu_0. \tag{13.39}$$

This relationship provides a range of values for ε^2. In particular,

a. if a small value of $\varepsilon^2 \ll \|G^*\|^2$ is desired to ensure that the underlying NOILC algorithm produces a large error reduction, then μ_0 is bounded from below by a value close to $\frac{1}{2}\|G^*\|^2$. That is, annihilation can only be attempted for those parts of the spectrum of GG^* in the range $(\frac{1}{2}\|G^*\|^2, \|G^*\|^2]$.
b. Alternatively, if complete freedom to choose $\mu_0 > 0$ arbitrarily is preferred, ε^2 must be larger than $\|G^*\|^2$. That is, the price of this flexibility is that the underlying NOILC problem cannot reduce the error too much.

To illustrate the magnitude of the effects described, suppose that the value of ε^2 is guided by Theorem 13.3. For example, if the process is to improve on the norm reduction achieved by NOILC and eliminate any limitations on the choice of σ^2, choose $\varepsilon^2 = \|G^*\|^2$. The eigenvalues of $(I+\varepsilon^{-2}GG^*)^{-1}$ are then $\{(I+\varepsilon^{-2}\sigma_j^2)^{-1}\}_{j\geq 1}$ which lie on the curve $(1 + (\mu/\|G^*\|^2))^{-1}$, $0 \leq \mu \leq \|G^*\|^2$, a curve that reduces monotonically from the value of unity to the value 0.5 when $\mu = \|G^*\|^2$. The effect of the choice of σ^2 is illustrated in the following examples,

1. A notch at $\sigma^2 = 0.5$ then ensures that approximate annihilation of the eigenvalues σ_j^2 is achieved in the vicinity of $\mu = 0.5\|G^*\|^2$. Improvements on NOILC error reductions are achieved elsewhere. For example, examining the cases of $\mu = \|G^*\|^2$ and $\mu = 0.25\|G^*\|^2$ gives $f(\|G^*\|^2, 0.5) = f(0.25\|G^*\|^2, 0.5) = 0.1$. Using the monotonicity properties of f gives $0 \leq f(\mu, 0.5) \leq 0.1$ on the interval $0.25\|G^*\|^2 \leq \mu \leq \|G^*\|^2$. The corresponding range of reductions for NOILC is [0.5, 0.8] in that same eigenvalue range. The conclusion is that, when compared with NOILC, a substantial norm reduction is achieved over a wide range of eigenvalue values.
2. A notch at $\sigma^2 = 1$ gives values $f(\|G^*\|^2, 1) = 0$, $f(0.5\|G^*\|^2, 1) = 1/9$ and $f(0.25\|G^*\|^2, 1) = 9/25$. This illustrates, again, the improvement on NOILC. It also shows that the range of eigenvalues with substantial reductions in magnitude depends on the choice of σ^2.
3. A notch at $\sigma^2 = 2$ is too large to provide annihilation properties on any eigenvalue but error reduction benefits are still seen as illustrated by the computed values $f(\|G^*\|^2, 2) = 0.1$, $f(0.5\|G^*\|^2, 2) = 1/3$ and $f(0.25\|G^*\|^2, 2) = 49/85$.

13.3.2 The Notch Algorithm and Iterative Control Using Successive Projection

The two step process and described and the properties presented in the previous Sect. 13.3.1 are central to what follows. First note that the computations can be continued using the pair $(e_1, u_1) \in S_1$ to create an algorithm. That is, the computation can be repeated from this data point to create $(e_2^{(1)}, u_2^{(1)}) \in S_2$ and, from this, the data $(e_2, u_2) \in S_1$. However, the value of σ^2 used in this second application need not be

that used in the first! With this in mind, an Iterative Algorithm based on the concept of introducing an iteration dependent notch can be described as given below.

Algorithm 13.3 (*A Notch Algorithm with Iteration Dependent Notch*) Using the notation of Sect. 13.3.1, suppose that $r \in \mathscr{Y}$, $\varepsilon^2 > 0$ and that the iterative process is initiated by the choice of an input $u_0 \in \mathscr{U}$ that produces the data $(e_0, u_0) \in H = \mathscr{Y} \times \mathscr{U}$ from the plant. Then, an Iterative *Notch Algorithm* is defined by the process, for $k \geq 0$, of sequentially/iteratively using data (e_k, u_k) in the three step process defined by,

STEP ONE: Choose a value $\sigma^2 = \tilde{\sigma}_{k+1}^2 > 0$ to create a notch in the desired part of the spectrum of $L(\tilde{\sigma}_{k+1}^2)$.

STEP TWO: Use off-line computations to find the input $u_{k+1}^{(1)}$ and associated tracking error $e_{k+1}^{(1)}$ that solves the optimization problem

$$\begin{aligned}(e_{k+1}^{(1)}, u_{k+1}^{(1)}) = \arg\min_{(\tilde{e},\tilde{u})\in S_2(\tilde{\sigma}_{k+1}^2)} & \left(\|\tilde{e} - e_k\|_{\mathscr{Y}}^2 + \varepsilon^2 \|\tilde{u} - u_k\|_{\mathscr{U}}^2\right) \\ \textit{subject to the constraints} \quad & \tilde{e} = -\tilde{\sigma}_{k+1}^{-2}(r - G\tilde{u} - d).\end{aligned} \tag{13.40}$$

Note: This is the projection of (e_k, u_k) onto $S(\tilde{\sigma}_{k+1}^2)$ and is, simply, either

1. one step of the NOILC Algorithm 9.1 where the data (r, G, d) for the original plant is replaced by $(-\tilde{\sigma}_{k+1}^{-2} r - e_k, -\tilde{\sigma}_{k+1}^{-2} G, -\tilde{\sigma}_{k+1}^{-2} d)$ for the modified dynamics.
2. Alternatively, it is one step of NOILC for the original plant but with reference replaced by $r + \tilde{\sigma}_{k+1}^2 e_k$ and ε^2 replaced by $\tilde{\sigma}_{k+1}^4 \varepsilon^2$. In this case, the computations might be possible on-line.

STEP THREE: Use on-line or off-line computations to construct the data (e_{k+1}, u_{k+1}) (for use in the next iteration) as the solution of the optimization problem

$$\begin{aligned}(e_{k+1}, u_{k+1}) = \arg\min_{(e,u)\in S_1} & \left(\|e - e_{k+1}^{(1)}\|_{\mathscr{Y}}^2 + \varepsilon^2 \|u - u_{k+1}^{(1)}\|_{\mathscr{U}}^2\right) \\ \textit{subject to the constraints} \quad & e = r - Gu - d.\end{aligned} \tag{13.41}$$

Note: This is the projection of $(e_{k+1}^{(1)}, u_{k+1}^{(1)})$ onto S_1. It is one step of the NOILC Algorithm 9.1 for the original plant using the minimization of $J(u, u_{k+1}^{(1)})$ but where the reference signal r is replaced by $r - e_{k+1}^{(1)}$.

The algorithm produces an error evolution expressed as

$$e_{k+1} = L(\tilde{\sigma}_{k+1}^2) e_k, \quad \textit{so that} \quad e_k = \prod_{j=1}^{k} L(\tilde{\sigma}_j^2) e_0 \quad \textit{for } k \geq 1. \tag{13.42}$$

In particular, using Theorem 13.2, the algorithm generates a monotonically decreasing error sequence satisfying

$$\|e_{k+1}\|_{\mathscr{Y}} \leq \|e_k\|_{\mathscr{Y}} \quad \textit{for } k \geq 0 \quad \textit{with} \tag{13.43}$$

strict inequality holding if $e_k \neq 0$ does not lie in $ker[G^*]$. If $e_0 \in ker[G^*]$, then $e_k = e_0$ for all $k \geq 0$.

The detailed properties of the algorithm can be deduced in several ways. The simplest convergence result is constructed using the annihilation properties of the basic notch process.

Theorem 13.5 (Sequential Eigenvector Annihilation/Finite Convergence) *Suppose that GG^* has strictly positive eigenvalues $\{\sigma_j^2\}_{j\geq 1}$ satisfying the order property $\|G^*\|^2 = \sigma_1^2 \geq \sigma_2^2 \geq \sigma_1^3 \geq \cdots$ and generating a complete set of orthonormal eigenvectors $\{v_j\}_{j\geq 1}$ spanning $\overline{\mathscr{R}[G]}$. Then, every initial error e_0 can be written in the form $e_0 = \sum_{j\geq 1} \gamma_j v_j + e_0^{(2)}$ with $\sum_{j\geq 0} \gamma_j^2 < \infty$ and the term $e_0^{(2)} \in ker[G^*]$ identified as the orthogonal projection $P_{ker[G^*]}e_0$ of e_0 onto $ker[G^*]$. The Notch Algorithm 13.3 then produces the tracking errors, for $k \geq 1$,*

$$e_k = \prod_{p=1}^{k} L(\tilde{\sigma}_p^2)e_0 = \sum_{j\geq 1} \gamma_j \left(\prod_{p=1}^{k} f(\sigma_j^2, \tilde{\sigma}_p^2) \right) v_j + P_{ker[G^*]}e_0. \tag{13.44}$$

Let the indices $\{j_k\}_{k\geq 1}$ be a re-ordering of the non-zero eigenvalues of GG^* and set $\tilde{\sigma}_k^2 = \varepsilon^{-2}\sigma_{j_k}^2$. Then, Notch Algorithm 13.3

1. *generates a monotonically reducing sequence of error norms and convergent errors satisfying*

$$\lim_{k\to\infty} e_k = P_{ker[G^*]}e_0. \tag{13.45}$$

2. *If, in addition, $\mathscr{Y}$ is finite dimensional, then, this limit is achieved in a finite number of iterations.*

Note: The finite convergence theoretically possible in this case is a parallel to Algorithm 7.4 presented in Sect. 7.2.2. Discrete state space systems satisfy the finite dimensionality assumption as does the Intermediate Point control problem (Sect. 10.5).

Note: The formula for e_k provides useful insight into the choices of $\tilde{\sigma}_j^2$ and underlines the rapid convergence that is possible if they are selected carefully.

Proof Every initial error e_0 can be written in the form $e_0 = \sum_{j\geq 1} \gamma_j v_j + e_0^{(2)}$ with the term $e_0^{(2)} \in ker[G^*]$ identified as the orthogonal projection of e_0 onto $ker[G^*]$. The formula for e_k then follows from the eigen-properties $L(\sigma_j^2, \sigma^2)v_j = f(\sigma_j^2, \sigma^2)v_j, j \geq 1$. Let $\delta > 0$ be arbitrary. As $\sum_{j\geq 1} \gamma_j^2 < \infty$, it is possible to choose an integer $N_1(\delta)$ such that $\sum_{j>N_1(\delta)} \gamma_j^2 < \delta$. In addition, for all sufficiently large integers $k \geq N_2(\delta)$ (say), the values $\varepsilon^{-2}\sigma_j^2$, $1 \leq j \leq N_1(\delta)$ have been used and the contribution of the associated eigenvectors to the error has been annihilated. The remaining error has norm $\|e_k\|_{\mathscr{Y}} \leq \delta$ for all $k \geq N_2(\delta)$. That is, $\limsup_{k\to\infty} \|e_k - e_0^{(2)}\|_{\mathscr{Y}} \leq \delta$ and convergence to zero follows as δ can be arbitrarily small. This process is finite if $\mathscr{Y}$ is finite dimensional. □

In practice, of course, the eigenvalues of GG^* will not be known and hence the exact annihilation property cannot be used. This does not prevent the successful application of Algorithm 13.3 as indicated by the following result that uses the information provided by Theorem 13.3.

Theorem 13.6 (Convergence Using a Limited Bandwidth) *Using the notation of Algorithm 13.3, suppose that, following N_0 iterations of unconstrained choice, the remaining values of $\tilde{\sigma}_{k+1}^2$ satisfy the condition*

$$1 + 2\tilde{\sigma}_{k+1}^2 \geq \varepsilon^{-2}\|G^*\|^2 \quad \textit{for all} \quad k \geq N_0 \ (\textit{say}). \tag{13.46}$$

Then, Notch Algorithm 13.3 generates a monotonically reducing sequence of error norms satisfying $\lim_{k\to\infty} e_k = P_{ker[G^*]}e_0$. In particular,

1. *if $N_0 = 0$, the error norm sequence reduces at a rate faster than that of NOILC Algorithm 9.1.*
2. *More generally, the algorithm converges faster than NOILC from the initial error e_{N_0}.*

Proof First note that the projection $P_{ker[G^*]}e_k = P_{ker[G^*]}e_0$ and, using Theorem 13.2, the error norm sequence is monotonically reducing for all k. It is possible to assume, without loss of generality, that $N_0 = 0$. Using Theorem 13.3 then gives $L(\tilde{\sigma}_{k+1}^2) \leq (I + \varepsilon^{-2}GG^*)^{-1}$ so that

$$\prod_{j=1}^{k} L(\tilde{\sigma}_j^2) \leq (I + \varepsilon^{-2}GG^*)^{-k} \tag{13.47}$$

Convergence properties of the Notch Algorithm then follow from those of Algorithm 9.1. □

The result allows situations where choices for the first iterations are flexible but, asymptotically, it requires emphasis on the interval $1 + 2\sigma^2 \geq \varepsilon^{-2}\|G^*\|^2$ which, if $\varepsilon^{-2}\|G^*\|^2 > 1$ does not permit "small" values of σ^2 to be used. This will limit the options available to influence convergence rates as part of the spectrum cannot be annihilated. In applications, two strategies will be required namely one to choose the first N_0 values, the second stage being the systematic selection of $\tilde{\sigma}_{k+1}^2$ satisfying $1 + 2\tilde{\sigma}_{k+1}^2 \geq \varepsilon^{-2}\|G^*\|^2$. For example

1. If $N_0 = 0$, the first stage is not needed. If $N_0 > 0$, a selection of values that covers a wide range of the spectrum is covered by the equally (linearly or logarithmically) spaced points. For example,

$$\sigma^2 = \varepsilon^{-2}\|G^*\|^2 \left(\frac{j}{N_0}\right), 1 \leq j \leq N_0. \tag{13.48}$$

The reader should note that, if N_0 is sufficiently large, the notch properties suggest that the contribution to e_{N_0} of eigenvalues in, for example, the range $N_0^{-1}\|G^*\|^2 \leq \mu \leq \|G^*\|^2$ will have been greatly reduced.

2. For $k \geq N_0$, there may be many more iterations to do. Although it is permitted to use values of $\sigma^2 > \varepsilon^{-2}\|G^*\|^2$, a focus on the interval of choice $[0.5(\varepsilon^{-2}\|G^*\|^2 - 1), \varepsilon^{-2}\|G^*\|^2]$ still leaves an infinity of choices including, for example,
   a. choosing an iteration independent value $\tilde{\sigma}_{k+1}^2 = \sigma^2$ in that range,
   b. choices randomly generated by a uniform pseudo-random number generator,
   c. choices selected from N_1 equally spaced points in the interval, or
   d. choices guided by "expert" knowledge of the plant and its observed behaviour (see Sect. 13.3.3 and the proposed use of frequency domain criteria).

Finally, with an iteration independent notch and a reference that can be tracked by some input signal, it is left as an exercise for the reader to prove the existence of a limit for the input sequence in the norm topology and, in addition, that the increased convergence rate does not influence the nature of this limit which is exactly the minimum energy/minimum norm solution obtained using NOILC without the notch modification.

13.3.3 A Notch Algorithm for Discrete State Space Systems

Notch Algorithm 13.3 applies quite generally and, being based on NOILC Algorithm 9.1, computations can be applied in any situation where NOILC, and its derivative algorithms, can be applied. It is expected that the most commonly used model in many applications is that of a state space model $S(A, B, C)$ with a specified initial condition $x(0) = x_0$. In this section, m-output, ℓ-input, linear, discrete state space models, operating on an interval $0 \leq t \leq N$, are considered. Algorithm 13.3 then has a very specific realization.

13.3.3.1 Algorithm Statement

More precisely, if $Q(t), 0 \leq t \leq N$ and $R(t), 0 \leq t \leq N$ are symmetric, positive definite weighting matrices,

Algorithm 13.4 (*A Notch Algorithm for Discrete State Space Systems*) Using the notation of Algorithm 13.3, suppose that the reference r has been specified and the iterative process initiated by the choice of u_0 that produces the data (e_0, u_0) from the plant. Then, an Iterative Notch Algorithm for the linear, discrete, state space system $S(A, B, C)$ is defined by the process, for $k \geq 0$, of sequentially using data (e_k, u_k) in the three step process,

STEP ONE: Choose a value $\sigma^2 = \tilde{\sigma}_{k+1}^2 > 0$ to create the desired notch for iteration $k + 1$.

STEP TWO: Use off-line computations to find the input time series $u_{k+1}^{(1)}$ and associated tracking error $e_{k+1}^{(1)}$ that minimizes the objective function

$$\begin{array}{c}\sum_{j=0}^{N}\ \left((\tilde{e}(t)-e_k(t))^T Q(t)(\tilde{e}(t)-e_k(t))+\varepsilon^2(\tilde{u}(t)-u_k(t))^T R(t)(\tilde{u}(t)-u_k(t))\right)\\ \textit{subject to the dynamic constraints}\quad -\tilde{\sigma}_{k+1}^2\tilde{e}=r-G\tilde{u}-d.\end{array} \tag{13.49}$$

For state space computations, the constraints can be written in the form $\tilde{e}(t) = -\tilde{\sigma}_{k+1}^{-2} r(t) - \tilde{y}(t)$, $0 \le t \le N$, where $\tilde{y}(t)$, $0 \le t \le N$, is the response of the system $S(A, B, -\tilde{\sigma}_{k+1}^{-2} C)$ to $\tilde{u}(t)$ from the initial condition $-\tilde{\sigma}_{k+1}^{-2} x_0$.

Note: As noted in Algorithm 13.3 there are two possible ways to approach this problem.

1. The approach described above is one step of the NOILC Algorithm 9.1 (perhaps in the form of Algorithm 9.4) from the input data u_k but where the data (r, C, x_0) for the original plant are replaced by $(-\tilde{\sigma}_{k+1}^{-2} r - e_k, -\tilde{\sigma}_{k+1}^{-2} C, x_0)$. This data change will, for example, change the form of both the Riccati equation and the equation for the predictive term ζ_{k+1} used in the computations.
2. The second approach uses the NOILC Algorithm 9.1 for the original plant model and initial condition, using the reference $r(t) + \tilde{\sigma}_{k+1}^2 e_k(t)$ and replacing ε^2 by $\tilde{\sigma}_{k+1}^4 \varepsilon^2$.

STEP THREE: Use on-line or off-line computations to construct the data (e_{k+1}, u_{k+1}) (for use in the next iteration) as the solution that minimizes the objective function

$$\begin{array}{l}\sum_{j=0}^{N}\ ((e(t)-e_{k+1}^{(1)}(t))^T Q(t)(e(t)-e_{k+1}^{(1)}(t))\\ \qquad\qquad +\varepsilon^2(u(t)-u_{k+1}^{(1)}(t))^T R(t)(u(t)-u_{k+1}^{(1)}(t)))\end{array} \tag{13.50}$$

subject to the constraints that $e(t) = r(t) - y(t)$, $0 \le t \le N$, where $y(t)$, $0 \le t \le N$, is the response of the system $S(A, B, C)$ to $u(t)$ from the initial condition x_0.

Note: This is one step of the NOILC Algorithm 9.1 from the data $u_{k+1}^{(1)}$ for the original plant but where the reference signal r is replaced by $r - e_{k+1}^{(1)}$. This data change requires only one Riccati equation but the equation for the predictive term ζ_{k+1} is driven by the input $u^{(1)}(t)$ and the "error corrected reference" $r - e_{k+1}^{(1)}$.

The algorithm has the monotonicity and convergence properties described by Algorithm 13.3 and illustrated by Theorems 13.5 and 13.6 and other material in Sect. 13.3.1 and 13.3.

The implementation of the algorithm is very similar to that of Norm Optimal Control and an understanding of that algorithm is essential. Together with Q and R, the choice of $\tilde{\sigma}_{k+1}^2$ is central to the prediction and achievement of significantly improved convergence rates.

13.3.3.2 Notch Values and the Frequency Domain

In what follows, it is seen that the choice of $\tilde{\sigma}^2_{k+1}$ can be supported by the frequency domain properties of the signal e_k. More precisely, suppose that the system $S(A, B, C)$ is asymptotically stable and that the matrices $\{Q(t)\}_{0\le t\le N}$ and $\{R(t)\}_{0\le t\le N}$ are independent of "t". In Sect. 7.2.3, an approximate eigenvector property was derived in terms of (complex) vectors $W_j(z_p)$ associated with the positive, real eigenvalues $\sigma_j^2(z_p)$ of the matrix $G(z)R^{-1}G^T(z^{-1})Q$ at frequencies $z = z_p = e^{(2\pi ip)/(N+1)}$, $0 \le p \le N$. These relationships are precisely those required by Theorem 13.1, as, for large values of N, it follows that $\|(\sigma_j^2(z)I - GG^*)W_j(z)\| \ll \|W_j(z)\|$ for all z on the unit circle in the complex plane. As a consequence, it is concluded that the choice of a "frequency" $z_{c,k}$ on the unit circle followed by setting

$$\tilde{\sigma}^2_{k+1} = \varepsilon^{-2}\sigma_j^2(z_{c,k}) \quad \textit{on each iteration} \quad (\textit{for } k \ge 0) \tag{13.51}$$

will have the effect of annihilating, approximately, the frequency content of the error at the frequency $z = z_{c,k}$ in the subspace generated by the eigenvector $w_j(z_{c,k})$ of $G(z)R^{-1}G^T(z^{-1})Q$ corresponding to the eigenvalue $\sigma_j^2(z_{c,k})$. Of course, for multi-output systems, other parts of the frequency content (corresponding to eigenvalues of $G(z)R^{-1}G^T(z^{-1})Q$ at other frequencies that are equal to $\tilde{\sigma}^2_{k+1}$) will also be suppressed. The notch is hence a notch based on "gain" values rather than individual frequencies. This interpretation is simplified for SISO systems as $w_j(z_{c,k}) = 1$ and the associated eigenvalue is exactly $G(z_{c,k})R^{-1}G^T(z_{c,k}^{-1})Q = QR^{-1}\left|G(z_{c,k})\right|^2$. This links the choice to the frequency domain and the values of Q and R in a familiar way but does not specify a suitable value of $z_{c,k}$. Intuitively, a good choice might be either

1. a frequency, generated from an analysis of e_k, representing a frequency range where error magnitudes are large or
2. a frequency representing physical phenomena that are ideally suppressed rapidly.
3. Theorem 13.6 may also play a role in limiting the choice. That is, if better convergence rates than those achieved by the NOILC Algorithm are required, the choice of $z_{c,k}$ is limited to a frequency range defined, approximately, by values satisfying

$$\sup_{|z|=1} |G(z)|^2 \le \varepsilon^2 RQ^{-1} + 2|G(z)|^2. \tag{13.52}$$

Finally, in theoretical terms, the frequency analysis of e_k has used the supervector description and is based on representations of the error as a finite summation of the basis vectors $\{W_j(z_k)\}$. In practice, it is more likely to be assessed using the familiar Fast Fourier Transform (FFT).

13.3.4 Robustness of the Notch Algorithm in Feedforward Form

Robustness of Notch Algorithm 13.3 is an important issue. Intuitively, it is expected to be less robust than NOILC Algorithm 9.1 as modelling errors will inevitably reduce the precision and effectiveness of the annihilation that can be achieved. Given a measured error signal e_k on iteration k, let $y = Gu + d$ be a model of the plant but suppose that the plant has a modelling error represented by a linear, bounded, multiplicative modelling error U. Off-line feedforward computations using this model ignore this error and produce the data

$$\begin{aligned}
u^{(1)}_{k+1} &= u_k - \sigma^{-2}\varepsilon^{-2}G^*\left(e^{(1)}_{k+1} - e_k\right),\\
e^{(1)}_{k+1} &= (\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)e_k,\\
u_{k+1} &= u^{(1)}_{k+1} + \varepsilon^{-2}G^*\left(e_{k+1} - e^{(1)}_{k+1}\right) \quad \textit{and predicted error}\\
e_{k+1} &= L(\sigma^2)e_k = \left(I + \varepsilon^{-2}GG^*\right)^{-1}(\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^2 e_k
\end{aligned} \tag{13.53}$$

Simple algebra then indicates that the implemented input signal takes the form

$$\begin{aligned}
u_{k+1} &= u_k + \varepsilon^{-2}G^*(I + \varepsilon^{-2}GG^*)^{-1}\Gamma(GG^*, \sigma^2)e_k\\
&\textit{where}\\
\Gamma(GG^*, \sigma^2) &= (1+\sigma^2)^2(\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}\\
\textit{and}\quad L(\sigma^2) &= I - (I - L)\Gamma(GG^*, \sigma^2) \quad \textit{where} \quad L = (I + \varepsilon^{-2}GG^*)^{-1}
\end{aligned} \tag{13.54}$$

is the familiar operator seen in NOILC (Chap. 10). Note that both $\Gamma(GG^*, \sigma^2) : \mathscr{Y} \to \mathscr{Y}$ and $\Gamma(G^*G, \sigma^2) : \mathscr{U} \to \mathscr{U}$ are self-adjoint and have bounded inverses.

Left Multiplicative Modelling Errors: Suppose that σ^2 is iteration independent and that U is a *left* multiplicative modelling error. It then follows that the error seen on the plant following application of u_{k+1} will be

$$e_{k+1} = (I - \varepsilon^{-2}UGG^*(I + \varepsilon^{-2}GG^*)^{-1}\Gamma(GG^*, \sigma^2))e_k = (I - U(I - L(\sigma^2)))e_k. \tag{13.55}$$

With a simple change in notation, this is just the expression seen in Sect. 9.2.4 with L replaced by $L(\sigma^2)$. Using the same approach as that section, an inner product on $\overline{\mathscr{R}[G]}$ can be defined to be

$$\langle e, w\rangle_{\sigma^2} = (1+\sigma^2)^{-2}\langle e, (I - L(\sigma^2))w\rangle_{\mathscr{Y}} \tag{13.56}$$

where the term $(1+\sigma^2)^{-2}$ is introduced to scale the inner product and associated norm. It has the benefit of simplifying the expression and plays only a minor role in the following examination of inequalities. Note that

Theorem 13.7 (Topological Equivalence of Norms) *With the definitions given above, the norm* $\|e\|_{\sigma^2} = \sqrt{(1+\sigma^2)^{-2}\langle e, (I - L(\sigma^2))w\rangle_{\mathscr{Y}}}$ *is topologically equivalent to* $\|e\|_0 = \sqrt{\langle e, (I-L)e\rangle_{\mathscr{Y}}}$ *where* $L = (I + \varepsilon^{-2}GG^*)^{-1}$ *as*

$$\frac{1}{\sigma^4 + \varepsilon^{-2}\|G^*\|^2}(I - L) \leq (1+\sigma^2)^{-2}(I - L(\sigma^2)) \leq \sigma^{-4}(I - L) \tag{13.57}$$

and hence, for any $e \in \mathscr{Y}$,

$$\left(\frac{1}{\sigma^4 + \varepsilon^{-2}\|G^*\|^2}\right)\|e\|_0^2 \leq \|e\|_{\sigma^2}^2 \leq \sigma^{-4}\|e\|_0^2 \tag{13.58}$$

Note: As consequence, convergence or boundedness in $\overline{\mathscr{R}[G]}$ *with respect to one norm implies that using the other norm.*

It is now easily shown that,

Theorem 13.8 (A Robustness Result for the Notch Algorithm with Constant σ^2) *Suppose that* $\tilde{\sigma}_{k+1}^2 = \sigma^2$ *is iteration independent. Then, Theorem 9.15, with L replaced by* $L(\sigma^2)$, *remains valid for the Notch Algorithm 13.3 if the norm* $\|\cdot\|_0$ *used is replaced by*

$$\|e\|_{\sigma^2}^2 = (1+\sigma^2)^{-2}\langle e, (I - L(\sigma^2))e\rangle_{\mathscr{Y}}, \quad e \in \mathscr{Y}. \tag{13.59}$$

More precisely, if there exists a real number $\varepsilon_0^2 > 0$ *such that, expressed in terms of the original topology in* $\mathscr{Y}$,

$$U + U^* \geq U^*(I - L(\sigma^2)U + \varepsilon_0^2 I \quad on \quad \mathscr{R}[G], \tag{13.60}$$

then ,

1. *the monotonicity condition* $\|e_{k+1} - e_0^{(2)}\|_{\sigma^2} \leq \|e_k - e_0^{(2)}\|_{\sigma^2}$ *is satisfied for all* $k \geq 0$.
2. *In addition, Theorem 13.7 then indicates that the sequence* $\{\|e_k\|_0\}_{k\geq 0}$ *is bounded. It converges to zero if, and only if, the sequence* $\{\|e_k\|_{\sigma^2}\}_{k\geq 0}$ *converges to zero.*

Simplifications are possible by noting that,

$$I - L(\sigma^2) = \Gamma(GG^*, \sigma^2)(I - L) \leq \frac{(1+\sigma^2)^2}{\sigma^4}(I - L) \tag{13.61}$$

so that a sufficient condition for robustness is that

$$U + U^* \geq \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right)U^*(I - L)U + \varepsilon_0^2 I \quad on \quad \mathscr{R}[G]. \tag{13.62}$$

which is the condition seen in Theorem 9.15 with an additional scaling factor of $\frac{(1+\sigma^2)^2}{\sigma^4} > 1$. This expression immediately suggests that the Notch Algorithm 13.3 is likely to be less robust than NOILC and that this robustness probably reduces as σ^2 gets smaller. Use of the notch algorithm will therefore be a balance between the benefits of the improved convergence rates and annihilation properties as measured against any loss in robustness if modelling errors are thought to be large.

Further simplifications of the robustness criterion follow directly by defining

$$\beta_I = \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right)\left(\frac{\varepsilon^{-2}\|G^*\|^2}{(1+\varepsilon^{-2}\|G^*\|^2)}\right) \quad \textit{and} \quad \beta_G = \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right)\varepsilon^{-2}, \tag{13.63}$$

from which, using the same norm definitions, it follows that

Theorem 13.9 (A Simplified Robustness Result for the Notch Algorithm) *Suppose that $\tilde{\sigma}^2_{k+1} = \sigma^2$ is iteration independent. Then, condition Four of Theorem 9.16, with β_I and β_G defined as above and with $\|\cdot\|_0$ replaced by $\|\cdot\|_{\sigma^2}$, remains valid as a predictor of robustness for the Notch Algorithm 13.3.*

In particular,

1. The parameters β_I and β_G depend on the choice of Q, R, ε^2 and σ^2 and reduce to values used in Theorem 9.16 as $\sigma^2 \to \infty$.
2. For application to discrete state space models, the results are easily translated into frequency domain criteria simply by using the new definitions of parameters. The details are left as an exercise for the reader and should be related to Theorems 9.17 and 9.18.

The assumption that σ^2 is iteration independent can be relaxed by noting that

$$(1+\sigma^2)^{-2}(I - L(\sigma^2)) - (1+\mu^2)^{-2}(I - L(\mu^2)) \;\geq\; 0 \quad \textit{if} \quad \mu^2 \geq \sigma^2 \tag{13.64}$$

so that $\|e\|_{\sigma^2}$ reduces, monotonically, as σ^2 increases.

Theorem 13.10 (Robustness using Monotonic Values of $\tilde{\sigma}^2_{k+1}$) *With the notation of the discussion above, assume that the sequence $\{\tilde{\sigma}^2_{k+1}\}_{k\geq 0}$ is monotonically increasing from an initial value $\tilde{\sigma}^2_1 > 0$ and bounded by the value $\tilde{\sigma}^2_\infty < \infty$. Then, if*

$$U + U^* \geq \left(\frac{(1+\tilde{\sigma}^2_1)^2}{\tilde{\sigma}^4_1}\right) U^*(I-L)U + \varepsilon_0^2 I \quad \textit{on} \quad \mathscr{R}[G], \tag{13.65}$$

the error sequence generated by the Notch Algorithm 13.3 satisfies the monotonicity condition

$$\|e_{k+1}\|^2_{\tilde{\sigma}^2_{k+2}} \leq \|e_{k+1}\|^2_{\tilde{\sigma}^2_{k+1}} \leq \|e_k\|^2_{\tilde{\sigma}^2_{k+1}}, \quad \textit{for} \quad k \geq 0. \tag{13.66}$$

In particular, there exists a real number $E_\infty \geq 0$ *such that*

$$\lim_{k\to\infty} \|e_k\|_{\tilde{\sigma}_{k+1}^2} = E_\infty \quad \textit{and} \quad \limsup_{k\to\infty} \|e_k\|_0 \leq E_\infty \sqrt{\sigma_\infty^4 + \varepsilon^{-2}\|G^*\|^2} \tag{13.67}$$

and hence $\lim_{k\to\infty} \|e_k\|_0 = 0$ *if* $E_\infty = 0$.

Proof The proof follows from the discussion that precedes the theorem and the observation that the condition (13.65) applied to iteration $k+1$ gives, using Theorem 13.8 with $\sigma^2 = \tilde{\sigma}_{k+1}^2 \geq \tilde{\sigma}_1^2$, $\|e_{k+1}\|^2_{\tilde{\sigma}_{k+1}^2} \leq \|e_k\|^2_{\tilde{\sigma}_{k+1}^2}$. The result is then easily proved using the monotonicity assumption to give $\|e_{k+1}\|^2_{\tilde{\sigma}_{k+2}^2} \leq \|e_{k+1}\|^2_{\tilde{\sigma}_{k+1}^2}$. The existence of the limits then follows with the bounds on $\|e_k\|_0$ being deduced using Theorem 13.7. □

The monotonicity assumption for $\{\tilde{\sigma}_j^2\}_{j\geq 1}$ is a useful assumption for analysis as it adds a structure that produces a simple argument. For state space systems, Sect. 13.3.3, it can be interpreted as choosing values of $\varepsilon^2\tilde{\sigma}_j^2$ that sweep a range of eigenvalue magnitudes from a defined minimum $\tilde{\sigma}_1^2$ to some defined maximum $\tilde{\sigma}_\infty^2$. As this maximum increases, the closer the asymptotic behaviour and robustness will be to that observed in NOILC. If $\tilde{\sigma}_1^2$ is too small, then robustness in the initial iterations may be compromised, an observation that can be interpreted as a warning that the notch procedure should not be used to "annihilate" high frequency components of the error.

Right Multiplicative Modelling Errors: Again assume that $\tilde{\sigma}_{k+1}^2 = \sigma^2$ is independent of iteration. Suppose also that U is now a right multiplicative modelling error so that, in a feedforward implementation, the observed error evolution is simply $e_{k+1} = (I - \varepsilon^{-2}GUG^*L\Gamma(GG^*, \sigma^2))e_k$. The techniques used in Sect. 9.2.2 for analysis of right multiplicative perturbations can now be used. The key to the analysis is to note that $L\Gamma(GG^*, \sigma^2) = \Gamma(GG^*, \sigma^2)L$ is self-adjoint and strictly positive and to use the inner product in $\mathscr{Y}$ defined by

$$\langle e, w\rangle_{\sigma^2} = (1+\sigma^2)^2\langle e, L\Gamma(GG^*, \sigma^2)w\rangle_{\mathscr{Y}}. \tag{13.68}$$

The norm induced by this inner product is equivalent to the norm $\|\cdot\|_0$ used in Sect. 9.2.2 as

$$\begin{array}{c}\left(\frac{1}{\sigma^4+\varepsilon^{-2}\|G^*\|^2}\right) L \leq (1+\sigma^2)^{-2}L\Gamma(GG^*, \sigma^2) \leq \sigma^{-4}L \quad \textit{implies that} \\ \left(\frac{1}{\sigma^4+\varepsilon^{-2}\|G^*\|^2}\right) \|e\|_0^2 \leq \|e\|_{\sigma^2}^2 \leq \sigma^{-4}\|e\|_0^2.\end{array} \tag{13.69}$$

With this definition, the analysis of Sect. 9.2.2 can be used and leads easily to the following result

Theorem 13.11 (A Robustness Result for the Notch Algorithm with Constant σ^2) *Suppose that $\tilde{\sigma}_{k+1}^2 = \sigma^2$ is iteration independent. Then, Theorem 9.15, with L replaced by $L\Gamma(GG^*, \sigma^2)$ and $\|\cdot\|_0$ replaced by $\|\cdot\|_{\sigma^2}$, remains valid for the Notch Algorithm 13.3. More precisely, Condition One is replaced by the condition*

$$U + U^* > \varepsilon^{-2} U^* G^* L \Gamma(GG^*, \sigma^2) GU, \quad on \quad \mathscr{R}[G^*]. \tag{13.70}$$

It ensures that $\|e_{k+1}\|_{\sigma^2} \leq \|e_k\|_{\sigma^2}$ for all $k \geq 0$. Boundedness of both $\{\|e_k\|_{\sigma^2}\}_{k\geq 0}$ and $\{\|e_k\|_0\}_{k\geq 0}$ then follow from topological equivalence of the norms as does the equivalence of their convergence to zero (if it occurs).

As with the case of left multiplicative perturbations, the inequality is satisfied if

$$U + U^* > \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right) \varepsilon^{-2} U^* G^* L G U, \quad on \quad \mathscr{R}[G^*] \tag{13.71}$$

which is precisely Theorem 9.6 with the added factor of $\left(\frac{(1+\sigma^2)^2}{\sigma^4}\right) > 1$ suggesting a reduction in robustness as σ^2 gets smaller. It is now an easy task to prove that monotonicity of the sequence $\{\|e_k\|_{\sigma^2}\}_{k\geq 0}$ for the notch algorithm with iteration independent values $\tilde{\sigma}_{k+1}^2 = \sigma^2 > 0$ follows in a similar way to Theorem 9.8 from the condition

$$\hat{U} + \hat{U}^* > \theta \beta_I U^* U + (1-\theta)\beta_G G^* G \quad on \ \mathscr{U} \quad for \ some \ \theta \in [0,1], \tag{13.72}$$

provided that β_I and β_G are replaced by

$$\beta_I = \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right) \frac{\varepsilon^{-2}\|G\|^2}{1+\varepsilon^{-2}\|G\|^2} \quad and \quad \beta_G = \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right) \varepsilon^{-2}. \tag{13.73}$$

The reminder of the analysis follows the pattern seen for the case of left multiplicative modelling errors and is left as an exercise for the reader who will note that $\|e\|_{\sigma^2}$ again reduces as σ^2 increases. As a consequence Theorem 13.10 remains valid in the form,

Theorem 13.12 (Robustness using Monotonic Values of $\tilde{\sigma}_{k+1}^2$) *With the notation of the discussion above, assume that the sequence $\{\tilde{\sigma}_{k+1}^2\}_{k\geq 0}$ is monotonically increasing from an initial value $\tilde{\sigma}_1^2 > 0$ and bounded by the value $\tilde{\sigma}_\infty^2 < \infty$. Then, if*

$$\hat{U} + \hat{U}^* > \left(\frac{1+\tilde{\sigma}_1^2}{\tilde{\sigma}_1^2}\right)^2 U^* G^* L G U \quad on \quad \mathscr{R}[G^*], \tag{13.74}$$

the error sequence generated by the Notch Algorithm 13.3 satisfies the monotonicity condition

$$\|e_{k+1}\|_{\tilde{\sigma}_{k+2}^2}^2 \leq \|e_{k+1}\|_{\tilde{\sigma}_{k+1}^2}^2 \leq \|e_k\|_{\tilde{\sigma}_{k+1}^2}^2, \quad for \quad k \geq 0. \tag{13.75}$$

In particular, there exists a real number $E_\infty \geq 0$ *such that*

$$\lim_{k\to\infty} \|e_k\|_{\tilde{\sigma}^2_{k+1}} = E_\infty \quad \textit{and} \quad \limsup_{k\to\infty} \|e_k\|_0 \leq E_\infty \sqrt{\sigma_\infty^4 + \varepsilon^{-2}\|G^*\|^2} \tag{13.76}$$

and hence $\lim_{k\to\infty} \|e_k\|_0 = 0$ *if* $E_\infty = 0$.

13.4 Discussion and Further Reading

Successive projection was defined in Chap. 12 based on the work in [95]. That reference provided a convincing indication that the geometry of the Hilbert space H can lead to slow convergence. In the case of NOILC Algorithm 9.1, this is easily seen in computational studies and is primarily associated with the use of large values of weight ε^2 (as well as other dynamic factors such as non-minimum-phase properties). Unfortunately, for state space models, small values of weight can lead to high gain feedback solutions. The case for using an acceleration mechanism in Iterative Control that does not require small weight values is therefore appealing, although, being model-based, issues of robustness need to be considered.

The work in [95] proposed an extrapolation approach based on Successive Projection. This was adopted in [83] for Iterative Control and its accelerating effects have been demonstrated in computational studies [27, 83]. Reductions in robustness, as compared with NOILC Algorithms such as Algorithm 9.1, can be expected in practice as the methodology is an extrapolation process using data from models and the plant. High values of the extrapolation factor λ_{k+1} could amplify the errors leading to erratic algorithm performance and, potentially, divergence. The most dangerous situation will occur if the sets S_1 and S_2 do not intersect. That is, there is no solution to the tracking problem. Reference [95] then suggests that wild oscillations and divergence could occur unless the design uses the flexibility in the choice of λ_{k+1} and puts practical limits $\lambda \in [1, \lambda_{max}]$ on the range used in the implementation!

The concept of successive projection seems to have considerable power in linking algorithms to practical experimental processes. There is no real reason why the sets involved need not vary from iteration to iteration. The question is "how should the sets vary and what effect will the choice and variation have on convergence rates and robustness?". This text has introduced the *Notch Algorithm* 13.3 which is new to the literature and uses this flexibility in the form of a parameterized family of sets in the product space $\mathscr{Y} \times \mathscr{U}$. Essentially, it applies a scaling factor $-\sigma^2$ to one term in the error definition, follows this by constructing a set $S_2(\sigma^2)$ to complement the dynamics S_1, and then applies successive projection. In the form presented, it is an algebraic construct but has properties of acceleration demonstrated by its improvements on NOILC as stated in Theorem 13.3 and, more fundamentally, the property of exact or approximate annihilation of spectral components of the error signal. Simple calculations based on the form of $f(\mu, \sigma^2)$, $0 < \mu \leq \|G^*\|^2$ provide

considerable insight into the benefits of this algorithm, particularly when the factor σ^2 is varied from iteration to iteration in an intelligent way.

The key to analysing the benefits on iteration $k+1$ lie in the consideration of functions defined by the products, with $k \geq 0$,

$$\prod_{j=0}^{k} f(\mu, \tilde{\sigma}_{j+1}^2) = \frac{1}{(1+\varepsilon^{-2}\mu)^{k+1}} \prod_{j=0}^{k} \frac{(\tilde{\sigma}_{j+1}^2 - \varepsilon^{-2}\mu)^2}{(\tilde{\sigma}_{j+1}^4 + \varepsilon^{-2}\mu)}, 0 < \mu \leq \|G^*\|^2. \quad (13.77)$$

The first factor describes the changes that would arise from $k+1$ iterations of basic NOILC Algorithm 9.1. The second factor

$$\prod_{j=0}^{k} \frac{(\tilde{\sigma}_{j+1}^2 - \varepsilon^{-2}\mu)^2}{(\tilde{\sigma}_{j+1}^4 + \varepsilon^{-2}\mu)}, 0 < \mu \leq \|G^*\|^2, \quad (13.78)$$

describes the way in which the various notches used on the iterations in Algorithm 13.3 affect the performance of the NOILC Algorithm. This term has well-defined "zeros" at $\varepsilon^2 \tilde{\sigma}_{j+1}^2, 0 \leq j \leq k$, providing the desired annihilation properties. In addition, the placing of the values opens up the opportunity, for a given value of "k", of ensuring that f has very small values over a chosen spectral range. The interested reader will be able to assess the effects and benefits by looking at a simple case with $\varepsilon^2 = \|G^*\|^2 = 1$, choosing $k = 2$ with $\tilde{\sigma}_1^2 = 1, \tilde{\sigma}_2^2 = 0.5$ and $\tilde{\sigma}_3^2 = 0.25$ and plotting the form of the expression above in the range $0 < \mu \leq 1$. A remarkable reduction in values will be observed and demonstrates the ability to create fast convergence without "high gain" control loops. In this case, for initial errors e_0 dominated by the contribution of the spectral values $\sigma^2 > 0.25$, rapid reductions in error magnitude can be achieved.

Finally, the robustness analysis has used the mathematical construct of introducing new norms to the problems and deriving operator inequalities that define tolerable multiplicative modelling errors. These inequalities are based on the artificial requirement of requiring monotonicity properties of the error sequence as measured by the new norms. The results provide some reassurance that bounded responses will be seen in practice. However, the precise behaviour of the errors as measured by the original norm $\|\cdot\|_{\mathscr{Y}}$ is difficult to ascertain from the properties of L and $L(\sigma^2)$, particularly when the "notch" σ^2 is varied from iteration to iteration.

Chapter 14
Parameter Optimal Iterative Control

This chapter continues with the use of optimization as the basis of Iterative Learning Control but considers a process of simplification of the computational structure. In the most general terms, the algorithms replace the search over "all" input signals by restricting the search to one defined by a finite number of parameters and previously recorded data. The free parameters are then computed as the solution of, what then becomes, a *finite dimensional optimization problem*. There are considerable benefits to be obtained by this process. In particular, monotonic reduction of error norms can be retained and computation is simplified as the parameters are defined by formulae obtained from an analytic solution. The representation is, however, nonlinear in the measured error data.

The presentation takes a quite general form emphasizing principles, properties and problems that can be met in applications. For example, care should be taken in the design of an algorithm as the choice of a poor parameterization can lead to flat-lining/plateauing effects very similar to those observed in gradient and NOILC algorithms when applied to non-minimum-phase state space systems. The problem can be avoided if an appropriate operator is strictly positive definite.

14.1 Parameterizations and Norm Optimal Iteration

The purpose of this section is twofold, namely

1. to show that suitable parametric representations of feedforward Iterative Control laws can be equivalent to the Norm Optimal Control Algorithm 9.1 and
2. to use this fact to create simplified algorithms that use only a few parameters.

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_14

Consider a process described by the dynamic relationship

$$y = Gu + d, \ \textit{with output} \ \ y \in \mathscr{Y} \ \ \textit{and} \ \ u \in \mathscr{U} \tag{14.1}$$

where $\mathscr{Y}$ and $\mathscr{U}$ are real Hilbert spaces and G is a bounded linear operator mapping $\mathscr{U} \to \mathscr{Y}$. Together with NOILC, this representation (with G replaced by "G_e") also includes Intermediate Point Algorithms, Multi-task Algorithms and other algorithms that are underpinned by the NOILC philosophy.

Definition 14.1 A general form of parameterization for feedforward algorithms is described by the update formula

$$u_{k+1} = u_k + \Gamma_{k+1} e_k \quad \textit{with} \quad e_k = r - y_k \tag{14.2}$$

(*Conceptual Parameterized Feedforward Iterative Control*).

It is parameterized by the iteration dependent, bounded, linear operator $\Gamma_{k+1} : \mathscr{Y} \to \mathscr{U}$ representing the process chosen for input updating. Rather than allow any choice, it is allowed to vary from iteration to iteration within limits defined by a constraint set Ω_Γ defined by the user.

Note: In what follows, optimization over signals $u \in \mathscr{U}$ is replaced by optimizing with respect to the choice of operator $\Gamma_{k+1} \in \Omega_\Gamma$.

The behaviour of the algorithm depends crucially on the interaction between the choice of Ω_Γ, $\Gamma_{k+1} \in \Omega_\Gamma$, the dynamics G and reference signal r. The problem of algorithm design is therefore one of choosing the constraints Ω_Γ and then, on each iteration, the operator Γ_{k+1} to achieve desired convergence characteristics. Algorithms such as NOILC Algorithm 9.1 create choices by optimization of $J(u, u_k)$ over input signals to give the result $\Gamma_{k+1} = \varepsilon^{-2} G^* (I + \varepsilon^{-2} G G^*)^{-1}$ whilst intermediate point, multi-task and other algorithms, including the notch algorithm, give other choices. In what follows, the optimization is that of choosing the "best" operator rather than the best signal. Using optimization for operator choice can be a complex mathematical task but the potential power of the idea is revealed in a special case described in what follows.

The Unconstrained Finite Dimensional Case: Suppose that both $\mathscr{Y}$ and $\mathscr{U}$ are finite dimensional and identify signals with column matrices and the operators G and Γ_{k+1} with matrices of the appropriate dimensions. Such an assumption includes the use of supervector descriptions of discrete state space models (Chap. 4). The inner products used can then be expressed in the form $\langle y, w \rangle_{\mathscr{Y}} = y^T Q w$ and $\langle u, v \rangle_{\mathscr{U}} = u^T R v$ where both Q and R are symmetric positive definite matrices. Note that, using these topologies, the adjoint operators are

$$G^* = R^{-1} G^T Q \quad \textit{and} \quad \Gamma^* = Q^{-1} \Gamma^T R. \tag{14.3}$$

The optimization proposed is expressed as the minimization of the "quadratic" objective function

$$\begin{aligned} J_{k+1}(\Gamma) &= \|e\|^2_{\mathscr{Y}} + tr[V_{k+1}\Gamma^T W_{k+1}\Gamma], \\ \text{where } V_{k+1} &= V_{k+1}^T > 0 \text{ and } W_{k+1} = W_{k+1}^T > 0 \end{aligned} \tag{14.4}$$

are, as yet unspecified, weighting matrices of appropriate dimension and the optimization is done subject to the dynamic constraints $e = r - Gu - d$ and the relationship $u = u_k + \Gamma e_k$. Otherwise the choice of Γ is unrestricted. That is,

$$\Gamma_{k+1} = \arg\min\{ J_{k+1}(\Gamma) \ : \ \textit{subject to } \ e = (I - G\Gamma)e_k\}. \tag{14.5}$$

The key to the existence and form of a solution is to note that

$$\langle \Gamma_1, \Gamma_2\rangle_F = tr[V_{k+1}\Gamma_1^T W_{k+1}\Gamma_2] \tag{14.6}$$

is an inner product on matrices and hence $\|\Gamma\|_F = \sqrt{tr[V_{k+1}\Gamma^T W_{k+1}\Gamma]}$ is a norm (a weighted form of Frobenius norm). The equation $e = (I-G\Gamma)e_k$ is therefore a linear, input-output relationship between two real Hilbert spaces but where the "input" is the matrix Γ. The problem of finding an unconstrained optimal Γ is therefore the linear quadratic optimization problem of minimizing $J_{k+1}(\Gamma)$ subject to $e = e_k - X_{k+1}\Gamma$, where X_{k+1} is a bounded, linear operator (between real Hilbert spaces) defined by $X_{k+1}\Gamma = G\Gamma e_k$. The problem therefore has a unique solution. More formally,

Theorem 14.1 (Parameter Optimization can be Equivalent to NOILC) *The notation of the discussion above is assumed and the choice of Γ is taken to be unconstrained. Under these conditions, the solution for the minimization of $J_{k+1}(\Gamma)$ subject to the constraint $e = (I - G\Gamma)e_k$ is the unique solution of the equation*

$$\Gamma_{k+1} = W_{k+1}^{-1}G^T Q(I - G\Gamma_{k+1})e_k e_k^T V_{k+1}^{-1} \tag{14.7}$$

and the input update law takes the form, for $k \geq 0$,

$$\begin{aligned} u_{k+1} &= u_k + \Gamma_{k+1}e_k \\ &= u_k + (I + \|e_k\|^2_{V_{k+1}} W_{k+1}^{-1}G^T QG)^{-1}\|e_k\|^2_{V_{k+1}} W_{k+1}^{-1}G^T Qe_k, \end{aligned} \tag{14.8}$$

where $\|e_k\|^2_{V_{k+1}} = e_k^T V_{k+1}^{-1} e_k$. Moreover, the norm sequence $\{\|e_k\|_{\mathscr{Y}}\}_{k\geq 0}$ is monotonically decreasing and, if W_{k+1} is chosen by the formula

$$W_{k+1} = \varepsilon^2\|e_k\|^2_{V_{k+1}} R, \tag{14.9}$$

the update law is precisely that obtained from the NOILC Algorithm 9.1. Under these circumstances the algorithm is identical to NOILC Algorithm 9.1 and inherits the properties of NOILC exactly.

Proof Writing the objective function in the form $J_{k+1} = \|e\|^2_{\mathscr{Y}} + \|\Gamma\|^2_F$ and the constraint in the form $e = e_k - X_{k+1}\Gamma$ gives the solution $\Gamma_{k+1} = X^*_{k+1}e_{k+1}$ where X^*_{k+1} is the adjoint of X_{k+1}. This is computed by considering, for any $e \in \mathscr{Y}$ and matrix Γ, the inner product

$$\begin{aligned}\langle e, X_{k+1}\Gamma\rangle_{\mathscr{Y}} &= e^T Q G \Gamma e_k = e^T Q G W_{k+1}^{-1} W_{k+1}\Gamma e_k = tr[e_k e^T Q G W_{k+1}^{-1} W_{k+1}\Gamma] \\ &= tr[V_{k+1}V_{k+1}^{-1} e_k e^T Q G W_{k+1}^{-1} W_{k+1}\Gamma] \\ &= tr[V_{k+1}(W_{k+1}^{-1} G^T Q e e_k^T V_{k+1}^{-1})^T W_{k+1}\Gamma] \\ &= \langle W_{k+1}^{-1} G^T Q e e_k^T V_{k+1}^{-1}, \Gamma\rangle_F = \langle X^* e, \Gamma\rangle_F,\end{aligned} \tag{14.10}$$

so that X^* is defined by the relationship $X^* e = W_{k+1}^{-1} G^T Q e e_k^T V_{k+1}^{-1}$. The result for $\Gamma_{k+1} = X^* e_{k+1}$ follows by replacing e_{k+1} by $(I - G\Gamma_{k+1})e_k$ to give

$$\Gamma_{k+1} = W_{k+1}^{-1} G^T Q (I - G\Gamma_{k+1}) e_k e_k^T V_{k+1}^{-1}. \tag{14.11}$$

The formula for u_{k+1} then follows by computing $\Gamma_{k+1}e_k$ and rearranging. The monotonicity of the error norms follows from the observation that $\Gamma = 0$ is suboptimal. The final observation follows trivially as a simple substitution gives $\Gamma_{k+1}e_k = (I + \varepsilon^{-2}G^*G)^{-1}G^* e_k = G^*(I + \varepsilon^{-2}GG^*)^{-1}e_k$ which are precisely the formula derived for feedforward NOILC. □

The result shows the, perhaps surprising, equivalence of a signal optimization problem and a class of "parameter optimization problems" if a suitable quadratic objective function is used and the weights in the norms associated with the parameters are carefully chosen. For an $N_1 \times N_2$ matrix Γ, the number of parameters used is precisely $N_1 \times N_2$ which, in Iterative Control, is typically very large indeed. The theorem will be used in Sects. 14.2 and 14.4 to motivate an approximation procedure based on the intuition that, if the number of free parameters is reduced, the resultant sub-optimal *Parameter Optimal Iterative Learning Control* (POILC) *Algorithm* may offer a simplified alternative to Norm Optimal ILC that avoids complex computation and retains the important, practical property of monotonic error norm reduction.

Choosing V_{k+1}: Theorem 14.1 is true for all symmetric, positive definite choices of V_{k+1}. It is natural to ask whether or not this non-uniqueness can be resolved using some other criterion. For example,

1. the choice of $V_{k+1} = I$ (the identity) produces the result that $\|e_k\|_{V_{k+1}} = \sqrt{e_k^T e_k}$ is the Euclidean norm. This has a certain simplicity but its numerical value will depend on the units used to describe the physical variables in e_k and hence the contribution of individual elements may not reflect their true importance in the application.

2. On the assumption that the matrix Q is chosen to reflect the physical units used and the relative importance of the variables, a definition of V_{k+1} that transforms to unit changes in the same way is obtained by the choice of

$$V_{k+1} = Q^{-1} \quad \textit{when, in particular,} \quad V_{k+1}^{-1}\Gamma_{k+1}R = \Gamma^*. \tag{14.12}$$

In this case $\|e_k\|_{V_{k+1}} = \|e_k\|_{\mathscr{Y}}$ and

Theorem 14.2 (The Optimal Value of Γ_{k+1}) *If $W_{k+1} = \varepsilon^2\|e_k\|^2_{V_{k+1}}R$ and $V_{k+1} = Q^{-1}$, then the optimal Γ_{k+1} is given by*

$$\begin{aligned}\Gamma_{k+1} &= (I + \varepsilon^{-2}G^*G)^{-1}\varepsilon^{-2}G^*e_ke_k^TQ\|e_k\|_{\mathscr{Y}}^{-2}\\ &= \varepsilon^{-2}G^*(I + \varepsilon^{-2}GG^*)^{-1}e_ke_k^TQ\|e_k\|_{\mathscr{Y}}^{-2}\end{aligned} \tag{14.13}$$

and

$$\varepsilon^2\|e_k\|^2_{\mathscr{Y}}tr\left[\Gamma_{k+1}^*\Gamma_{k+1}\right] = \varepsilon^2\|\varepsilon^{-2}G^*(I + \varepsilon^{-2}GG^*)^{-1}e_k\|^2_{\mathscr{U}}. \tag{14.14}$$

Note: The optimal value of the trace term can therefore be identified with the optimal value of $\varepsilon^2\|u - u_k\|^2_{\mathscr{U}}$ as computed using the NOILC Algorithm of Chap. 9. This observation provides an additional strengthening of the intuitive link between parameter optimization and NOILC-based signal optimization.

Proof Using Theorem 14.1 with the values assumed, write $\Gamma_{k+1} = \Psi e_ke_k^TQ$ for some matrix Ψ to obtain

$$\begin{aligned}\varepsilon^2\|e_k\|^2_{\mathscr{Y}}\Gamma_{k+1} &= G^*\left(I - G\Gamma_{k+1}\right)e_ke_k^TQ\\ &= G^*\left(I - G\Psi e_ke_k^TQ\right)e_ke_k^TQ\\ &= G^*e_ke_k^TQ - \|e_k\|^2_{\mathscr{Y}}G^*G\Gamma_{k+1}.\end{aligned} \tag{14.15}$$

The expression for Γ_{k+1} follows by rearranging. Next, note that the orthogonal projection $f \mapsto e_k\frac{e_k^TQf}{\|e_k\|^2_{\mathscr{Y}}}$ is self adjoint as a map from $\mathscr{Y}$ into $\mathscr{Y}$ and use the notation $L = (I + \varepsilon^{-2}GG^*)^{-1}$ to write

$$\begin{aligned}\varepsilon^2\|e_k\|^2_{\mathscr{Y}}tr\left[\Gamma_{k+1}^*\Gamma_{k+1}\right] &= \varepsilon^{-2}\|e_k\|^{-2}_{\mathscr{Y}}tr\left[e_ke_k^TQLGG^*Le_ke_k^TQ\right]\\ &= \varepsilon^{-2}\|e_k\|^{-2}_{\mathscr{Y}}e_k^TQLGG^*Le_k(e_k^TQe_k)\\ &= \varepsilon^{-2}\langle e_k, LGG^*Le_k\rangle_{\mathscr{Y}}\\ &= \varepsilon^{-2}\|G^*Le_k\|^2_{\mathscr{U}} = \varepsilon^2\|\varepsilon^{-2}G^*Le_k\|^2_{\mathscr{U}}\end{aligned} \tag{14.16}$$

which is the required expression. □

It is the second choice that is assumed in this text. Noting that $\Gamma^* = Q^{-1}\Gamma^TR$ then indicates that the objective function has the simple form

$$\begin{aligned}&J(\Gamma) = \|e\|^2_{\mathscr{Y}} + w_{k+1}tr[\Gamma^*\Gamma] && \textit{with weighting parameter}\\ &w_{k+1} = w_0 + \varepsilon^2\|e_k\|^2_{\mathscr{Y}} && \textit{where} \quad w_0 = 0.\end{aligned} \tag{14.17}$$

The iteration dependent parameter w_{k+1} represents the changing weight given to the operator Γ (squared) norm term as iterations progress.

Note: The apparently redundant introduction of a parameter w_0 followed, immediately, by setting it to zero will play a role in what follows when this constraint will be removed to allow choices $w_0 \geq 0$. Intuitively, choosing $w_0 > 0$ can be seen as a simple mechanism to avoid numerical ill-conditioning and, perhaps, retaining a degree of robustness by preventing the weight from reducing to zero as $\|e_k\|$ becomes small. During iterations where $w_0 \ll \varepsilon^2 \|e_k\|^2_{\mathscr{Y}}$, the link to NOILC is retained.

14.2 Parameter Optimal Control: The Single Parameter Case

The notion of *Parameter Optimal Iterative Learning Control* (POILC) is motivated by the analysis of the previous section with the addition of a constraint set Ω_Γ. As an introduction to the ideas, the case of a single parameter β is considered in this section. Even at this simple level, several technical issues arise, not least of which are the choice of parameterization and the selection of an objective function.

14.2.1 Alternative Objective Functions

The choice of parameterization and its consequences for the form of the objective function offers a number of alternatives.

14.2.1.1 Approach One

Consider the class of control laws $u = u_k + \Gamma e_k$ and the optimization problem of minimizing (14.17) over all possible operators $\Gamma \in \Omega_\Gamma$, with

$$\Omega_\Gamma = \{\Gamma \ : \ \Gamma = \beta \Gamma_0\}, \tag{14.18}$$

where $\Gamma_0 : \mathscr{Y} \to \mathscr{U}$ is a fixed (iteration independent) operator chosen by the user and β is a free scalar "gain" parameter. The vector $\Gamma_0 e_k$ can be regarded as the chosen search direction for the update rule on iteration $k+1$. The objective function motivated by (14.17) is

$$\begin{array}{ll} J(\Gamma) = \|e\|^2_{\mathscr{Y}} + w_{k+1}\beta^2 & \textit{with weight} \\ w_{k+1} = w(e_k) \qquad \textit{where} & \\ w(e) = \left(w_0 + \varepsilon^2 \|e\|^2_{\mathscr{Y}}\right) tr[\Gamma_0^* \Gamma_0] & \textit{and} \quad w_0 \geq 0. \end{array} \tag{14.19}$$

The parameter $\varepsilon^2 > 0$ is the notional link to NOILC but if, this link is unimportant, it is possible to allow the choice of ε^2 and w_0 satisfying

$$w_0 \geq 0, \quad \varepsilon^2 \geq 0 \quad \textit{and} \quad w_0 + \varepsilon^2 > 0 \quad (\textit{that is } \; w_1 > 0 \; \textit{if} \; \; e_0 \neq 0). \tag{14.20}$$

In addition, it may make practical sense to set

$$tr[\Gamma_0^* \Gamma_0] = 1, \qquad (\textit{trace normalization}) \tag{14.21}$$

by absorbing its real value into the choice of w_0 and ε^2.

Note: The algorithms described in Chaps. 6 and 7 represent examples of this constraint with $\Gamma_0 = G^{-1}$ (inverse model algorithms) and $\Gamma_0 = G^*$ (gradient-based algorithms). In those cases, β was regarded as a fixed, user-specified gain, but it is now regarded as a parameter that can be varied in an optimization problem.

14.2.1.2 Approach Two

Examination of the form of the optimal choice in (14.13) suggests that Ω_Γ should be error dependent. The natural generalization takes the form

$$\Omega_\Gamma(e) = \{\Gamma(e, \beta) \; : \; \Gamma(e, \beta) = \beta \Gamma_0 e e^T Q \|e\|_{\mathcal{Y}}^{-2}\}, \tag{14.22}$$

The control laws considered then take the form

$$u = u_k + \Gamma(e_k, \beta) e_k = u_k + \beta \Gamma_0 e_k. \tag{14.23}$$

That is, despite the different parameterization, the control update rule is identical to that used on Approach One. However, the objective function does change as

$$\begin{aligned} tr[\Gamma^*(e, \beta)\Gamma(e, \beta)] &= \beta^2 tr[e e^T Q \Gamma_0^* \Gamma_0 e e^T Q \|e\|_{\mathcal{Y}}^{-4}] \\ &= \beta^2 \|e\|_{\mathcal{Y}}^{-2} \langle e, \Gamma_0^* \Gamma_0 e \rangle_{\mathcal{Y}} = \beta^2 \|e\|_{\mathcal{Y}}^{-2} \|\Gamma_0 e\|_{\mathcal{U}}^2. \end{aligned} \tag{14.24}$$

The objective function therefore naturally takes the form

$$\begin{aligned} J(\Gamma) &= \|e\|_{\mathcal{Y}}^2 + \varepsilon^2 \|e_k\|_{\mathcal{Y}}^2 tr[\Gamma^*(e_k, \beta)\Gamma(e_k, \beta)] \\ &= \|e\|_{\mathcal{Y}}^2 + \varepsilon^2 \beta^2 \|\Gamma_0 e_k\|_{\mathcal{U}}^2 \end{aligned} \tag{14.25}$$

More generally, an option is to let

$$J(\Gamma) = \|e\|_{\mathcal{Y}}^2 + w_{k+1} \beta^2 \quad \textit{with weighting parameter} \quad w_{k+1} > 0, \tag{14.26}$$

where an additional parameter $w_0 \geq 0$ is included using the definition,

$$w_{k+1} = w(e_k) \quad \textit{where} \quad w(e) = w_0 + \varepsilon^2 \|\Gamma_0 e\|^2_{\mathscr{U}}. \tag{14.27}$$

The constraints on w_0 and ε^2 to ensure that $w_{k+1} > 0$ for all non-zero $e \in \mathscr{Y}$ are that

$$\begin{array}{rl} w_0 > 0 & \textit{if} \quad ker[\Gamma_0] \neq \{0\} \quad \textit{and} \\ w_0 + \varepsilon^2 > 0 & \textit{otherwise}. \end{array} \tag{14.28}$$

14.2.1.3 Approach Three

A third approach to creation of a parameter optimization problem is to use the inequality $\|\Gamma_0 e_k\|_{\mathscr{U}} \leq \|\Gamma_0\|^2 \|e_k\|^2_{\mathscr{Y}}$ and replace (14.26) by

$$\begin{array}{l} J(\Gamma) = \|e\|^2_{\mathscr{Y}} + w_{k+1}\beta^2 \qquad \textit{with weight} \quad w_{k+1} = w(e_k), \ \textit{and} \\ w(e) = w_0 + \varepsilon^2 \|\Gamma_0\|^2 \|e\|^2_{\mathscr{Y}}, \quad \textit{where} \\ \qquad w_0 \geq 0, \ \ \varepsilon^2 \geq 0 \ \ \textit{and} \ \ w_0 + \varepsilon^2 > 0. \end{array} \tag{14.29}$$

This expression includes an additional degree of approximation but note that

1. The increased value of w_{k+1} increases the weighting placed on β in J.
2. The expression can be further simplified by setting $\|\Gamma_0\| = 1$ and scaling ε^2 appropriately. In this case, ε^2 no longer has the same relationship to the underlying NOILC problem.

14.2.2 Problem Definition and Convergence Characterization

In all three cases discussed in Sects. 14.2.1.1–14.2.1.3, the user must choose an operator Γ_0 and to select, for iteration $k+1$, the optimizing parameter value in a class of one parameter update laws

$$u = u_k + \beta \Gamma_0 e_k \tag{14.30}$$

using the optimization problem

$$\beta_{k+1} = \arg\min_{\beta} J(\Gamma), \quad \textit{where} \quad J(\Gamma) = \|e\|^2_{\mathscr{Y}} + w_{k+1}\beta^2, \tag{14.31}$$

subject to the dynamical constraint $e = r - y$, $y = Gu + d$. The resultant input is denoted by $u_{k+1} = u_k + \beta_{k+1}\Gamma_0 e_k$ and the error response is denoted by e_{k+1}. The error arising from using any value is denoted by $e = r - y = r - Gu - d = (I - \beta G \Gamma_0) e_k$.

Substituting for e in $J(\Gamma)$ gives

$$J(\Gamma) = \|e_k\|^2_{\mathscr{Y}} - 2\beta\langle e_k, G\Gamma_0 e_k\rangle_{\mathscr{Y}} + \left(\|G\Gamma_0 e_k\|^2_{\mathscr{Y}} + w_{k+1}\right)\beta^2, \tag{14.32}$$

which is minimized by $\beta_{k+1} = \beta(e_k)$ where the function $\beta(e)$ is defined by

$$\beta(e) = \begin{bmatrix} \left(w(e) + \|G\Gamma_0 e\|^2_{\mathscr{Y}}\right)^{-1}\langle e, G\Gamma_0 e\rangle_{\mathscr{Y}} & , \quad if \quad e \neq 0 \\ 0 & , \quad if \quad e = 0. \end{bmatrix} \tag{14.33}$$

(*The Function Defining Optimal Parameter Values*).

Note that $\beta(e) = 0$ if, and only if, either $e = 0$ or $e \neq 0$ and $\langle e, G\Gamma_0 e\rangle_{\mathscr{Y}} = 0$. Also, $\beta(e)$ is continuous everywhere if $w_0 > 0$ and continuous everywhere but at the origin $e = 0$ if $w_0 = 0$.

The following algorithm exploits the discussion provided above. It can be applied to any of the formulations described in Sects. 14.2.1.1–14.2.1.3 by using the relevant expression for w_{k+1} and constraints on w_0 and ε^2.

Algorithm 14.1 (*Parameter Optimal Iteration: Single Parameter Case*) Given a system $y = Gu + d$ with output $y \in \mathscr{Y}$, input $u \in \mathscr{U}$ and reference signal $r \in \mathscr{Y}$, suppose that both $\mathscr{Y}$ and $\mathscr{U}$ are finite dimensional with signals represented by column vectors and operators by matrices and inner products $\langle e, w\rangle_{\mathscr{Y}} = e^T Q w$ and $\langle u, v\rangle_{\mathscr{U}} = u^T R v$ respectively. Then, using the notation of the preceding discussion, choose an operator (matrix/dynamical system) Γ_0, parameters $\varepsilon^2 \geq 0$, $w_0 \geq 0$ and initial input signal u_0 and obtain its associated tracking error e_0. Then the resultant single parameter, *Parameter Optimal Iterative Learning Control (POILC) Algorithm*, for all $k \geq 0$, sets the input for iteration $k+1$ equal to that obtained with parameter β equal to the optimizing value $\beta_{k+1} = \beta(e_k)$ computed from Eq. (14.33).

Note: The algorithm could, in principle, be used for cases where either or both of the spaces $\mathscr{Y}$ and $\mathscr{U}$ are infinite dimensional. This is true for the two approaches in Sects. 14.2.1.2 and 14.2.1.3 as the form of w_{k+1} has a natural interpretation in infinite dimensions. Problems can occur using the approach of Sect. 14.2.1.1 as the coefficient $tr[\Gamma_0^*\Gamma_0]$ can be seen as a source of technical problems as it is linked to matrix descriptions. The reader will note, however, that the algorithm can make mathematical sense of the trace computation in some cases including

1. if $\mathscr{Y}$ is finite dimensional as, then, $\Gamma_0^*\Gamma_0 : \mathscr{Y} \to \mathscr{Y}$ is a matrix.
2. If $\mathscr{U}$ is finite dimensional, then the range of G is finite dimensional and the matrix identity $tr[\Gamma_0^*\Gamma_0] = tr[\Gamma_0\Gamma_0^*]$ (where $\Gamma_0\Gamma_0^*$ is now a matrix) allows appropriate trace computation.
3. If both spaces are infinite dimensional, then $tr[\Gamma_0^*\Gamma_0]$ can be replaced by any scalar $T_0 > 0$, the choice of this scalar being an issue for the user to resolve. A default choice could be unity by absorbing T_0 into the choice of parameters in w_{k+1}. In this case the link to NOILC is lost but the algorithm still has computational validity.

The algorithm is feedforward and is implemented by plant tests using u_k to compute/measure the error e_k. This is followed by inter-task computation of the signal $G\Gamma_0 e_k$ either by

1. using a model and off-line simulation (the simplest option) or
2. by computing $v_k = \Gamma_0 e_k$ off-line and then finding $G\Gamma_0 e_k = Gv_k$ as the output from the plant using v_k as an input and using zero initial state conditions.

The inner product and norm evaluations are then possible for the evaluation of β_{k+1} and then u_{k+1}. Being a single parameter approximation to NOILC probably means that, for a given topology and choice of ε^2, convergence rates will be slower but the simplicity of the computations offers considerable potential benefits in the implementation. Details of convergence properties follow,

Theorem 14.3 (POILC: General Monotonicity and Convergence Properties) *Using the notation defined above, Algorithm 14.1 generates error and parameter sequences* $\{e_k\}_{k\geq 0}$ *and* $\{\beta_{k+1}\}_{k\geq 0}$ *that satisfy the monotonicity and convergence conditions*

$$\begin{array}{ll} \|e_{k+1}\|_{\mathscr{Y}} \leq \|e_k\|_{\mathscr{Y}}, \quad \textit{for all } k \geq 0 & (\textit{Monotonicity}), \\ \quad \textit{and} \quad \sum_{k=0}^{\infty} w_{k+1}\beta_{k+1}^2 < \infty & (\textit{Parameter Convergence}). \end{array} \tag{14.34}$$

In particular,

1. *equality* $\|e_{k+1}\|_{\mathscr{Y}} = \|e_k\|_{\mathscr{Y}}$ *holds if, and only if,* $\beta_{k+1} = 0$.
2. *Finally, if* $e_0 \neq 0$, *then* $e_k \neq 0$ *for all* $k \geq 0$. *That is, convergence to zero error in a finite number of iterations cannot occur.*

Proof As $\beta = 0$ is sub-optimal and the corresponding error is e_k, it follows that $\|e_{k+1}\|_{\mathscr{Y}}^2 + w_{k+1}\beta_{k+1}^2 \leq \|e_k\|_{\mathscr{Y}}^2$ for all $k \geq 0$ (proving monotonicity and the comment on conditions for equality). Recursive computation gives

$$\|e_0\|_{\mathscr{Y}}^2 \geq \|e_k\|_{\mathscr{Y}}^2 + \sum_{j=1}^{k} w_j\beta_j^2 \geq 0, \quad \textit{for all } k \geq 1. \tag{14.35}$$

The parameter convergence result follows easily from the positivity of norms and parameters. Finally, suppose that $e_k \neq 0$ but that $e_{k+1} = 0$. It follows that $\beta_{k+1} \neq 0$ and $G\Gamma_0 e_k = \beta_{k+1}^{-1} e_k$. Substituting into the formula for β_{k+1} gives

$$\beta_{k+1} = \left(w_{k+1} + \beta_{k+1}^{-2}\|e_k\|_{\mathscr{Y}}^2\right)^{-1} \beta_{k+1}^{-1}\|e_k\|_{\mathscr{Y}}^2 \tag{14.36}$$

which is just $w_{k+1} = 0$. That is, for the three cases considered,

$$\begin{array}{r} \left(w_0 + \varepsilon^2\|e_k\|_{\mathscr{Y}}^2\right) tr[\Gamma_0^*\Gamma_0] = 0, \\ w_0 + \varepsilon^2\|\Gamma_0 e_k\|_{\mathscr{Y}}^2 = 0 \quad or \\ w_0 + \varepsilon^2\|\Gamma_0\|^2\|e_k\|_{\mathscr{Y}}^2 = 0. \end{array} \tag{14.37}$$

With $e_k \neq 0$, these conditions are inconsistent with the specified constraints on the choices of w_0 and ε^2. □

The monotonicity property of NOILC Algorithm 9.1 is therefore retained but the price paid for the simplicity of the parameterization is that the relationship between parameter choice and convergence properties is more complex. The simplest differences arise as a consequence of the value of w_0 chosen and the properties of the quadratic form $\langle e, G\Gamma_0 e\rangle_{\mathscr{Y}}$ when $e \neq 0$.

14.2.3 Convergence Properties: Dependence on Parameters

Case ONE: If $w_0 = 0$ and $e_k \neq 0$, then the construction seen in Sect. 14.2.1.2 is excluded. For the cases explained in Sects. 14.2.1.1 and 14.2.1.3, ε^2 is necessarily non-zero and e_k can be replaced by its normalized value $h_k = e_k/\|e_k\|_{\mathscr{Y}}$ to give $\|h_k\|^2_{\mathscr{Y}} = 1$.

$$\beta_{k+1} = \frac{\langle h_k, G\Gamma_0 h_k\rangle_{\mathscr{Y}}}{(\varepsilon^2 T_0 + \|G\Gamma_0 h_k\|^2_{\mathscr{Y}})}, \tag{14.38}$$

for some real number $T_0 > 0$. It follows that β_{k+1} lies in a closed and bounded range $\underline{\beta} \leq \beta \leq \overline{\beta}$ for all $e_k \neq 0$. Without loss of generality, assume that $\underline{\beta}$ (respectively, $\overline{\beta}$) is the largest (respectively, smallest) such value. The situations considered are as follows,

1. If $\underline{\beta}\overline{\beta} > 0$, then $\beta^2_{k+1} \geq \min\{\underline{\beta}^2, \overline{\beta}^2\} > 0$ for all $k \geq 0$. The convergence of the infinite series in Theorem 14.3 then implies the ideal outcome that

$$\lim_{k\to\infty} \|e_k\|^2_{\mathscr{Y}} = 0 \quad (\textit{Error Convergence to Zero}) \tag{14.39}$$

   as $\min\{\underline{\beta}^2, \overline{\beta}^2\}\varepsilon^2 \sum_{k=0}^{\infty} \|e_k\|^2_{\mathscr{Y}} \leq \sum_{k=0}^{\infty} w_{k+1}\beta^2_{k+1} < \infty$.
2. If $\underline{\beta}\overline{\beta} < 0$, then there exists at least one non-zero vector $\hat{e} \in \mathscr{Y}$ such that $\beta(\hat{e}) = 0$ and hence that $\langle \hat{e}, G\Gamma_0\hat{e}\rangle_{\mathscr{Y}} = 0$. Under these conditions, the algorithm converges immediately if the initial error $e_0 = \hat{e}$ as $\beta_1 = \beta(e_0) = 0$ from which $u_1 = u_0$ and $e_1 = e_0$. The algorithm produces no change as it follows that $e_k = e_0$ and $\beta_{k+1} = 0$ for all $k \geq 0$. More generally, if e_0 is arbitrarily chosen, the parameter sequence $\{\beta_{k+1}\}_{k\geq 0}$ remains bounded but, although the error norms reduce monotonically, the error sequence $\{e_k\}_{k\geq 0}$ may not necessarily converge to zero.
3. If $\underline{\beta}\overline{\beta} = 0$, then either $0 = \underline{\beta} < \overline{\beta}$ or $\underline{\beta} < \overline{\beta} = 0$. For simplicity, the first case of $\underline{\beta} = 0$ is considered and the reader is invited to complete the analysis for the second case of $\overline{\beta} = 0$. Two situations should be considered. The first is when there exists a non-zero vector $\hat{e} \in \mathscr{Y}$ such that $\langle \hat{e}, G\Gamma_0\hat{e}\rangle_{\mathscr{Y}} = 0$. It is easily seen that the behaviour has the same properties as that deduced for $\underline{\beta}\overline{\beta} < 0$.

This leaves the case where $\underline{\beta} = 0$ but where there is no vector $\hat{e} \neq 0$ such that $\langle \hat{e}, G\Gamma_0 \hat{e} \rangle_{\mathscr{Y}} = 0$. This can only occur if both $\mathscr{Y}$ and $\mathscr{U}$ are infinite dimensional, a case not considered further in this text.

The discussion indicates the benefits of choosing Γ_0 to ensure that $\underline{\beta}\overline{\beta} > 0$. A consideration of the form of $\beta(e)$ indicates that this is achieved if, and only if, $G\Gamma_0$ is sign definite in the sense that, for some $\varepsilon_0^2 > 0$, either

$$\langle e, G\Gamma_0 e \rangle_{\mathscr{Y}} \geq \varepsilon_0^2 \|e\|_{\mathscr{Y}}^2 \quad or \quad \langle e, G\Gamma_0 e \rangle_{\mathscr{Y}} \leq -\varepsilon_0^2 \|e\|_{\mathscr{Y}}^2 \quad for\ all\ e \in \mathscr{Y}. \tag{14.40}$$

In terms of the operators involved, these conditions are expressed as

$$G\Gamma_0 + (G\Gamma_0)^* \geq 2\varepsilon_0^2 I \quad or \quad G\Gamma_0 + (G\Gamma_0)^* \leq -2\varepsilon_0^2 I. \tag{14.41}$$

One condition becomes the other if Γ_0 is replaced by $-\Gamma_0$ so, without loss of generality, the positivity condition describes the situation completely.

Case Two: If $w_0 > 0$, then all three of the approaches seen in Sects. 14.2.1.1–14.2.1.3 can be considered together. Error norm monotonicity is present and there exists a scalar $\tilde{w}_0 > 0$ such that $w_{k+1} \geq \tilde{w}_0$ for all $k \geq 0$. Using Eq. (14.34), it follows that

$$\sum_{k=0}^{\infty} \tilde{w}_0 \beta_{k+1}^2 < \sum_{k=0}^{\infty} w_{k+1} \beta_{k+1}^2 < \infty \quad so\ that \quad \lim_{k \to \infty} \beta_{k+1} = 0. \tag{14.42}$$

The nature of the asymptotic behaviour of the algorithm is, again, described entirely by the behaviour of $\langle e, G\Gamma_0 e \rangle_{\mathscr{Y}}$. The analysis is very similar to that of $w_0 = 0$. In particular,

1. if $G\Gamma_0$ is sign-definite in the sense of Eq. (14.40) then convergence of the $\{\beta_{k+1}\}_{k \geq 0}$ to zero implies that $\lim_{k \to \infty} e_k = 0$.
2. If there exists a non-zero vector $\hat{e} \in \mathscr{Y}$ such that $\langle \hat{e}, G\Gamma_0 \hat{e} \rangle_{\mathscr{Y}} = 0$, then $\beta(\hat{e}) = 0$ and the initial error $e_0 = \hat{e}$ is unchanged by the algorithm. That is, the algorithm ensures monotonic reductions in error norm but the error does not necessarily converge to zero.

The introduction of $w_0 > 0$ therefore does not change the basic nature of the convergence of the algorithm. Its role lies, primarily, in avoiding any ill-conditioning of the computations when the error norms are small. For example, a useful choice may be to make it sufficiently small so that it has little effect on initial iterations and only comes into play when the error is near to the desired tracking accuracy.

14.2.4 Choosing the Compensator

Section 14.2.3 has shown the benefits of choosing Γ_0 to satisfy a sign-definiteness condition (14.40). Four examples are described in what follows based on the Inverse Model Algorithm 6.1, the Gradient Algorithm 7.2 and the Norm Optimal Algorithm 9.1 for discrete systems (Algorithm 9.4).

Inverse Model Compensation: The choice of a *right inverse* G_R (when it exists) of G as the operator Γ_0 gives $G\Gamma_0 = I$ and the positivity condition (Eq. 14.41) is satisfied. Algorithm 14.1 hence produces monotonic errors converging to zero and

$$\beta_{k+1} = \frac{\|e_k\|^2_{\mathscr{Y}}}{\left(w_{k+1} + \|e_k\|^2_{\mathscr{Y}}\right)}. \tag{14.43}$$

Gradient-based Compensation: The choice of the *adjoint* operator G^* (of G) as the operator Γ_0 gives $G\Gamma_0 = GG^*$ and

$$\beta_{k+1} = \frac{\|G^*e_k\|^2_{\mathscr{Y}}}{\left(w_{k+1} + \|GG^*e_k\|^2_{\mathscr{Y}}\right)}. \tag{14.44}$$

The orthogonal subspace decomposition $\mathscr{Y} = ker[G^*] \oplus \overline{\mathscr{R}[G]}$ and the evolution $e_{k+1} = (I - \beta_{k+1}GG^*)e_k$ then indicates that the orthogonal projection $P_{ker[G^*]}e_0$ of e_0 onto $ker[G^*]$ is unchanged from iteration to iteration and that all error evolution is confined to $\overline{\mathscr{R}[G]}$. For analysis purposes $\mathscr{Y}$ can be replaced by $\overline{\mathscr{R}[G]}$. The positivity condition $GG^* > 0$ is therefore always satisfied on the Hilbert subspace $\overline{\mathscr{R}[G]}$ and, in the finite dimensional case, $GG^* \geq (1/2)\varepsilon_0^2 I$ for some $\varepsilon_0 > 0$. Hence, the error norms are monotonically reducing and

$$\lim_{k\to\infty} e_k = P_{ker[G^*]}e_0. \tag{14.45}$$

Approximate Inverses and Gradients: In the above, Γ_0 can be replaced by an approximation to either an inverse plant or the adjoint. The motivation for such approximations is primarily to simplify the computation of $\Gamma_0 e_k$. There is an infinity of such simplifications but, for successful algorithm performance, the positivity conditions should, ideally, be satisfied.

Norm Optimal Iteration: NOILC Algorithm 9.1 suggests the choice of $\Gamma_0 = \varepsilon^{-2}G^*(I + \varepsilon^{-2}GG^*)^{-1}$.

1. For the parameterization introduced in Sect. 14.2.1.1, eigenvalue-based matrix analysis then shows that

$$\frac{\varepsilon^{-4}}{(1 + \varepsilon^{-2}\|G^*\|^2)^2} tr[GG^*] \;\leq\; tr[\Gamma_0^*\Gamma_0] \;\leq\; \varepsilon^{-4} tr[GG^*] \tag{14.46}$$

which bounds the trace by that seen in the case of gradient algorithms.

2. For the parameterization used in Sect. 14.2.1.2 with $\varepsilon^2 > 0$, the computation of $\|\Gamma_0 e_k\|^2_{\mathscr{U}}$ is precisely the evaluation of the norm $\|u_{k+1} - u_k\|^2_{\mathscr{U}}$ computed using the NOILC Algorithm using the specified value of ε^2.
Note: If the value of $w_0 = 0$ were to be used on any iteration, then the material in Sect. 14.1 indicates that the optimal value of the parameter $\beta_{k+1} = 1$ on that iteration. Using $w_0 > 0$ will change this situation.
3. For the parameterization used in Sect. 14.2.1.3, an estimate of the norm of Γ_0 is required and can be undertaken using computational and/or theoretical eigenvalue analysis and the spectral mapping theorem. The eigenvalues are unknown but, as

$$\Gamma_0^* \Gamma_0 = \varepsilon^{-4}(I + \varepsilon^{-2} GG^*)^{-1} GG^* (I + \varepsilon^{-2} GG^*)^{-1} \geq 0, \tag{14.47}$$

it can be bounded by analysis of the properties of the function $\varepsilon^{-2}x(1+x)^{-2}$ in the range $0 \leq x \leq \varepsilon^{-2}\|G^*\|^2$. This gives

$$\|\Gamma_0\|^2 \leq \sup_{0 \leq x \leq \varepsilon^{-2}\|G^*\|^2} \varepsilon^{-2} x(1+x)^{-2}. \tag{14.48}$$

The supremum over $[0, \infty)$ occurs at $x = 1$ from which

$$\begin{aligned} \|\Gamma_0\|^2 &= \varepsilon^{-4}\|G^*\|^2(1 + \varepsilon^{-2}\|G^*\|^2)^{-2}, \ \textit{if} \quad \varepsilon^2 \geq \|G^*\|^2 \quad \textit{and} \\ |\Gamma_0\|^2 &\leq \tfrac{1}{4}\varepsilon^{-2} \quad \textit{otherwise}. \end{aligned} \tag{14.49}$$

In any situation, the user has the choice of three forms of w_{k+1}. This choice can be reduced as there are problems implicit in the approach suggested in Sect. 14.2.1.1. This is illustrated by considering the frequency domain estimation of the trace of GG^* and G^*G for discrete state space systems in the next section.

14.2.5 Computing $tr[\Gamma_0^* \Gamma_0]$: Discrete State Space Systems

The value of $tr[\Gamma_0^* \Gamma_0]$ used in the expressions for β_{k+1} must be either assumed, estimated or computed if the approach of Sect. 14.2.1.1 is selected. This section considers the problem of estimating $tr[\Gamma_0^* \Gamma_1]$ where Γ_0 (respectively, Γ_1) can be represented by a linear, discrete, state space model $S(A, B, C, D)$ (respectively, $S(A_1, B_1, C_1, D_1)$) and associated discrete transfer function matrix $\Gamma_0(z)$ (respectively, $\Gamma_1(z)$). The trace computation can then be approximated as follows,

Theorem 14.4 (Estimating the Trace of $\Gamma_0^* \Gamma_1$ for Discrete State Space Systems) *Using the notation above, suppose that both $\Gamma_0 : \mathscr{Y} \to \mathscr{U}$ and $\Gamma_1 : \mathscr{Y} \to \mathscr{U}$ are asymptotically stable, discrete, ℓ-output, m-input, state space systems operating on*

the discrete interval $0 \le t \le N$ *and that* $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ *and* $\mathscr{U} = \mathscr{R}^{\ell(N+1)}$ *with inner products*

$$\langle e, w\rangle_{\mathscr{Y}} = \sum_{t=0}^{N} e^T(t)Qw(t) \quad \text{and} \quad \langle u, v\rangle_{\mathscr{U}} = \sum_{t=0}^{N} u^T(t)Rv(t) \tag{14.50}$$

where R *(respectively,* Q*) is an* $\ell \times \ell$ *(respectively* $m \times m$*) symmetric, positive definite matrix. Then, keeping the sample interval constant and extending the time interval by increasing* N *gives*

$$\lim_{N\to\infty} (N+1)^{-1} tr[\Gamma_0^* \Gamma_1] = tr\left[\frac{1}{2\pi i}\oint_{|z|=1} Q^{-1}\Gamma_0^T(z^{-1})R\Gamma_1(z)\frac{dz}{z}\right]. \tag{14.51}$$

Note: The contour integral integrates on the unit circle in the complex plane in a counterclockwise direction represented by $z = e^{i\theta}$, $0 \le \theta \le 2\pi$.

Proof Using the supervector description (Chap. 4) of the system, the adjoint operator is the supervector description of $S(A^T, C^T R, Q^{-1}B^T, Q^{-1}D^T R)$ with a zero terminal condition at $t = N$. Using the structure of supervector descriptions, the trace of $\Gamma_0^*\Gamma_1$ is then given by

$$tr[\Gamma_0^*\Gamma_1] = \sum_{j=0}^{N} (N+1-j)tr[Q^{-1}X_j^T RY_j] \tag{14.52}$$

where $X_0 = D$, $Y_0 = D_1$ and $X_j = CA^{j-1}B$, $Y_j = C_1 A_1^{j-1}B_1$ for $j \ge 1$. The asymptotic stability assumption, the Cauchy-Schwarz Inequality and the fact that the trace is an inner product on matrices then implies the existence of scalars $\lambda \in (0, 1)$ and $M > 0$ such that $0 \le \left|tr[Q^{-1}X_j^T RY_j]\right| \le M\lambda^{j-1}$ for all $j \ge 1$. It follows that,

$$\left|\lim_{N\to\infty} \sum_{j=0}^{N} j\, tr[Q^{-1}X_j^T RY_j]\right| \le M \lim_{N\to\infty} \sum_{j=1}^{N} j\lambda^{j-1} = \frac{M}{(1-\lambda)^2} < \infty \tag{14.53}$$

and hence that,

$$\begin{aligned} \lim_{N\to\infty} (N+1)^{-1} tr[\Gamma_0^*\Gamma_1] &= \lim_{N\to\infty} \sum_{j=0}^{N} \tfrac{(N+1-j)}{(N+1)} tr[Q^{-1}X_j^T RY_j] \\ &= \sum_{j=0}^{\infty} tr[Q^{-1}X_j^T RY_j] \\ &= tr\left[\sum_{j=0}^{\infty} Q^{-1}X_j^T RY_j\right] < \infty \end{aligned} \tag{14.54}$$

The result follows by writing $\Gamma_0(z) = \sum_{j\ge 0} z^{-j}X_j$, $\Gamma_1(z) = \sum_{j\ge 0} z^{-j}Y_j$ and evaluating the contour integral. □

The result provides an estimate of the trace when N is finite but large. Namely,

$$tr[\Gamma_0^* \Gamma_1] \approx tr\left[\frac{N+1}{2\pi i} \oint_{|z|=1} Q^{-1} \Gamma_0^T(z^{-1}) R \Gamma_1(z) \frac{dz}{z}\right], \tag{14.55}$$

the fractional error in this estimate becoming arbitrarily small as N increases. More structure is revealed by considering the example of $\Gamma_0 = \Gamma_1$ with

$$\Gamma_0(z) = \tfrac{z(1-\lambda)}{z-\lambda} \quad \textit{when} \quad \tfrac{1}{2\pi i} \oint_{|z|=1} \Gamma_0(z^{-1}) \Gamma_0(z) \tfrac{dz}{z} = \tfrac{1-\lambda}{1+\lambda}, \tag{14.56}$$

where $\lambda \in (-1, 1)$ ensures stability and $\Gamma_0(1) = 1$ normalizes low frequency gains. If the pole $\lambda = e^{ah}$ where $h = N^{-1}T$ is the sample interval and the system evolves on an underlying continuous time interval $[0, T]$, then, assuming that h is small, the integral $\approx -ah/2$ and hence is small. Multiplying by $N+1$ then shows that $tr[\Gamma_0^* \Gamma_0]$ is approximately proportional to T.

Finally, approximate gradient-based methodologies (Chap. 7) would choose Γ_0 as the adjoint K^* of an asymptotically stable, discrete, ℓ-input, m-output, state space system $K : \mathscr{U} \to \mathscr{Y}$ (with transfer function matrix $K(z)$) operating on the discrete interval $0 \le t \le N$. If Γ_1 is the adjoint of an asymptotically stable, discrete, ℓ-input, m-output, state space system $K_1 : \mathscr{U} \to \mathscr{Y}$ (with transfer function matrix $K_1(z)$), Theorem 14.4 immediately gives,

Theorem 14.5 ($tr[\,\Gamma_1^* \Gamma_0]$ for Adjoints of Discrete State Space Systems) *Using the notation and assumptions preceding this theorem,*

$$\lim_{N\to\infty} (N+1)^{-1} tr[\Gamma_1^* \Gamma_0] = tr\left[\frac{1}{2\pi i} \oint_{|z|=1} R^{-1} K^T(z^{-1}) Q K_1(z) \frac{dz}{z}\right]. \tag{14.57}$$

Proof The techniques used in the proof of Theorem 14.4 can be used. If K has a state space model $S(A, B, C, D)$, K^* then has the model $S(A^T, C^T Q, R^{-1}B^T, R^{-1}D^T Q)$ with zero terminal condition at $t = N$. The result follows by writing $\Gamma_0 = K^*$ and $\Gamma_1^* = (K_1^*)^* = K_1$ so that

$$tr[\Gamma_1^* \Gamma_0] = tr[K_1 K^*] = tr[K^* K_1] \tag{14.58}$$

and noting the interchanged roles of R and Q in the algebra. □

14.2.6 Choosing Parameters in $J(\beta)$

As presented in previous sections, the choice of parameters w_0 and ε^2 is an essential part of algorithm design. Section 14.1 provides an interpretation of ε^2 as the parameter appearing in the NOILC objective function where, in addition, $w_0 = 0$. An alternative approach is simply to regard w_0 and ε^2 as parameters for

influencing the performance of POILC. This relationship is complex and no general result is available. The simplest approach to seeing the issues is to consider a simple example.

Consider the case of linear discrete time state space systems on an interval $0 \leq t \leq N$. Let $\Gamma_0 = G_R$ where G_R is a right inverse of G, and hence $GG_R = I$. Writing $\beta(e)$ in terms of the "mean square value" $\overline{\|e\|^2_{\mathscr{Y}}} = (N+1)^{-1}\|e\|^2_{\mathscr{Y}}$,

$$\beta(e) = \frac{\overline{\|e\|^2_{\mathscr{Y}}}}{\left(w(e)(N+1)^{-1} + \overline{\|e\|^2_{\mathscr{Y}}}\right)}. \tag{14.59}$$

The factor $w(e)(N+1)^{-1}$ provides some guidance as, in the three cases considered in Sects. 14.2.1.1–14.2.1.3 which give

$$\begin{aligned}
&(A) \quad w(e)(N+1)^{-1} = \left(w_0(N+1)^{-1} + \varepsilon^2\overline{\|e\|^2_{\mathscr{Y}}}\right) tr[G_R^* G_R],\\
&(B) \quad w(e)(N+1)^{-1} = \left(w_0(N+1)^{-1} + \varepsilon^2\overline{\|G_R e\|^2_{\mathscr{U}}}\right) \quad \textit{and}\\
&(C) \quad w(e)(N+1)^{-1} = \left(w_0(N+1)^{-1} + \varepsilon^2\|G_R\|^2\overline{\|e\|^2_{\mathscr{Y}}}\right)
\end{aligned} \tag{14.60}$$

These expressions suggest the following observations

1. In all cases, if the parameter w_0 is to have any effect, it will need to increase as N increases.
2. Using (A) with a long time interval N yields, from Theorem 14.4,
   a. the result that $\beta(e)$ is inversely proportional to $N+1$ so that, for a given mean square error and parameters w_0 and ε^2, convergence may be slow. This problem can be removed by using the substitution

   $$tr[\Gamma_0^* \Gamma_0] = 1 \quad (\textit{Trace Normalization}) \tag{14.61}$$

   and then to regard $\varepsilon^2 \geq 0$ and $w_0 \geq 0$ as independent parameters to be chosen to achieve the desired convergence rates and robustness.
   b. This approach is reinforced when G_R is unstable as then the trace can be very large.
3. Using (C) with $w_0 = 0$ gives a constant value

   $$\beta(e) = \frac{1}{\varepsilon^2\|G_R\|^2 + 1} \tag{14.62}$$

   and the analysis reduces to the choice of gain for the inverse algorithm analysed in Chap. 6. The choice of a non-zero value $w_0 > 0$ could be based on the selection of both w_0 and ε^2 to make β_1 take a desired value $\beta_{init} < 1$ and to take the value $\beta_{tol} < \beta_{init}$ when the error norm reaches a value E_{tol} with $E^2_{tol} < \|e_0\|^2_{\mathscr{Y}}$. That is the parameters are the solutions of the two equations

$$\beta_{init} = \frac{\|e_0\|^2_{\mathscr{Y}}}{(w_0+\varepsilon^2\|G_R\|^2\|e_0\|^2_{\mathscr{Y}})+\|e_0\|^2_{\mathscr{Y}}} \quad and$$
$$\beta_{tol} = \frac{E^2_{tol}}{(w_0+\varepsilon^2\|G_R\|^2E^2_{tol})+E^2_{tol}} \tag{14.63}$$

which are just two linear simultaneous equations in w_0 and ε^2. If positive solutions exist, the effect of the procedure is to implement a faster convergence rate in the initial iterations but to reduce this rate progressively as the error norm gets smaller. The benefits of this are expected to be increased robustness of the iterative process in the latter stages of the algorithm.

4. For case (B), the analysis is less clear as, for a given mean square error, the mean square value of $G_R e_k$ can take a range of values dependent on e_k.

The effect of parameter choice on robustness is considered in Sect. 14.3 with the general result that increased robustness will be associated with increased values.

14.2.7 Iteration Dynamics

Suppose that both $\mathscr{Y}$ and $\mathscr{U}$ are finite dimensional. Parameter optimal iteration dynamics can be seen as a finite dimensional nonlinear dynamical system defined by the equations

$$e_{k+1} = (I - \beta_{k+1}G\Gamma_0)e_k, \qquad \beta_{k+1} = \beta(e_k), \qquad for\ \ k \geq 0, \tag{14.64}$$

with parameter evolution,

$$\beta_{k+1} = \frac{\langle e_k, G\Gamma_0 e_k\rangle_{\mathscr{Y}}}{(w_{k+1} + \|G\Gamma_0 e_k\|^2_{\mathscr{Y}})}, \tag{14.65}$$

where w_{k+1} is constructed in one of the ways described in Sects. 14.2.1.1–14.2.1.3.

In the ideal case, when the sign-definiteness conditions (14.40) are satisfied, convergence of the error to zero is guaranteed. In addition, the parameter sequence $\{\beta_{k+1}\}_{k\geq 0}$ is bounded if $w_0 = 0$ and converges to zero if $w_0 > 0$. If, however, (14.41) is not satisfied, the situation is more complex. It is still possible for the error to converge to zero but the dynamical behaviour has more components.

14.2.8 Plateauing/Flatlining Phenomena

Convergence problems exist only at error data points e in the set

$$S_0 = \{e\ :\ \langle e, G\Gamma_0 e\rangle_{\mathscr{Y}} = 0\}, \quad (that\ is, \quad S_0 = \{e\ :\ \beta(e) = 0\}) \tag{14.66}$$

in the sense that initial errors $e_0 \in S_0$ are equilibrium points of the iteration dynamics. Note that,

1. S_0 always contains the kernel/null space $ker[G\Gamma_0]$ of $G\Gamma_0$.
2. Starting at any nonzero point $e_0 \in S_0$, the resultant iteration response has the form $e_k = e_0$ and $\beta_{k+1} = 0$, for all $k \geq 0$. That is, Algorithm 14.1 does not improve on the initial error.
3. If $S_0 \neq \{0\}$ and e is close to S_0, then $\beta(e)$ is small.
4. If $e \in S_0$ then $\lambda e \in S_0$ for all real scalars λ and any $S_0 \neq \{0\}$ contains error signals of arbitrarily large and small magnitudes.
5. The case when (14.40) holds corresponds to $S_0 = \{0\}$.

Consider, therefore, the case where S_0 contains at least one nonzero element. Starting at any nonzero point $e_0 \in \mathscr{Y}$, the resultant iteration error sequence $\{e_k\}_{k\geq 0}$ can have several characteristics including convergence to zero error. However, the presence of the set $S_0 \neq \{0\}$ means that, if an error e_k in the sequence, at any time, moves close to S_0, the resultant small value of β_{k+1} leads to a small change in error which then leads to a small value of β_{k+2}. Continuing with this argument, it is deduced therefore, that, close to S_0,

1. small changes in error occur over several, or even many, iterations. That is, the algorithm will lead to an iteration response with a period of very slow convergence represented by an almost constant value of error norm. Hence,
2. the closer an iterate e_k gets to S_0, the greater the number of iterations that will be needed to escape from the slow convergence phenomenon (if, indeed, escape is possible).

The period of infinitesimally slow convergence if the error sequence gets close to S_0 is an analogue, for Parameter Optimal Control, of the behaviour seen in Gradient and Norm Optimal Algorithms applied to non-minimum-phase discrete state space systems (see Sects. 8.2.4 and 9.3). The period of poor convergence could, in principle, consist of just a few iterations but, very close to S_0, the period could consist of hundreds or even thousands of iterations. Such a situation would be indistinguishable, in practice, from actual convergence of the algorithm to a non-zero limit error. Using the dynamical system interpretation of Eq. (14.64) suggests several possibilities based on the interpretation of S_0 as the set of equilibrium points. That is,

1. S_0 could *attract* the sequence $\{e_k\}_{k\geq 0}$ with convergence $\lim_{k\to\infty} e_k = e_\infty \in S_0$ where e_∞ need not be zero, or
2. it could attract the sequence which, rather than converging to a limit, *slides* monotonically and very slowly over the surface of S_0. Alternatively,
3. S_0 could *repel* the error sequence in the sense that, despite a period near to S_0, the iterates ultimately move away and, improved convergence rates reappear.
4. Note that, even if the sequence moves away from S_0, it is still possible that it could return to be close to it again. That is periods of slow convergence could reappear several times before final convergence (or "sliding") is achieved.

The possible behaviours described above are deduced from the dynamical system interpretation (14.64) of iteration evolution. The dynamics are evolving, typically, in a high dimensional space which makes detailed computational analysis difficult. For practical purposes therefore, the discussion underlines the benefits of ensuring that the positivity condition (14.40) is satisfied. If, however, it is not satisfied, the discussion throws light on the range of behaviours and difficulties that can be met.

A natural question to ask is whether or not conditions can be derived that describe the "attracting" and "repelling" behaviours in terms of subsets of S_0 where they occur. A semi-quantitative analysis of the situation now follows. The proximity of $e \in \mathscr{Y}$ to S_0 is associated with the value of $\beta(e)$. S_0 is characterized by $\beta(e) = 0$ and is the frontier of two disjoint sets in $\mathscr{Y}$ where, respectively, $\beta(e) > 0$ and $\beta(e) < 0$. Suppose that e_k is non-zero and not in S_0.

A simple characterization of situations where S_0 attracts iterates is represented by movement towards S_0 represented by the inequality $\beta_{k+2}^2 < \beta_{k+1}^2$. This is just

$$0 > (\beta_{k+2} + \beta_{k+1})(\beta_{k+2} - \beta_{k+1}) = (2\beta_{k+1} + (\beta_{k+2} - \beta_{k+1}))(\beta_{k+2} - \beta_{k+1}). \tag{14.67}$$

If e_k is non-zero and infinitesimally close to S_0, then $e_{k+1} - e_k$ is infinitesimally small and, using a first order power series expansion of the function $\beta(e)$ about the point e_k gives

$$\beta_{k+2} - \beta_{k+1} = \beta(e_{k+1}) - \beta(e_k) \approx \frac{\partial \beta(e)}{\partial e}|_{e=e_k}(e_{k+1} - e_k) \tag{14.68}$$

where the vector $\frac{\partial \beta(e)}{\partial e}|_{e=e_k}$ represents the Frechet derivative of $\beta(e)$ at e_k. Using the iteration dynamics $e_{k+1} - e_k = -\beta_{k+1} G\Gamma_0 e_k$ then characterizes movement to S_0 by the condition

$$0 < \left(2\beta_{k+1} - \beta_{k+1}\frac{\partial \beta(e)}{\partial e}|_{e=e_k} G\Gamma_0 e_k\right)\beta_{k+1}\frac{\partial \beta(e)}{\partial e}|_{e=e_k} G\Gamma_0 e_k. \tag{14.69}$$

Considering the situation where e_k is infinitesimally close to a point $e \in S_0$ and dividing by $\beta_{k+1}^2 > 0$ suggests that iterations close to a point $e \in S_0$ will tend to move towards S_0 if

$$0 < \left(2 - \frac{\partial \beta(e)}{\partial e} G\Gamma_0 e\right)\frac{\partial \beta(e)}{\partial e} G\Gamma_0 e, \tag{14.70}$$

which is just

$$0 < \frac{\partial \beta(e)}{\partial e} G\Gamma_0 e < 2. \tag{14.71}$$

Consideration of repelling behaviour starting with the inequality $\beta_{k+2}^2 > \beta_{k+1}^2$ (representing movement away from S_0) suggests, using a similar argument that this occurs near to $e \in S_0$ if

$$\textit{either} \quad \frac{\partial \beta(e)}{\partial e} G\Gamma_0 e \; < \; 0 \quad \textit{or} \quad \frac{\partial \beta(e)}{\partial e} G\Gamma_0 e > 2. \tag{14.72}$$

A more complete characterization of behaviours requires the computation of $\frac{\partial \beta(e)}{\partial e}$ with $e \in S_0$. This is now illustrated by writing

$$\left(w(e) + \|G\Gamma_0 e\|^2_{\mathscr{Y}}\right) \beta(e) = \langle e, G\Gamma_0 e \rangle_{\mathscr{Y}}. \tag{14.73}$$

Differentiating and using the fact that $\beta(e) = 0$ then gives

$$\left(w(e) + \|G\Gamma_0 e\|^2_{\mathscr{Y}}\right) \frac{\partial \beta(e)}{\partial e} G\Gamma_0 e = \langle e, \left(G\Gamma_0 + (G\Gamma_0)^*\right) G\Gamma_0 e \rangle_{\mathscr{Y}}. \tag{14.74}$$

Using this in Eq. (14.71), iterations close to a point $e \in S_0$ will move towards S_0 if

$$0 < \frac{\|G\Gamma_0 e\|^2_{\mathscr{Y}} + \langle e, (G\Gamma_0)^2 e \rangle_{\mathscr{Y}}}{\left(w(e) + \|G\Gamma_0 e\|^2_{\mathscr{Y}}\right)} < 2, \tag{14.75}$$

(*A Characterization of Attracting Regions of* S_0).

The nonlinear and high dimensionality of the relationship make a more detailed analysis very complex. It is not attempted here and is unlikely to have any real value in practice other than as a tool for understanding observed behaviours when analyzing data. Note also that behaviours near to points where equality holds in Eq. (14.75) are excluded. An analysis of these special situations would require second (or higher order) order term(s) in the power series expansion of $\beta(e)$ to be included.

Note that satisfaction of (14.75) is equivalent to the two conditions

$$\begin{array}{ll} (A) & \|G\Gamma_0 e\|^2_{\mathscr{Y}} + \langle e, (G\Gamma_0)^2 e \rangle_{\mathscr{Y}} > 0 \qquad \textit{and} \\ (B) & \|G\Gamma_0 e\|^2_{\mathscr{Y}} - \langle e, (G\Gamma_0)^2 e \rangle_{\mathscr{Y}} + 2w(e) > 0. \end{array} \tag{14.76}$$

The first inequality depends only on the properties of $G\Gamma_0$ and hence is a consequence of the choice of Γ_0 only. The second, however, depends also on the user-specified parameters w_0 and ε^2. For example,

1. using the approach of Sect. 14.2.1.1 with the trace normalization assumption of Eq. (14.61), (B) is replaced by

$$(B_1) \qquad \|G\Gamma_0 e\|^2_{\mathscr{Y}} - \langle e, (G\Gamma_0)^2 e \rangle_{\mathscr{Y}} + 2(w_0 + \varepsilon^2 \|e\|^2_{\mathscr{Y}}) > 0. \tag{14.77}$$

   Given a choice of Γ_0 and a data vector $e \in S_0$ satisfying (A),

   a. increasing w_0 increases the range of values of ε^2 that imply that $e \in S_0$ satisfies (B_1) and hence is an attracting point.
   b. A (conservative) estimate of the range of ε^2 that satisfies (B_1) at e is obtained by noting that $|\langle e, (G\Gamma_0)^2 e \rangle_{\mathscr{Y}}| \leq \|G\Gamma_0\|^2 \|e\|^2_{\mathscr{Y}}$ so that the second inequality

(B_1) is satisfied if

$$\|G\Gamma_0\|^2\|e\|^2_{\mathscr{Y}} \;<\; 2\left(w_0+\varepsilon^2\|e\|^2_{\mathscr{Y}}\right). \tag{14.78}$$

This inequality is true if

$$\|G\Gamma_0\|^2 \;<\; 2\varepsilon^2. \tag{14.79}$$

c. More generally, consider the POILC Algorithm starting from e_0 and let $S(e_0)$ be the subset of $e \in S_0$ that satisfy $\|e\|_{\mathscr{Y}} \le \|e_0\|_{\mathscr{Y}}$ and for which condition (A) holds but where

$$\|G\Gamma_0 e\|^2_{\mathscr{Y}} - \langle e, (G\Gamma_0)^2 e\rangle_{\mathscr{Y}} + 2\varepsilon^2\|e\|^2_{\mathscr{Y}} \;\le\; 0. \tag{14.80}$$

Then these points will be added to the set of attracting points if

$$2w_0 > \sup_{e\in S(e_0)}\left(\langle e, (G\Gamma_0)^2 e\rangle_{\mathscr{Y}} - 2\varepsilon^2\|e\|^2_{\mathscr{Y}} - \|G\Gamma_0 e\|^2_{\mathscr{Y}}\right) \tag{14.81}$$

2. For the case considered in Sect. 14.2.1.2, (B) is replaced by

$$(B_2) \qquad \|G\Gamma_0 e\|^2_{\mathscr{Y}} - \langle e, (G\Gamma_0)^2 e\rangle_{\mathscr{Y}} + 2(w_0+\varepsilon^2\|\Gamma_0 e\|^2_{\mathscr{U}}) > 0. \tag{14.82}$$

As above, increasing the values of w_0 and ε^2 increases the size of the attractive component of S_0. It is left as an exercise for the reader to use similar techniques to assess the effect of parameter choices on the size of the attractive set.

3. The reader is also invited to complete the cases considered by considering the case of $w(e)$ as specified in Sect. 14.2.1.3.

Example Finally, the following example suggests that both attracting and repelling surfaces can co-exist. Consider the simple situation where $\mathscr{Y} = \mathscr{R}^2$ with the Euclidean inner product $\langle y, w\rangle = y^T w$ and with

$$G\Gamma_0 = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \quad \textit{so that} \quad (G\Gamma_0)^2 = \begin{bmatrix} 1 & 0 \\ 8 & 1 \end{bmatrix} = 2G\Gamma_0 - I. \tag{14.83}$$

S_0 is therefore characterized, with $e = [e_1, e_2]^T$, by

$$\langle e, G\Gamma_0 e\rangle = e_1^2 + e_2^2 + 4e_1e_2 = 0 \quad \textit{and hence} \quad e_2 = e_1(-2 \pm \sqrt{3}) \tag{14.84}$$

which has two distinct components corresponding to the two signs $\pm$. Next, from Eq. 14.76, compute the first part (A) as

$$\|G\Gamma_0 e\|^2 + \langle e, (G\Gamma_0)^2 e\rangle = 8e_1(2e_1 + e_2) = \pm\, 8e_1^2\sqrt{3}. \tag{14.85}$$

Note that both negative and positive values can be achieved. The negative sign corresponds to part of a repelling component of S_0 whilst the positive sign suggests the possibility of the existence of an attracting component. This is verified by noting that the second part (B) is also positive for all ε^2 and w_0 as

$$\|G\Gamma_0 e\|_{\mathscr{Y}}^2 - \langle e, (G\Gamma_0)^2 e\rangle_{\mathscr{Y}} = \|G\Gamma_0 e\|_{\mathscr{Y}}^2 + \|e\|_{\mathscr{Y}}^2 > 0. \tag{14.86}$$

It is concluded that S_0 contains both repelling and attracting subsets for all choice of w_0 and ε^2. The interested reader may find it interesting and/or instructive to plot these sets in the (e_1, e_2) plane and to confirm, by simulation, that the error update equation $e_{k+1} = (I - \beta_{k+1} G\Gamma_0)e_k$ has different properties depending on the initial error. In particular, unless the initial error lies precisely on the repelling set, the iterations typically have a non-zero limit that lies on the attracting set. Also, if the initial condition lies extremely close to the repelling set, the algorithm starts very slowly with many iterations that result in infinitesimal changes in error until, finally, the iterates "escape" and move more rapidly to a limit on the attractive set.

14.2.9 Switching Algorithms

Sections 14.2.7 and 14.2.8 have pointed out the possibility that, despite the monotonicity property, the failure to satisfy positivity conditions of the form of (14.40) can lead to a flat-lining, plateau-like behaviour represented by convergence to a strictly positive residual error norm. The behaviour is induced by the set S_0 defined in Eq. (14.66) and occurs in situations where convergence rates slow because the error becomes very close to S_0 and, as a consequence, the values of β_{k+1} remain infinitesimally small for many iterations.

Once "trapped" near to an attracting component of S_0, the error sequence may fail, in practical terms, to converge further. This situation cannot then be improved upon as it is a consequence of the properties of $G\Gamma_0$ and, in particular, the choice of Γ_0. This suggests that iterates will be released if S_0 is "moved" away from e_k by changing Γ_0 on iteration $k+1$. This may allow an increase in the values of β_{k+1} that will be computed, begin the process of moving iterates away from S_0 and ultimately improving the achievable error norm reduction. If such a change is applied on each iteration, it may be possible to eliminate the flat-lining problem altogether.

The above discussion leads to the question—*how can Γ_0 be changed from iteration to iteration to eliminate the flat-lining phenomenon?* A complete answer to this question is not possible but an indication of the possibilities can be obtained using some simple constructions. Consider therefore the case of G as a linear, discrete time, state space system operating on a time interval $0 \leq t \leq N$. Choose $M \geq 1$ (non-zero) operators (matrices) $\{\Gamma_0^{(j)}\}_{1 \leq j \leq M}$ each of which is associated with its own set $S_0^{(j)} = \{e \ : \ \langle e, G\Gamma_0^{(j)} e\rangle_{\mathscr{Y}} = 0\}$ of equilibrium points. All contain the points in the intersection

$$S_\infty = \bigcap_{j=1}^{M} S_0^{(j)}. \tag{14.87}$$

On iteration k, Γ_0 is taken to have the value $\Gamma_0(k) = \Gamma_0^{(j_k)}$ where j_k is chosen from the range $1 \leq j \leq M$ in such a way that any index j in this set will be chosen, for some $\tilde{M} \geq M$, at least once in each index set $k, k+1, \ldots, k+\tilde{M}$ for all $k \geq 0$. The intuition behind the intersection property is that convergence to any set $S_0^{(j)}$ is avoided by the changes leaving motion to S_∞ as the only remaining option. If $S_\infty = \{0\}$, then, intuitively, the error converges to zero. This is formalized as follows using the objective function used in Sect. 14.2.1.1 with the trace normalization assumption of Eq. (14.61).

Algorithm 14.2 *(Single Parameter Optimal Iteration with Switching)* Using the assumptions and construction provided above, a Parameter Optimal Iterative Control Algorithm with switching can be defined by the input update for iteration $k+1$,

$$u_{k+1} = u_k + \beta_{k+1}\Gamma_0(k+1)e_k, \tag{14.88}$$

where

$$\beta_{k+1} = \frac{\langle e_k, G\Gamma_0(k+1)e_k\rangle_{\mathscr{Y}}}{\left((w_0 + \varepsilon^2\|e_k\|_{\mathscr{Y}}^2) + \|G\Gamma_0(k+1)e_k\|_{\mathscr{Y}}^2\right)} \tag{14.89}$$

is the value of β that minimizes the objective function

$$\begin{aligned} &J_{k+1}(\beta) = \|e\|_{\mathscr{Y}}^2 + w_{k+1}\beta^2, \\ \textit{with} \quad &w_{k+1} = w_0 + \varepsilon^2\|e_k\|_{\mathscr{Y}}^2, \quad w_0 \geq 0, \quad \varepsilon^2 \geq 0, \textit{ and } w_1 > 0, \end{aligned} \tag{14.90}$$

subject to the constraint $e = (I - \beta G\Gamma_0(k+1))\, e_k$.

The resulting iteration dynamics has the properties of generating a monotonically decreasing error norm sequence with error sequence $\{e_k\}_{k\geq 0}$ that asymptotically gets arbitrarily close to S_∞. If $S_\infty = \{0\}$, then the error sequence converges to zero.

Proof of the Stated Convergence Properties From optimality, the choice of $\beta = 0$ is sub-optimal so monotonicity follows from the inequality

$$\|e_{k+1}\|_{\mathscr{Y}}^2 + w_{k+1}\beta_{k+1}^2 \leq \|e_k\|_{\mathscr{Y}}^2, \tag{14.91}$$

from which, for all $k \geq 0$,

$$\|e_{k+1}\|_{\mathscr{Y}}^2 \leq \|e_0\|_{\mathscr{Y}}^2 - \sum_{j=0}^{k} w_{j+1}\beta_{j+1}^2. \tag{14.92}$$

Monotonicity implies that the limit $\lim_{k\to\infty} \|e_k\|_{\mathscr{Y}} = E_\infty \geq 0$ exists. Either $E_\infty = 0$ (when the error sequence converges to zero and the proof is complete) or $E_\infty > 0$.

If $E_\infty > 0$, then, for all parameter choices, $\lim_{k\to\infty} \beta_{k+1} = 0$. In particular, as the error differences $e_{j+1} - e_j$ then converge to zero, it follows that, for $1 \le j \le M$, $\lim_{k\to\infty} \langle e_k, G\Gamma_0^{(j)} e_k\rangle_{\mathscr{Y}} = 0$. Continuity arguments then prove the asymptotic proximity of the error sequence to S_∞ and convergence to zero if $S_\infty = \{0\}$. □

The use of switching has the theoretical potential to significantly improve the convergence characteristics of POILC although the choice of M, the $\{\Gamma_0^{(j)}\}_{1\le j\le M}$ and their sequencing to create $\{\Gamma_0(k+1)\}_{k\ge 0}$ requires careful consideration. Ultimately, their choice will be guided by two factors namely, reducing the asymptotic value of the error norm and ensuring that S_∞ only contains error signals of an acceptable dynamic form. The practical feasibility of the algorithm will then depend on acceptable convergence properties over the, probably relatively small, number of iterations that are ideally used in practice.

Finally, the following examples link properties of S_∞ to the choice of $\{\Gamma_0^{(j)}\}_{1\le j\le M}$.

1. If, for example, $\sum_{j=1}^{M} \alpha_j \left(G\Gamma_0^{(j)} + (G\Gamma_0^{(j)})^*\right)$ is strictly positive definite for some choice of scalars $\{\alpha_j\}_{1\le j\le M}$, then the reader can verify that $S_\infty = \{0\}$. This suggests that the richer the set used, the more likely it is that convergence will be achieved.
2. For an asymptotically stable, single-input, single-output, discrete time, state space system with transfer function $G(z)$ with Γ_0 the adjoint of an asymptotically stable, SISO, discrete time, state space system with transfer function $K(z)$, the inner product can be identified intuitively if N is very large, with

$$\langle e, GK^*e\rangle_{\mathscr{Y}} \approx \frac{1}{2\pi i}\oint_{|z|=1} |e(z)|^2 Re\left[G(z)K(z^{-1})\right]\frac{dz}{z}, \tag{14.93}$$

   which suggests that the associated set $S_0 = \{e : \langle e, GK^*e\rangle_{\mathscr{Y}} = 0\}$ requires a balance between the "energy" in the frequency range where the phase of $G(z)K(z^{-1})$ lies in the interval $(-\frac{\pi}{2}, \frac{\pi}{2})$ and the energy at other frequencies. The introduction of several such relationships then defines S_∞ as the region where M such energy balances are satisfied. If both G and K are low pass systems, this will require a balance between the energies at "low" and "high" frequencies. Combining this with the monotonicity properties then suggests that S_∞ may often be associated with signals of pre-dominantly higher frequency content.
3. The operators $\{\Gamma_0^{(j)}\}_{1\le j\le M}$ need not be dynamically complex. To illustrate this fact, suppose that G and K are linear, SISO, asymptotically stable, discrete systems with state space models $S(A, B, C, D)$ and $S(A_K, B_K, C_K, D_K)$. The iterations are assumed to operate on the interval $0 \le t \le N$ and the relative degree-based, shift techniques of Chap. 4 have been used to ensure that the matrices, G and K, in the supervector description are invertible (that is, $D \ne 0$ and $D_K \ne 0$). The switching mechanism is introduced using a simple first order filter $F(z) = (1 - \lambda z^{-1})^{-1}$ with state space model $S(\lambda, \lambda, 1, 1)$ and supervector description associated with the matrix

$$F(\lambda) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \lambda & 1 & 0 \cdots & 0 & \\ \vdots & & & & \vdots \\ \lambda^{N-1} & \lambda^{N-2} & \cdots & 1 & 0 \\ \lambda^N & \lambda^{N-1} & \cdots & \lambda & 1 \end{bmatrix} \tag{14.94}$$

The choice of $\lambda = 0$ generates the "filter" $F(z) \equiv 1$ when no filtering occurs. More generally, assume the Euclidean inner product $\langle e, w\rangle_{\mathscr{Y}} = e^T w$ in both $\mathscr{U} = \mathscr{Y} = \mathscr{R}^{N+1}$ and set

$$\Gamma_0^{(j)} = F^*(\lambda_j)K^* = \left(KF(\lambda_j)\right)^*, \qquad 1 \le j \le M, \tag{14.95}$$

where $\{\lambda_j\}_{1\le j\le M}$ is a set of real, distinct scalars in the open interval $(-1, 1)$ and $M \ge N + 1$. Note that the adjoint in this case is simply the transpose of the matrix and, for single-input, single-output systems, K commutes with all matrices $F(\lambda_j),\ \ 1 \le j \le M$.

Algorithm 14.3 (*SISO POILC: Switching and Anti-causal Filters*) A switching algorithm based on the above construction uses the update rule

$$\begin{gathered} u_{k+1} = u_k + \beta_{k+1}\Gamma_0(k+1)e_k = u_k + \beta_{k+1}K^* e_k^F \\ \textit{where} \qquad e_k^F = F^*(\lambda_{j_k})e_k \end{gathered} \tag{14.96}$$

where the indices $\{j_k\}_{k\ge 1}$ satisfy the assumptions of Algorithm 14.2.

In essence the algorithm uses the adjoint K^* as an operator on the filtered error and the bandwidth of the filter is systematically changed from iteration to iteration. The filtering operation is done in reverse time using a state space representation of the adjoint with zero terminal state condition.

Properties of the above algorithm are stated formally as follows,

Theorem 14.6 (SISO POILC: Switching and Anti-causal Filters) *The effect of the choices made above is to ensure that* $S_\infty = \{0\}$ *and hence that the tracking error sequence converges to zero.*

Proof S_∞ is defined by M equations as

$$S_\infty = \{e \ : \ \langle e, GF^*(\lambda_j)K^* e\rangle_{\mathscr{Y}} = 0, \quad 1 \le j \le M\}. \tag{14.97}$$

Equivalently, for $1 \le j \le M$, with $w = G^* e$ and $\hat{G}$ as the inverse system of G,

$$\begin{aligned} \langle G^*e, F^*(\lambda_j)K^*\hat{G}^*G^*e\rangle_{\mathscr{Y}} &= \langle G^*e, F^*(\lambda_j)K^*\hat{G}^*G^*e\rangle_{\mathscr{Y}} \\ &= \langle w, \left(\hat{G}KF(\lambda_j)\right) w\rangle_{\mathscr{Y}} = 0. \end{aligned} \tag{14.98}$$

That is, using the form of $F(\lambda)$, $\psi(\lambda, w) = \langle w, \left(\hat{G}KF(\lambda)\right) w\rangle_{\mathscr{Y}}$ is a polynomial in λ of degree N with $M \ge N + 1$ distinct roots. It is therefore identically zero

and, in particular the coefficients in the polynomial are all zero. Noting that $\hat{G}KF(\lambda)$ represents a linear dynamical system and that $w = [w(0), w(1), \ldots, w(N)]^T$ is a time series, the coefficient of λ^N is easily computed to be $D^{-1}D_K w(0)w(N) = 0$. That is, $w(0) = 0$ and/or $w(N) = 0$. In all situations, the time interval can be reduced by one or two samples leaving a characterization of S_∞ that is structurally identical but of lower dimension. Applying an inductive argument sequentially shortens the interval to indicate that $w(t) = 0$ at all samples except, possibly, at one point $t = t^*$ where $DD_K w^2(t^*) = 0$. That is $w(t^*) = 0$ which proves that $w = 0$, that $e = 0$ and hence that $S_\infty = \{0\}$. □

Note: The example illustrates the theoretical validity of the switching concept using simple control elements. The price paid for convergence is that $M \geq N + 1$ filters are needed and N is typically very large in practice. At least one sweep of the filter set may be required to ensure satisfactory accuracy so that convergence rates may be slow. An alternative viewpoint is that the switching is simply there to achieve improvements on the performance to be expected in the absence of switching.

14.3 Robustness of POILC: The Single Parameter Case

In single parameter Optimal Iteration Algorithm 14.1, $\Gamma_0 e_k$ can be interpreted as a tentative descent direction for the problem with optimization being used to choose the step length β_{k+1}. Intuitively, this will be tolerant of some modelling errors as multiplicative modelling errors will introduce perturbations of $G\Gamma_0$ that retain the descent characteristic provided that the value of β is small enough. Examination of the form of $\beta(e)$ suggests that this "smallness" will be associated with, in particular, larger values of ε^2 and/or w_0.

A general analysis of the robustness of Algorithm 14.1 is not available but an indication of robustness properties is possible. The simplest approach examines a special case of interest.

14.3.1 Robustness Using the Right Inverse

More precisely, suppose that G is a right-invertible model of the plant dynamics in the form of an asymptotically stable, ℓ-input, m-output, discrete, state space system with associated transfer function matrix $G(z)$. Plant dynamics is assumed to take place on an interval $0 \leq t \leq N$ and the inner product in $\mathscr{Y} = \mathscr{R}^{m(N+1)}$ is

$$\langle y, w \rangle_{\mathscr{Y}} = \sum_{t=0}^{N} y^T(t) Q w(t), \quad \textit{where} \quad Q = Q^T > 0 \tag{14.99}$$

is the chosen $m \times m$, positive definite weight matrix.

A Parameter Optimal Algorithm can then be constructed using a right inverse G_R of a model G in a similar manner to Algorithm 6.1 in Chap. 6. The input update relation takes the form

$$u_{k+1} = u_k + \beta_{k+1}\Gamma_0 e_k \quad with \quad \Gamma_0 = G_R \quad and \quad GG_R = I \quad (the \quad identity). \tag{14.100}$$

The objective function used has the form

$$J(\beta) = \|e\|_{\mathscr{Y}}^2 + w_{k+1}\beta^2 \quad with \quad w_{k+1} = w(e_k) \tag{14.101}$$

where $w(e)$ is constructed using the ideas of Sect. 14.2.1.1 (with or without trace normalization), Sect. 14.2.1.2 or 14.2.1.3. The optimizing value of β_{k+1} predicted by the model is obtained as

$$\beta_{k+1} = \beta(e_k) \quad where \quad \beta(e) = \frac{\|e\|_{\mathscr{Y}}^2}{w(e) + \|e\|_{\mathscr{Y}}^2} \in [0, 1]. \tag{14.102}$$

This value can be computed from measured error data e_k on each iteration even if modelling errors are present. The result of the calculation can then be used in input updating. Suppose that the modelling error is a left multiplicative modelling error U with associated $m \times m$ transfer function matrix $U(z)$. The actual plant is then characterized by the operator UG and the error evolution is described by

$$e_{k+1} = (I - \beta_{k+1}U)e_k \tag{14.103}$$

The robustness analysis of Chap. 6, Sect. 6.2 then shows that the monotonicity property $\|e_{k+1}\|_{\mathscr{Y}} < \|e_k\|_{\mathscr{Y}}$ is retained for all $k \geq 0$ if, in the topology induced by $\langle \cdot, \cdot \rangle_{\mathscr{Y}}$,

$$U + U^* > \beta_{k+1}U^*U, \quad for \ all \quad k \geq 0, \tag{14.104}$$

a condition that is satisfied only if, during the iterations,

$$U + U^* > \left(\sup_{k \geq 0} \beta_{k+1}\right) U^*U. \tag{14.105}$$

The sequences $\{\beta_{k+1}\}_{k \geq 0}$ and $\{e_k\}_{k \geq 0}$ are unknown before iterations are initiated but a sufficient condition is easily obtained by replacing the condition by

$$U + U^* > \beta^* U^*U \quad where \quad \beta^* = \sup_{\|e\| \leq \|e_0\|} \beta(e) \in (0, 1]. \tag{14.106}$$

This can be written as a robustness condition expressed in terms of the spectral radius

$$r\left((I - \beta^* U)^*(I - \beta^* U)\right) < 1. \tag{14.107}$$

As in Sect. 6.2, this then provides the frequency domain condition for discrete state space systems

$$\sup_{|z|=1} r\left(Q^{-1}(I-\beta^* U^T(z^{-1}))Q(I-\beta^* U(z))\right) < 1. \tag{14.108}$$

The interpretation of the result is identical to that of the fixed parameter case and, in particular, again requires U to have positivity properties. Note the additional information provided from the equation for $\beta(e)$ and the fact that robustness of the algorithm will increase as β^* decreases. The details depend on the chosen form for $w(e)$. More precisely,

1. for the choices in Sects. 14.2.1.1 and 14.2.1.3, $\beta^* = \beta(e_0) = \beta_1$. That is the value β_1 computed on the first iteration decides the monotonic robustness of all of the following iterations.
2. Using $w(e)$ from Sect. 14.2.1.2,

$$\beta^* = \sup_{\|e\| \le \|e_0\|} \left(\frac{\|e\|^2_{\mathscr{Y}}}{w_0 + \varepsilon^2 \|G_R e\|^2_{\mathscr{U}} + \|e\|^2_{\mathscr{Y}}} \right), \tag{14.109}$$

   the value of which depends on the form and properties of G_R.

All of the above verify the intuition that robustness increases if either or both parameter values w_0 and ε^2 increase. In addition, the tracking accuracy, as measured by $\|e_0\|^2_{\mathscr{Y}}$, achieved using u_0 on the zeroth iteration, influences the robustness of the algorithm and provides a convincing motivation for the user to choose it carefully.

14.3.2 Robustness: A More General Case

The analysis of the robustness of POILC can be related, more generally, to that of the fixed parameter case. Suppose that,

1. G_R is replaced by an operator Γ_0 and that the positivity conditions in Eq. (14.40) are satisfied. Suppose also that the plant is described by the model G subjected to a left multiplicative modelling error U and that U satisfies a positivity property of the form

$$U + U^* \ge \varepsilon_0^2 I \quad \textit{for some real scalar} \quad \varepsilon_0^2 > 0. \tag{14.110}$$

2. As U may be unknown, the gains $\{\beta_{k+1}\}_{k\ge 0}$ used in POILC will be computed using the model G of the plant.
3. It follows that the gains that can be used in both the fixed parameter and POILC cases are positive. The gain in POILC is given by $\beta_{k+1} = \beta(e_k)$ where $\beta(e)$ is now given by

$$\beta(e) = \frac{\langle e, G\Gamma_0 e\rangle_{\mathscr{Y}}}{w(e) + \|G\Gamma_0 e\|^2_{\mathscr{Y}}}. \tag{14.111}$$

4. Let $\beta_f > 0$ be the largest gain such that the fixed parameter, perturbed process UG has the property of robust monotonic reduction of the error norm sequence using any value of β in the range $0 < \beta < \beta_f$.
5. It follows that the variations in gain value from iteration to iteration generated by POILC will also produce monotonic convergence provided that

$$\sup_{k\geq 0} \beta_{k+1} < \beta_f. \tag{14.112}$$

This immediately yields an alternative criterion for single-parameter POILC problems in the form of properties of $\beta(e)$, namely,

Theorem 14.7 (Robustness of POILC: A General Result) *Using the notation and assumptions of the preceding discussion, a sufficient condition for robust monotonic convergence of the error norm sequence in the presence of the left-multiplicative modelling error U is that* $\|e_0\|_{\mathscr{Y}}$ *should be small enough and/or* w_0 *and* ε^2 *large enough to ensure that*

$$\beta^* = \sup_{\|e\|\leq\|e_0\|} \beta(e) < \beta_f. \tag{14.113}$$

This again indicates that robustness conditions for the fixed parameter case can be used to generate robustness conditions for the related POILC approach and that robustness can be influenced by parameter choice and judicious choice of u_0. If β_f is not known, the result provides trend information. If β_f can reasonably be estimated, then an evaluation of sup $\beta(e)$ may be considered. For a gradient-based POILC algorithm where $\Gamma_0 = G^*$, the analysis requires evaluation of β_f followed by an investigation of the effect of parameter choice and u_0 to satisfy the condition

$$\sup_{0<\|e\|_{\mathscr{Y}}\leq\|e_0\|_{\mathscr{Y}}} \frac{\|G^*e\|^2}{\left(w(e) + \|GG^*e\|^2_{\mathscr{Y}}\right)} < \beta_f. \tag{14.114}$$

The supremum is achieved for some vector e satisfying $\|e\|_{\mathscr{Y}} = \|e_0\|_{\mathscr{Y}}$ so that evaluation of the left hand side is an optimization problem with an equality constraint. The solution is obtainable as a stationary point of the Lagrangian

$$\mathscr{L}[e, \lambda] = \beta(e) + \lambda(\|e_0\|^2_{\mathscr{Y}} - \|e\|^2_{\mathscr{Y}}) \tag{14.115}$$

with scalar Lagrange Multiplier λ.

14.4 Multi-Parameter Learning Control

Parameter Optimal Iterative Learning Control (POILC) is motivated by the analysis of Sect. 14.1. In contrast to the single parameter approach of Sect. 14.2, this section uses a constraint set Ω_Γ of a very specific, *multi-parameter* form containing n_p free parameters $\{\beta^{(j)}\}_{1\le j\le n_p}$ regarded as elements of a parameter vector

$$\beta = [\beta^{(1)}, \beta^{(2)}, \ldots, \beta^{(n_p)}]^T \in \mathscr{R}^{n_p}. \tag{14.116}$$

14.4.1 The Form of the Parameterization

The operator $\Gamma(\beta)$ in the input update rule is now dependent on β and the optimization for iteration $k+1$ searches over inputs

$$u = u_k + \Gamma(\beta)e_k \quad \textit{with} \quad \beta \in \mathscr{R}^{n_p}. \tag{14.117}$$

The $n_p \times 1$ parameter vector β is a design variable with the value chosen for iteration $k+1$ denoted by β_{k+1} leading to an error response e_{k+1}. The spaces $\mathscr{Y}$ and $\mathscr{U}$ are finite dimensional with signals and operators represented by matrices. The spaces are given the inner products $\langle y, w\rangle_{\mathscr{Y}} = y^T Q w$ and $\langle u, v\rangle_{\mathscr{U}} = u^T R v$ where both Q and R are symmetric positive definite matrices. The operator $\Gamma(\beta) : \mathscr{Y} \to \mathscr{U}$ can in principle take any functional dependence on the parameters but, in the "linear" spirit of this text, the structure is simplified by

Linearity and No Redundancy Assumptions: *More precisely,* $\Gamma(\beta)$ is *assumed to depend linearly on* β *so that*

$$\Gamma(\lambda_1\beta + \lambda_2\beta') = \lambda_1\Gamma(\beta) + \lambda_2\Gamma(\beta'), \quad (\textit{and hence } \Gamma(0) = 0), \tag{14.118}$$

for all $\beta, \beta' \in \mathscr{R}^{n_p}$ *and real scalars* λ_1 *and* λ_2. *In addition, it is assumed that*

$$\Gamma(\beta) = 0 \quad \textit{implies that} \quad \beta = 0, \tag{14.119}$$

which ensures that there is no redundancy in the parameterization.

An important consequence is that there exists an alternative representation described by

$$\Gamma(\beta)e = M(e)\beta, \quad \textit{for all} \quad e \in \mathscr{Y} \quad \textit{and} \quad \beta \in \mathscr{R}^{n_p}, \tag{14.120}$$

where $M(e)$ is linear in e with $M(0) = 0$. For example, if $n_p = 2$ and $e = [e_1, e_2]^T$ and

$$\Gamma(\beta) = \begin{bmatrix} \beta^{(1)} & -\beta^{(2)} \\ 2\beta^{(1)} + 3\beta^{(2)} & 4\beta^{(1)} - 5\beta^{(2)} \end{bmatrix}, \quad \textit{then} \quad M(e) = \begin{bmatrix} e_1 & -e_2 \\ 2e_1 + 4e_2 & 3e_1 - 5e_2 \end{bmatrix}. \tag{14.121}$$

More generally, writing, with suitable choice of coefficient matrices $\{\tilde{\Gamma}_j\}_{1 \le j \le n_p}$,

$$\Gamma = \sum_{j=1}^{n_p} \tilde{\Gamma}_j \beta^{(j)}, \quad \textit{then} \quad M(e) = \left[\tilde{\Gamma}_1 e, \tilde{\Gamma}_2 e, \ldots, \tilde{\Gamma}_{n_p} e \right]. \tag{14.122}$$

The input and error update equations hence have the form

$$u = u_k + M(e_k)\beta \quad \textit{and} \quad e = e_k - GM(e_k)\beta. \tag{14.123}$$

14.4.2 Alternative Forms for Ω_Γ and the Objective Function

The discussion follows the form used in the single parameter case in Sects. 14.2.1.1–14.2.1.3.

14.4.2.1 Approach One

Consider the class of control laws $u = u_k + \Gamma e_k$ and the optimization problem of minimizing (14.17) over all possible operators $\Gamma \in \Omega_\Gamma$, with

$$\Omega_\Gamma = \{\Gamma \ : \ \Gamma = \Gamma(\beta)\}, \tag{14.124}$$

where $\Gamma(\beta) : \mathscr{Y} \to \mathscr{U}$ is a parameter dependent operator chosen by the user and β is a free parameter vector. The first calculation replaces $tr[\Gamma^* \Gamma]$ by the quadratic form

$$tr[\Gamma^*(\beta)\Gamma(\beta)] = \beta^T W \beta, \tag{14.125}$$

where the $n_p \times n_p$ matrix W is symmetric with elements,

$$W_{ij} = tr[\tilde{\Gamma}_i^* \tilde{\Gamma}_j] = tr[Q^{-1} \tilde{\Gamma}_i^T R \tilde{\Gamma}_j] = W_{ji}. \tag{14.126}$$

The non redundancy assumption and the positivity of the trace norm on matrices ensures that W is positive definite.

The objective function suggested by this expression and Eq. (14.17) takes the form

$$\begin{array}{ll} J(\Gamma) = \|e\|^2_{\mathscr{Y}} + \beta^T W_{k+1}\beta & \textit{with weight matrix} \\ W_{k+1} = W(e_k) \qquad \textit{where} & \\ W(e) = \left(w_0 + \varepsilon^2\|e\|^2_{\mathscr{Y}}\right) W & \textit{and} \quad w_0 \geq 0. \end{array} \tag{14.127}$$

As in Sect. 14.2.1.1, the parameter $\varepsilon^2 > 0$ is the notional link to NOILC but if, this link is unimportant, it is possible to allow any choice of ε^2 and w_0 satisfying

$$w_0 \geq 0, \quad \varepsilon^2 \geq 0 \quad \textit{and} \quad w_0 + \varepsilon^2 > 0. \tag{14.128}$$

In addition, it may make practical sense

1. to avoid problems such as those seen in Sect. 14.4 by using *trace normalization*. This could take many forms but the most natural is the replacement

$$W \mapsto n_p \left(\frac{1}{tr[W]}\right) W, \qquad (\textit{multi-parameter trace normalization}), \tag{14.129}$$

   a mapping that takes the identity in $\mathscr{R}^{n_p}$ into itself.
2. Alternatively, W may simply be a simple and convenient choice for the user.

14.4.2.2 Approach Two

Following Sect. 14.2.1.2, set

$$\Omega_\Gamma(e) = \{\Gamma(e,\beta) \ : \ \Gamma(e,\beta) = \Gamma(\beta)ee^T Q\|e\|^{-2}_{\mathscr{Y}}\}. \tag{14.130}$$

The control laws considered then take the form

$$u = u_k + \Gamma(e_k,\beta)e_k = u_k + \Gamma(\beta)e_k. \tag{14.131}$$

That is, despite the different parameterization, the control update rule is identical to that used on Approach One. Computing

$$\begin{array}{rl} tr[\Gamma^*(e,\beta)\Gamma(e,\beta)] &= tr[ee^T Q\Gamma^*(\beta)\Gamma(\beta)ee^T Q\|e\|^{-4}_{\mathscr{Y}}] \\ &= \|e\|^{-2}_{\mathscr{Y}}\langle e, \Gamma^*(\beta)\Gamma(\beta)e\rangle_{\mathscr{Y}} \\ &= \|e\|^{-2}_{\mathscr{Y}}\langle \Gamma(\beta)e, \Gamma(\beta)e\rangle_{\mathscr{U}} \\ &= \|e\|^{-2}_{\mathscr{Y}}\beta^T W_1(e)\beta, \end{array} \tag{14.132}$$

where $W_1(e)$ is symmetric as it has elements

$$(W_1(e))_{ij} = \langle\tilde{\Gamma}_i e, \tilde{\Gamma}_j e\rangle_{\mathscr{U}} = (W_1(e))_{ji} \tag{14.133}$$

It is also positive as the trace norm is positive on matrices.

Theorem 14.8 *Using the above construction, $W_1(e)$ is positive definite if, and only if, the vectors $\{\tilde{\Gamma}_j e\}_{1\le j\le n_p}$ are linearly independent.*

Proof The result is a consequence of noting that $\beta^T W_1(e)\beta = \|v\|^2_{\mathscr{U}}$ where $v = \Gamma(\beta)e = \sum_{j=1}^{n_p} \beta^{(j)}\tilde{\Gamma}_j e$. □

The objective function therefore naturally takes the form

$$\begin{aligned} J(\Gamma) &= \|e\|^2_{\mathscr{Y}} + \varepsilon^2\|e\|^2_{\mathscr{Y}} tr[\Gamma^*(e_k,\beta)\Gamma(e_k,\beta)] \\ &= \|e\|^2_{\mathscr{Y}} + \varepsilon^2\beta^T W_1(e_k)\beta. \end{aligned} \tag{14.134}$$

However, the preferred choice has the form,

$$J(\Gamma) = \|e\|^2_{\mathscr{Y}} + \beta^T W_{k+1}\beta \quad \textit{with weighting matrix} \quad W_{k+1} > 0, \tag{14.135}$$

defined by

$$W_{k+1} = W(e_k) \quad \textit{where} \quad W(e) = W_0 + \varepsilon^2 W_1(e). \tag{14.136}$$

To ensure that an optimization problem is well-conditioned, it is advisable to ensure that $W_{k+1} > 0$ for all non-zero $e \in \mathscr{Y}$. This is achieved if the $n_p \times n_p$ matrix W_0 is both symmetric and positive definite and $\varepsilon^2 \ge 0$.

14.4.2.3 Approach Three

A third approach to creation of a parameter optimization problem is to use a parallel process to that used in Sect. 14.2.1.3. Formally, the idea is to replace W_{k+1} in Eq. (14.136) by

$$W_{k+1} = W(e_k) \quad \textit{where} \quad W(e) = W_0 + \varepsilon^2\|e\|^2_{\mathscr{Y}} W_\infty \tag{14.137}$$

and W_∞ is symmetric and positive definite and satisfies the inequality,

$$W_1(e) \le \|e\|^2_{\mathscr{Y}} W_\infty \quad \textit{for all} \quad e \in \mathscr{Y}. \tag{14.138}$$

Such a matrix is non-unique and a suitable candidate might be difficult to compute. The set of candidates

$$S_W = \{\tilde{W} \ : \ W(e) \le \tilde{W}, \quad \textit{for all} \ \|e\| = 1\} \tag{14.139}$$

is easily seen to be closed and convex and can be regarded as being bounded as

$$\beta^T W_1(e)\beta = \|\Gamma(\beta)e\|^2_{\mathscr{U}} \leq \|\Gamma(\beta)\|^2\|e\|^2_{\mathscr{Y}} \leq \beta^T\beta\left(\sup_{\beta^T\beta=1}\|\Gamma(\beta)\|^2_{\mathscr{U}}\right)\|e\|^2_{\mathscr{Y}} \tag{14.140}$$

and hence it is only necessary to consider W_∞ satisfying a bound such as

$$W_\infty \leq \sup_{\beta^T\beta=1}\|\Gamma(\beta)\|^2_{\mathscr{U}} I. \tag{14.141}$$

An alternative viewpoint is simply to regard W_0 and W_∞ as weight matrices chosen by the user to achieve the required algorithm performance.

It is of interest to note that S_W contains elements with minimality properties. The proof of the statement depends the notion of partial and total orders on sets and the application of Zorn's Lemma, which is beyond the scope of this text. The details are left for the reader to explore if he or she so wishes. The computation of such a "minimal" element is non-trivial and may have little practical benefit.

Theorem 14.9 (Choices of W_∞ with Minimality Properties) *Using the notation of the discussion above, suppose that there exists a vector e such that the vectors* $\{\Gamma_j e\}_{1\leq j\leq n_p}$ *are linearly independent. Then, the closed, convex set* S_W *contains only positive definite, symmetric matrices. In particular, there exists at least one choice of* $W_\infty \in S_W$ *with the minimality property that, if* $W \in S_W$ *and* $W \leq W_\infty$*, then* $W = W_\infty$.

Proof Using Theorem 14.8 indicates the positive definiteness of every element of S_W. Next, the symbol $\leq$ is a *partial order* on S_W that relates some (but not all) pairs of matrices by the statement $A \leq B$. A *totally ordered subset* T_W of S_W is a subset of S_W for which, either $A \leq B$ or $B \leq A$ for all pairs (A, B) in T_W. It is easy to prove that every totally ordered subset T_W of S_W has a *lower bound* W_{T_W} in S_W satisfying $W_{T_W} \leq W$ for all $W \in T_W$. The result now follows from Zorn's Lemma. □

14.4.3 The Multi-parameter POILC Algorithm

The previous sections have considered various forms of objective functions motivated by the link to NOILC explored in Sect. 14.1. Whatever the chosen form of $W(e)$, and hence W_{k+1}, the formula for $J(\Gamma)$ becomes

$$J(\Gamma) = \|e\|^2_{\mathscr{Y}} + \|\beta\|^2_{W_{k+1}} \quad (\textit{the Multi-parameter Objective Function}) \tag{14.142}$$

This is a quadratic objective function with iteration dependent weight matrix W_{k+1} where the norm $\|\beta\|_{W_{k+1}}$ in $\mathscr{R}^{n_p}$ is induced by the inner product $\langle\beta, \gamma\rangle_{W_{k+1}} =$

$\beta^T W_{k+1}\gamma$. It has been constructed using the relationships that were created in the discussion and proof of Theorem 14.1. In this sense, minimization of $J(\Gamma)$ with respect to $\beta \in \mathscr{R}^{n_p}$ can be viewed as generating a parametric approximation to an iteration of NOILC Algorithm 9.1 when the underlying spaces are finite dimensional. The general form of *Parameter Optimal Iterative Learning Control* (POILC) algorithm that this creates is as follows.

Algorithm 14.4 (*Multi-Parameter Optimal Iterative Learning Control*) Given a system $y = Gu + d$ with output $y \in \mathscr{Y}$, input $u \in \mathscr{U}$ and reference signal $r \in \mathscr{Y}$, suppose that

1. a linear parameterization $\Gamma(\beta)$ has been chosen by the user to form candidate inputs using Eq. (14.117) and that
2. a form $W(e)$ for the weight used to generate the matrix $W_{k+1} = W(e_k)$ on each and every iteration $k+1$ has been selected.

The iterative process is initiated by a control signal u_0 with associated tracking error e_0. The resultant iterative control algorithm constructs, for all $k \geq 0$, the input u_{k+1} to be used on iteration $k+1$ by using the parameter β_{k+1} defined as the solution of the optimization problem

$$\beta_{k+1} = \arg \min_{\beta \in \mathscr{R}^{n_p}} \{J(\Gamma) \ : \ \textit{subject to } \ e = r - Gu - d = e_k - G\Gamma(\beta)e_k\} \quad (14.143)$$

where $J(\Gamma)$ is defined by Eq. (14.142).

Note that the optimization problem is well-defined and has a unique solution as $W_{k+1} > 0$ whenever $e_k \neq 0$.

Despite the formal algorithm definition, the choice of β_{k+1} is based on a formula derived below

Theorem 14.10 (POILC: General Monotonicity and Convergence Properties) *Using the notation defined above, Algorithm 14.4 can be implemented using the formula* $\beta_{k+1} = \beta(e_k),\ k \geq 0$, *for the optimal parameter choice where*

$$\beta(e) = (W(e) + M^T(e)G^T QGM(e))^{-1} M^T(e)G^T Qe. \quad (14.144)$$

The resultant error and parameter sequences $\{e_k\}_{k\geq 0}$ *and* $\{\beta_k\}_{k\geq 1}$ *satisfy the monotonicity and convergence conditions*

$$\begin{array}{c} \|e_{k+1}\|_{\mathscr{Y}} \ \leq \ \|e_k\|_{\mathscr{Y}}, \quad \textit{for all} \ \ k \geq 0 \\ \textit{and} \quad \sum_{k=0}^{\infty} \ \|\beta_{k+1}\|^2_{W_{k+1}} < \infty. \end{array} \quad (14.145)$$

Proof Write the constraint $e = e_k - G\Gamma(\beta)e_k = e_k - GM(e_k)\beta$ as a linear map of β into e. Note that $GM(e_k)$ maps $\mathscr{R}^{n_p}$ into $\mathscr{Y}$ and hence has an adjoint $W_{k+1}^{-1} M^T(e_k)G^T Q$. The optimizing solution

$$\beta_{k+1} = (GM(e_k))^* e_{k+1} = W_{k+1}^{-1} M^T(e_k)G^T Q(e_k - GM(e_k)\beta_{k+1}). \quad (14.146)$$

The formula for β_{k+1} follows by rearranging. Monotonicity of $\{\|e_k\|_{\mathscr{Y}}\}_{k\geq 0}$ and convergence of the series follows as in the proof of Theorem 14.3. □

In computational terms, the optimal parameter vector β_{k+1} is defined by a matrix computation constructed from R, G, the functional form of $M(e)$ and $W(e)$ and the data vector e_k. The columns of $M(e_k)$ are computed by evaluation of the operations $\tilde{\Gamma}_j e_k$, $1 \leq j \leq n_p$. Each column of $GM(e_k)$ can then be computed as the response of the model G from zero initial conditions to the equivalent column of $M(e_k)$. Similar calculations combined with evaluation of the norm of e_k form the basis of evaluation of $W_{k+1} = W(e_k)$.

14.4.4 Choice of Multi-parameter Parameterization

Algorithm 14.4 has a well-defined solution for every chosen parameterization. Most of the properties seen in the discussion for the single parameter case hold in the multi-parameter case and the role of the weights are relatively unchanged. It is left as an exercise for the reader to explore these observations. The most complex property to generalize to the multi-parameter case is that of flat-lining/plateauing of the plot of $\|e_k\|_{\mathscr{Y}}$ against iteration index k and its removal by switching (Sect. 14.2.9). For the case where switching is not included, the set defining any plateau effects is

$$\begin{aligned} S_0 &= \{e \,:\, \beta(e) = 0\} = \{e \,:\, M^T(e)G^T Qe = 0\} \\ &= \{e \,:\, e^T \tilde{\Gamma}_j^T G^T Qe = 0,\ 1 \leq j \leq n_p\}. \end{aligned} \tag{14.147}$$

If $S_0 = \{0\}$, then the error converges to zero. If $S_0 \neq \{0\}$, convergence to a non zero error is likely. By analogy with the single parameter case, S_0 then, typically, contains regions that attract iterations whilst others repel iterations. Plots of $\|e_k\|_{\mathscr{Y}}$ against k will hence have one or more periods of slow convergence before the limit is reached.

In what follows, the case of $S_0 = \{0\}$ is considered using a number of examples.

Example ONE: Approximations to $\varepsilon^{-2}G^*(I + \varepsilon^{-2}GG^*)^{-1}$ The formal power series expansion

$$\varepsilon^{-2}G^*(I + \varepsilon^{-2}GG^*)^{-1} = \varepsilon^{-2}G^* \sum_{j=0}^{\infty} (-1)^j (\varepsilon^{-2}GG^*)^j \tag{14.148}$$

of the operator seen in NOILC Algorithm 9.1 suggests the parameterization

$$\begin{aligned} &\Gamma(\beta) = \varepsilon^{-2}G^* \textstyle\sum_{j=1}^{n_p} \beta^{(j)} (\varepsilon^{-2}GG^*)^{j-1} \\ &(\textit{with}\ \ \tilde{\Gamma}_j = \varepsilon^{-2}G^*(\varepsilon^{-2}GG^*)^{j-1},\ 1 \leq j \leq n_p), \end{aligned} \tag{14.149}$$

based on truncation of the series after n_p terms and choosing the coefficients $\{\beta^{(j)}\}_{1\leq j\leq n_p}$ using a POILC methodology. The signal $(GG^*)^j e_k$, $j \geq 1$ is computed

from $2j$ off-line simulations starting with G^*e_k and then computing $GG^*e_k = G(G^*e_k)$, $G^*GG^*e_k = G^*(GG^*e_k)$ etc. Note that,

1. The case of $n_p = 1$ corresponds to a gradient methodology whilst choosing $n_p > 1$ provides greater variability in descent direction and offers the potential for faster convergence rates.
2. As $\tilde{\Gamma}_1 = \varepsilon^{-2}G^*$, then $S_0 = \{0\}$ if $ker[G^*] = \{0\}$ and convergence of the error to zero is guaranteed.

The reader will note that, if $K : \mathscr{Y} \to \mathscr{U}$ is an additional compensator chosen to modify plant dynamics (in a similar manner to the methods of Sect. 8.2.2), a more general structure takes the form

$$\begin{array}{c} \Gamma(\beta) = \varepsilon^{-2}(GK)^* \sum_{j=1}^{n_p} \beta^{(j)}(\varepsilon^{-2}GK(GK)^*)^{j-1} \\ (with \;\; \tilde{\Gamma}_j = \varepsilon^{-2}(GK)^*(\varepsilon^{-2}GK(GK)^*)^{j-1}, \;\; 1 \le j \le n_p), \end{array} \tag{14.150}$$

Example TWO: Simpler Approximations to $G^*(I + \varepsilon^{-2}GG^*)^{-1}$ A further simplification of the argument used in Example One suggests the choice of

$$\Gamma(\beta) = \sum_{j=1}^{n_p} \beta^{(j)}K_j^*, \quad (i.e., \;\; \tilde{\Gamma}_j = K_j^*, \;\; 1 \le j \le n_p) \tag{14.151}$$

where K_j^* is the adjoint of an operator $K_j : \mathscr{U} \to \mathscr{Y}$. For example, for discrete state space systems, K_j will have a state space model $S(A_j.B_j, C_j, D_j)$. A sufficient condition for $S_0 = \{0\}$ is that $GK_j^* + K_jG^*$ is positive for one or more indices j.

Example THREE: Approximations to the Inverse Compensator For (square) m-input, m-output, linear, discrete time state space systems $S(A, B, C, D)$ operating on $0 \le t \le N$, the case where the associated matrix operator G in the supervector description (Chap. 4) is invertible makes the choice $\Gamma_0 = G^{-1}$ the natural choice for single parameter, inverse model-based POILC algorithms. Inverse models, however, tend to be sensitive to high frequency modelling errors. A general approximation that avoids the use of inverses is constructed as follows: choosing another m-input, m-output, linear, discrete time state space system K, the inverse compensator G^{-1} can be written in the form $K(GK)^{-1}$. From the Cayley-Hamilton Theorem, if GK has characteristic polynomial $\rho_{GK}(z) = \sum_{j=0}^{m(N+1)} a_{m(N+1)-j}z^j$ with $a_0 = 1$, then invertibility ensures that $a_{m(N+1)} \neq 0$ so that

$$\begin{array}{c} (GK)^{-1} = -a_{m(N+1)}^{-1} \sum_{j=1}^{m(N+1)} a_{m(N+1)-j}(GK)^{j-1}, \quad and\ hence \\ G^{-1} = -K \sum_{j=1}^{m(N+1)} a_{m(N+1)}^{-1}a_{m(N+1)-j}(GK)^{j-1}. \end{array} \tag{14.152}$$

Truncation of the series suggests a multi-parameter controller defined by approximating G^{-1} with

$$\Gamma(\beta) = K \sum_{j=1}^{n_p} \beta^{(j)} (GK)^{j-1}, \quad \tilde{\Gamma}_j = K(GK)^{j-1}, \quad 1 \leq j \leq n_p. \tag{14.153}$$

This parameterization generates a POILC algorithm that searches for a descent direction over a subspace spanned by $\{K(GK)^{j-1} e_k\}_{1 \leq j \leq n_p}$. There is considerable freedom to choose K to influence convergence and robustness. Possible examples include

1. $K = G_A^{-1}$ where G_A is an approximation to G when a sufficient condition for $S_0 = \{0\}$ is that $GG_A^{-1} + (GG_A^{-1})^*$ is positive.
2. Alternatively, it could be a filter to condition the algorithm to reduce sensitivity to frequency ranges where modelling errors are thought to be most significant.

Finally, the strengths of POILC lie in the essential simplicity of the computations and the guarantee of monotonicity of error norms. The number of degrees of freedom available to the control design engineer is however large. The NOILC Algorithm 9.1 provides, for example, some insight into the choice of Q and R but the choice of n_p and parameterization $\Gamma(\beta)$ opens up a wide variety of choices. These can be guided by the examples given above and moderated by the simplicity and/or ease of implementation required by the user.

14.5 Discussion and Further Reading

14.5.1 Chapter Overview

The presentation chosen for this chapter does not follow the time-line of its development in the literature from its original introduction in [89] motivated by links to a linear form of "Hebbian" neural search algorithm [58, 77]. In that approach, the only real link to NOILC is the linear quadratic nature of the optimization problems. In the alternative presentation chosen for this text, the ideas are linked to Norm Optimal Iterative Control through the equivalence stated in Theorem 14.1 (a generalization of the work reported in [82]). The construction of Parameter Optimal Iterative Learning Control from this result could go in several directions. The one chosen retains both the linear and quadratic nature of the optimization problem and automatically leads to monotonic error norms and formulae for off-line calculation of the required parameter vectors for each iteration.

The algorithms can be seen in two ways. Using the single parameter algorithm as a model, the first approach uses an approximation to a specified, underlying NOILC problem that generates the values of w_0 and ε^2 to be used, with $\|e_k\|^2_{\mathscr{Q}}$, in w_{k+1}. Being suboptimal, the resultant convergence rate cannot be predicted. The second

approach regards the link to NOILC as being secondary to actual performance and chooses these parameters to achieve the desired properties using empirical and/or trial and error methods. POILC can perform well in practice but situations can be created where convergence properties are unacceptable. This is particularly clear in the single parameter Algorithm 14.1 where (Sect. 14.2.8) practical convergence to a non-zero error norm can be observed (the flat-lining/plateauing phenomenon). It is associated with the failure to ensure that the set of equilibrium points S_0 is the singleton set $\{0\}$ or, more generally, to ensure that S_0 only contains signals of acceptable form. The analysis of switching algorithms in Sect. 14.2.9 suggests that a practical approach to amelioration of this behaviour is to change the input update formulae on each iteration. See also [96].

Improved convergence and reductions in the negative effects of the flat-lining phenomenon can also be attempted using multi-parameter POILC as in Algorithm 14.4. In optimization terms, multi-parameter algorithms aim to achieve improved norm reductions by searching over an n_p dimensional subspace of possible descent directions. These algorithms inherit many of the properties, problems and simplicity of the single-parameter case but are more complex to analyze due to the number of parameters available and the matrix-valued, nonlinear nature of the optimizing parameter vector β_{k+1} as a function of e_k. The approach suggests multi-parameter parameterizations by using approximations to NOILC, gradient and inverse-model algorithms (see also [85, 87, 94]).

The analysis of the robustness of POILC is more complex than that of the fixed parameter case, primarily because of the nonlinear dependence of the optimal parameters on error data. Available results are provided in Sect. 14.3 and in [55] for the case of single parameter, inverse model compensators. The theory works well here as conditions for monotonic error norm reductions fit naturally with the monotonic dependence of $\beta(e)$ on $\|e\|_{\mathscr{Y}}$. Similar ideas work to a certain extent for gradient-based iteration, the original ideas being presented in [94], and Theorem 14.7 provides a general link between the robustness of fixed parameter cases and POILC although the computation of the range of parameters that can be used is more complex. The results reinforce the intuition that increased robustness of the fixed parameter algorithm will tend to increase the robustness of the POILC version. The derivation of useful robustness conditions for the multi-parameter case is an open problem but an investigation of any link to the fixed parameter case may be the vital key to any progress.

Finally, there are many open questions in POILC that could be worthy of further research. Robustness is perhaps the most urgent but parameter choice is also crucial. The choice/computation of W_∞ is particularly intriguing. Theorem 14.9 suggests the existence of "optimal" choices. Information on Zorn's Lemma can be found in many texts including [107].

14.5.2 High Order POILC: A Brief Summary

The literature also contains related notions of "high-order" Iterative Control, [17, 25, 78, 114] a topic not covered in this text as it seems to add little to the scientific understanding of the algorithms. A more direct description of this idea can start from the observation that the theory of POILC could be formulated from the description/parameterization

$$u_{k+1} = u_k + M(e_k)\beta, \quad with \quad \beta \in \mathscr{R}^{n_p}. \tag{14.154}$$

If this is done then linearity with respect to the parameter vector is retained and the basic theory in Sect. 14.4 is unchanged other than a need to consider the form of the weight $W(e)$ used to generate W_{k+1}. In this scenario, $M(e)$ can have any functional form including nonlinearities. The case where linearity is retained but with $M(0) \neq 0$ has been analysed in [57] and interpreted as a representation of control signal elements that are error independent but included to reflect perceived needs for defined forms of control action on specific time intervals where, for example, large input signals are (temporarily) expected.

High order update rules and algorithms can take many forms, including

$$u = u_k + M(e_k, e_{k-1}, \ldots, e_{k+1-n_e})\beta. \tag{14.155}$$

Clearly, $M(\cdot)$ depends on n_e past error signals $e_k, e_{k-1}, \ldots, e_{k+1-n_e}$. With $n_e = n_p$, the update rule has the form

$$\begin{array}{c} u_{k+1} = u_k + \sum_{j=1}^{n_p} \beta^{(j)} \tilde{\Gamma}_j e_{k+1-j}, \\ (so\ that \quad M(e_k, e_{k-1}, \ldots, e_{k+1-n_e}) = \left[\tilde{\Gamma}_1 e_k, \ldots, \tilde{\Gamma}_{n_p} e_{k+1-n_p}\right]) \end{array} \tag{14.156}$$

generates descent directions using n_p compensators $\{\tilde{\Gamma}_j\}_{1\leq j\leq n_p}$ and error signals $\{e_{k+1-j}\}_{1\leq j\leq n_p}$ from n_p previous iterations. There may be benefits for convergence rates in the initial stages of the algorithm but, if the error converges to a non-zero value, the errors become aligned and the algorithm is then expected to behave in a similar way to the multi-parameter algorithm.

References

The following references aim to support the reader whilst studying areas related to the work described in this text. This includes classical control methodologies historical aspects of Iterative and Repetitive Control, useful links to the mathematical techniques used and keys to published material in Iterative Control that has an optimization content. References to published material are primarily placed at the end of each chapter in a "Discussion and Further Reading" section.

1. H. Abou-Kandil, G. Freiling, V. Ionescu, G. Jank, *Matrix Riccati Equations in Control and Systems Theory* (Birkhuaser, Basel, 2003)
2. H.-S. Ahn, Y.Q. Chen, K.L. Moore, Iterative learning control: brief survey and categorization. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. **37**(6), 1099–1121 (2007)
3. H.-S. Ahn, K.L. Moore, Y.Q. Chen, *Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems*. Series on Communications and Control Engineering (Springer, 2007)
4. P. Albertos, A. Sala, *Multivariable Control Systems: An Engineering Approach* (Springer, 2004)
5. N. Amann, D.H. Owens, E. Rogers, Iterative learning control for discrete-time systems with exponential rate of convergence. Proc. IEE Control Theory Appl. **143**(23), 217–224 (1996)
6. N. Amann, D.H. Owens, E. Rogers, Iterative learning control using optimal feedback and feedforward actions. Int. J. Control **65**(2), 277–293 (1996)
7. N. Amann, D.H. Owens, E. Rogers, Predictive optimal iterative learning control. Int. J. Control **69**(2), 203–266 (1998)
8. B.D.O. Anderson, J.B. Moore, *Linear Optimal Control* (Prentice-Hall, New Jersey, 1971)
9. B.D.O. Anderson, J.B. Moore, *Optimal Control: Linear Quadratic Methods* (Prentice-Hall, New Jersey, 1989)
10. S. Arimoto, S. Kawamura, F. Miyazaki, Bettering operation of robots by learning. J. Robot. Syst. **1**(2), 123–140 (1984)
11. M. Athans, P.L. Falb, *Optimal Control* (McGraw Hill, 1966)
12. G. Bachman, L. Narici, *Functional Analysis* (Academic Press, 1966)
13. K.L. Barton, A.G. Alleyne, A norm optimal approach to time-varying ilc with application to a multi-axis robotic testbed. IEEE Trans. Control Syst. Technol. **19**(1), 166–180 (2011)
14. H.H. Bauschke, J.M. Borwein, On the convergence of von neumann's alternating projection algorithm for two sets. Set Valued Anal. **1**(2), 185–212 (1993)
15. H.H. Bauschke, J.M. Borwein, On projection algorithms for solving convex feasibility problems. SIAM Rev. **38**(3), 367–426 (1996)

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3

16. G. Bengtsson, Minimal system inverses for linear multivariable systems. J. Math. Anal. Appl. **46**(2), 261–274 (1974)
17. Z. Bien, J.-X. Xu, *Iterative Learning Control: Analysis, Design, Integration and Applications* (Springer, New York, 1998)
18. L.M. Bregman, The method of successive projection for finding a common point of convex sets. Soviet Math. Doklady **6**(2), 688–692 (1965)
19. D.A. Bristow, M. Tharayil, A.G. Alleyne, A survey of iterative learning control: a learning-based method for high-performance tracking control. IEEE Control Syst. Mag. **26**(3), 96–114 (2006)
20. A.E. Bryson, *Applied Linear Optimal Control* (Cambridge University Press, Cambridge, 2002)
21. J.B. Burt, *Linear Optimal Control: H2 and H-infinity Methods*, 1st edn. (Prentice Hall, 1998)
22. E.F. Camacho, C. Bordons, *Model Predictive Control* (Springer, London, 1999)
23. B.M. Chen, *Robust and H-infinity Control* (Springer, London, 2000)
24. T. Chen, B. Francis, *Optimal Sampled-Data Control Systems* (Springer, London, 1995)
25. Y. Chen, C. Wen, *Iterative Learning Control: Convergence, Robustness and Applications*. Springer Lecture Notes in Control and Information Sciences No. 48 (1999)
26. B. Chu, D.H. Owens, Iterative learning control for constrained linear systems. Int. J. Control **83**(7), 1397–1413 (2010)
27. B. Chu, D.H. Owens, Accelerated predictive norm-optimal iterative learning control. Proc. Inst. Mech. Eng. **225**(6), 744–759 (2011)
28. L. Collatz, *Functional Analyis and Numerical Mathematics* (Academic Press, New York, 1966)
29. S. Daley, D.H. Owens, J. Hatonen, Application of optimal iterative learning control to the dynamic testing of mechanical structures. Proc. IMechE Part I: J. Syst. Control Eng. **221**, 211–222 (2007)
30. D.H. Owens, *Multivariable and Optimal Systems* (Academic Press, London, 1981)
31. J. Dieudonne, *Foundations of Modern Analysis* (Academic Press, New York, 1969)
32. N. Dunford, J.T. Schwartz, *Linear Operators: Part 1 General Theory* (Wiley, New York, 1988)
33. N. Dunford, J.T. Schwartz, *Linear Operators: Part 2, Spectral Theory*, Self Adjoint Operators in Hilbert Space (Wiley, New York, 1988)
34. J.B. Edwards, D.H. Owens, *Analysis and Control of Multipass Processes* (Research Studies Press, Wiley, 1982)
35. G. Ellis, *Observers in Control Systems*, 1st edn. (Elsevier, 2002)
36. R. Escalante, M.Raydan, *Alternating Projection Methods*. SIAM books (SIAM, Philadelphia, 2011)
37. P.L. Falb, W.A. Wolovich, Decoupling in the design and synthesis of multivariable control systems. IEEE Trans. Sut. Control **12**(6), 651–659 (1969)
38. B.A. Francis, W.M. Wonham, The internal model principle of control theory. Automatica **12**, 457–465 (1976)
39. G.F. Franklin, *Digital Control of Dynamic Systems* (Prentice-Hall, 1997)
40. C.T. Freeman, A.M. Hughes, J.H. Burridge, P.H. Chappell, P.L. Lewin, E. Rogers, Iterative learning control of FES applied to the upper extremity for rehabilitation. Control Eng. Pract. **17**(3), 368–381 (2009)
41. C.T. Freeman, Z. Cai, E. Rogers, P.L. Lewin, Iterative learning control for multiple point-to-point tracking application. IEEE Trans. Control Syst. Technol. **99**, 1–11 (2010)
42. C.T. Freeman, E. Rogers, A. Hughes, J.H. Burridge, K.L. Meadmore, Iterative learning control in healthcare: electrical stimulation and robotic-assisted upper-limb stroke rehabilitation. IEEE Control Syst. **32**(1), 18–43 (2012)
43. B. Friedland, *Control Systems Design: An Introduction to State Space Methods* (Dover, New York, 1986)
44. J. Frueh, M. Phan, Linear quadratic optimal learning control (lql). Int. J. Control **73**(10), 832–839 (2000)

45. F.R. Gantmacher, *The Theory of Matrices*, vol. I (Chelsea Publishing Company, New York, 1959)
46. F.R. Gantmacher, *The Theory of Matrices*, vol. II (Chelsea Publishing Company, New York, 1959)
47. E.G. Gilbert, The decoupling of multivariable systems by state feedback. SIAM J. Control **7**(1), 50–63 (1969)
48. G.C. Goodwin, S.F. Graebe, M.E. Salgado, *Control Systems Design* (Prentice-Hall, 2000)
49. M. Gopal, *Control Systems: Principles and Design* (Tata McGraw-Hill Education, 2002)
50. H. Gorecki, S. Kuksa, P. Grabowski, A. Korytowski, *Analysis and Synthesis of Tiem Delay Systems* (Wiley, 1989)
51. U. Grenander, G. Szego, *Toeplitz Forms* (AMS CHelsea Publishing, New York, 1984)
52. D.H. Griffel, *Applied Functional Analysis* (Dover, 2003)
53. P.R. Halmos, *Finite Dimensional Vector Spaces* (Springer, New York, 1958)
54. P.R. Halmos, *A Hilbert Space Problem Book*, 2nd edn. (Springer, New York, 1982)
55. T.J. Harte, J. Hatonen, D.H. Owens, Discrete-time inverse model-based iterative learning control: stability, monotonicity and robustness. Int. J. Control **78**(8), 577–586 (2005)
56. J. Hatonen, D.H. Owens, Convex modifications to an iterative learning control law. Automatica **78**(7), 1213–1220 (2004)
57. J. Hatonen, D.H. Owens, K. Feng, Basis functions and parameter optimization in high-order iterative learning control. Automatica **78**, 287–294 (2006)
58. S. Haykin, *Neural Networks: A Comprehensive Foundation* (Prentice-Hall, New Jersey, 1999)
59. D. Hinrichsen, A.J. Pritchard, *Mathematical Systems Theory I* (Springer, Berlin, 2005)
60. G. James, *Modern Engineering Mathematics*, 4th edn. (Prentice -Hall, 2010)
61. K. Kinosita, T. Sogo, N. Adachi, Iterative learning control using adjoint systems and stable inversion. Asian J. Control **4**(1), 60–67 (2002)
62. V. Kucera, A review of the matrix riccati equation. Kybernetika **1**(9), 42–61 (2006)
63. I.D. Landau, G. Zito, *Digital Control Systems: Design, Identification and Implementation* (Springer, London, 2006)
64. J.H. Lee, K.S. Lee, W.C. Kim, Model based iterative learning control with a quadratic criterion for time varying linear systems. Automatica **36**(5), 641–657 (2000)
65. W.S. Levine, (ed.), *The Control Handbook*, 2nd edn. Control System Fundamentals (CRC Press, 2011)
66. Y. Li, Y.Q. Chen, H.-S. Ahn, G. Tian, A survey on fractional order iterative learning. J. Optim Theory Appl **156**, 127–140 (2013)
67. T. Lin, D.H. Owens, J. Hatonen, Newton method based iterative learning control for discrete nonlinear systems. Int. J. Control **79**(10), 1263–1276 (2006)
68. R.W. Longman, Iterative learning control and repetitive control for engineering practice. Int. J. Control **73**(10), 930–954 (2000)
69. D.G. Luenberger, *Optimization Using Vector Space Methods* (Wiley, New York, 1969)
70. D.G. Luenberger, An introduction to observers. IEEE GTrans. Aut. Control **16**(6), 596–602 (1971)
71. J.M. Maciejowski, *Multivariable Feedback Design* (Addison Wesley, 1989)
72. J.M. Maciejowski, *Predictive Control with Constraints* (Pearson Education Ltd, England, 2002)
73. I.J. Maddox, *Elements of Functional Analysis*, 2nd edn. (Cambridge University Press, Cambridge, 1988)
74. K.L. Moore, *Iterative Learning Control for Deterministic Systems* (Springer, London, 1993)
75. A.S. Morse, Structural invariants of linear multivariable systems. SIAM J. Control **11**(3), 446–465 (1973)
76. K.S. Narendra, A.M. Annaswamy, *Stable Adaptive Systems* (Prentice-Hall, New Jersey, 1989)
77. M. Norgaard, O. Ravn, N.K. Poulsen, L.K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems* (Springer, London, 2000)
78. M. Norrlof, Iterative Learning Control: Analysis, Design and Experiments. Linkoping Studies in Science and Technology. Dissertation No. 653 (2000)

79. J.M. Ortega, W.C. Rheinboldt, *Functional Analyis and Numerical Mathematics* (Academic Press, London, 1970)
80. D.H. Owens, Dyadic expansion for the analysis of linear multivariable systems. Proc. IEE **121**(7), 713–716 (1974)
81. D.H. Owens, *Feedback and Multivariable Systems* (Peter Peregrinus, Stevenage, 1978)
82. D.H. Owens, Multivariable norm optimal and parameter optimal iterative learning control: a unified formulation. Int. J. Control **85**(8), 1010–1025 (2012)
83. D.H. Owens, B. Chu. Int. J. Control (8), 1469–1484
84. D.H. Owens, B. Chu. Modelling of non-minimum phase effects in discrete-time norm optimal iterative learning control. Int. J. Control **83**(10), 2012–2027 (2010)
85. D.H. Owens, B. Chu, Combined inverse and gradient iterative learning control: performance, monotonicity, robustness and non-minimum-phase zeros. Int. J. Robust Nonlinear Control (2), 203–226 (2012). doi:10.1002/rnc.2893
86. D.H. Owens, B. Chu, E. Rogers, C.T. Freeman, P.L. Lewin, Influence of nonminimum phase zeros on the performance of continuous-time optimal iterative learning control. IEEE Trans. Control Syst. Technol. **22**(3), 1151–1158 (2013)
87. D.H. Owens, B. Chu, M. Songjun, Parameter-optimal iterative learning control using polynomial representations of the inverse plant. Int. J. Control **85**(5), 533–544 (2012)
88. D.H. Owens, S. Daley, Iterative learning control—monotonicity and optimization. Int. J. Appl. Math. Conmput. Sci. **18**(3), 279–293 (2008)
89. D.H. Owens, K. Feng, Parameter optimization in iterative learning control. Int. J. Control **76**(11), 1059–1069 (2003)
90. D.H. Owens, C.F. Freeman, B. Chu, Multivariable norm optimal iterative learning control with auxiliary optimization. Int. J. Control **86**(6), 1026–1045 (2013)
91. D.H. Owens, C.F. Freeman, B. Chu, An inverse-modelapproach to multivariable norm optimal iterative learning control with auxiliary optimization. Int. J. Control **87**(8), 1026–1045 (2014)
92. D.H. Owens, C.F. Freeman, T. Van Dinh, Norm-optimal iterative learning control with intermediate point weighting: theory, algorithms and experimental evaluation. IEEE Trans Control Syst. Technol. (3), 999–1007 (2012)
93. D.H. Owens, J. Hatonen, Iterative learning control—an optimization paradigm. Annu. Rev. Control **29**, 57–70 (2005)
94. D.H. Owens, J.J. Hatonen, S. Daley, Robust monotone gradient-based discrete-time iterative learning control. Int. J. Robust Nonlinear Control **19**, 634–661 (2009)
95. D.H. Owens, R.P. Jones, Iterative solution of constrained differential/algebraic systems. Int. J. Control **27**(6), 957–964 (1978)
96. D.H. Owens, M. Tomas-Rodriguez, S. Daley, Limit sets and switching strategies in parameter-optimal iterative learning control. Int. J. Control **81**(4), 626–640 (2008)
97. J.D. Ratcliffe, P.L. Lewin, E. Rogers, J. Hatonen, D.H. Owens, Norm-optimal iterative learning control applied to gantry robots for automation applications. IEEE Trans. Robot. **22**(6), 1303–1307 (2006)
98. J.B. Rawlings, D.Q. Mayne, *Model Predictive Control: Theory and Design* (Nob Hill Publishing, 2009)
99. E. Rogers, K. Galkowski, D.H. Owens, *Control Systems Theory and Applications for Linear Repetitive Processes*. Springer Lecture Notes in Control and Information Science No. 349 (2007)
100. E. Rogers, D.H. Owens, H. Werner, C.T. Freeman, P.L. Lewin, S. Kirchhoff, C. Schmidt, G. Lichtenberg, Norm-optimal iterative learning control with application to problems in acceleration-based free electron lasers and rehabilitation robotics. Euro. J. Control **5**, 496–521 (2010)
101. W. Rudin, *Principles of Mathematical Analysis*, 3rd edn. (McGraw-Hill, New York, 1976)
102. Y. Shengyue, Z. Qu, X. Fan, X. Nian, Novel iterative learning controls for linear discrete-time systems based on a performance index over iterations. Automatica **44**, 1366–1372 (2008)
103. L.M. Silverman, Inversion of multivariable linear systems. IEEE Trans. Aut. Control **14**(3), 270–276 (1969)

104. E.D. Sontag, *Mathematical Control Theory: Deterministic, Finite Dimensional Systems* (Springer, New York, 1998)
105. K.A. Stroud, D.J. Booth, *Advanced Engineering Mathematics*, 5th edn. (Palgrave-Macmillan, 2011)
106. K.A. Stroud, D.J. Booth, *Engineering Mathematics*, 7th edn. (Palgrave-Macmillan, 2013)
107. A.E. Taylor, *Introduction to Functional analysis*, 2nd edn. (Wiley, London, 1980)
108. R. Tousain, E. van der Meche, O. Bosgra, Design strategies for iterative learning control based on optimal control. Sel. Top. Signals Syst. Control **12**(12), 1–8 (2001)
109. C.-C. Tsui, Observer design—a survey. Int. J. Autom. Comput. **12**(1), 50–61 (2015)
110. L. Wang, *Model Predictive Control System Design and Implementation using MATLAB* (Springer, London, 2009)
111. Y. Wang, F.J. Furong Gao, I.I.I. Doyle, Survey on iterative learning control repetitive control and run-to run control. J. Process Control **19**, 1589–1600 (2009)
112. W.M. Wonham, *Linear Multivariable Control: A Geometric Approach*, 3rd edn. (Springer, London, 1985)
113. Xu J.-X., A survey on iterative learning control for nonlinear systems. Int. J. Control **84**(7), 1275–1294 (2011)
114. J.-X. Xu, S.K. Panda, T.H. Lee, *Real Time Iterative Learning Control: Design and Applications*, Series on Advances in Industrial Control (Springer, 2008)
115. Y. Ye, D. Wang, Zero phase learning control using reversed time input runs. J. Dyn. Syst. Meas. Control **127**, 133–139 (2005)

Index

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3

D

E

F

G

H

I