Lecture Notes in Electrical Engineering 80

Paul Van Dooren
Shankar P. Bhattacharyya
Raymond H. Chan
Vadim Olshevsky
Aurobinda Routray
Editors

Numerical Linear Algebra in Signals, Systems and Control

Springer

Lecture Notes in Electrical Engineering

Volume 80

For further volume
http://www.springer.com/series/7818

Paul Van Dooren · Shankar P. Bhattacharyya
Raymond H. Chan · Vadim Olshevsky
Aurobinda Routray
Editors

Numerical Linear Algebra in Signals, Systems and Control

Springer

Editors
Dr. Paul Van Dooren
CESAME
Universite Catholique de Louvain
Batiment Euler 4, avenue Georges Lemaitre
1348 Louvain la Neuve
Belgium
e-mail: paul.vandooren@uclouvain.be

Prof. Vadim Olshevsky
Department of Mathematics
University of Connecticut
Auditorium Road 196
Storrs Connecticut
U-9 06269-3009
USA
e-mail: olshevsky@math.uconn.edu

Prof. Shankar P. Bhattacharyya
Department of Electrical and Computer Engineering
Texas A&M University
Zachry Engineering Center
College Station
TX 77843-3128
USA
e-mail: bhatt@ee.tamu.edu

Dr. Aurobinda Routray
Department of Electrical Engineering
Indian Institute of Technology
Kharagpur 721302
India
e-mail: aurobinda.routray@gmail.com

Prof. Raymond H. Chan
Department of Mathematics
The Chinese University of Hong Kong
Shatin, New Territories,
Hong Kong SAR
e-mail: rchan@math.cuhk.edu.hk

ISSN 1876-1100 e-ISSN 1876-1119

ISBN 978-94-007-0601-9 e-ISBN 978-94-007-0602-6

DOI 10.1007/978-94-007-0602-6

Springer Dordrecht Heidelberg London New York

Cover design: eStudio Calamar, Berlin/Figueres

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

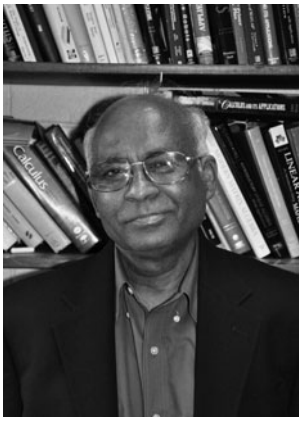
Foreword

A Special Volume Dedicated to Biswa Nath Datta

The Indian Institute of Technology-Kharagpur (IIT-KGP) hosted an international workshop on "Numerical Linear Algebra in Signal, Systems, and Control", during January 9–11, 2007. The conference was sponsored by IEEE Kharagpur Section, Department of Electrical Engineering of IIT-KGP, and Systems Society of India-IIT Kharagpur Chapter. The convener of the workshop was Professor Aurobinda Routray of Electrical Engineering Department of IIT-KGP.

The workshop was interdisciplinary in nature blending linear and numerical linear algebra with control and systems theory, and signal processing. Though a few such conferences, such as the AMS conference on "Linear Algebra and its Role in Systems Theory", and a series of four SIAM conferences on "Linear Algebra in Signals, Systems, and Control", were held before in USA, this is the first time an interdisciplinary conference of this type was held in India. About one hundred mathematicians, computational scientists, and engineers from several countries of the world, including Australia, Belgium, Brazil, Cyprus-Turkey, Germany, Hong Kong, India, Sri Lanka, USA, and Venezuela, participated in this workshop. The picture below shows the group of attendees.

At the banquet of this workshop, Professor Biswa Nath Datta was honored by the IEEE for his numerous contributions in the area of numerical linear algebra with control and systems theory, and signal processing. The ceremony was presided by Professor N. Kishore, the-then President of the IEEE Kharagpur Chapter and Profressor Rajendra Bhatia, an eminent India mathematician from Indian Statistical Institute, was the principal banquet speaker. The other speakers were: the late Gene Golub, Professor of Stanford University, and Professors Paul Van Dooren of Universiťe Catholique de Louvain, Belgium, Volker Mehrmann of Technische Universiẗat Berlin, Germany and V. K. Mohan of Indian Institute of Technology, Kharagpur, India. It was also decided at the end of this meeting to dedicate a special volume to Biswa Datta. This volume contains papers presented at the workshop, as well as contributed and invited papers sent in by authors working in this area. All of these papers went through a regular refereeing process.



As editors of this volume we are pleased to convey our best wishes to Biswa.

Shankar Bhattacharyya, Raymond Chan, Vadim Olshevsky, Aurobinda Routray and Paul Van Dooren.

Biosketch of Biswa Nath Datta

Biswa Datta was born in the village of Bighira, West Bengal, India in 1941. After completing his high school education from his village school and bachelor and master degrees from Calcutta University in India, he moved to London, England to pursue his higher studies in 1967. Primarily due to economic reasons, he could not complete his studies in London and in 1968, moved to Hamilton, Ontario, Canada from where he completed his masters degree in mathematics in 1970.

Based on his Masters Thesis, written under Joseph Csima, he published his first paper, *DAD Theorem for Nonnegative Symmetric Matrices* in Journal of Combinatorial Theory in 1972.

He then joined University of Ottawa, Canada and received his Ph.D. degree in 1972, under the direction of the late James H. Howland.

Immediately after his graduation from University of Ottawa, he got married to Karabi Datta, who at that time was also a fellow graduate student in mathematics at University of Ottawa and working under the direction of the late Professor Howland.

Later that year, he met with Air Vice Marshal S. Roychoudhury, the-then Director of Gas Turbine Research Establishment (GTRE), Bangalore, India, who was visiting North America at that time with the mission of recruiting "Indian-talents" from abroad. Mr. Roychoudhury offered both Biswa and Karabi the positions as scientific officers at GTRE. Additionally, Datta was appointed as the Head of the Computational Mathematics Group in that organization. They both enthusiastically went back to India in 1973 with their new positions. Unfortunately, this non-academic government job was not stimulating and challenging enough for Datta and he decided to leave India again. After spending a few months at Ahmadu Bello University, Zaria, Nigeria, as a lecturer (1974) and about four years (1975–1980) at State University of Campinas, Campinas, Brazil as an Associate Professor of Computational Mathematics, he finally moved back to North America in 1980 as a visiting Associate Professor of Computer Science at Pennsylvania State University.

Their two children, Rajarshi and Rakhi, both were born in Campinas, in 1976 and 1979, respectively. Biswa and Karabi now have two grandsons: Jayen(4) and Shaan(2)—thier parents are Rajarshi and Swati. Datta joined Northern Illinois University in 1981 as a Full Professor in Mathematical Sciences Department. He was nominated by *Hans Schneider* and several other prominent mathematicians for this position, There he, along with some of his colleagues, developed the current Computational Mathematics Program at NIU. He also played a key role in the planning and development of the Ph.D. Program in the Mathematical Sciences Department. He also served as the acting director of the "Applications Involvement Component (AIC)" of the Ph.D. Program in mathematical sciences from 1993 to 1996.

In 2001, he was appointed as a Presidential Research Professor at NIU and was elevated to the position of "Distinguished Research Professor" in 2005, which he currently holds. Datta has also held Visiting Professorship at University of Illinois (1985) and University of California, San Diego (1987–1988) and numerous short-term Visiting Professorship (including several *Distinguished Visiting Professorship*) at institutes and research organizations in various countries around the world. These include Australia, Brazil, Chile, China, England, France, Hong Kong, Greece, India, Mexico, Malaysia, Portugal, Spain, Taiwan, and Venezuela.

Though Datta spent most of his academic career outside India, he maintained a close scientific relationship with India. In 1993, he was appointed as a *member of overseas panel of scientists* for the Council of Scientific and Industrial Research, Government of India. During 1993–2003, he made many short-term scientific visits to research laboratories and prominent institutes in India as a scientific advisor. He also contributed to the development of the scientific software for India's first supercomputer "Param".

In recognition of his contributions to science and engineering, Datta has received several awards and honors. He was honored in a special IEEE sponsored honoring ceremony during the banquet of the *International Workshop on Numerical Linear Algebra in Signals, Systems and Control*, IIT-Kharagpur, 2007, and more recently, in another honoring ceremony on the occasion of the *First International Conference on Power, Control, Signals and Computations*, held at the Vidya Academy of Science and Technology, Thrissur, India, 2010, where he was awarded a *Gold Medal of Honor*. Datta was also recognized by some of the world's leading linear and numerical linear algebraists in a special banquet honoring ceremony held during the *IMA sponsored International Conference on Linear and Numerical Linear Algebra: Theory, Methods and Applications*, held at Northern Illinois University, DeKalb, Illinois, August 12, 2009. Besides these professional recognitions, Datta has also received several other important professional awards and honors. He is an *IEEE Distinguished Lecturer, an IEEE Fellow, an Academician of the Academy of Nonlinear Sciences*, and is a recipient of *NIU's Presidential Research Professorship, International Federation of Nonlinear Analysis Medal of Honor, Senior Fulbright Specialist award by US State Department* and *several Plaques of Honor* awarded by local IEEE chapters of IIT-Kharagpur, India, and NIU. A special issue on *"Inverse Problems in Science and Industry"* of the Journal **Numerical Linear Algebra with Applications** will be published in his honor in 2011.

Datta has versatile research interests ranging from theoretical and computational linear algebra to control and systems theory and vibration engineering.

- Contributions to inertia, stability, and D-stability
  In the early part of his research career, he worked on the stability, inertia, and D-stability of matrices. He collaborated with several leading matrix theorists, including David Carlson, Paul Fuhrmann, Charles Johnson and Hans Schneider on this endeavor. In a series of papers published between 1976 and 1980, he developed an unified matrix theoretic framework, via the historical Lypunov stability theory, for many apparently diverse classical root-separation results for polynomials, which were obtained by celebrated mathematicians in the early twentieth century. In the process of doing so, he derived simple and elementary matrix theoretic proofs of several of these classical root-separation criteria (e.g., proofs of the Routh–Hurwitz–Fujiwara root-separation criterion and stability criterion of Lienard–Chipart via Bezoutian). Most of the original proofs were based on function-theory and were too involved and long. A key result proved by him in this context, was that the Bezoutian of two polynomials is a symmetrizer of its associated companion matrix.

  In 1979, in a paper published in Numerische Mathematik, he and David Carlson, developed a numerical algorithm for computing the inertia and stability of a non-hermitian matrix and in 1980, along with David Carlson, and Charles Johnson, developed a new characterization of D-stability for symmetric tridiagonal matrices. The concept of D-stability arising in economics was originally formulated by Nobel Laureate, Kenneth Arrow. An effective characterization of

D-stability still does not exist. Notably, their result is still one of the state-of-the-art results on this problem in the literature.

- Contributions to Computational Control Theory
  In the last two decades, Datta and several other prominent researchers, including Paul Van Dooren, Alan Laub, Rajni Patel, Andras Varga, Volker Mehrmann, and Peter Benner, and others have developed computationally effective algorithms for control systems design and analysis using state-of-the-art numerical linear algebra techniques. While there existed a high-level mathematical theory, practically applicable algorithms were lacking in this area. These and other previously existing algorithms have been the basis of one of Datta's books, *Numerical Methods for Linear Control Systems Design and Analysis* and two software packages, *MATHEMATICA based Control Systems Professional-Advanced Numerical Methods* and *MATLAB* Toolkit *MATCONTROL*. Datta has delivered several invited workshops on *Computer-aided Control Systems Design and Analysis* based on this book and the software packages, at some of the leading IEEE and other conferences and at universities and research organizations around the world. His work on "Large-Scale Computations in Control" has been in the forefront of attempts made by him and several others to develop parallel and high-performance algorithms for control systems design. His paper *Large-scale and Parallel Computations in Control*, LAA, 1989, is one of the early research papers in this area. A few other important papers authored/co-authored by him in this area include, *Arnoldi methods for large Sylvester-like observer matrix equations and an associated algorithm for partial spectrum assignment* (with Youcef Saad), LAA (1991), *A parallel algorithm for Sylvester-observer equation* (with Chris Bischof and A. Purkayastha), SIAM *Journal of Scientific Computing* (1996), *Parallel algorithms for certain matrix computations* (with B. Codenotti and M. Leoncini), *Theoretical Computer Science* (1997), *Parallel Algorithms in Control*, Proc. IEEE Conf. Decision and Control (1991), and *High performance computing in control*, SIAM book on *Parallel Processing for Scientific Computing* (1993). While parallel and high performance algorithms were developed in many disciplines in science and engineering, control engineering was lagging behind.

- Contributions to Quadratic Inverse Eigenvalue Problems
  The quadratic inverse eigenvalue problem (QIEP) is an emerging topic of research. Because of intrinsic mathematical difficulties and high computational complexities, research on QIEP has not been very well developed. Datta, in collaboration with several vibration engineers, numerical linear algebraists, and optimization specialists, including Z.-J. Bai, Moody Chu, Eric Chu, Sien Deng, Sylvan Elhay, Abhijit Gupta, Wen-Wei Lin, Yitshak Ram, Marcos Raydan, Kumar V. Singh, Jesse Prat, Jenn Nan Wang, C.S. Wang, and some of his former and current students, Sanjoy Brahma, Joao Carvalho, Joali Moreno, Daniil Sarkissian, and Vadim Sokolov, have made some significant contributions to the development of numerically effective and practically applicable algorithms and associated mathematical theory for several important QIEPs

arising in active vibration control and finite element model updating. A new orthogonality relation for the quadratic matrix pencil derived in the much cited paper *Orthogonality and partial pole assignment for the symmetric definite quadratic pencil* (with S. Elhay and Y. Ram), LAA (1997) has played a key role in these works.

Effective and practically applicable solutions to these problems pose some mathematically challenging and computationally difficult issues. The two major constraints are: (i) *the problems must be solved using only a small number of eigenvalues and eigenvectors of the associated quadratic pencil which are computable using the state-of-the-art computational techniques*, and (ii) *the no spill-over phenomenon (that is keeping the large number of unassigned eigenvalues and eigenvectors of the original pencil unchanged) must be ascertained by means of mathematical theory.* Furthermore, solutions of the robust and minimum-norm quadratic partial eigenvalue assignment and some practical aspects of model updating problem lead to difficult nonlinear (some cases nonconvex) optimization problems, which give rise to additional challenges for computing the required gradient formulas with only a small part of computable eigenvalues and eigenvectors. Most of the current industrial techniques are adhoc and lack strong mathematical foundations. Datta and his collaborators have adequately addressed some of these challenges in their work. One of the important thrusts of their work has been to develop a mathematical theory justifying some of the existing industrial techniques for which such a theory does not exist. These results have been published in some of the leading mathematics and vibration engineering journals, such as *Journal of Sound and Vibration, Mechanical Systems and Signal Processing* (MSSP), *AIAA Journal*. His recent joint paper (with Z.-J. Bai and J. Wang), *Robust and minimum-norm partial eigenvalue assignment in vibrating systems*, MSSP (2010) is a significant paper on the solution of robust and minimum-norm quadratic partial eigenvalue assignment arising in active vibrating control.

Based on his current work on QIEP, Datta has delivered many plenary and key-note talks (and several more are to be delivered this year) at interdisciplinary conferences blending mathematics, computational mathematics and optimization with vibration and control engineering. He has also served on the editorial board of more than a dozen of mathematics and engineering journals, including, *SIAM J. Matrix Analysis, Lin. Alg. Appl.* (Special Editor), *Num. Lin. Alg. Appl., Computational and Applied Mathematics, Dynamics of Continuous, Discrete and Impulsive Systems, Mechanical Systems and Signal Processing* (MSSP), *Computational and Applied Mathematics* (Brazil) and others. He also edited/co-edited several special issues of these journals, the most recent one is on "Inverse Problems in Mechanical Systems and Signal Processing" for the Journal, *MSSP* (with John Mottershead as a co-editor), published in 2009.

For more than 25 years, Datta, through his research, books, and other academic activities, has been actively contributing to promote interdisciplinary research blending linear and numerical linear algebra with control, systems, and signal

processing, which has been a primary mission in his career. He has authored more than 110 research papers, two books, and three associated software packages, all of which are highly interdisciplinary. His book, Numerical Linear Algebra and Applications contains a wide variety of applications drawn from numerous disciplines of science and engineering. His other book, Numerical Methods for Linear Control Systems Design and Analysis, describes how sophisticated numerical linear algebra techniques can be used to develop numerically reliable and computationally effective algorithms for linear control systems design and analysis. This book provides an interdisciplinary framework for studying linear control systems. The software packages, *MATCOM* and *MATCONTROL* are widely used for classroom instructions and *Control Systems Professional-Advanced Numerical Methods* is used for both industrial applications and classroom instructions.

Datta took a leading role in organizing a series of interdisciplinary conferences. The first such conference, chaired by Datta, was the *AMS Summer Research Conference on the Role of Linear Algebra in Signals, Systems, and Control* in 1984, which was participated by many leading researchers in linear and numerical algebra, control and system theory and signal processing. Remarkably, this was the first conference ever supported by the AMS in linear algebra. Subsequently, Datta, on invitation by Edward Block, the-then managing director of SIAM, chaired and organized the first *SIAM Conference on Linear Algebra in Signals Systems, and Control* in 1986. The huge success of this conference led to three more SIAM conferences in this series held, respectively, in San Francisco, Seattle and Boston, in 1990, 1993, and 2001, which were chaired or co-chaired by Datta. He also organized and co-chaired the interdisciplinary conference blending mathematics with systems theory, *Mathematical Theory of Networks and Systems* (MTNS) in 1996. He also served as a member of the international *Steering Committee* of MTNS.

Datta has served as an editor of three interdisciplinary books that grew out of some of these conferences: *The Role of Linear Algebra in Systems theory*, AMS Contemporary Mathematics, volume 47, 1985; *Linear Algebra in Signals, Systems and Control*, SIAM, 1988; and *Systems and Control in the Twenty-First Century*, Birkhauser, 1997. He was also the editor-in-chief of the two books in the series: *Applied and Computational Control, Signals, and Circuits*, vol I published by Birkhauser in 1999, and vol II by Kluwer Academic Publisher, 2001. He also took an initiative in founding the well-known *SIAM J. Matrix Analysis and Applications*, and served as one of the founding editors (with the late Gene Golub as the first Managing Editor) of this journal. He served as the Vice-Chair of the *SIAM Linear Algebra Activity Group* from 1993 to 1998, and chaired the award committee of the Best SIAM Linear Algebra Papers in 1994 and 1997. He has also chaired or served as a member of several panels in linear algebra and control, including the NSF Panel on *Future Directions of Research and Teaching in Mathematical Systems Theory*, University of Notre Dame, 2002 of which he was the chair.

So far, Datta has advised ten interdisciplinary Ph.D. Dissertations and numerous masters theses. During their graduate studies, these students acquired

interdisciplinary training by taking advanced courses from Datta on numerical aspects of control and vibration engineering, and working on interdisciplinary projects and dissertation topics. Almost all of these students picked up their dissertation while taking the advanced interdisciplinary courses from him. Such interdisciplinary expertise is in high demand in both academia and industries worldwide, but is hard to find. Indeed, several of Datta's former students are now working as industrial mathematicians and researchers in research laboratories: *Samar Choudhury* at IBM, *Vadim Sokolov* at Argonne National Laboratory, *Avijit Purkayastha* at Texas Advanced Computing Center of University of Texas, *Dan'l Pierce*, formerly of the Boeing Company and now the CEO of Access Analytics International, and *M. Lagadapati* at Caterpillar.

Contents

Chapter 1
The Anti-Reflective Transform and Regularization by Filtering

A. Aricò, M. Donatelli, J. Nagy and S. Serra-Capizzano

Abstract Filtering methods are used in signal and image restoration to reconstruct an approximation of a signal or image from degraded measurements. Filtering methods rely on computing a singular value decomposition or a spectral factorization of a large structured matrix. The structure of the matrix depends in part on imposed boundary conditions. Anti-reflective boundary conditions preserve continuity of the image and its (normal) derivative at the boundary, and have been shown to produce superior reconstructions compared to other commonly used boundary conditions, such as periodic, zero and reflective. The purpose of this paper is to analyze the eigenvector structure of matrices that enforce anti-reflective boundary conditions. In particular, a new anti-reflective transform is introduced, and an efficient approach to computing filtered solutions is proposed. Numerical tests illustrate the performance of the discussed methods.

A. Aricò
Dipartimento di Matematica, Università di Cagliari, Viale Merello 92, 09123 Cagliari, Italy
e-mail: arico@unica.it

M. Donatelli · S. Serra-Capizzano
Dipartimento di Fisica e Matematica, Università dell'Insubria - Sede di Como, Via Valleggio 11, 22100 Como, Italy
e-mail: marco.donatelli@uninsubria.it

S. Serra-Capizzano
e-mail: stefano.serrac@uninsubria.it

J. Nagy (✉)
Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA
e-mail: nagy@mathcs.emory.edu

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_1,

1.1 Introduction

In this paper we consider structured matrices that arise from the discretization of large scale ill-posed inverse problems,

$$\mathbf{g} = A\mathbf{f} + \eta. \tag{1.1}$$

Given the vector $\mathbf{g}$ and matrix A, the aim is to compute an approximation of the unknown vector $\mathbf{f}$. The vector η represents unknown errors (e.g., measurement or discretization errors and noise) in the observed data. These problems arise in many applications, including image reconstruction, image deblurring, geophysics, parameter identification and inverse scattering; cf. [2, 9, 11, 12, 19]. We are mainly interested in problems that arise in spatially invariant signal and image restoration, where the observed data is

$$g_i = \sum_{j \in \mathbf{Z}^d} f_j h_{i-j} + \eta_i,$$

and the dimension $d = 1$ for signals (such as voice), and $d = 2$ or 3 for images. The d-dimensional tensor $\mathbf{h} = [h_i]$ represents the blurring operator, and is called the *point spread function* (PSF). Notice that we have an infinite summation because a true signal or image scene does not have a finite boundary. However, the data g_i is collected only at a finite number of values, and thus represents only a finite region of an infinite scene. Boundary conditions (BCs) are used to artificially describe the scene outside the viewable region. The PSF and the imposed BCs together define the matrix A.

Typically the matrix A is very ill-conditioned and the degenerating subspace largely intersects the high frequency space: consequently regularization techniques are used to compute stable approximations of $\mathbf{f}$ with controlled noise levels [9–11, 19]. Many choices of regularization can be employed, such as TSVD, Tikhonov, and total variation [9, 11, 19]. Analysis and implementation of regularization methods can often be simplified by computing a spectral (or singular value) decomposition of A. Unfortunately this may be very difficult for large scale problems, unless the matrix has exploitable structure. For example, if A is circulant then the spectral decomposition can be computed efficiently with the fast Fourier transform (FFT) [5]. In image deblurring, circulant structures arise when enforcing periodic boundary conditions. Periodic boundary conditions are convenient for computational reasons, but it is difficult to justify their use in a physical sense for most problems.

Other boundary conditions, which better describe the scene outside the viewable region have been proposed. For example, reflective boundary conditions assume the scene outside the viewable region is a reflection of the scene inside the viewable region. In this case the matrix A has a Toeplitz-plus-Hankel structure. If the blur satisfies a strong symmetry condition, $h_i = h_{|i|}$ for all $i \in \mathbf{Z}^d$, then the spectral decomposition of A can be computed very efficiently using the fast discrete cosine transform (DCT) [14].

More recently, new *anti-reflective* boundary conditions (AR-BCs) have been proposed [17] and studied [1, 7, 8, 15, 18], which have the advantage that continuity of the image, and of the normal derivative, are preserved at the boundary. This regularity, which is not shared with zero or periodic BCs, and only partially shared with reflective BCs, significantly reduces ringing artifacts that may occur with other boundary conditions. The matrix structure arising from the imposition of AR-BCs is Toeplitz-plus-Hankel (as in the case of reflective BCs), plus an additional structured low rank matrix. By linearity of the boundary conditions with respect to the PSF, it is evident that the set of AR-BC matrices is a vector space. Unfortunately it is not closed under multiplication or inversion. However, if we restrict our attention to strongly symmetric PSFs and assume that the PSF satisfies a mild finite extent condition (more precisely $h_i = 0$ if $|i_j| \geq n-2, i = (i_1, \ldots, i_d)$ for some $j \in \{1, \ldots, d\}$), then any of the resulting AR-BC matrices belong to a d-level commutative matrix algebra denoted by $\mathcal{AR}^{(d)}$, see [1]. Certain computations involving matrices in $\mathcal{AR}^{(d)}$ can be done efficiently. For example, matrix–vector products can be implemented using FFTs, and solution of linear systems and eigenvalue computation involving these matrices can be done efficiently using mainly fast sine transforms (FSTs), see [1].

The new contribution of this paper is the analysis of the eigenvector structure of $\mathcal{AR}^{(d)}$ matrices. The main result concerns the definition of the AR-transform, which carries interesting functional information, is fast ($O(n^d \log(n))$ real operations) and structured, but it is not orthogonal. Then we use the resulting eigenvector structure to define filtering-based regularization methods to reconstruct approximate solutions of (1.1).

The paper is organized as follows. In Sect. 1.2, we review the main features of the $\mathcal{AR}$ matrix algebra that are essential for describing and analyzing AR-BC matrices. In Sect. 1.3 we introduce AR-BCs and in Sect. 1.4 we discuss the spectral features of the involved matrices and define an AR-transform. In Sect. 1.5 we describe an efficient approach to compute a filtered (regularized) solution of (1.1). In Sect. 1.6 some 1D numerical results validate the theoretical analysis. A concise treatment of the multidimensional case is given in Sect. 1.7, together with a 2D numerical example.

In the rest of the paper we consider essentially only the one-dimensional problem (that is, $d = 1$), to simplify the notation and mathematical analysis. However, the results generalize to higher dimensions, $d > 1$; comments and results for extending the analysis are provided in Sect. 1.7.

1.2 The Algebra of Matrices Induced by AR-BCs

This section is devoted to describing the algebra $\mathcal{AR}_n \equiv \mathcal{AR}, n \geq 3$. The notation introduced in this section will be used throughout the paper, and is essential for the description, given in Sect. 1.3, of the $n \times n$ matrices arising from the imposition of AR-BCs.

1.2.1 The τ Algebra

Let Q be the type I sine transform matrix of order n (see [3]) with entries

$$[Q]_{i,j} = \sqrt{\frac{2}{n+1}} \sin\left(\frac{ji\pi}{n+1}\right), \quad i,j = 1,\ldots,n. \tag{1.2}$$

It is known that the real matrix Q is orthogonal and symmetric ($Q^{-1} = Q^T = Q$). For any n-dimensional real vector $\mathbf{v}$, the matrix–vector multiplication $Q\mathbf{v}$ (DST-I transform) can be computed in $O(n\log(n))$ real operations by using the algorithm FST-I.

Let τ be the space of all the matrices that can be diagonalized by Q:

$$\tau = \{QDQ : D \text{ is a real diagonal matrix of size } n\}. \tag{1.3}$$

Let $X = QDQ \in \tau$, then $QX = DQ$. Consequently, if we let $\mathbf{e}_1$ denote the first column of the identity matrix, then the relationship $QX\mathbf{e}_1 = DQ\mathbf{e}_1$ implies that the eigenvalues $[D]_{i,i}$ of X are given by $[D]_{i,i} = [Q(X\mathbf{e}_1)]_i/[Q\mathbf{e}_1]_i$, $i = 1,\ldots,n$. Therefore the eigenvalues of X can be obtained by applying a DST-I transform to the first column of X and, in addition, any matrix in τ is uniquely determined by its first column.

Now we report a characterization of the τ class, which is important for analyzing the structure of AR-BC matrices. Let us define the shift of any vector $\mathbf{h} = [h_0,\ldots,h_{n-1}]^T$ as $\sigma(\mathbf{h}) = [h_1, h_2, \ldots, h_{n-1}, 0]^T$. According to a Matlab like notation, we define $T(\mathbf{x})$ to be the n-by-n symmetric Toeplitz matrix whose first column is $\mathbf{x}$ and $H(\mathbf{x},\mathbf{y})$ to be the n-by-n Hankel matrix whose first and last column are $\mathbf{x}$ and $\mathbf{y}$, respectively. Every matrix of the class (1.3) can be written as (see [3])

$$T(\mathbf{h}) - H(\sigma^2(\mathbf{h}), J\sigma^2(\mathbf{h})), \tag{1.4}$$

where $\mathbf{h} = [h_0,\ldots,h_{n-1}]^T \in \mathbf{R}^n$ and J is a matrix with entries $[J]_{s,t} = 1$ if $s + t = n + 1$ and zero otherwise. We refer to J as a "flip" matrix because the multiplication $J\mathbf{x}$ has the effect of flipping the entries of the vector $\mathbf{x}$ in an up/down direction. The structure defined by (1.4) means that matrices in the τ class are special instances of Toeplitz-plus-Hankel matrices.

Moreover, the eigenvalues of the τ matrix in (1.4) are given by the cosine function $h(y)$ evaluated at the grid points vector $G_n = [\frac{k\pi}{n+1}]_{k=1}^n$, where

$$h(y) = \sum_{|j|\le n-1} h_j \exp(\mathbf{i}jy), \tag{1.5}$$

$\mathbf{i}^2 = -1$, and $h_j = h_{|j|}$ for $|j| \le n-1$. The τ matrix in (1.4) is usually denoted by $\tau(h)$ and is called the τ matrix generated by the function or *symbol* $h = h(\cdot)$ (see the seminal paper [3] where this notation was originally proposed).

1.2.2 The AR-Algebras $\mathcal{AR}$

Let $h = h(\cdot)$ be a real-valued cosine polynomial of degree l and let $\tau_k(h) \equiv Q\,\mathrm{diag}(h(G_k))Q$ (note that $\tau_k(h)$ coincides with the matrix in (1.4–1.5), when $l \leq k-1$). Then the Fourier coefficients of h are such that $h_i = h_{-i} \in \mathbf{R}$ with $h_i = 0$ if $|i| > l$, and for $k = n-2$ we can define the one-level $AR_n(\cdot)$ operator

$$AR_n(h) = \begin{bmatrix} h(0) & & \\ \mathbf{v}_{n-2}(h) & \tau_{n-2}(h) & J\mathbf{v}_{n-2}(h) \\ & & h(0) \end{bmatrix}, \tag{1.6}$$

where J is the flip matrix, $\mathbf{v}_{n-2}(h) = \tau_{n-2}((\phi(h))(\cdot))\mathbf{e}_1$ and

$$(\phi(h))(y) = \frac{h(y) - h(0)}{2(\cos(y) - 1)}. \tag{1.7}$$

It is interesting to observe that $h(y) - h(0)$ has a zero of order at least 2 at zero (since h is a cosine polynomial) and therefore $\phi(h) = (\phi(h))(\cdot)$ is still a cosine polynomial of degree $l-1$, whose value at zero is $-h''(0)/2$ (in other words the function is well defined at zero).

As proved in [1], with the above definition of the operator $AR_n(\cdot)$, we have

1. $\alpha AR_n(h_1) + \beta AR_n(h_2) = AR_n(\alpha h_1 + \beta h_2)$,
2. $AR_n(h_1)AR_n(h_2) = AR_n(h_1 h_2)$,

for real α and β and for cosine functions $h_1 = h_1(\cdot)$ and $h_2 = h_2(\cdot)$.

These properties allow us to define $\mathcal{AR}$ as the algebra (closed under linear combinations, product and inversion) of matrices $\mathcal{AR}_n(h)$, with h being a cosine polynomial. By standard interpolation arguments it is easy to see that $\mathcal{AR}$ can be defined as the set of matrices $AR_n(h)$, where h is a cosine polynomial of degree at most $n-3$. Therefore, it is clear that $\dim(\mathcal{AR}) = n-2$. Moreover the algebra $\mathcal{AR}$ is commutative thanks to point 2, since $h_1(y)h_2(y) = h_2(y)h_1(y)$ at every y. Consequently, if matrices of the form $AR_n(h)$ are diagonalizable, then they must have the same set of eigenvectors [13]. This means there must exist an "anti-reflective transform" that diagonalizes the matrices in $\mathcal{AR}$. Unfortunately this transform fails to be unitary, since the matrices in $\mathcal{AR}$ are generically not normal. However the AR transform is close in rank to an orthogonal linear mapping. Development of this transform is the main contribution of this paper, and is discussed in detail in Sect. 1.4.

1.3 AR-BCs and the AR-BC Matrices

In this section we describe the anti-reflective BCs. We have already mentioned that, in the generic case, periodic and zero Dirichlet BCs introduce a discontinuity in the signal, while the reflective BCs preserve the continuity of the signal, but

introduce a discontinuity in the derivative. Our approach is to use an anti-reflection: in this way, at the boundaries, instead of having a mirror-like symmetry (reflective BCs), we impose a global symmetry around the boundary points. This latter choice corresponds to a central symmetry around the considered boundary point. If f_1 is the left boundary point and f_n is the right one, then the external points f_{1-j} and $f_{n+j}, j \geq 1$, are computed as a function of the internal points according to the rules $f_{1-j} - f_1 = -(f_{j+1} - f_1)$ and $f_{n+j} - f_n = -(f_{n-j} - f_n)$. If the support of the centered blurring function is $q = 2m + 1 \leq n$, then for $j = 1, \ldots, m$, we have

$$f_{1-j} = 2f_1 - f_{j+1}, \quad f_{n+j} = 2f_n - f_{n-j}.$$

Following the analysis given in [17], if the blurring function (the PSF) $\mathbf{h}$ is symmetric (i.e., $h_i = h_{-i}, \forall i \in \mathbf{Z}$), if $h_i = 0$ for $|i| \geq n - 2$ (degree condition), and if $\mathbf{h}$ is normalized so that $\sum_{i=-m}^{m} h_i = 1$, then the structure of the $n \times n$ anti-reflective blurring matrix A is

$$A = \begin{bmatrix} z_0 & \mathbf{0}^T & \mathbf{0} \\ z_1 & & z_m \\ \vdots & \widehat{A} & \vdots \\ z_m & & z_1 \\ \mathbf{0} & \mathbf{0}^T & z_0 \end{bmatrix}, \tag{1.8}$$

where $A_{1,1} = A_{n,n} = 1, z_i = h_i + 2\sum_{k=i+1}^{m} h_k, \widehat{A}$ has order $n - 2$ and

$$\widehat{A} = T(\mathbf{h}) - H(\sigma^2(\mathbf{h}), J\sigma^2(\mathbf{h})), \tag{1.9}$$

with $\mathbf{h} = [h_0, h_1, \ldots, h_m, 0, \ldots, 0]^T$. According to the brief discussion of Sect. 1.2.1, relation (1.9) implies that $\widehat{A} = \tau_{n-2}(h)$ with $h(y) = h_0 + 2\sum_{k=1}^{m} h_k \cos(ky)$ (see (1.4) and (1.5)). Moreover in [1] it is proved that $A = AR_n(h)$.

1.4 Eigenvalues and Eigenvectors of AR-BC Matrices

In this section we first describe the spectrum of AR-BC matrices, under the usual mild degree condition (that is, the PSF $\mathbf{h}$ has finite support), with symmetric, normalized PSFs. Then we describe the eigenvector structure and we introduce the AR-transform. In the next section, we will use these results to efficiently compute filtered solutions of (1.1) when the blurring matrix A is in the $\mathcal{AR}$ algebra.

1.4.1 Eigenvalues of $AR_n(\cdot)$ Operators

The spectral structure of any AR-BC matrix, with symmetric PSF $\mathbf{h}$, is concisely described in the following result.

Theorem 1 [1] *Let the blurring function (PSF)* $\mathbf{h}$ *be symmetric (i.e.,* $h_s = h_{-s}$*), normalized, and satisfying the usual degree condition. Then the eigenvalues of the* $n \times n$ *AR-BC blurring matrix A in* (1.8)*,* $n \geq 3$*, are given by* $h(0) = 1$ *with multiplicity two and* $h(G_{n-2})$*.*

The proof can be easily derived by (1.6) which shows that the eigenvalues of $AR_n(h)$ are $h(0)$ with multiplicity 2 and those of $\tau_{n-2}(h)$, i.e. $h(G_{n-2})$, with multiplicity 1 each.

1.4.2 The AR-Transform and Its Functional Interpretation

Here we will determine the eigenvectors of every matrix $AR_n(h)$. In particular, we show that every AR-BC matrix is diagonalizable, and we demonstrate independence of the eigenvectors from the symbol h. With reference to the notation in (1.2–1.5), calling $\mathbf{q}_j^{(n-2)}$ the jth column of Q_{n-2}, and $y_j^{(n-2)}$ the jth point of $G_{n-2}, j = 1, \ldots, n-2$, we have

$$AR_n(h)\begin{bmatrix} 0 \\ \mathbf{q}_j^{(n-2)} \\ 0 \end{bmatrix} = \begin{bmatrix} h(0) & & \\ \mathbf{v}_{n-2}(h) & \tau_{n-2}(h) & J\mathbf{v}_{n-2}(h) \\ & & h(0) \end{bmatrix}\begin{bmatrix} 0 \\ \mathbf{q}_j^{(n-2)} \\ 0 \end{bmatrix}$$
$$= \begin{bmatrix} 0 \\ \tau_{n-2}(h)\mathbf{q}_j^{(n-2)} \\ 0 \end{bmatrix} = h\left(y_j^{(n-2)}\right)\begin{bmatrix} 0 \\ \mathbf{q}_j^{(n-2)} \\ 0 \end{bmatrix}, \qquad (1.10)$$

since $\mathbf{q}_j^{(n-2)}$ is an eigenvector of $\tau_{n-2}(h)$ and $h\left(y_j^{(n-2)}\right)$ is the related eigenvalue. Due to the centro-symmetry of the involved matrix, if $[1, \mathbf{p}^T, 0]^T$ is an eigenvector of $AR_n(h)$ related to the eigenvalue $h(0)$, then the other is its flip, i.e., $[0, (J\mathbf{p})^T, 1]^T$. Let us look for this eigenvector, by imposing the equality

$$AR_n(h)\begin{bmatrix} 1 \\ \mathbf{p} \\ 0 \end{bmatrix} = h(0)\begin{bmatrix} 1 \\ \mathbf{p} \\ 0 \end{bmatrix}$$

which is equivalent to seeking a vector $\mathbf{p}$ that satisfies

$$\mathbf{v}_{n-2}(h) + \tau_{n-2}(h)\mathbf{p} = h(0)\mathbf{p}.$$

Since $\mathbf{v}_{n-2}(h) = \tau_{n-2}(\phi(h))\mathbf{e}_1$ by definition of the operator $\mathbf{v}_{n-2}(\cdot)$ (see (1.6) and the lines below), and, because of the algebra structure of τ_{n-2} and thanks to (1.7), we deduce that the vector $\mathbf{p}$ satisfies the relation

$$\tau_{n-2}(h - h(0))\left[-L_{n-2}^{-1}\mathbf{e}_1 + \mathbf{p}\right] = \mathbf{0} \qquad (1.11)$$

where L_{n-2} is the discrete one-level Laplacian, i.e., $L_{n-2} = \tau_{n-2}(2 - 2\cos(\cdot))$. Therefore by (1.11) the solution is given by $\mathbf{p} = L_{n-2}^{-1}\mathbf{e}_1 + \mathbf{r}$ where $\mathbf{r}$ is any vector belonging to the kernel of $\tau_{n-2}(h - h(0))$. If $\tau_{n-2}(h - h(0))$ is invertible (as it happens for every nontrivial PSF, since the unique maximum of the function is reached at $y = 0$, which is not a grid point of G_{n-2}), then the solution is unique. Otherwise $\mathbf{r}$ will belong to the vector space generated by those vectors $\mathbf{q}_j^{(n-2)}$ for which the index j is such that $h(y_j^{(n-2)}) = h(0)$. However, the contribution contained in $\mathbf{r}$ was already considered in (1.10), and therefore $\mathbf{p} = L_{n-2}^{-1}\mathbf{e}_1$ is the only solution that carries new information. Hence, independently of h, we have

$$AR_n(h)\begin{bmatrix} 1 & & \\ \mathbf{p} & Q_{n-2} & J\mathbf{p} \\ & & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ \mathbf{p} & Q_{n-2} & J\mathbf{p} \\ & & 1 \end{bmatrix} \begin{bmatrix} h(0) & & \\ & \operatorname{diag}(h(G_{n-2})) & \\ & & h(0) \end{bmatrix}.$$

Now we observe that the jth eigenvector is unitary, $j = 2, \ldots, n-1$, because Q_{n-2} is unitary: we wish to impose the same condition on the first and the last eigenvector. The interesting fact is that $\mathbf{p}$ has an explicit expression. By using standard finite difference techniques, it follows that $p_j = 1 - j/(n-1)$ so that the first eigenvector is exactly the sampling of the function $1 - x$ on the grid $j/(n-1)$ for $j = 0, \ldots, n-1$. Its Euclidean norm is $\alpha_n = \sqrt{\sum_{j=0}^{n-1} j^2}/(n-1) \sim \sqrt{n/3}$, where, for nonnegative sequences β_n, γ_n, the relation $\gamma_n \sim \beta_n$ means $\gamma_n = \beta_n(1 + o(1))$. In this way, the (normalized) AR-transform can be defined as

$$T_n = \begin{bmatrix} \alpha_n^{-1} & & \\ \alpha_n^{-1}\mathbf{p} & Q_{n-2} & \alpha_n^{-1}J\mathbf{p} \\ & & \alpha_n^{-1} \end{bmatrix}. \tag{1.12}$$

Remark 1 With the normalization condition in (1.12), all the columns of T_n are unitary. However orthogonality is only partially fulfilled since it holds for the central columns, while the first and last columns are not orthogonal to each other, and neither one is orthogonal to the central columns. We can solve the first problem: the sum of the first and of the last column (suitably normalized) and the difference of the first and the last column (suitably normalized) become orthonormal, and are still eigenvectors related to the eigenvalue $h(0)$. However, since $\mathbf{q}_1^{(n-2)}$ has only positive components and the vector space generated by the first and the last column of T_n contains positive vectors, it follows that T_n cannot be made orthonormal just by operating on the first and the last column. Indeed, we do not want to change the central block of T_n since it is related to a fast $O(n \log(n))$ real transform and hence, necessarily, we cannot get rid of this quite mild lack of orthogonality.

Remark 2 There is a suggestive functional interpretation of the transform T_n. When considering periodic BCs, the transform of the related matrices is the Fourier transform: its jth column vector, up to a normalizing scalar factor, can be

viewed as a sampling, over a suitable uniform gridding of $[0, 2\pi]$, of the frequency function $\exp(-\mathbf{i}jy)$. Analogously, when imposing reflective BCs with a symmetric PSF, the transform of the related matrices is the cosine transform: its jth column vector, up to a normalizing scalar factor, can be viewed as a sampling, over a suitable uniform gridding of $[0, \pi]$, of the frequency function $\cos(jy)$. Here the imposition of the anti-reflective BCs can be functionally interpreted as a linear combination of sine functions and of linear polynomials (whose use is exactly required for imposing C^1 continuity at the borders).

The previous observation becomes evident in the expression of T_n in (1.12). Indeed, by defining the one-dimensional grid $\widetilde{G}_n = [0,\ G_{n-2}^T,\ \pi]^T = [j\pi/(n-1)]_{j=0}^{n-1}$, which is a subset of $[0, \pi]$, we infer that the first column of T_n is given by $\alpha_n^{-1}(1 - y/\pi)|_{\widetilde{G}_n}$, the jth column of $T_n, j = 2, \ldots, n-1$, is given by $\sqrt{2/(n-1)}(\sin(jy))|_{\widetilde{G}_n}$, and finally the last column of T_n is given by $\alpha_n^{-1}(y/\pi)|_{\widetilde{G}_n}$ i.e.

$$T_n = \left[1 - \frac{y}{\pi}, \sin(y), \ldots, \sin((n-2)y), \frac{y}{\pi}\right]\Big|_{\widetilde{G}_n} \cdot \Delta_n, \tag{1.13}$$

$$\Delta_n = \operatorname{diag}\left(\alpha_n^{-1}, \sqrt{\frac{2}{n-1}} I_{n-2}, \alpha_n^{-1}\right).$$

Finally, it is worth mentioning that the inverse transform is also described in terms of the same block structure since

$$T_n^{-1} = \begin{bmatrix} \alpha_n & & \\ -Q_{n-2}\mathbf{p} & Q_{n-2} & -Q_{n-2}J\mathbf{p} \\ & & \alpha_n \end{bmatrix}. \tag{1.14}$$

Theorem 2 [$AR_n(\cdot)$ Jordan Canonical Form.] *With the notation and assumptions of Theorem 1, the $n \times n$ AR-BC blurring matrix A in (1.8), $n \geq 3$, coincides with*

$$AR_n(h) = T_n \operatorname{diag}(h(\widehat{G}_n))\, T_n^{-1}, \tag{1.15}$$

where T_n and T_n^{-1} are defined in (1.13) and (1.14), while $\widehat{G}_n = [0, G_{n-2}^T, 0]^T$.

1.5 Filtering Methods for AR-BC Matrices

As mentioned in Sect. 1.1, regardless of the imposed boundary conditions, matrices A that arise in signal and image restoration are typically severely ill-conditioned, and regularization is needed in order to compute a stable approximation of the

solution of (1.1). A class of regularization methods is obtained through spectral filtering [11, 12]. Specifically, if the spectral decomposition of A is

$$A = T_n \operatorname{diag}(\mathbf{d})\, T_n^{-1}, \quad T_n = [\,\mathbf{t}_1 \quad \mathbf{t}_2 \quad \cdots \quad \mathbf{t}_n\,], \quad T_n^{-1} = \begin{bmatrix} \tilde{\mathbf{t}}_1^T \\ \tilde{\mathbf{t}}_2^T \\ \vdots \\ \tilde{\mathbf{t}}_n^T \end{bmatrix},$$

with $\mathbf{d} = h(\widehat{G}_n)$, then a spectral filter solution is given by

$$\mathbf{f}_{\text{reg}} = \sum_{i=1}^{n} \phi_i \frac{\tilde{\mathbf{t}}_i^T \mathbf{g}}{d_i} \mathbf{t}_i, \tag{1.16}$$

where ϕ_i are filter factors that satisfy

$$\phi_i \approx \begin{cases} 1 & \text{if } d_i \text{ is large,} \\ 0 & \text{if } d_i \text{ is small.} \end{cases}$$

The small eigenvalues correspond to eigenvectors with high frequency components, and are typically associated with the noise space, while the large eigenvalues correspond to eigenvectors with low frequency components, and are associated with the signal space. Thus filtering methods attempt to reconstruct signal space components of the solution, while avoiding reconstruction of noise space components.

For example, the filter factors for two well known filtering methods, truncated spectral value decomposition (TSVD) and Tikhonov regularization, are

$$\phi_i^{\text{tsvd}} = \begin{cases} 1 & \text{if } d_i \geq \delta, \\ 0 & \text{if } d_i < \delta \end{cases} \quad \text{and} \quad \phi_i^{\text{tik}} = \frac{d_i^2}{d_i^2 + \lambda}, \ \lambda > 0, \tag{1.17}$$

where the problem dependent *regularization parameters* δ and λ must be chosen [12]. Several techniques can be used to estimate appropriate choices for the regularization parameters when the SVD is used for filtering (i.e., d_i are the singular values), including generalized cross validation (GCV), L-curve, and the discrepancy principle [9, 11, 19].

In our case, the notation in (1.17) defines a slight abuse of notation, because the eigenvalues d_i are not the singular values: in fact the Jordan canonical form (CF) in (1.15) is different from the singular value decomposition (SVD), since the transform T_n is not orthogonal (indeed it is a rank-2 correction of a symmetric orthogonal matrix). Therefore note that the use of ϕ_i^{tsvd} in (1.16) defines the filtering of the eigenvalues in the Jordan canonical form instead of the more classical filtering of the singular values in the SVD. However, we note that in general computing the SVD can be computationally very expensive, especially in the multidimensional case and also in the strongly symmetric case. Moreover, quite surprisingly, a recent and quite exhaustive set of numerical tests, both in the case of signals and images (see [16, 18]), has shown that the truncated Jordan

canonical form is more or less equivalent to the truncated SVD in terms of quality of the restored object: indeed this is a delicate issue that deserves more attention in the future.

Furthermore, also the so-called Tikhonov regularization needs a further discussion in this direction. Indeed its definition is related to the solution of the linear system $(A^TA + \lambda^2 I)\mathbf{f}_{\text{tik}} = A^T\mathbf{g}$, but the $\mathcal{AR}$ algebra is not closed under transposition. Hence we cannot use the Jordan canonical form for computing the solution of this linear system in a fast and stable way. In [6] it was proposed to replace A^T by A: in such a way the associated Tikhonov-like linear system becomes $(A^2 + \lambda I)\mathbf{f}_{\text{reg}} = A\mathbf{g}$, and now $\mathbf{f}_{\text{reg}}$ is the one in (1.16) with the filter factors ϕ_i^{tik}. In [8] it has been shown that the considered approach, called reblurring, arises when looking at the regularized solution of the continuous problem and then followed by the anti-reflective approximation of each operator separately.

Our final aim is to compute (1.16) in a fast and stable way. We can follow two strategies.

Strategy 1. This is the classic approach implemented for instance with periodic BCs by using three FFTs. In our case we employ the AR-transform, its inverse, and a fast algorithm for computing the eigenvalues.

Algorithm 1

1. $\tilde{\mathbf{g}} = T_n^{-1}\mathbf{g}$,
2. $\mathbf{d} = [h(0), \hat{\mathbf{d}}^T, h(0)]^T$, where $\hat{\mathbf{d}} = [d_2, \ldots, d_{n-1}]^T$ are the eigenvalues of $\tau_{n-2}(h)$ that can be computed by a fast sine transform (FST),
3. $\tilde{\mathbf{f}} = (\boldsymbol{\phi}./\mathbf{d}).*\tilde{\mathbf{g}}$, where the dot operations are component-wise,
4. $\mathbf{f}_{\text{reg}} = T_n\tilde{\mathbf{f}}$.

The product $T_n\tilde{\mathbf{f}}$ can be clearly computed in a fast and stable way by one FST. Indeed for all $\mathbf{x} \in \mathbf{R}^n$

$$T_n\mathbf{x} = \alpha_n^{-1}x_1\begin{bmatrix}1\\ \mathbf{p}\\ 0\end{bmatrix} + \begin{bmatrix}0\\ Q_{n-2}\mathbf{x}(2:n-1)\\ 0\end{bmatrix} + \alpha_n^{-1}x_n\begin{bmatrix}0\\ J\mathbf{p}\\ 1\end{bmatrix},$$

where $\mathbf{x}(2:n-1)$ in Matlab notation is the vector $\mathbf{x}$ with components indexed from 2 to $n-1$. A similar strategy can be followed for computing the matrix–vector product $T_n^{-1}\mathbf{g}$. Instead of $\alpha_n^{-1}\mathbf{p}$ there is $\mathbf{u} = -Q_{n-2}\mathbf{p}$ and instead of $\alpha_n^{-1}J\mathbf{p}$ there is $\mathbf{w} = -Q_{n-2}J\mathbf{p}$. Recalling that $\mathbf{p} = L_{n-2}^{-1}\mathbf{e}_1$ the two vectors $\mathbf{u}$, and $\mathbf{w}$ can be explicitly computed obtaining $u_i = (2n-2)^{-1/2}\cot(\frac{i\pi}{2n-2})$, for $i = 1, \ldots, n-2$ and $\mathbf{w} = \text{diag}_{i=1,\ldots,n-2}(-1)^{i+1}\mathbf{u}$.

Strategy 2. Now we describe a slightly different approach for computing filtered solutions. In particular, we see that the eigenvalues $d_1 = d_n = h(0)$ have eigenvectors essentially belonging to the signal space (for more details, refer to the subsequent Remark 4). Hence we set a priori $\phi_1 = \phi_n = 1$, and rewrite the filtered solution as

$$\mathbf{f}_{\text{reg}} = \frac{\tilde{\mathbf{t}}_1^T \mathbf{g}}{d_1}\mathbf{t}_1 + \frac{\tilde{\mathbf{t}}_n^T \mathbf{g}}{d_n}\mathbf{t}_n + \sum_{i=2}^{n-1} \phi_i \frac{\tilde{\mathbf{t}}_i^T \mathbf{g}}{d_i}\mathbf{t}_i.$$

Now observe that $\tilde{\mathbf{t}}_1 = \mathbf{e}_1, \tilde{\mathbf{t}}_n = \mathbf{e}_n$, and for $i = 2, 3, \ldots, n-1$, $\mathbf{t}_i = [0, \mathbf{q}_{i-1}^T, 0]^T$, where $\mathbf{q}_j$ are columns of the DST-I matrix Q_{n-2}. Thus, the filtered solution can be written as

$$\mathbf{f}_{\text{reg}} = \frac{1}{h(0)}(g_1\mathbf{t}_1 + g_n\mathbf{t}_n) + \begin{bmatrix} 0 \\ \hat{\mathbf{f}}_{\text{reg}} \\ 0 \end{bmatrix}.$$

Let $\mathbf{g} = [g_1, \hat{\mathbf{g}}^T, g_n]^T$, then

$$\begin{aligned} \hat{\mathbf{f}}_{\text{reg}} &= \sum_{i=2}^{n-1} \phi_i \frac{\tilde{\mathbf{t}}_i^T \mathbf{g}}{d_i}\mathbf{q}_{i-1} \\ &= \sum_{i=2}^{n-1} \frac{\phi_i}{d_i}([-Q_{n-2}\mathbf{p}]_{i-1}g_1 + q_{i-1}^T\hat{\mathbf{g}} - [Q_{n-2}J\mathbf{p}]_{i-1}g_n)\mathbf{q}_{i-1} \\ &= Q_{n-2}\mathbf{y}, \end{aligned}$$

where

$$\mathbf{y} = \text{diag}_{i=2,\ldots,n-1}\left(\frac{\phi_i}{d_i}\right)(Q_{n-2}\hat{\mathbf{g}} - g_1 Q_{n-2}\mathbf{p} - g_n Q_{n-2}J\mathbf{p}).$$

Therefore

$$\hat{\mathbf{f}}_{\text{reg}} = Q_{n-2}\,\text{diag}_{i=2,\ldots,n-1}\left(\frac{\phi_i}{d_i}\right)Q_{n-2}\tilde{\mathbf{g}}, \qquad \text{where } \tilde{\mathbf{g}} = \hat{\mathbf{g}} - g_1\mathbf{p} - g_n J\mathbf{p}.$$

The algorithm can be summarized as following

Algorithm 2

1. $\tilde{\mathbf{g}} = \hat{\mathbf{g}} - g_1\mathbf{p} - g_n J\mathbf{p}$,
2. $\hat{\mathbf{f}}_{\text{reg}} = Q_{n-2}\,\text{diag}_{i=2,\ldots,n-1}\left(\frac{\phi_i}{d_i}\right)Q_{n-2}\,\tilde{\mathbf{g}}$ by three FSTs,
3. $\mathbf{f}_{\text{reg}} = \frac{1}{h(0)}\left(\begin{bmatrix} 0 \\ \hat{\mathbf{f}}_{\text{reg}} \\ 0 \end{bmatrix} + g_1\begin{bmatrix} 1 \\ \mathbf{p} \\ 0 \end{bmatrix} + g_n\begin{bmatrix} 0 \\ J\mathbf{p} \\ 1 \end{bmatrix}\right)$.

We can compare the two strategies when ϕ_i are the two classic choices in (1.17). Concerning the spectral truncation performed with ϕ_i^{tsvd}, for every choice of δ we have $\delta \le d_1 = d_n = \max_{i=1,\ldots,n} d_i$ and then $\phi_1^{\text{tsvd}} = \phi_n^{\text{tsvd}} = 1$. Therefore the two strategies are exactly the same. On the other hand, for Tikhonov regularization $\phi_1^{\text{tik}} = \phi_n^{\text{tik}} = \frac{h(0)^2}{h(0)^2+\lambda} \neq 1$ and the two strategies are slightly different as already observed in [4]. Indeed the first one arises from the reblurring approach proposed in [6], while the second one is the same as described in [4] (see Remark 5).

We close this section with a few remarks.

Remark 3 The difference between the two strategies increases with λ, hence it is more evident when the problem requires a substantial amount of regularization, while it is negligible for small values of λ. Furthermore the second approach imposes $\phi_1 = \phi_n = 1$ a priori. Hence, for the implementation, we can also use Algorithm 1 where at step 3 we add $\phi_1 = \phi_n = 1$. In this way the same vector $\mathbf{f}_{\text{reg}}$ is computed, as in Algorithm 2.

Remark 4 As discussed in Remark 2, there is a natural interpretation in terms of frequencies when considering one-dimensional periodic and reflective BCs. The eigenvalue obtained as a sampling of the symbol h at a grid-point close to zero, i.e. close to the maximum point of h, has an associated eigenvector that corresponds to low frequency (signal space) information, while the eigenvalue obtained as a sampling of the symbol h at a grid-point far away from zero (and, in particular, close to π), has an associated eigenvector that corresponds to high frequency (noise space) information. Concerning anti-reflective BCs, the same situation occurs when dealing with the frequency eigenvectors $\sqrt{2/(n-1)}(\sin(jy))|_{\widetilde{G}_n}$, $j = 2, \ldots, n-1$. The other two exceptional eigenvectors generate the space of linear polynomials and therefore they correspond to low frequency information: this intuition is well supported by the fact that the related eigenvalue is $h(0)$, i.e. the maximum and the infinity norm of h, and by the fact that AR-BCs are more precise than other classical BCs.

Remark 5 The matrix $Q_{n-2}\,\text{diag}_{i=2,\ldots,n-1}(\phi_i/d_i)Q_{n-2}$ is the τ matrix with eigenvalues ϕ_i/d_i, for $i = 2, \ldots, n-1$. Therefore step 2 in Algorithm 2 is equivalent to regularizing a linear system with coefficient matrix $\tau_{n-2}(h)$ corresponding to the inner part of $A = AR_n(h)$. It is straightforward that this strategy is exactly the approach used in [4] with homogeneous AR-BCs. Obviously, as already discussed in Remark 1, the two eigenvectors that complete the sine basis can be chosen in several ways: for instance in [4], instead of $[1, \mathbf{p}^T, 0]^T$ and $[0, (J\mathbf{p})^T, 1]^T$, the authors prefer to consider the first vector and the vector $\mathbf{e}$ with all components equal to one, since

$$\mathbf{e} = \begin{bmatrix} 1 \\ \mathbf{p} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ J\mathbf{p} \\ 1 \end{bmatrix}.$$

1.6 A Numerical Comparison of the Two Strategies

For Tikhonov regularization, a comparison between the two strategies described in the previous section is already provided in [4] Sect. 1.6. However, we report an example where, according to Remark 3, we explicitly compare both strategies

varying λ. Since from a computational point of view they are equivalent, we will compare only the quality of the restored signals.

In our example we use the true signal and the out of focus PSF shown in Fig. 1.1. The out of focus blurring is well-known to be severely ill-conditioned.

The two dotted vertical lines shown in the figure of the true signal denote the field of view of our signal; that is, the signal inside the dotted lines represents that part of the signal that can be directly observed, while the signal extending outside the dotted lines represents information that cannot be directly observed, and which must be approximated through boundary conditions. To the blurred signal we add white Gaussian noise (i.e., normally distributed random values with mean 0 and variance 1) with a percentage $||\eta_2/||\mathbf{f}_{\text{blur}}||_2$, where $\mathbf{f}_{\text{blur}}$ is the (noise free) blurred signal and η is the noise. We consider two different levels of noise, 1 and 10%. The observed signals are shown in Fig. 1.2.

Clearly the problem with 10% noise requires a stronger regularization than the problem with 1% noise. For both strategies, in Fig. 1.3 we show the restored signals, while in Fig. 1.4 we report the logarithmic plot of the relative restoration

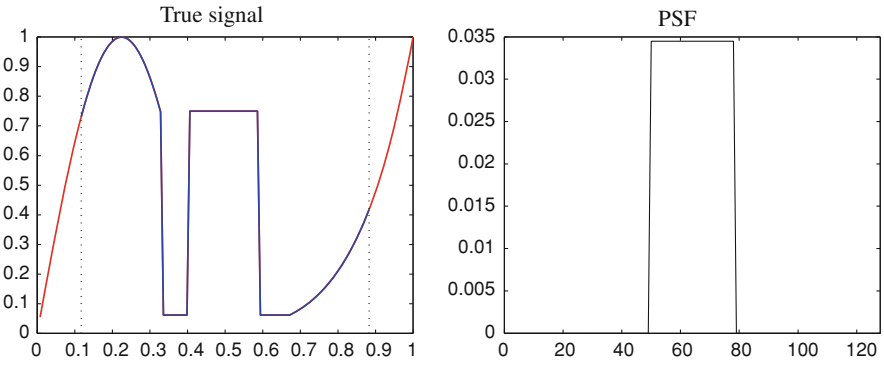


Fig. 1.1 True signal and out of focus PSF

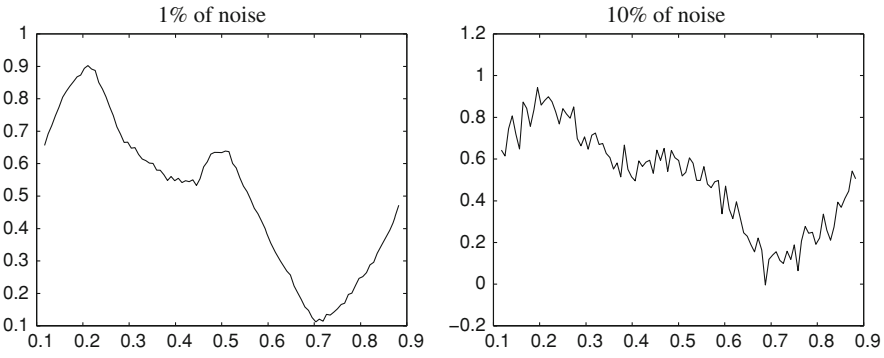


Fig. 1.2 Observed signals

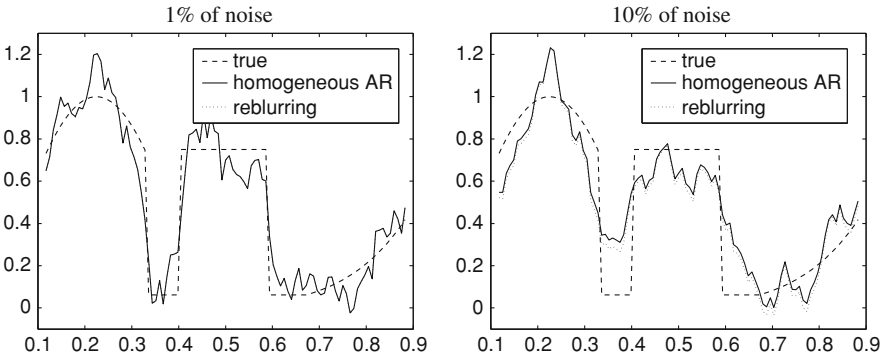


Fig. 1.3 Restored signals for both strategies

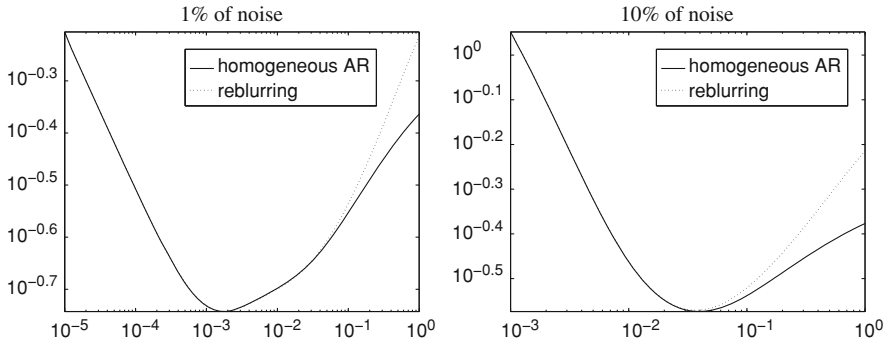


Fig. 1.4 Relative restoration errors vs λ

errors varying λ. In the legends the first strategy is called "re-blurring" according to the terminology of [6] where this idea was proposed, while the second strategy is called "homogeneous AR" according to the terminology in [4], where this variation of the theme was discussed. For this example, we note that for a low level of noise, i.e. in our case 1%, the two strategies are equivalent. Indeed the two restored signals are not distinguishable in Fig. 1.3. Moreover in Fig. 1.4 we observe that the second strategy becomes superior with respect to the first only for $\lambda > 10^{-2}$, while the optimum is reached for $\lambda < 10^{-2}$. On the other hand, for higher levels of noise, i.e. for instance 10%, we need a more substantial regularization and hence a larger value of λ. From Fig. 1.4, by looking in a neighborhood of the optimal value of λ, we notice that the second procedure becomes more precise and in fact the restored signal is computed with a lower error norm. However, in Fig. 1.3 we can see that the quality of the restored signal is not sensibly improved.

Finally, we compare the two strategies by varying the noise level. Figure 1.5 shows in logarithmic scale the optimal relative restoration error changing the noise level for both techniques. When the noise is lower than 10%, the two strategies

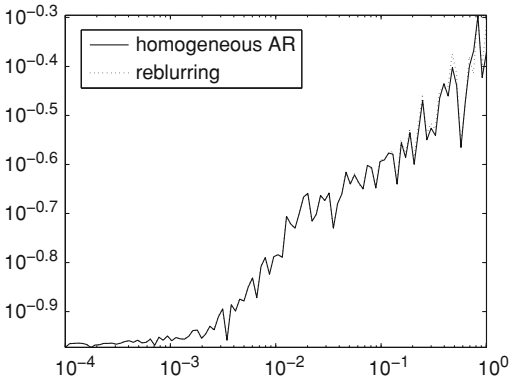


Fig. 1.5 Optimal relative restoration errors vs noise

achieve about the same minimal restoration error. For a noise level greater than 10%, we observe a different minimum and the second proposal seems to be slightly better.

In the previous example, the second strategy seems to be slightly superior with respect to the first one, when the problem requires stronger regularization. On the other hand, when the optimal value of λ is small, the considered procedures are essentially similar, according to the theoretical discussion in Sect. 1.5.

1.7 Multilevel Extension

Here we provide some comments on the extension of our findings to d-dimensional objects with $d > 1$. Note when $d = 1, \mathbf{h}$ is a vector, when $d = 2, \mathbf{h}$ is a 2D array, when $d = 3, \mathbf{h}$ is a 3D tensor, etc. For $d = 1$ and with reference to the previous sections, we have proved that, thanks to the definition of a (fast) AR-transform, it is possible to define a truncated spectral decomposition. However we are well-aware that the real challenge is represented by a general extension to the multi-dimensional setting. This is the topic that we briefly discuss in the rest of the section.

With reference to Sect. 1.2.2 we propose a (canonical) multidimensional extension of the algebras $\mathcal{AR}$ and of the operators $AR_{\boldsymbol{n}}(\cdot), \boldsymbol{n} = (n_1, \ldots, n_d)$: the idea is to use tensor products. If $h = h(\cdot)$ is d-variate real-valued cosine polynomial, then its Fourier coefficients form a real d-dimensional tensor which is strongly symmetric. In addition, $h(\mathbf{y}), \mathbf{y} = (y_1, \ldots, y_d)$, can be written as a linear combination of independent terms of the form $m(\mathbf{y}) = \prod_{j=1}^{d} \cos(c_j y_j)$ where any c_j is a nonnegative integer. Therefore, we define

$$AR_{\boldsymbol{n}}(m(\mathbf{y})) = AR_{n_1}(\cos(c_1 y_1)) \otimes \cdots \otimes AR_{n_d}(\cos(c_d y_d)), \tag{1.18}$$

where $\otimes$ denotes Kronecker product, and we force

$$AR_{\boldsymbol{n}}(\alpha h_1 + \beta h_2) = \alpha AR_{\boldsymbol{n}}(h_1) + \beta AR_{\boldsymbol{n}}(h_2) \tag{1.19}$$

for every real α and β and for every d-variate real-valued cosine polynomials $h_1 = h_1(\cdot)$ and $h_2 = h_2(\cdot)$. It is clear that the request that $AR_{\boldsymbol{n}}(\cdot)$ is a linear operator (for $d > 1$, we impose this property in (1.19) by definition) is sufficient for defining completely the operator in the d-dimensional setting.

With the above definition of the operator $AR_{\boldsymbol{n}}(\cdot)$, we have

1. $\alpha AR_{\boldsymbol{n}}(h_1) + \beta AR_{\boldsymbol{n}}(h_2) = AR_{\boldsymbol{n}}(\alpha h_1 + \beta h_2)$,
2. $AR_{\boldsymbol{n}}(h_1)AR_{\boldsymbol{n}}(h_2) = AR_{\boldsymbol{n}}(h_1 h_2)$,

for real α and β and for cosine functions $h_1 = h_1(\cdot)$ and $h_2 = h_2(\cdot)$.

The latter properties of algebra homomorphism allows to define a commutative algebra $\mathcal{AR}$ of the matrices $AR_{\boldsymbol{n}}(h)$, with $h(\cdot)$ being a d-variate cosine polynomial. By standard interpolation arguments it is easy to see that $\mathcal{AR}$ can be defined as the set of matrices $AR_{\boldsymbol{n}}(h)$, where h is a d-variate cosine polynomial of degree at most $n_j - 3$ in the jth variable for every j ranging in $\{1, \ldots, d\}$: we denote the latter polynomial set by $\mathcal{P}^{(d,\text{even})}_{\boldsymbol{n}-2\boldsymbol{e}}$, with $\boldsymbol{e}$ being the vector of all ones. Here we have to be a bit careful in specifying the meaning of algebra when talking of polynomials. More precisely, for $h_1, h_2 \in \mathcal{P}^{(d,\text{even})}_{\boldsymbol{n}-2\boldsymbol{e}}$ the product $h_1 \cdot h_2$ is the unique polynomial $h \in \mathcal{P}^{(d,\text{even})}_{\boldsymbol{n}-2\boldsymbol{e}}$ satisfying the following interpolation condition

$$h(y) = z_y, \quad z_y \equiv h_1(y)h_2(y), \quad \forall y \in G^{(d)}_{n-2}. \tag{1.20}$$

If the degree of h_1 plus the degree of h_2 in the jth variable does not exceed $n_j - 2, j = 1, \ldots, d$, then the uniqueness of the interpolant implies that h coincides with the product between polynomials in the usual sense. The uniqueness holds also for $d \geq 2$ thanks to the tensor form of the grid $G^{(d)}_{n-2}$ (see [1] for more details). The very same idea applies when considering inversion. In conclusion, with this careful definition of the product/inversion and with the standard definition of addition, $\mathcal{P}^{(d,\text{even})}_{\boldsymbol{n}-2\boldsymbol{e}}$ has become an algebra, showing the vector-space dimension equal to $(n_1 - 2) \cdot (n_2 - 2) \cdots (n_d - 2)$ which coincides with that of $\mathcal{AR}_{\boldsymbol{n}}$.

Without loss of generality and for the sake of notational clarity, in the following we assume $n_j = n$ for $j = 1, \ldots, d$. Thanks to the tensor structure emphasized in (1.18–1.19), and by using Theorem 2 for every term $AR_n(\cos(c_j y_j)), j = 1, \ldots, d$, of $AR_{\boldsymbol{n}}(m)$ the d-level extension of such a theorem easily follows. More precisely, if h is a d-variate real-valued cosine symbol related to a d-dimensional strongly symmetric and normalized mask $\mathbf{h}$, then

$$AR_{\boldsymbol{n}}(h) = T^{(d)}_{\boldsymbol{n}} D_{\boldsymbol{n}}(T^{(d)}_{\boldsymbol{n}})^{-1}, \quad T^{(d)}_{\boldsymbol{n}} = T_n \otimes \cdots \otimes T_n, \tag{1.21}$$

(d times) where $D_{\boldsymbol{n}}$ is the diagonal matrix containing the eigenvalues of $AR_{\boldsymbol{n}}(h)$. The description of $D_{\boldsymbol{n}}$ in d dimensions is quite involved when compared with the case $d = 1$, implicitly reported in Theorem 1.

For a complete analysis of the spectrum of $AR_{\boldsymbol{n}}(h)$ we refer the reader to [1]. Here we give details on a specific aspect. More precisely we attribute a correspondence in a precise and simple way among eigenvalues and eigenvectors, by making recourse only to the main d-variate symbol $h(\cdot)$. Let $\mathbf{x}_{\boldsymbol{n}} = \mathbf{x}_n^{(1)} \otimes \mathbf{x}_n^{(2)} \otimes \cdots \otimes \mathbf{x}_n^{(d)}$ be a column of $T_{\boldsymbol{n}}^{(d)}$, with $\mathbf{x}_n^{(j)} \in \{\alpha_n^{-1}[1, \mathbf{p}^T, 0]^T, \alpha_n^{-1}[0, (J\mathbf{p})^T, 1]^T\}$ or $\mathbf{x}_n^{(j)} = [0, \mathbf{q}_{s_j}^T, 0]^T, 1 \le s_j \le n-2$ and $\mathbf{q}_{s_j}$ is the (s_j)th column of Q_{n-2}, for $j = 1, \ldots, d$. Let

$$\mathcal{F}_{\mathbf{x}_n} = \{j \mid \mathbf{x}_n^{(j)} = \alpha_n^{-1}[1, \mathbf{p}^T, 0]^T \quad \text{or} \quad \mathbf{x}_n^{(j)} = \alpha_n^{-1}[0, (J\mathbf{p})^T, 1]^T\} \subset \{1, \ldots, d\},$$

with $\mathbf{x}_{\boldsymbol{n}}$ being the generic eigenvector, i.e., the generic column of $T_{\boldsymbol{n}}^{(d)}$. The eigenvalue related to $\mathbf{x}_{\boldsymbol{n}}$ is

$$\lambda = h\left(y_1^{(n)}, \ldots, y_d^{(n)}\right) \tag{1.22}$$

where $y_j^{(n)} = 0$ for $j \in \mathcal{F}_{\boldsymbol{x}_n}$ and $y_j^{(n)} = \frac{\pi v_j}{n-1}$ for $j \notin \mathcal{F}_{\mathbf{x}_n}$. We define the d-dimensional grid

$$\widehat{G}_{\boldsymbol{n}}^{(d)} = \widehat{G}_n \tilde{\otimes} \cdots \tilde{\otimes} \widehat{G}_n \quad d \text{ times}, \tag{1.23}$$

as a vector of length n^d whose entries are d-tuples. More precisely given two vectors $\mathbf{x}$ and $\mathbf{y}$ of size p and q, respectively, whose entries are l-tuples and m-tuples, respectively, the operation $\mathbf{z} = \mathbf{x} \tilde{\otimes} \mathbf{y}$ produces a new vector of size pq containing all possible chains $x_i \circ y_j, i = 1, \ldots, p, j = 1, \ldots, q$: in this way the entries of $\mathbf{z}$ are tuples of length $l + m$ and the ordering of the entries in $\mathbf{z}$ is the same as that of the standard Kronecker product. In other words and shortly, we can claim that the operation $\tilde{\otimes}$ is the same as $\otimes$ where the standard product between elements is replaced by the chain operation between tuples. Hence we can evaluate the d-variate function h on $\widehat{G}_{\boldsymbol{n}}^{(d)}$ since its entries are points belonging to the definition domain of h. Using this notation the following compact and elegant result can be stated (its proof is omitted since it is simply the combination of the eigenvalue analysis in [1], of Theorem 2, and of the previous tensor arguments).

Theorem 3 [$AR_{\boldsymbol{n}}(\cdot)$ Jordan Canonical Form.] *The $n^d \times n^d$ AR-BC blurring matrix A, obtained when using a strongly symmetric d-dimensional mask $\mathbf{h}$ such that $h_i = 0$ if $|i_j| \ge n-2$ for some $j \in \{1, \ldots, d\}$ (the d-dimensional degree condition), $n \ge 3$, coincides with*

$$AR_{\boldsymbol{n}}(h) = T_{\boldsymbol{n}}^{(d)} \operatorname{diag}(h(\widehat{G}_{\boldsymbol{n}}^{(d)}))\, (T_{\boldsymbol{n}}^{(d)})^{-1}, \tag{1.24}$$

where $T_{\boldsymbol{n}}^{(d)}$ and $\widehat{G}_{\boldsymbol{n}}^{(d)}$ are defined in (1.21) *and* (1.23).

This shows that the study of the anti-reflective transform is not only of interest in itself both from theoretical and computational viewpoints, but it is also useful in

simplifying the analysis and the interpretation of the eigenvalues studied in [1]: in this sense compare the elegant result in Theorem 3 and the quite tricky combinatorial representation in [1].

It is finally worth observing that, except for more involved multi-index notations, both Algorithms 1 and 2 in Sect. 1.5 are plainly generalized in the multilevel setting, maintaining a cost proportional to three d-level FSTs of size $(n-2)^d$, and the key tool is the simplified eigenvalue–eigenvector correspondence concisely indicated in Theorem 3. Indeed, for Algorithm 1 the only difficult task is the computation in step 2, where we have to compute the eigenvalues in the right order. For this task we refer to [1], where an algorithm is proposed and studied: more specifically the related procedure in [1] is based on a single d-level FST of size $(n-2)^d$ plus lower order computations. For the second strategy in Sect. 1.5, we can operate as suggested in Remark 3. We employ Algorithm 1 where we fix a priori $\phi_{\boldsymbol{k}} = 1$, when the corresponding grid point in $\widehat{G}_{\boldsymbol{n}}^{(d)}$ is equal to zero (notice that, according to (1.23) and to the definition of $\widehat{G}_n$, there exist 2^d of such points).

We conclude this section with an example illustrating the approach discussed above for a two-dimensional imaging problem. We do not take an extensive comparison of the AR-BCs with other classic BCs, like periodic or reflective, since the topic and related issues have been already widely discussed in several works (see e.g. [4, 6, 8]), where the advantage on some classes of images, in terms of the restored image quality, of the application of AR-BCs has been emphasized. Here we propose only a 2D image deblurring example with Gaussian blur and various levels of white Gaussian noise.

The true and the observed images are in Fig. 1.6, where the observed image is affected by a Gaussian blur and 1% noise. We compare the AR-BCs only with the reflective BCs since for this test other BCs like periodic or Dirichlet do not produce satisfactory restorations. In Fig. 1.7 we observe a better restoration and reduced ringing effects at the edges for AR-BCs with respect to reflective BCs.

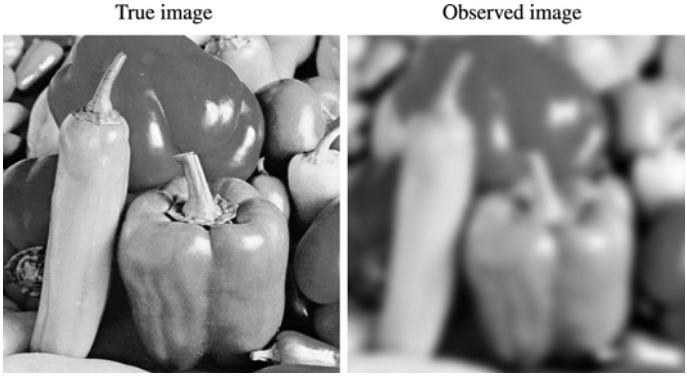


Fig. 1.6 Test problem with Gaussian blur and 1% white Gaussian noise

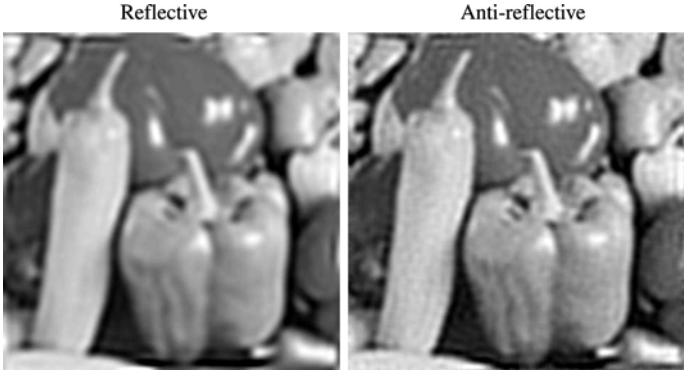


Fig. 1.7 Restored images for the test problem in Fig. 1.6

Table 1.1 Minimum relative restoration errors for the image restoration problem

Noise (%)	Reflective	AR—strategy 1	AR—strategy 2
10	0.1284	0.1261	0.1271
1	0.1188	0.1034	0.1034
0.1	0.1186	0.0989	0.0989

Restored images in Fig. 1.7 are obtained with the minimum relative restoration error varying several values of the regularization parameter λ.

In the previous example, the choice between the two strategies for AR-BCs is not important since, as already observed in Sect. 1.6, they provide different restorations only for a high noise level. This fact is evident in Table 1.1 where we compare the minimum relative restoration errors for the reflective BCs and the two strategies for AR-BCs. As already noted, the two approaches differ only for the four values of the filter factors corresponding to the vertices of the image. We note that for the 10% noise case, all of the approaches give comparable restorations. On the other hand, decreasing the noise, i.e., passing to 1% and then to 0.1% noise, the AR-BCs improve the restoration while the reflective BCs are not able to do that, due to the barrier of the ringing effects.

Acknowledgements The work of the first, second and fourth authors was partially supported by MIUR, grant numbers 2004015437 and 2006017542. The work of the third author was supported in part by the NSF under grant DMS-05-11454.

References

1. Aricò A, Donatelli M, Serra-Capizzano S (2008) Spectral analysis of the anti-reflective algebra. Linear Algebra Appl 428:657–675
2. Bertero M, Boccacci P (1998) Introduction to inverse problems in imaging. IOP Publishing Ltd, London

3. Bini D, Capovani M (1983) Spectral and computational properties of band symmetric Toeplitz matrices. Linear Algebra Appl 52/53:99–125
4. Christiansen M, Hanke M (2008) Deblurring methods using antireflective boundary conditions. SIAM J Sci Comput 30:855–872
5. Davis PJ (1979) Circulant matrices. Wiley, New York
6. Donatelli M, Serra-Capizzano S (2005) Anti-reflective boundary conditions and re-blurring. Inverse Probl 21:169–182
7. Donatelli M, Estatico C, Nagy J, Perrone L, Serra-Capizzano S (2004) Anti-reflective boundary conditions and fast 2d deblurring models. In: Luk FT (ed) Advanced signal processing algorithms, architectures, and implementations VIII, vol 5205. SPIE, pp 380–389
8. Donatelli M, Estatico C, Martinelli A, Serra-Capizzano S (2006) Improved image deblurring with anti-reflective boundary conditions and re-blurring. Inverse Probl 22:2035–2053
9. Engl HW, Hanke M, Neubauer A (2000) Regularization of inverse problems. Kluwer Academic Publishers, Dordrecht
10. Groetsch CW (1984) The theory of Tikhonov regularization for fredholm integral equations of the first kind. Pitman, Boston
11. Hansen PC (1997) Rank-deficient and discrete ill-posed problems. SIAM, Philadelphia
12. Hansen PC, Nagy JG, O'Leary DP (2006) Deblurring images: matrices, spectra, and filtering. SIAM, Philadelphia
13. Horn R, Johnson C (1999) Matrix analysis. Cambridge University Press, Cambridge
14. Ng MK, Chan RH, Tang W (1999) A fast algorithm for deblurring models with Neumann boundary conditions. SIAM J Sci Comput 21:851–866
15. Perrone L (2006) Kronecker product approximations for image restoration with anti-reflective boundary conditions. Numer. Linear Algebra Appl. 13:1–22
16. Rossi F (2006) Tecniche di Filtraggio nella Ricostruzione di Immagini con CC Antiriflettenti (in Italian). Basic Degree Thesis, U. Milano Bicocca, Milano
17. Serra-Capizzano S (2003) A note on anti-reflective boundary conditions and fast deblurring models. SIAM J Sci Comput 25:1307–1325
18. Tablino Possio C (2010) Truncated decompositions and filtering methods with Reflective/ Anti-Reflective boundary conditions: a comparison. In Olshevsky V, Tyrtyshnikov E (eds) Matrix methods: theory, algorithms and applications. Dedicated to the Memory
19. Vogel CR (2002) Computational methods for inverse problems. SIAM, Philadelphia

Chapter 2
Classifications of Recurrence Relations via Subclasses of (H, m)-quasiseparable Matrices

T. Bella, V. Olshevsky and P. Zhlobich

Abstract The results on characterization of orthogonal polynomials and Szegö polynomials via tridiagonal matrices and unitary Hessenberg matrices, respectively, are classical. In a recent paper we observed that tridiagonal matrices and unitary Hessenberg matrices both belong to a wider class of $(H, 1)$-quasiseparable matrices and derived a complete characterization of the latter class via polynomials satisfying certain EGO-type recurrence relations. We also established a characterization of polynomials satisfying three-term recurrence relations via $(H, 1)$-well-free matrices and of polynomials satisfying the Szegö-type two-term recurrence relations via $(H, 1)$-semiseparable matrices. In this paper we generalize all of these results from *scalar* (H,1) to the *block* (H, m) case. Specifically, we provide a complete characterization of (H, m)-quasiseparable matrices via polynomials satisfying *block* EGO-type two-term recurrence relations. Further, (H, m)-semiseparable matrices are completely characterized by the polynomials obeying *block* Szegö-type recurrence relations. Finally, we completely characterize polynomials satisfying m-term recurrence relations via a new class of matrices called (H, m)-well-free matrices.

T. Bella (✉)
Department of Mathematics, University of Rhode Island, Kingston RI 02881-0816, USA
e-mail: tombella@math.uri.edu

V. Olshevsky · P. Zhlobich
Department of Mathematics, University of Connecticut, Storrs CT 06269-3009, USA
e-mail: olshevsky@uconn.edu

P. Zhlobich
e-mail: zhlobich@math.uconn.edu

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_2,

2.1 Introduction

2.1.1 Classical Three-term and Two-term Recurrence Relations and Their Generalizations

It is well known that real-orthogonal polynomials $\{r_k(x)\}$ satisfy three-term recurrence relations of the form

$$r_k(x) = (\alpha_k x - \delta_k) r_{k-1}(x) - \gamma_k \cdot r_{k-2}(x), \quad \alpha_k \neq 0, \quad \gamma_k > 0. \tag{2.1}$$

It is also well known that Szegö polynomials $\{\phi_k^{\#}(x)\}$, or polynomials orthogonal not on a real interval but on the unit circle, satisfy slightly different three-term recurrence relations of the form

$$\phi_k^{\#}(x) = \left[\frac{1}{\mu_k} \cdot x + \frac{\rho_k}{\rho_{k-1}} \frac{1}{\mu_k}\right] \phi_{k-1}^{\#}(x) - \frac{\rho_k}{\rho_{k-1}} \frac{\mu_{k-1}}{\mu_k} \cdot x \cdot \phi_{k-2}^{\#}(x) \tag{2.2}$$

Noting that the essential difference between these two sets of recurrence relations is the presence or absence of the x dependence in the $(k-2)$th polynomial, it is natural to consider the more general three-term recurrence relations of the form

$$r_k(x) = (\alpha_k x - \delta_k) \cdot r_{k-1}(x) - (\beta_k x + \gamma_k) \cdot r_{k-2}(x), \tag{2.3}$$

containing both (2.1) and (2.2) as special cases, and to classify the polynomials satisfying such three-term recurrence relations.

Also, in addition to the three-term recurrence relations (2.2), Szegö polynomials satisfy two-term recurrence relations of the form

$$\begin{bmatrix} \phi_k(x) \\ \phi_k^{\#}(x) \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} 1 & -\rho_k^* \\ -\rho_k & 1 \end{bmatrix} \begin{bmatrix} \phi_{k-1}(x) \\ x\phi_{k-1}^{\#}(x) \end{bmatrix} \tag{2.4}$$

for some auxiliary polynomials $\{\phi_k(x)\}$ (see, for instance, [18, 20]). By relaxing these relations to the more general two-term recurrence relations

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k & \beta_k \\ \gamma_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ (\delta_k x + \theta_k) r_{k-1}(x) \end{bmatrix} \tag{2.5}$$

it is again of interest to classify the polynomials satisfying these two-term recurrence relations.

In [8], these questions were answered, and the desired classifications were given in terms of the classes of matrices $A = [a_{i,j}]_{i,j=1}^{n}$ related to the polynomials $\{r_k(x)\}$ via

$$r_k(x) = \frac{1}{a_{1,0} a_{2,1} \cdots a_{k+1,k}} \det(xI - A)_{(k \times k)}, \quad k = 0, \ldots, n, \tag{2.6}$$

where $A_{(k \times k)}$ denotes the $k \times k$ principal submatrix of A. Note that this relation involves the entries of the matrix A and two additional parameters $a_{1,0}$ and $a_{n+1,n}$

Table 2.1 Correspondence between recurrence relations satisfied by polynomials and related subclasses of quasiseparable matrices, from [8]

Recurrence relations	Matrices
Real-orthogonal three-term (2.1)	Irreducible tridiagonal matrices
Szegö two-term (2.4)/ three-term (2.4)	Almost unitary Hessenberg matrices
General three-term (2.3)	$(H, 1)$-well-free (Definition 4)
Szegö-type two-term (2.5)	$(H, 1)$-semiseparable (Definition 3)
EGO-type two-term (2.16)	$(H, 1)$-quasiseparable (Definition 1)

outside the range of parameters of A. In the context of this paper, these parameters not specified by the matrix A can be any nonzero numbers[1]. These classifications generalized the well-known facts that real-orthogonal polynomials and Szegö polynomials were related to irreducible tridiagonal matrices and almost unitary Hessenberg matrices, respectively, via (2.6). These facts as well as the classifications of polynomials satisfying (2.3), (2.5) and a third set to be introduced later, respectively, are given in Table 2.1.

Furthermore, the classes of matrices listed in Table 2.1 (and formally defined below) were shown in [8] to be related as is shown in Fig. 2.1.

While it is likely that the reader is familiar with tridiagonal and unitary Hessenberg matrices, and perhaps quasiseparable and semiseparable matrices, the class of well-free matrices is less well-known. We take a moment to give a brief description of this class (a more rigorous description is provided in Sect. 2.5.1). A matrix is well-free provided it has **no** columns that consist of all zeros above (but not including) the main diagonal, unless that column of zeros lies to the left of a block of **all** zeros. That is, no columns of the form shown in Fig. 2.2 appear in the matrix.

As stated in Table 2.1, it was shown in [8] that the matrices related to polynomials satisfying recurrence relations of the form (2.3) are not just well-free, but $(H, 1)$-well-free; i.e., they are well-free and also have an $(H, 1)$-quasiseparable structure, which is defined next.

2.1.2 Main Tool: Quasiseparable Structure

In this section we give a definition of the structure central to the results of this paper, and explain one of the results shown above in Table 2.1. We begin with the definition of (H, m)-quasiseparability.

Definition 1 ((H, m)-*quasiseparable matrices*). Let A be a strongly upper Hessenberg matrix (i.e. upper Hessenberg with nonzero subdiagonal elements: $a_{i,j} = 0$

[1] More details on the meaning of these numbers will be provided in Sect. 2.2.1.

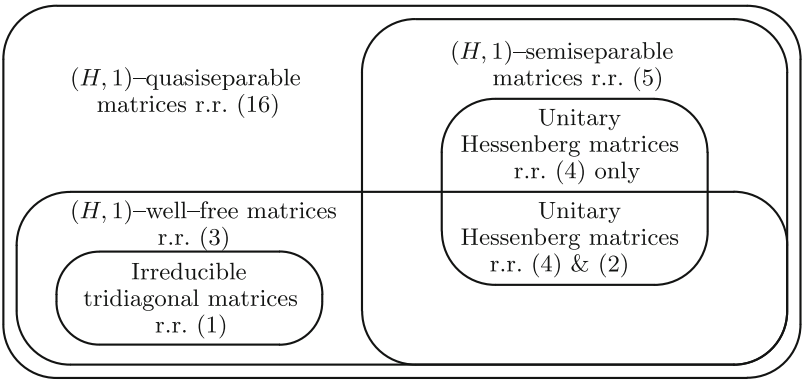
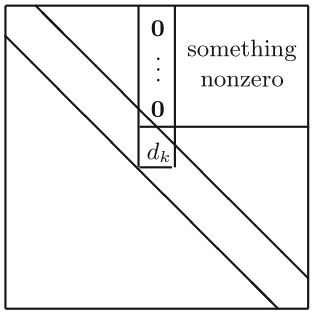


Fig. 2.1 Relations between subclasses of $(H, 1)$-quasiseparable matrices, from [8]

Fig. 2.2 Illustration of a well



for $i > j + 1$, and $a_{i+1,i} \neq 0$ for $i = 1, \ldots, n-1$). Then over all symmetric[2] partitions of the form

$$A = \left[\begin{array}{c|c} * & A_{12} \\ \hline * & * \end{array}\right],$$

(i) if $\max \operatorname{rank} A_{12} = m$, then A is (H, m)-quasiseparable, and
(ii) if $\max \operatorname{rank} A_{12} \leq m$, then A is *weakly* (H, m)-quasiseparable.

For instance, the rank m blocks (respectively rank at most m blocks) of a 5×5 (H, m)-quasiseparable matrix (respectively weakly (H, m)-quasiseparable matrix) would be those shaded below:

$$\left[\begin{array}{c|cccc} \star & \star & \star & \star & \star \\ \hline \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star \\ 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star \end{array}\right] \quad \left[\begin{array}{cc|ccc} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star \\ 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star \end{array}\right] \quad \left[\begin{array}{ccc|cc} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star \\ \hline 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star \end{array}\right] \quad \left[\begin{array}{cccc|c} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star \\ 0 & 0 & \star & \star & \star \\ \hline 0 & 0 & 0 & \star & \star \end{array}\right]$$

[2] $A_{12} = A(1 : k, k + 1 : n), k = 1, \ldots, n - 1$ in the MATLAB notation.

2.1.3 Motivation to Extend Beyond the (H, 1) Case

In this paper, we extend the results of these classifications to include more general recurrence relations. Such generalizations are motivated by several examples for which the results of [8] are inapplicable as they are not order $(H, 1)$; one of the simplest of such is presented next.

Consider the three-term recurrence relations (2.1), one could ask what classes of matrices are related to polynomials satisfying such recurrence relations if more than three terms are included. More specifically, consider recurrence relations of the form

$$x \cdot r_{k-1}(x) = -a_{k,k} r_k(x) - a_{k-1,k} r_{k-1}(x) - \cdots - a_{k-(l-1),k} \cdot r_{k-(l-1)}(x) \tag{2.7}$$

It will be shown that this class of so-called l-recurrent polynomials is related via (2.6) to $(1, l-2)$-banded matrices (i.e., one nonzero subdiagonal and $l-2$ nonzero superdiagonals) of the form

$$A = \begin{bmatrix} a_{0,1} & \cdots & a_{0,l-1} & 0 & \cdots & 0 \\ a_{1,1} & a_{1,2} & \cdots & a_{1,l} & \ddots & \vdots \\ 0 & a_{2,2} & & & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & & a_{n-(l-1),n} \\ \vdots & & \ddots & a_{n-2,n-2} & & \vdots \\ 0 & \cdots & \cdots & 0 & a_{n-1,n-1} & a_{n-1,n} \end{bmatrix}. \tag{2.8}$$

This equivalence cannot follow from the results of [8] as summarized in Table 2.1 because those results are limited to the simplest $(H, 1)$-quasiseparable case. As we shall see in a moment, the matrix A of (2.8) is $(H, l-2)$-quasiseparable.

Considering the motivating example of the matrix A of (2.8), it is easy to see that the structure forces many zeros into the blocks A_{12} of Definition 1 (the shaded blocks above), and hence the ranks of these blocks can be small compared to their size. It can be seen that in the case of an $(1, m)$-banded matrix, the matrices A_{12} have rank at most m, and so are (H, m)-quasiseparable.

This is only one simple example of a need to extend the results listed in Table 2.1 from the scalar $(H, 1)$-quasiseparable case to the block (H, m)-quasiseparable case.

2.1.4 Main Results

The main results of this paper are summarized next by Table 2.2 and Fig. 2.3, analogues of Table 2.1 and Fig. 2.1 above, for the most general case considered in this paper.

Table 2.2 Correspondence between polynomial systems and subclasses of (H, m)-quasiseparable matrices

	Recurrence relations	Matrices
Classical	Real-orthogonal three-term (1)	Irreducible tridiagonal
	Szegö two-term (4)/three-term (2)	Almost unitary Hessenberg
[8]	General three-term (3)	$(H, 1)$-well-free (Definition 4)
	Szegö-type two-term (5)	$(H, 1)$-semiseparable (Definition 3)
	EGO-type two-term (16)	$(H, 1)$-quasiseparable (Definition 1)
This paper	General l-term (45)	(H, m)-well-free (Definition 5)
	Szegö-type two-term (32)	(H, m)-semiseparable (Definition 3)
	EGO-type two-term (2.15)	(H, m)-quasiseparable (Definition 1)

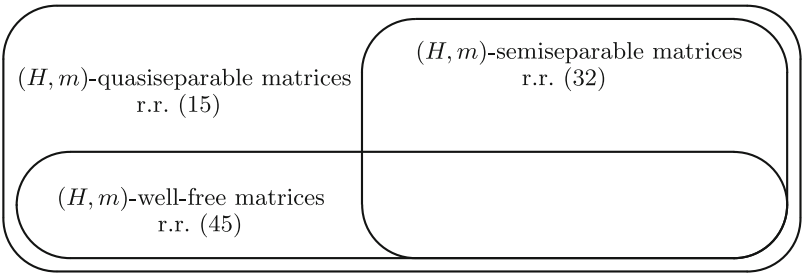


Fig. 2.3 Relations between subclasses of (H, m)-quasiseparable matrices

Table 2.2 and Fig. 2.3 both mention (H, m)-well-free matrices. It is not immediately obvious from the definition of $(H, 1)$-well-free matrices how one should define an (H, m)-well-free matrix in a natural way. In Sect. 2.5, the details of this extension are given, but we briefly describe the new definition here. A matrix is (H, m)-well-free if

$$\operatorname{rank} B_i^{(m)} = \operatorname{rank} B_i^{(m+1)} \quad i = 1, 2, \ldots \tag{2.9}$$

where the matrices $B_i^{(m)}$ are formed from the columns of the partition A_{12} of Definition 1, as

$$A_{12} = \overbrace{\quad\quad}^{B_1^{(m+1)}} \cdots = \overbrace{\quad\quad}^{B_2^{(m+1)}} \cdots$$
$$\underbrace{\quad\quad}_{B_1^{(m)}} \qquad\qquad \underbrace{\quad\quad}_{B_2^{(m)}}$$

We show in this paper that (H, m)-well-free matrices and polynomials satisfying

$$r_k(x) = \underbrace{(\delta_{k,k}x + \varepsilon_{k,k})r_{k-1}x + \cdots + (\delta_{k+m-2,k}x + \varepsilon_{k+m-2,k})r_{k+m-3}(x)}_{m+1 \text{ terms}}, \tag{2.10}$$

provide a complete characterization of each other.

Next, consider briefly the $m = 1$ case to see that this generalization reduces properly in the $(H, 1)$ case. For $m = 1$, this relation implies that no wells of width $m = 1$ form as illustrated in Fig. 2.2.

2.2 Correspondences Between Hessenberg Matrices and Polynomial Systems

In this section we give details of the correspondence between (H, m)-quasiseparable matrices and systems of polynomials defined via (2.6), and explain how this correspondence can be used in classifications of quasiseparable matrices in terms of recurrence relations and vice versa.

2.2.1 A Bijection Between Invertible Triangular Matrices and Polynomial Systems

Let $\mathcal{T}$ be the set of invertible upper triangular matrices and $\mathcal{P}$ be the set of polynomial systems $\{r_k\}$ with $\deg r_k = k$. We next demonstrate that there is a bijection between $\mathcal{T}$ and $\mathcal{P}$. Indeed, given a polynomial system $R = \{r_0(x), r_1(x), \ldots, r_n(x)\} \in \mathcal{P}$ satisfying $\deg(r_k) = k$, there exist unique n-term recurrence relations of the form

$$r_0(x) = a_{0,0}, x \cdot r_{k-1}(x) = a_{k+1,k} \cdot r_k(x) - a_{k,k} \cdot r_{k-1}(x) - \cdots - a_{1,k} \cdot r_0(x), \quad a_{k+1,k} \neq 0, \quad k = 1, \ldots, n \tag{2.11}$$

because this formula represents $x \cdot r_{k-1} \in \mathbb{P}_k$ ($\mathbb{P}_k$ being the space of all polynomials of degree at most k) in terms of $r_k, r_{k-1}, r_{k-2}, \ldots, r_0$, which form a basis in $\mathbb{P}_k$, and hence these coefficients are unique. Forming a matrix $B \in \mathcal{T}$ from these coefficients as $B = \left[a_{i,j}\right]_{i,j=0}^{n}$ (with zeros below the main diagonal), it is clear that there is a bijection between $\mathcal{T}$ and $\mathcal{P}$, as they share the same unique parameters.

It is shown next that this bijection between invertible triangular matrices and polynomials systems (satisfying $\deg r_k(x) = k$) can be viewed as a bijection between strongly Hessenberg matrices together with two free parameters and polynomial systems (satisfying $\deg r_k(x) = k$). Furthermore, the strongly Hessenberg matrices and polynomial systems of this bijection are related via (2.6). Indeed, it was shown in [24] that the *confederate matrix* A, the strongly upper Hessenberg matrix defined by

$$A = \begin{bmatrix} a_{0,1} & a_{0,2} & a_{0,3} & \cdots & a_{0,n} \\ a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{n-2,n} \\ 0 & \cdots & 0 & a_{n-1,n-1} & a_{n-1,n} \end{bmatrix}, \tag{2.12}$$

or in terms of B as

$$B = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & \cdots & a_{0,n} \\ 0 & a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ 0 & 0 & a_{2,2} & a_{2,3} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & a_{n-2,n} \\ \vdots & 0 & \cdots & 0 & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & 0 & a_{n,n} \end{bmatrix} = \begin{bmatrix} a_{0,0} & & & & \\ 0 & & & & \\ \vdots & & A & & \\ 0 & & & & \\ 0 & 0 & \cdots & 0 & a_{n,n} \end{bmatrix}, \tag{2.13}$$

is related to the polynomial system R via (2.6). This shows the desired bijection, with $a_{0,0}$ and $a_{n,n}$ serving as the two free parameters.

Remark 1 Based on this discussion, if $R = \{r_0, r_1, \ldots, r_{n-1}, r_n\}$ is related to a matrix A via (2.6), then $R_{a,b} = \left\{ar_0, \frac{1}{a}r_1, \ldots, \frac{1}{a}r_{n-1}, br_n\right\}$ for any nonzero parameters a and b provides a full characterization of all polynomial systems related to the matrix A.

2.2.2 Generators of (H, m)-quasiseparable Matrices

It is well known that Definition 2, given in terms of ranks is equivalent to another definition in terms of a sparse representation of the elements of the matrix called *generators* of the matrix, see, e.g., [13] and the references therein. Such sparse representations are often used as inputs to fast algorithms involving such matrices. We give next this equivalent definition.

Definition 2 (*Generator definition for* (H, m)*-quasiseparable matrices*). A matrix A is called (H, m)-quasiseparable if (i) it is upper Hessenberg, and (ii) it can be represented in the form

$$A = \begin{array}{|lll} d_1 & & g_i b_{ij}^{\times} h_j \\ & \ddots & \\ p_2 q_1 & & \ddots \\ & \ddots & \\ 0 & & p_n q_{n-1} \quad d_n \end{array} \tag{2.14}$$

where $b_{ij}^{\times} = b_{i+1} \cdots b_{j-1}$ for $j > i+1$ and $b_{ij}^{\times} = 1$ for $j = i+1$. The elements

$$\{p_k, q_k, d_k, g_k, b_k, h_k\},$$

called the generators of the matrix A, are matrices of sizes

	$p_{k+1}q_k$	d_k	g_k	b_k	h_k
Sizes	1×1	1×1	$1 \times u_k$	$u_{k-1} \times u_k$	$u_{k-1} \times 1$
Range	$k \in [1, n-1]$	$k \in [1, n]$	$k \in [1, n-1]$	$k \in [2, n-1]$	$k \in [2, n]$

subject to $\max_k u_k = m$. The numbers $u_k, k = 1, \ldots, n-1$ are called the *orders* of these generators.

Remark 2 The generators of an (H, m)-quasiseparable matrix give us an $\mathcal{O}(nm^2)$ representation of the elements of the matrix. In the $(H, 1)$-quasiseparable case, where all generators can be chosen simply as scalars, this representation is $\mathcal{O}(n)$.

Remark 3 The subdiagonal elements, despite being determined by a single value, are written as a product $p_{k+1}q_k, k = 1, \ldots, n-1$ to follow standard notations used in the literature for quasiseparable matrices. We emphasize that this product acts as a single parameter in the Hessenberg case to which this paper is devoted.

Remark 4 The generators in Definition 2 can be always chosen to have sizes $u_k = m$ for all k by padding them with zeros to size m.

Also, the ranks of the submatrices A_{12} of Definition 1 represent the smallest possible sizes of the corresponding generators. That is, denoting by $A_{12}^{(k)} = A(1 : k, k+1 : n)$ the partition A_{12} of the k-th symmetric partition, then

$$\operatorname{rank} A_{12}^{(k)} \leq u_k, \quad k = 1, \ldots, n.$$

Furthermore, if generators can be chosen such that

$$\max_k \operatorname{rank} A_{12}^{(k)} = \max_k u_k = m,$$

then A is an (H, m)-quasiseparable matrix, whereas if

$$\max_k \operatorname{rank} A_{12}^{(k)} \leq \max_k u_k = m,$$

then A is a *weakly* (H, m)-quasiseparable matrix, following the terminology of Definition 1. As stated above, we will avoid making explicit distinctions between (H, m)-quasiseparable matrices and weakly (H, m)-quasiseparable matrices.

For details on the existence of minimal size generators, see [16].

2.2.3 A Relation Between Generators of Quasiseparable Matrices and Recurrence Relations for Polynomials

One way to establish a bijection (up to scaling as described in Remark 1) between subclasses of (H, m)-quasiseparable matrices and polynomial systems specified by

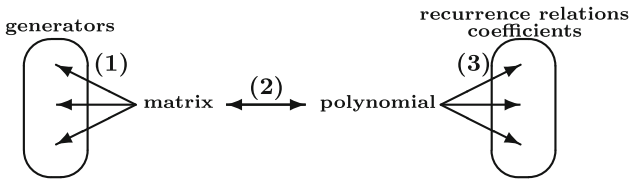


Fig. 2.4 Relations between subclasses of (H, m)-quasiseparable matrices and polynomials

recurrence relations is to deduce conversion rules between generators of the classes of matrices and coefficients of the recurrence relations. In this approach, a difficulty is encountered which is described by Fig. 2.4.

The difficulty is that the relation (2.2) shown in the picture is one-to-one correspondence but (2.1) and (2.3) are not. This fact is illustrated by the next two examples.

Example 1 (Nonuniqueness of recurrence relation coefficients). In contrast to the n-term recurrence relations (2.11), other recurrence relations such as the l-term recurrence relations (2.45) corresponding to a given polynomial system are not unique. As a simple example of a system of polynomials satisfying more than one set of recurrence relations of the form (2.45), consider the monomials $R = \{1, x, x^2, \ldots, x^n\}$, easily seen to satisfy the recurrence relations

$$r_0(x) = 1, \quad r_k(x) = x \cdot r_{k-1}(x), \quad k = 1, \ldots, n$$

as well as the recurrence relations

$$\begin{aligned} r_0(x) = 1, \quad r_1(x) = x \cdot r_{k-1}(x), \\ r_k(x) = (x+1) \cdot r_{k-1}(x) - x \cdot r_{k-2}(x), \quad k = 2, \ldots, n. \end{aligned}$$

Hence a given system of polynomials may be expressed using the same recurrence relations but with different coefficients of those recurrence relations.

Example 2 (Nonuniqueness of (H, m)-quasiseparable generators). Similarly, given an (H, m)-quasiseparable matrix, there is a freedom in choosing the set of generators of Definition 2. As a simple example, consider the matrix

$$\begin{bmatrix} 0 & \frac{1}{2} & 0 & \cdots & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & \ddots & \vdots \\ 0 & \frac{1}{2} & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \frac{1}{2} \\ 0 & \cdots & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

corresponding to a system of Chebyshev polynomials. It is obviously $(H, 1)$-quasiseparable and can be defined by different sets of generators, with either $g_k = 1, h_k = \frac{1}{2}$ or $g_k = \frac{1}{2}, h_k = 1$.

Remark 5 To overcome the difficulties of the nonuniqueness demonstrated here, we can define equivalence classes of generators describing the same matrix and equivalence classes of recurrence relations describing the same polynomials. Working with representatives of these equivalence classes resolves the difficulty.

We begin classification of recurrence relations of polynomials with considering EGO-type two-term recurrence relations (2.15) in Sect. 2.3 and associating the set of all (H, m)-quasiseparable matrices with them. Section 2.4 covers the correspondence between polynomials satisfying (2.32) and (H, m)-semiseparable matrices. In Sect. 2.5 we consider l-term recurrence relations (2.45) and (H, m)-well-free matrices.

2.3 (H, m)-quasiseparable Matrices and EGO-type Two-term Recurrence Relations (2.15)

In this section, we classify the recurrence relations corresponding to the class of (H, m)-quasiseparable matrices. The next theorem is the main result of this section.

Theorem 1 *Suppose A is a strongly upper Hessenberg matrix. Then the following are equivalent.*

(i) *A is (H, m)-quasiseparable.*
(ii) *There exist auxiliary polynomials $\{F_k(x)\}$ for some α_k, β_k, and γ_k of sizes $m \times m$, $m \times 1$ and $1 \times m$, respectively, such that the system of polynomials $\{r_k(x)\}$ related to A via (2.6) satisfies the EGO-type two-term recurrence relations*

$$\left[\begin{array}{c|c} F_0(x) & 0 \\ \hline r_0(x) & a_{0,0} \end{array}\right] = \left[\begin{array}{c|c} 0 & \\ \vdots & \\ \hline 0 & \end{array}\right], \quad \left[\begin{array}{c|c} F_k(x) & \\ \hline r_k(x) & \end{array}\right] = \left[\begin{array}{c|c} \alpha_k & \beta_k \\ \hline \gamma_k & \delta_k x + \theta_k \end{array}\right] \left[\begin{array}{c|c} F_{k-1}(x) & \\ \hline r_{k-1}(x) & \end{array}\right]. \tag{2.15}$$

Remark 6 Throughout the paper, we will not distinguish between (H, m)-quasiseparable and weakly (H, m)-quasiseparable matrices. The difference is technical; for instance, considering an $(H, 2)$-quasiseparable matrix as a weakly $(H, 3)$-quasiseparable matrix corresponds to artificially increasing the size of the vectors $F_k(x)$ in (2.15) by one. This additional entry corresponds to a polynomial system that is identically zero, or otherwise has no influence on the other polynomial systems. In a

similar way, any results stated for (H, m)-quasiseparable matrices are valid for weakly (H, m)-quasiseparable matrices through such trivial modifications.

This theorem, whose proof will be provided by the lemma and theorems of this section, is easily seen as a generalization of the following result for the $(H, 1)$-quasiseparable case from [8].

Corollary 1 *Suppose A is a strongly Hessenberg matrix. Then the following are equivalent.*

(i) *A is $(H, 1)$-quasiseparable.*
(ii) *There exist auxiliary polynomials $\{F_k(x)\}$ for some scalars α_k, β_k, and γ_k such that the system of polynomials $\{r_k(x)\}$ related to A via (2.6) satisfies the EGO-type two-term recurrence relations*

$$\begin{bmatrix} F_0(x) \\ r_0(x) \end{bmatrix} = \begin{bmatrix} 0 \\ a_{0,0} \end{bmatrix}, \quad \begin{bmatrix} F_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k & \beta_k \\ \gamma_k & \delta_k x + \theta_k \end{bmatrix} \begin{bmatrix} F_{k-1}(x) \\ r_{k-1}(x) \end{bmatrix}. \tag{2.16}$$

In establishing the one-to-one correspondence between the class of polynomials satisfying (2.15) and the class of (H, m)-quasiseparable matrices, we will use the following lemma which was given in [7] and is a consequence of Definition 2 and [24].

Lemma 1 *Let A be an (H, m)-quasiseparable matrix specified by its generators as in Definition 2. Then a system of polynomials $\{r_k(x)\}$ satisfies the recurrence relations*

$$r_k(x) = \frac{1}{p_{k+1} q_k} \left[(x - d_k) r_{k-1}(x) - \sum_{j=0}^{k-2} g_{j+1} b^{\times}_{j+1,k} h_k r_j(x) \right], \tag{2.17}$$

if and only if $\{r_k(x)\}$ is related to A via (2.6).

Note that we have not specified the sizes of matrices g_k, b_k and h_k in (2.17) explicitly but the careful reader can check that all matrix multiplications are well defined. We will omit explicitly listing the sizes of generators where it is possible.

Theorem 2 (EGO-type two-term recurrence relations $\Rightarrow$ (H, m)-quasiseparable matrices.) *Let R be a system of polynomials satisfying the EGO-type two-term recurrence relations (2.15). Then the (H, m)-quasiseparable matrix A defined by*

$$\begin{bmatrix} -\frac{\theta_1}{\delta_1} & -\frac{1}{\delta_2}\gamma_2\beta_1 & -\frac{1}{\delta_3}\gamma_3\alpha_2\beta_1 & -\frac{1}{\delta_4}\gamma_4\alpha_3\alpha_2\beta_1 & \cdots & -\frac{1}{\delta_n}\gamma_n\alpha_{n-1}\alpha_{n-2}\cdots\alpha_3\alpha_2\beta_1 \\ \frac{1}{\delta_1} & -\frac{\theta_2}{\delta_2} & -\frac{1}{\delta_3}\gamma_3\beta_2 & -\frac{1}{\delta_4}\gamma_4\alpha_3\beta_2 & \cdots & -\frac{1}{\delta_n}\gamma_n\alpha_{n-1}\alpha_{n-2}\cdots\alpha_3\beta_2 \\ 0 & \frac{1}{\delta_2} & -\frac{\theta_3}{\delta_3} & -\frac{1}{\delta_4}\gamma_4\beta_3 & \ddots & -\frac{1}{\delta_n}\gamma_n\alpha_{n-1}\cdots\alpha_4\beta_3 \\ 0 & 0 & \frac{1}{\delta_3} & -\frac{\theta_4}{\delta_4} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & -\frac{1}{\delta_n}\gamma_n\beta_{n-1} \\ 0 & \cdots & 0 & 0 & \frac{1}{\delta_{n-1}} & -\frac{\theta_n}{\delta_n} \end{bmatrix} \tag{2.18}$$

with generators

$$d_k = -\frac{\theta_k}{\delta_k},\ k = 1, \ldots, n, \quad \boxed{g_k} = \boxed{\beta_k^T},\ k = 1, \ldots, n-1,$$

$$p_{k+1} q_k = \frac{1}{\delta_k},\ k = 1, \ldots, n-1, \quad \boxed{h_k} = -\frac{1}{\delta_k} \boxed{\gamma_k^T},\ k = 2, \ldots, n,$$

$$\boxed{b_k} = \boxed{\alpha_k^T},\ k = 2, \ldots, n-1,$$

corresponds to the system of polynomials R via (2.6).

Proof Considering EGO-type recurrence relations (2.15) we begin with

$$r_k(x) = (\delta_k x + \theta_k) r_{k-1}(x) + \gamma_k F_{k-1}(x). \tag{2.19}$$

Using the relation $F_{k-1}(x) = \alpha_{k-1} F_{k-2}(x) + \beta_{k-1} r_{k-2}(x)$, (2.19) becomes

$$r_k(x) = (\delta_k x + \theta_k) r_{k-1}(x) + \gamma_k \beta_{k-1} r_{k-2}(x) + \gamma_k \alpha_{k-1} F_{k-2}(x) \tag{2.20}$$

The Eq. (2.20) contains $F_{k-2}(x)$ which can be eliminated as it was done on the previous step. Using the relation $F_{k-2}(x) = \alpha_{k-2} F_{k-3}(x) + \beta_{k-2} r_{k-3}(x)$ we get

$$\begin{aligned} r_k(x) = {} & (\delta_k x + \theta_k) r_{k-1}(x) + \gamma_k \beta_{k-1} r_{k-2}(x) + \gamma_k \alpha_{k-1} \beta_{k-2} r_{k-3}(x) \\ & + \gamma_k \alpha_{k-1} \alpha_{k-2} F_{k-3}(x). \end{aligned}$$

Continue this process and noticing that F_0 is the vector of zeros we will obtain the n-term recurrence relations

$$\begin{aligned} r_k(x) = {} & (\delta_k x + \theta_k) r_{k-1}(x) + \gamma_k \beta_{k-1} r_{k-2}(x) + \gamma_k \alpha_{k-1} \beta_{k-2} r_{k-3}(x) \\ & + \gamma_k \alpha_{k-1} \alpha_{k-2} \beta_{k-3} r_{k-4}(x) + \cdots + \gamma_k \alpha_{k-1} \ldots \alpha_2 \beta_1 r_0(x), \end{aligned} \tag{2.21}$$

which define the matrix (2.18) with the desired generators by using the n-term recurrence relations (2.17).

Theorem 3 ((H, m)-quasiseparable matrices $\Rightarrow$ EGO-type two-term recurrence relations.) *Let A be an (H, m)-quasiseparable matrix specified by the generators $\{p_k, q_k, d_k, g_k, b_k, h_k\}$. Then the polynomial system R corresponding to A satisfies*

$$\left[\begin{array}{c} F_0(x) \\ \hline r_0(x) \end{array}\right] = \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline a_{0,0} \end{array}\right], \quad \left[\begin{array}{c} F_k(x) \\ \hline r_k(x) \end{array}\right] = \left[\begin{array}{cc|c} & & \\ \multicolumn{2}{c|}{\alpha_k} & \beta_k \\ & & \\ \hline \multicolumn{2}{c|}{\gamma_k} & \delta_k x + \theta_k \end{array}\right] \left[\begin{array}{c} F_{k-1}(x) \\ \hline r_{k-1}(x) \end{array}\right], \tag{2.22}$$

with

$$\alpha_k = \frac{p_k}{p_{k+1}} b_k^T, \quad \beta_k = -\frac{1}{p_{k+1}} g_k^T, \quad \gamma_k = \frac{p_k}{p_{k+1} q_k} h_k^T, \quad \delta_k = \frac{1}{p_{k+1} q_k}, \quad \theta_k = -\frac{d_k}{p_{k+1} q_k}.$$

Proof It is easy to see that every system of polynomials satisfying $\deg r_k = k$ (e.g. the one defined by (2.22)) satisfy also the n-term recurrence relations

$$r_k(x) = (\alpha_k x - a_{k-1,k}) \cdot r_{k-1}(x) - a_{k-2,k} \cdot r_{k-2}(x) - \ldots - a_{0,k} \cdot r_0(x) \tag{2.23}$$

for some coefficients $\alpha_k, a_{k-1,k}, \ldots, a_{0,k}$. The proof is presented by showing that these n-term recurrence relations in fact coincide with (2.17), so these coefficients coincide with those of the n-term recurrence relations of the polynomials R. Using relations for $r_k(x)$ and $F_{k-1}(x)$ from (2.22), we have

$$r_k(x) = \frac{1}{p_{k+1} q_k} \left[(x - d_k) r_{k-1}(x) - g_{k-1} h_k r_{k-2}(x) + p_{k-1} h_k^T b_{k-1}^T F_{k-2}(x) \right]. \tag{2.24}$$

Notice that again using (2.22) to eliminate $F_{k-2}(x)$ from the Eq. (2.24) will result in an expression for $r_k(x)$ in terms of $r_{k-1}(x), r_{k-2}(x), r_{k-3}(x), F_{k-3}(x)$, and $r_0(x)$ without modifying the coefficients of $r_{k-1}(x)$, $r_{k-2}(x)$, or $r_0(x)$. Again applying (2.22) to eliminate $F_{k-3}(x)$ results in an expression in terms of $r_{k-1}(x)$, $r_{k-2}(x), r_{k-3}(x), r_{k-4}(x), F_{k-4}(x)$, and $r_0(x)$ without modifying the coefficients of $r_{k-1}(x)$, $r_{k-2}(x)$, $r_{k-3}(x)$, or $r_0(x)$. Continuing in this way, the n-term recurrence relations of the form (2.23) are obtained without modifying the coefficients of the previous ones.

Suppose that for some $0 < j < k-1$ the expression for $r_k(x)$ is of the form

$$\begin{aligned} r_k(x) = \frac{1}{p_{k+1} q_k} [& (x - d_k) r_{k-1}(x) - g_{k-1} h_k r_{k-2}(x) \\ & - \cdots - g_{j+1} b_{j+1,k}^{\times} h_k r_j(x) + p_{j+1} h_k^T (b_{j,k}^{\times})^T F_j(x)]. \end{aligned} \tag{2.25}$$

Using (2.22) for $F_j(x)$ gives the relation

$$F_j(x) = \frac{1}{p_{j+1} q_j} \left(p_j q_j b_j^T F_{j-1}(x) - q_j g_j^T r_{j-1}(x) \right). \tag{2.26}$$

Inserting (2.26) into (2.25) gives

$$r_k(x) = \frac{1}{p_{k+1}q_k}\Big[(x-d_k)r_{k-1}(x) - g_{k-1}h_k r_{k-2}(x) - \cdots - g_j b^{\times}_{j,k} h_k r_{j-1}(x) + p_j h_k^T (b^{\times}_{j-1,k})^T F_{j-1}(x)\Big]. \tag{2.27}$$

Therefore since (2.24) is the case of (2.25) for $j = k-2$, (2.25) is true for each $j = k-2, k-3, \ldots, 0$, and for $j = 0$, using the fact that $F_0 = 0$ we have

$$r_k(x) = \frac{1}{p_{k+1}q_k}\Big[(x-d_k)r_{k-1}(x) - g_{k-1}h_k r_{k-2}(x) - \cdots - g_1 b^{\times}_{1,k} h_k r_0(x)\Big]. \tag{2.28}$$

Since these coefficients coincide with (2.17) that are satisfied by the polynomial system R, the polynomials given by (2.22) must coincide with these polynomials. This proves the theorem.

These last two theorems provide the proof for Theorem 1, and complete the discussion of the recurrence relations related to (H, m)-quasiseparable matrices.

2.4 (H, m)-semiseparable Matrices and Szegö-type Two-term Recurrence Relations (2.32)

In this section we consider a class of (H, m)-semiseparable matrices defined next.

Definition 3 (*(H, m)-semiseparable matrices*) A matrix A is called (H, m)-semiseparable if (i) it is strongly upper Hessenberg, and (ii) it is of the form

$$A = B + triu(A_U, 1)$$

with $\text{rank}(A_U) = m$ and a lower bidiagonal matrix B, where following the MATLAB command `triu`, $triu(A_U, 1)$ denotes the strictly upper triangular portion of the matrix A_U.

Paraphrased, an (H, m)-semiseparable matrix has arbitrary diagonal entries, arbitrary nonzero subdiagonal entries, and the strictly upper triangular part of a rank m matrix. Obviously, an (H, m)-semiseparable matrix is (H, m)-quasiseparable. Indeed, let A be (H, m)-semiseparable and $n \times n$. Then it is clear that, if $A_{12}^{(k)}$ denotes the matrix A_{12} of the k-th partition of Definition 1, then

$$\text{rank } A_{12}^{(k)} = \text{rank } A(1:k, k+1:n) = \text{rank } A_U(1:k, k+1:n) \le m, \quad k = 1, \ldots, n-1,$$

and A is (H, m)-quasiseparable by Definition 1.

Example 3 (Unitary Hessenberg matrices are $(H, 1)$-semiseparable). Consider again the unitary Hessenberg matrix

$$H = \begin{bmatrix} -\rho_0^*\rho_1 & -\rho_0^*\mu_1\rho_2 & -\rho_0^*\mu_1\mu_2\rho_3 & \cdots & -\rho_0^*\mu_1\mu_2\mu_3\cdots\mu_{n-1}\rho_n \\ \mu_1 & -\rho_1^*\rho_2 & -\rho_1^*\mu_2\rho_3 & \cdots & -\rho_1^*\mu_2\mu_3\cdots\mu_{n-1}\rho_n \\ 0 & \mu_2 & -\rho_2^*\rho_3 & \cdots & -\rho_2^*\mu_3\cdots\mu_{n-1}\rho_n \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mu_{n-1} & -\rho_{n-1}^*\rho_n \end{bmatrix} \tag{2.29}$$

which corresponds to a system of Szegö polynomials. Its strictly upper triangular part is the same as in the matrix

$$B = \begin{bmatrix} -\rho_0^*\rho_1 & -\rho_0^*\mu_1\rho_2 & -\rho_0^*\mu_1\mu_2\rho_3 & \cdots & -\rho_0^*\mu_1\mu_2\mu_3\cdots\mu_{n-1}\rho_n \\ -\frac{\rho_1\rho_1^*}{\mu_1} & -\rho_1^*\rho_2 & -\rho_1^*\mu_2\rho_3 & \cdots & -\rho_1^*\mu_2\mu_3\cdots\mu_{n-1}\rho_n \\ -\frac{\rho_1\rho_2^*}{\mu_1\mu_2} & -\frac{\rho_2\rho_2^*}{\mu_2} & -\rho_2^*\rho_3 & \cdots & -\rho_2^*\mu_3\cdots\mu_{n-1}\rho_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{\rho_1\rho_{n-1}^*}{\mu_1\mu_2\cdots\mu_{n-1}} & -\frac{\rho_2\rho_{n-1}^*}{\mu_2\mu_3\cdots\mu_{n-1}} & -\frac{\rho_3\rho_{n-1}^*}{\mu_3\mu_4\cdots\mu_{n-1}} & \cdots & -\rho_{n-1}^*\rho_n \end{bmatrix}. \tag{2.30}$$

which can be constructed as, by definition,[3] $\mu_k \neq 0$, $k = 1, \ldots, n-1$. It is easy to check that the rank of the matrix B is one.[4] Hence the matrix (2.29) is $(H, 1)$-semiseparable. Recall that any unitary Hessenberg matrix (2.29) uniquely corresponds to a system of Szegö polynomials satisfying the recurrence relations

$$\begin{bmatrix} \phi_0(x) \\ \phi_0^{\#}(x) \end{bmatrix} = \frac{1}{\mu_0}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \phi_k(x) \\ \phi_k^{\#}(x) \end{bmatrix} = \frac{1}{\mu_k}\begin{bmatrix} 1 & -\rho_k^* \\ -\rho_k & 1 \end{bmatrix}\begin{bmatrix} \phi_{k-1}(x) \\ x\phi_{k-1}^{\#}(x) \end{bmatrix}. \tag{2.31}$$

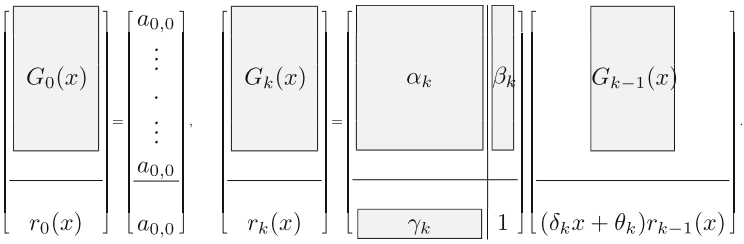
The next theorem gives a classification of the class of (H, m)-semiseparable matrices in terms of two-term recurrence relations that naturally generalize the Szegö-type two term recurrence relations. Additionally, it gives a classification in terms of their generators as in Definition 2.

Theorem 4 *Suppose A is a strongly upper Hessenberg $n \times n$ matrix. Then the following are equivalent.*

(i) *A is (H, m)-semiseparable.*
(ii) *There exists a set of generators of Definition 2 corresponding to A such that b_k is invertible for $k = 2, \ldots, n$.*
(iii) *There exist auxiliary polynomials $\{G_k(x)\}$ for some α_k, β_k, and γ_k of sizes $m \times m$, $m \times 1$ and $1 \times m$, respectively, such that the system of polynomials $\{r_k(x)\}$ related to A via (2.6) satisfies the Szegö-type two-term recurrence relations*

[3] The parameters μ_k associated with the Szegö polynomials are defined by $\mu_k = \sqrt{1 - |\rho_k|^2}$ for $0 \leq |\rho_k| < 1$ and $\mu_k = 1$ for $|\rho_k| = 1$, and since $|\rho_k| \leq 1$ for all k, we always have $\mu_k \neq 0$.

[4] Every i-th row of B equals the row number $(i-1)$ times $\rho_{i-1}^*/\rho_{i-2}^*\mu_{i-1}$.



(2.32)

This theorem, whose proof follows from the results later in this section, leads to the following corollary, which summarizes the results for the simpler class of $(H, 1)$-semiseparable matrices as given in [8].

Corollary 2 *Suppose A is an $(H, 1)$-quasiseparable matrix. Then the following are equivalent.*

(i) *A is $(H, 1)$-semiseparable.*
(ii) *There exists a set of generators of Definition 2 corresponding to A such that $b_k \neq 0$ for $k = 2, \ldots, n$.*
(iii) *There exist auxiliary polynomials $\{G_k(x)\}$ for some scalars α_k, β_k, and γ_k such that the system of polynomials $\{r_k(x)\}$ related to A via (2.6) satisfies the Szegö-type two-term recurrence relations*

$$\begin{bmatrix} G_0(x) \\ r_0(x) \end{bmatrix} = \begin{bmatrix} a_{0,0} \\ a_{0,0} \end{bmatrix}, \quad \begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k & \beta_k \\ \gamma_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ (\delta_k x + \theta_k) r_{k-1}(x) \end{bmatrix}. \tag{2.33}$$

2.4.1 (H, m)-semiseparable Matrices: Generator Classification

We next give a lemma that provides a classification of (H, m)-semiseparable matrices in terms of generators of an (H, m)-quasiseparable matrix.

Lemma 2 *An (H, m)-quasiseparable matrix is (H, m)-semiseparable if and only if there exists a choice of generators $\{p_k, q_k, d_k, g_k, b_k, h_k\}$ of the matrix such that matrices b_k are nonsingular*[5] *for all $k = 2, \ldots, n-1$.*

Proof Let A be (H, m)-semiseparable with $triu(A, 1) = triu(A_U, 1)$, where $\text{rank}(A_U) = m$. The latter statement implies that there exist row vectors g_i and column vectors h_j of sizes m such that $A_U(i, j) = g_i h_j$ for all i, j, and therefore we have $A_{ij} = g_i h_j,\ i < j$ or $A_{ij} = g_i b_{ij}^{\times} h_j,\ i < j$ with $b_k = I_m$.

Conversely, suppose the generators of A are such that b_k are invertible matrices for $k = 2, \ldots, n-1$. Then the matrices

[5] The invertibility of b_k implies that all b_k are square $m \times m$ matrices.

$$A_U = \begin{cases} g_i b_{i,j}^{\times} h_j & \text{if } 1 \le i < j \le n \\ g_i b_i^{-1} h_i & \text{if } 1 < i = j < n \\ g_i (b_{j-1,i+1}^{\times})^{-1} h_j & \text{if } 1 < j < i < n \\ 0 & \text{if } j = 1 \text{ or } i = n \end{cases} \quad B = \begin{cases} d_i & \text{if } 1 \le i = j \le n \\ p_i q_j & \text{if } 1 \le i+1 = j \le n \\ 0 & \text{otherwise} \end{cases}$$

are well defined, $\text{rank}(A_U) = m, B$ is lower bidiagonal, and $A = B + triu(A_U, 1)$.

Remark 7 We emphasize that the previous lemma guarantees the existence of generators of a (H, m)-semiseparable matrix with invertible matrices b_k, and that this condition need not be satisfied by all such generator representations. For example, the following matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 2 & 2 & 2 & 0 \\ & 1 & 1 & 3 & 3 & 0 \\ & & 1 & 1 & 4 & 0 \\ & & & 1 & 1 & 0 \\ & & & & 1 & 1 \end{bmatrix}$$

is $(H, 1)$-semiseparable, however it is obviously possible to choose a set of generators for it with $b_5 = 0$.

2.4.2 (H, m)-semiseparable Matrices. Recurrence Relations Classification

In this section we present theorems giving the classification of (H, m)-semiseparable matrices as those corresponding to systems of polynomials satisfying the Szegö-type two-term recurrence relations (2.32).

Theorem 5 (Szegö-type two-term recurrence relations $\Rightarrow$ (H, m)-semiseparable matrices) *Let $R = \{r_0(x), \ldots, r_{n-1}(x)\}$ be a system of polynomials satisfying the recurrence relations (2.32) with $\text{rank}(\alpha_k^T - \beta_k \gamma_k) = m$. Then the (H, m)-semiseparable matrix A defined by*

$$\begin{bmatrix} -\frac{\theta_1+\gamma_1\beta_0}{\delta_1} & -\frac{1}{\delta_2}\gamma_2(\alpha_1-\beta_1\gamma_1)\beta_0 & \cdots & -\frac{1}{\delta_n}\gamma_n(\alpha_{n-1}-\beta_{n-1}\gamma_{n-1})\cdots(\alpha_1-\beta_1\gamma_1)\beta_0 \\ \frac{1}{\delta_1} & -\frac{\theta_2+\gamma_2\beta_1}{\delta_2} & \ddots & -\frac{1}{\delta_n}\gamma_n(\alpha_{n-1}-\beta_{n-1}\gamma_{n-1})\cdots(\alpha_2-\beta_2\gamma_2)\beta_1 \\ 0 & \frac{1}{\delta_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\frac{\theta_n+\gamma_n\beta_{n-1}}{\delta_n} \\ 0 & \cdots & 0 & \frac{1}{\delta_n} \end{bmatrix} \tag{2.34}$$

with generators

$$d_k = -\frac{\theta_k + \gamma_k \beta_{k-1}}{\delta_k}, \quad k = 1, \ldots, n, \qquad p_{k+1} q_k = \frac{1}{\delta_k}, \quad k = 1, \ldots, n-1,$$

$$g_k = \beta_{k-1}^T, \quad k = 1, \ldots, n-1,$$

$$b_k^T = \alpha_{k-1} - \beta_{k-1} \gamma_{k-1}, \quad k = 2, \ldots, n-1,$$

$$\beta_0 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \qquad h_k = -\frac{1}{\delta_k} b_k \gamma_k^T, \quad k = 2, \ldots, n,$$

corresponds to the R via (2.6).

Proof Let us show that the polynomial system satisfying the Szegö-type two-term recurrence relations (2.32) also satisfies EGO-type two-term recurrence relations (2.15). By applying the given two-term recursion, we have

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k G_{k-1}(x) + \beta_k(\delta_k + \theta_k) r_{k-1}(x) \\ \gamma_k G_{k-1}(x) + (\delta_k + \theta_k) r_{k-1}(x) \end{bmatrix}. \tag{2.35}$$

Multiplying the second equation in (2.35) by β_k and subtracting from the first equation we obtain

$$G_k(x) - \beta_k r_k(x) = (\alpha_k - \beta_k \gamma_k) G_{k-1}(x). \tag{2.36}$$

Denoting in (2.36) G_{k-1} by F_k and shifting indices from k to $k-1$ we get the recurrence relation

$$F_k(x) = (\alpha_{k-1} - \beta_{k-1}\gamma_{k-1}) F_{k-1}(x) + \beta_{k-1} r_{k-1}(x). \tag{2.37}$$

In the same manner substituting (2.36) in the second equation of (2.35) and shifting indices one can be seen that

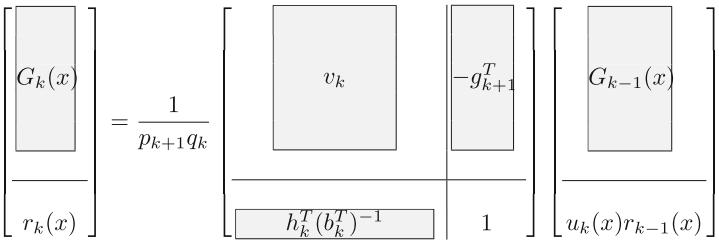
$$r_k(x) = \gamma_k(\alpha_{k-1} - \beta_{k-1}\gamma_{k-1}) F_{k-1}(x) + (\delta_k x + \theta_k + \gamma_k \beta_{k-1}) r_{k-1}(x). \tag{2.38}$$

Equations (2.37) and (2.38) together give necessary EGO-type two-term recurrence relations for the system of polynomials:

$$\begin{bmatrix} F_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_{k-1} - \beta_{k-1}\gamma_{k-1} & \beta_{k-1} \\ \gamma_k(\alpha_{k-1} - \beta_{k-1}\gamma_{k-1}) & \delta_k x + \theta_k + \gamma_k \beta_{k-1} \end{bmatrix} \begin{bmatrix} F_{k-1}(x) \\ r_{k-1}(x) \end{bmatrix}. \tag{2.39}$$

The result follows from Theorem 2 and (2.39).

Theorem 6 ((H, m)-semiseparable matrices ⇒ Szegö-type two-term recurrence relations) *Let A bea (H, m)-semiseparable matrix. Then for a set of generators $\{p_k, q_k, d_k, g_k, b_k, h_k\}$ of A such that each b_k is invertible, the polynomial system R corresponding to A satisfies* (2.32); *specifically*,



(2.40)

with $u_k(x) = x - d_k + g_k b_k^{-1} h_k, \quad v_k = p_{k+1} q_k b_{k+1}^T - g_{k+1}^T h_k^T (b_k^T)^{-1}$.

Proof According to the definition of (H, m)-semiseparable matrices the given polynomial system R must satisfy EGO-type two-term recurrence relations (2.15) with b_k invertible for all k. For the recurrence relations

$$\begin{bmatrix} F_k(x) \\ r_k(x) \end{bmatrix} = \frac{1}{p_{k+1}q_k} \begin{bmatrix} p_k q_k b_k^T & -q_k g_k^T \\ p_k h_k^T & x - d_k \end{bmatrix} \begin{bmatrix} F_{k-1}(x) \\ r_{k-1}(x) \end{bmatrix}, \tag{2.41}$$

let us denote $p_{k+1}F_k(x)$ in (2.41) as $G_k(x)$, and then we can rewrite these equations as

$$\begin{aligned} G_{k-1}(x) &= b_k^T G_{k-2}(x) - g_k^T r_{k-1}(x), \\ r_k(x) &= \frac{1}{p_{k+1}q_k}\left[h_k^T G_{k-2}(x) + (x - d_k) r_{k-1}(x)\right]. \end{aligned} \tag{2.42}$$

Using the invertibility of b_k we are able to derive the $G_{k-2}(x)$ from the first equation of (2.42) and inserting it in the second equation we obtain

$$r_k(x) = \frac{1}{p_{k+1}q_k}\left[h_k^T (b_k^T)^{-1} G_{k-1}(x) + (x - d_k + g_k b_k^{-1} h_k) r_{k-1}(x)\right]. \tag{2.43}$$

The second necessary recurrence relation can be obtained by substituting (2.43) in the first equation of (2.42) and shifting indices from $k-1$ to k.

$$\begin{aligned} G_k(x) = \frac{1}{p_{k+1}q_k}\Big[&(p_{k+1} q_k b_{k+1}^T - g_{k+1}^T h_k^T (b_k^T)^{-1}) G_{k-1}(x) \\ &- g_{k+1}^T (x - d_k + g_k b_k^{-1} h_k) r_{k-1}(x)\Big] \end{aligned} \tag{2.44}$$

This completes the proof.

This completes the justification of Theorem 4.

2.5 (H, m)-well-free Matrices and Recurrence Relations (2.45)

In this section, we begin by considering the l-term recurrence relations of the form

$$\begin{aligned} r_0(x) &= a_{0,0}, \quad r_k(x) = \sum_{i=1}^{k} (\delta_{ik}x + \varepsilon_{ik}) r_{i-1}(x), \quad k = 1, 2, \ldots, l-2, \\ r_k(x) &= \sum_{i=k-l+2}^{k} (\delta_{ik}x + \varepsilon_{ik}) r_{i-1}(x), \quad k = l-1, l, \ldots, n. \end{aligned} \tag{2.45}$$

As we shall see below, the matrices that correspond to (2.45) via (2.6) form a new subclass of (H, m)-quasiseparable matrices. As such, we then can also give a generator classification of the resulting class. This problem was addressed in [8] for the $l = 3$ case; that is, for (2.3),

$$r_0(x) = a_{0,0}, \quad r_1(x) = (\alpha_1 x - \delta_1) \cdot r_0(x), \tag{2.46}$$

$$r_k(x) = (\alpha_k x - \delta_k) \cdot r_{k-1}(x) - (\beta_k x + \gamma_k) \cdot r_{k-2}(x). \tag{2.47}$$

and was already an involved problem. To explain the results in the general case more clearly, we begin by recalling the results for the special case when $l = 3$.

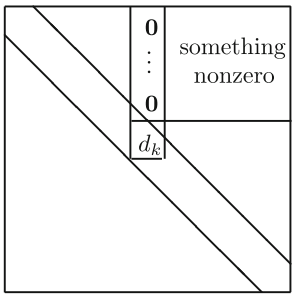
2.5.1 General Three-term Recurrence Relations (2.3) and $(H, 1)$-well-free Matrices

In [8], it was proved that polynomials that satisfy the general three-term recurrence relations (2.46) were related to a subclassf $(H, 1)$-quasiseparable matrices denoted $(H, 1)$-well-free matrices. A definition of this class is given next.

Definition 4 ($(H, 1)$-*well-free matrices*)

- An $n \times n$ matrix $A = (A_{i,j})$ is said to have a **well of size one** in column $1 < k < n$ if $A_{i,k} = 0$ for $1 \le i < k$ and there exists a pair (i, j) with $1 \le i < k$ and $k < j \le n$ such that $A_{i,j} \neq 0$.
- A $(H, 1)$-quasiseparable matrix is said to be $(H, 1)$**-well-free** if none of its columns $k = 2, \ldots, n-1$ contain wells of size one.

Verbally, a matrix has a well in column k if all entries above the main diagonal in the k-th column are zero, **except** if all entries in the upper-right block to the right of these zeros are also zeros, as shown in the following illustration.



The following theorem summarizes the results of [8] that will be generalized in this section.

Theorem 7 *Suppose A is a strongly upper Hessenberg $n \times n$ matrix. Then the following are equivalent.*

(i) *A is $(H, 1)$-well-free.*
(ii) *There exists a set of generators of Definition 2 corresponding to A such that $h_k \neq 0$ for $k = 2, \ldots, n$.*
(iii) *The system of polynomials related to A via (2.6) satisfies the general three-term recurrence relations (2.46).*

Having provided these results, the next goal is, given the l-term recurrence relations (2.45), to provide an analogous classification. A step in this direction can be taken using a formula given by Barnett in [4] that gives for such recurrence relations a formula for the entries of the related matrix. For the convenience of the reader, a proof of this lemma is given at the end of this section (no proof was given in [4]).

Lemma 3 *Let $R = \{r_0(x), \ldots, r_{n-1}(x)\}$ be a system of polynomials satisfying the recurrence relations (2.45). Then the strongly Hessenberg matrix*

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ \frac{1}{\delta_{11}} & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & \frac{1}{\delta_{22}} & a_{33} & \cdots & a_{3n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{1}{\delta_{n-1,n-1}} & a_{nn} \end{bmatrix} \tag{2.48}$$

with entries

$$\begin{aligned} a_{ij} &= -\frac{1}{\delta_{jj}} \left(\frac{\delta_{i-1,j}}{\delta_{i-1,i-1}} + \varepsilon_{ij} + \sum_{s=i}^{j-1} a_{is} \delta_{sj} \right) \\ \frac{\delta_{0j}}{\delta_{00}} &= 0, \quad \forall j; \quad \delta_{ij} = \varepsilon_{ij} = 0, \quad i < j - l + 2 \end{aligned} \tag{2.49}$$

corresponds to R via (2.6).

Remark 8 While Lemma 3 describes the entries of the matrix A corresponding to polynomials satisfying the l-term recurrence relations (2.45), the structure of A is not explicitly specified by (2.49). Indeed, as surveyed in this section, even in the simplest case of generalized three-term recurrence relations (2.3), the latter do not transparently lead to the characteristic quasiseparable and well-free properties of the associated matrices.

2.5.2 (H, m)-well-free Matrices

It was recalled in Sect. 2.5.1 that in the simplest case of three-term recurrence relations the corresponding matrix was $(H, 1)$-quasiseparable, and moreover, $(H, 1)$-well-free. So, one might expect that in the case of l-term recurrence relations (2.45), the associated matrix might turn out to be $(H, l-2)$-quasiseparable, but how does one generalize the concept of $(H, 1)$-well-free? The answer to this is given in the next definition.

Definition 5 (*(H, m)-well-free matrices*)

- Let A be an $n \times n$ matrix, and fix constants $k, m \in [1, n-1]$. Define the matrices

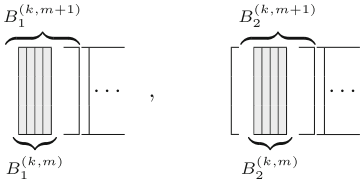
$$B_j^{(k,m)} = A(1:k, j+k:j+k+(m-1)), \quad j = 1, \ldots, n-k-m.$$

  Then if for some j,

$$\operatorname{rank}(B_j^{(k,m+1)}) > \operatorname{rank}(B_j^{(k,m)}),$$

  the matrix A is said to have a **well of size** m **in partition** k.
- A (H, m)-quasiseparable matrix is said to be (H, m)**-well-free** if it contains no wells of size m.

One can understand the matrices $B_j^{(k,m)}$ of the previous definition as, for constant k and m and as j increases, a sliding window consisting of m consecutive columns. Essentially, the definition states that as this window is slid through the partition A_{12} of Definition 1, if the ranks of the submatrices increase at any point by adding the next column, this constitutes a well. So a (H, m)-well-free matrix is such that each column of all partitions A_{12} is the linear combination of the m previous columns of A_{12}.



Notice that Definition 5 reduces to Definition 4 in the case when $m = 1$. Indeed, if $m = 1$, then the sliding windows are single columns, and an increase in rank is the result of adding a nonzero column to a single column of all zeros. This is shown next in (2.50).

$$\begin{array}{c|c|c|c|} & B_{j-1} & B_j & B_{j+1} \\ \hline & * & 0 & * \\ & * & 0 & * \\ & \vdots & \vdots & \vdots \\ & * & 0 & * \\ \hline \end{array} \tag{2.50}$$

In order for a matrix to be $(H, 1)$-quasiseparable, any column of zeros in A_{12} must be the first column of A_{12}; that is, in (2.50) $j = 1$. Thus a well of size one is exactly a column of zeros above the diagonal, and some nonzero entry to the right of that column, exactly as in Definition 4.

With the class of (H, m)-well-free matrices defined, we next present a theorem containing the classifications to be proved in this section.

Theorem 8 *Suppose A is a strongly upper Hessenberg $n \times n$ matrix. Then the following are equivalent.*

(i) *A is (H, m)-well-free.*
(ii) *There exists a set of generators of Definition 2 corresponding to A such that b_k are companion matrices for $k = 2, \ldots, n-1$, and $h_k = e_1$ for $k = 2, \ldots, n$, where e_1 is the first column of the identity matrix of appropriate size.*
(iii) *The system of polynomials related to A via (2.6) satisfies the general three-term recurrence relations (2.45).*

This theorem is an immediate corollary of Theorems 9, 10 and 11.

2.5.3 (H, m)-well-free Matrices: Generator Classification

Theorem 9 *An (H, m)-quasiseparable matrix is (H, m)-well-free if and only if there exists a choice of generators $\{p_k, q_k, d_k, g_k, b_k, h_k\}$ of the matrix that are of the form*

$$b_k = \begin{bmatrix} 0 & 0 & \cdots & 0 & \xi_{k,1} \\ 1 & 0 & \ddots & \vdots & \vdots \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \xi_{k,m-1} \\ 0 & \cdots & 0 & 1 & \xi_{k,m} \end{bmatrix}, \quad k = 2, \ldots, n-1, \quad h_k = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}, \quad k = 2, \ldots, n. \tag{2.51}$$

Proof Let $A = (a_{ij})$ be an (H, m)-well-free matrix. Then due to the low rank property of off-diagonal blocks, its entries satisfy

$$a_{ij} = \sum_{s=j-m}^{j-1} a_{is}\alpha_{sj}, \quad \text{if} \quad i < j - m. \tag{2.52}$$

It is easy to see that an (H, m)-well-free matrix B with

$$d_k = a_{kk}, \quad k = 1, \ldots, n, \quad p_{k+1}q_k = a_{k+1,k}, \quad k = 1, \ldots, n-1,$$
$$g_k = \begin{bmatrix} a_{k,k+1} \cdots a_{k,k+m} \end{bmatrix}, \quad k = 1, \ldots, n-1, \quad h_k = [1\, 0 \cdots 0]^T, \quad k = 2, \ldots, n,$$

$$b_k = \begin{bmatrix} 0 & 0 & \cdots & 0 & \alpha_{k-m+1,k+1} \\ 1 & 0 & \ddots & \vdots & \vdots \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \alpha_{k-1,k+1} \\ 0 & \cdots & 0 & 1 & \alpha_{k,k+1} \end{bmatrix}, \quad k = 2, \ldots, n-1. \tag{2.53}$$

coincides with A.

Conversely, suppose A is an (H, m)-quasiseparable matrix whose generators satisfy (2.51). Applying (2.14) from Definition 2 it follows that

$$a_{ij} = g_i b_{i,j}^{\times} h_j = \begin{cases} v_{i,j-i} & i = 1, \ldots, n \quad j = i, \ldots, i+m, \\ \sum_{s=j-m}^{j-1} a_{is}\xi_{j-m,s-j+m+1} & i = 1, \ldots, n \quad j = i+m+1, \ldots, n. \end{cases} \tag{2.54}$$

This is equivalent to a summation of the form (2.52), demonstrating the low-rank property, and hence the matrix A is (H, m)-well-free according to Definition 5.

This result generalizes the generator classification of $(H, 1)$-well-free matrices as given in [8], stated as a part of Theorem 7.

2.5.4 (H, m)-well-free Matrices. Recurrence Relation Classification

In this section, we will prove that it is exactly the class of (H, m)-well-free matrices that correspond to systems of polynomials satisfying l-term recurrence relations of the form (2.45)

Theorem 10 (l-term recurrence relations $\Rightarrow (H, l-2)$-well-free matrices) *Let $A = (a_{ij})_{i,j=1}^{n}$ be a matrix corresponding to a system of polynomials $R = \{r_0(x), \ldots, r_{n-1}(x)\}$ satisfying (2.45). Then A is (H, m)-well-free.*

Proof The proof is presented by demonstrating that A has a set of generators of the form (2.51), and hence is (H, m)-well-free. In particular, we show that

$$d_k = a_{kk}, k = 1, \ldots, n, \quad p_{k+1}q_k = \frac{1}{\delta_{kk}}, \quad k = 1, \ldots, n-1,$$

$$g_k = \left[a_{k,k+1} \cdots a_{k,k+l-2}\right], \quad k = 1, \ldots, n-1,$$

$$h_k = [\underbrace{1\,0 \cdots 0}_{l-2}\,0]^T, \quad k = 2, \ldots, n,$$

$$b_k = \begin{bmatrix} 0 & 0 & \cdots & 0 & -\frac{\delta_{k,k+l-2}}{\delta_{k+l-2,k+l-2}} \\ 1 & 0 & \ddots & \vdots & \vdots \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & -\frac{\delta_{k+l-4,k+l-2}}{\delta_{k+l-2,k+l-2}} \\ 0 & \cdots & 0 & 1 & -\frac{\delta_{k+l-3,k+l-2}}{\delta_{k+l-2,k+l-2}} \end{bmatrix}, \quad k = 2, \ldots, n-1, \tag{2.55}$$

with $\frac{\delta_{ij}}{\delta_{jj}} = 0$ if $i > n - l + 2$ forms a set of generators of A. We show that with this choice, the entries of the matrix A coincide with those of (2.49). From Definition 2, the choice of d_k as the diagonal of A and choice of $p_{k+1}q_k$ as the subdiagonal entries of (2.48) produces the desired result in these locations. We next show that the generators g_k, b_k and h_k define the upper triangular part of the matrix A correctly.

Consider first the product $g_i b_{i+1} b_{i+2} \cdots b_{i+t}$, and note that

$$g_i b_{i+1} b_{i+2} \cdots b_{i+t} = \left[a_{i,i+t+1} \cdots a_{i,i+t+l-2}\right]. \tag{2.56}$$

Indeed, for $t = 0$, (2.56) becomes

$$g_i = \left[a_{i,i+1} \cdots a_{i,i+l-2}\right],$$

which coincides with the choice in (2.55) for each i, and hence the relation is true for $t = 0$. Suppose next that the relation is true for some t. Then using the lower shift structure of the choice of each b_k of (2.55) and the formula (2.49), we have

$$\begin{aligned} g_i b_{i+1} b_{i+2} \cdots b_{i+t+1} &= \left[a_{i,i+t+1} \cdots a_{i,i+t+l-2}\right] b_{i+t+1} \\ &= \left[a_{i,i+t+2} \cdots a_{i,i+t+l-2} \sum_{p=i+t+1}^{i+t+l-2} \frac{-a_{ip}\delta_{p,i+t+l-1}}{\delta_{i+t+l-1,i+t+l-1}}\right] \\ &= \left[a_{i,i+t+2} \cdots a_{i,i+t+l-1}\right]. \end{aligned} \tag{2.57}$$

And therefore

$$g_i b_{ij}^{\times} h_j = \left[a_{ij} \cdots a_{i,j+s-1}\right] h_j = a_{ij}, \quad j > i$$

so (2.55) are in fact generators of the matrix A as desired.

Theorem 11 ((H, m)-well-free matrices $\Rightarrow$ $(m+2)$-term recurrence relations) *Let A be an (H, m)-well-free matrix. Then the polynomials system related to A via (2.6) satisfies the l-term recurrence relations (2.45).*

Proof By Theorem 9, there exists a choice of generators of A of the form

$$d_k = v_{k,0}, k = 1, \ldots, n, \quad p_{k+1}q_k = \mu_k, \quad k = 1, \ldots, n-1,$$

$$g_k = \begin{bmatrix} v_{k,1} \cdots v_{k,m} \end{bmatrix}, \quad k = 1, \ldots, n-1,$$

$$h_k = [\underbrace{1\,0 \cdots 0}_{m}\,0]^T, \quad k = 2, \ldots, n,$$

$$b_k = \begin{bmatrix} 0 & 0 & \cdots & 0 & \xi_{k,1} \\ 1 & 0 & \ddots & \vdots & \vdots \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \xi_{k,m-1} \\ 0 & \cdots & 0 & 1 & \xi_{k,m} \end{bmatrix}, \quad k = 2, \ldots, n-1. \tag{2.58}$$

We present a procedure to compute from these values the coefficients of (2.45).

1. Take

$$\delta_{ij} = \begin{cases} \frac{1}{\mu_j} & i = j, \\ -\frac{\xi_{j-m,i-j+m+1}}{\mu_{jj}} & j = m+2, \ldots, n \quad i = j-m, \ldots, \ j-1. \end{cases} \tag{2.59}$$

2. Calculate ε_{ij} and δ_{ij} for $j = 2, \ldots, m+1, \quad i = 1, \ldots, j-1$ as any solution of the following system of equations:

$$\begin{cases} v_{i,j-i} = -\mu_j \left(\varepsilon_{ij} + \sum_{s=1}^{j-1} v_{i,s-i}\delta_{sj} \right) & i = 1, \\ & j = 2, \ldots, m+1, \\ v_{i,j-i} = -\mu_j \left(\delta_{i-1,j}\mu_{i-1} + \varepsilon_{ij} + \sum_{s=1}^{j-1} v_{i,s-i}\delta_{sj} \right) & i = 2, \ldots, m, \\ & j = i+1, \ldots, m+1. \end{cases} \tag{2.60}$$

3. Find the remaining ε_{ij}-coefficients using

$$\varepsilon_{ij} = \begin{cases} -\frac{v_{1,0}}{\mu_1} & i = j = 1, \\ -\frac{v_{j,0}}{\mu_j} - \delta_{j-1,j}\mu_{j-1} & i = j > 1, \\ -\frac{v_{i,j-i}}{\mu_j} - \delta_{i-1,j}\mu_{i-1} - \sum_{s=i}^{j-1} v_{i,s-i}\delta_{sj} & j = m+2, \ldots, n \\ & i = j-m, \ldots, j-1. \end{cases} \tag{2.61}$$

The proof immediately follows by comparing (2.55), (2.58) and using (2.49). Note that the coefficients of the l-term recurrence relations depend on the solution of the system of equations (2.60), which consists of

$$\sum_{i=1}^{m} i = \frac{m(m+1)}{2}$$

equations and defines $m(m+1)$ variables. So for the generators (2.58) of an (H, m)-well-free matrix there is a freedom in choosing coefficients of the recurrence relations (2.45) for the corresponding polynomials.

This completes the justification of Theorem 8 stated above. In the $m = 1$ case, this coincides with the result given in [8], stated as Theorem 7.

2.5.5 Proof of Lemma 3

In this section we present a proof of Lemma 3, stated without proof by Barnett in [4].

Proof (Proof of Lemma 3) The results of [24] allow us to observe the bijection between systems of polynomials and dilated strongly Hessenberg matrices. Indeed, given a polynomial system $R = \{r_0(x), \ldots, r_{n-1}(x)\}$, there exist unique n-term recurrence relations of the form

$$\begin{aligned} x \cdot r_{j-1}(x) &= a_{j+1,j} \cdot r_j(x) + a_{j,j} \cdot r_{j-1}(x) + \cdots + a_{1,j} \cdot r_0(x), \quad a_{j+1,j} \neq 0, \\ & j = 1, \ldots, n-1. \end{aligned} \tag{2.62}$$

and $a_{1,j}, \ldots, a_{j+1,j}$ are coefficients of the j-th column of the correspondent strongly Hessenberg matrix A.

Using $\delta_{ij} = \varepsilon_{ij} = 0, i < j - l + 2$, we can assume that the given system of polynomials $R = \{r_0(x), \ldots, r_{n-1}(x)\}$ satisfies full recurrence relations:

$$r_j(x) = \sum_{i=1}^{j} (\delta_{ij} x + \varepsilon_{ij}) r_{i-1}(x), \quad j = 1, \ldots, n-1 \tag{2.63}$$

The proof of (2.49) is given by induction on j. For any i, if $j = 1$, it is true that $a_{11} = -\frac{\varepsilon_{11}}{\delta_{11}}$. Next, assuming that (2.49) is true for all $j = 1, \ldots, k-1$. Taking $j = k$ in (2.63) we can write that

$$x r_{k-1}(x) = \frac{1}{\delta_{kk}} r_k(x) - \frac{\varepsilon_{kk}}{\delta_{kk}} r_{k-1}(x) - \frac{1}{\delta_{kk}} \sum_{i=1}^{k-1} (\delta_{ik} x + \varepsilon_{ik}) r_{i-1}(x). \tag{2.64}$$

From the induction hypothesis and Eq. (2.62) we can substitute the expression for $x r_{i-1}$ into (2.64) to obtain

$$x r_{k-1}(x) = \frac{1}{\delta_{kk}} r_k(x) - \frac{\varepsilon_{kk}}{\delta_{kk}} r_{k-1}(x) - \frac{1}{\delta_{kk}} \sum_{i=1}^{k-1} \left[\delta_{ik} \sum_{s=1}^{i+1} a_{si} r_{s-1}(x) + \varepsilon_{ik} r_{i-1}(x) \right] \tag{2.65}$$

After grouping coefficients in (2.65) we obtain

$$xr_{k-1}(x) = \frac{1}{\delta_{kk}} r_k(x) - \frac{1}{\delta_{kk}} \sum_{i=1}^{k} \left[\frac{\delta_{i-1,k}}{\delta_{i-1,i-1}} + \varepsilon_{ik} + \sum_{s=i}^{k-1} a_{is}\delta_{sk} \right] r_{i-1}(x). \tag{2.66}$$

Comparing (2.62) and (2.66) we get (2.49) by induction.

2.6 Relationship Between These Subclasses of (H, m)-quasiseparable Matrices

Thus far it has been proved that the classes of (H, m)-semiseparable and (H, m)-well-free matrices are subclasses of the class of (H, m)-quasiseparable matrices. The only unanswered questions to understand the interplay between these classes is whether these two subclasses have common elements or not, and whether either class properly contains the other or not.

It was demonstrated in [8] that there is indeed a nontrivial intersection of the classes of $(H, 1)$-semiseparable and $(H, 1)$-well-free matrices, and so there is at least some intersection of the (weakly) (H, m) versions of these classes. In the next example it will be shown that such a nontrivial intersection exists in the rank m case; that is, there exist matrices that are both (H, m)-semiseparable and (H, m)-well-free.

Example 4 Let A be an (H, m)-quasiseparable matrix whose generators satisfy

$$b_k = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \ddots & \vdots & 1 \\ 0 & 1 & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 & 1 \\ 0 & \cdots & 0 & 1 & 1 \end{bmatrix} \in \mathbb{C}^{m\times m}, \quad h_k = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{C}^m.$$

Regardless of the other choices of generators, one can see that these generators satisfy both Lemma 2 and Theorem 9, and hence the matrix A is both (H, m)-well-free and (H, m)-semiseparable.

The next example demonstrates that an (H, m)-semiseparable matrix need not be (H, m)-well-free.

Example 5 Consider the (H, m)-quasiseparable matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Because of the shaded block of zeros, it can be seen that the matrix is not $(H, 2)$-well-free. However, one can observe that $\text{rank}(triu(A, 1)) = 2$, and hence A is $(H, 2)$-semiseparable. Thus the class of (H, m)-semiseparable matrices does not contain the class of (H, m)-well-free matrices.

To see that an (H, m)-well-free matrix need not be (H, m)-semiseparable, consider the banded matrix (2.8) from the introduction. It is easily verified to not be (H, m)-semiseparable (for $m < n - l$), however it is $(H, l - 2)$-well-free.

This completes the discussion on the interplay of the subclasses of (H, m)-quasiseparable matrices, as it has been shown that there is an intersection, but neither subclass contains the other. Thus the proof of Fig. 2.3 is completed.

2.7 Conclusion

To conclude, appropriate generalizations of real orthogonal polynomials and Szegö polynomials, as well as several subclasses of $(H, 1)$-quasiseparable polynomials, were used to classify the larger class of (H, m)-quasiseparable matrices for arbitrary m. Classifications were given in terms of recurrence relations satisfied by related polynomial systems, and in terms of special restrictions on the quasiseparable generators.

References

1. Ammar G, Calvetti D, Reichel L (1993) Computing the poles of autoregressive models from the reflection coefficients. In: Proceedings of the Thirty-First Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, October, pp 255–264
2. Ammar G, He C (1995) On an inverse eigenvalue problem for unitary Hessenberg matrices. Linear Algebra Appl 218:263–271
3. Bakonyi M, Constantinescu T (1992) Schur's algorithm and several applications. Pitman Research Notes in Mathematics Series, vol 61, Longman Scientific and Technical, Harlow
4. Barnett S (1981) Congenial matrices. Linear Algebra Appl 41:277–298
5. Bella T, Eidelman Y, Gohberg I, Koltracht I, Olshevsky V (2007) A Björck-Pereyra-type algorithm for Szegö-Vandermonde matrices based on properties of unitary Hessenberg matrices. Linear Algebra and its Applications (to appear)
6. Bella T, Eidelman Y, Gohberg I, Koltracht I, Olshevsky V. A fast Björck-Pereyra-like algorithm for solving Hessenberg-quasiseparable-Vandermonde systems (submitted)
7. Bella T, Eidelman Y, Gohberg I, Olshevsky V, Tyrtyshnikov E. Fast inversion of Hessenberg-quasiseparable Vandermonde matrices (submitted)
8. Bella T, Eidelman Y, Gohberg I, Olshevsky V. Classifications of three-term and two-term recurrence relations and digital filter structures via subclasses of quasiseparable matrices. SIAM J Matrix Anal (SIMAX) (submitted)
9. Bunse-Gerstner A, He CY (1995) On a Sturm sequence of polynomials for unitary Hessenberg matrices. (English summary) SIAM J Matrix Anal Appl 16(4):1043–1055
10. Bruckstein A, Kailath T (1986) Some matrix factorization identities for discrete inverse scattering. Linear Algebra Appl 74:157–172
11. Bruckstein A, Kailath T (1987) Inverse scattering for discrete transmission line models. SIAM Rev 29:359–389

12. Bruckstein A, Kailath T (1987) An inverse scattering framework for several problems in signal processing. IEEE ASSP Mag 4(1):6–20
13. Eidelman Y, Gohberg I (1999) On a new class of structured matrices. Integr Equ Oper Theory 34:293–324
14. Eidelman Y, Gohberg I (1999) Linear complexity inversion algorithms for a class of structured matrices. Integr Equ Oper Theory 35:28–52
15. Eidelman Y, Gohberg I (2002) A modification of the Dewilde-van der Veen method for inversion of finitestructured matrices. Linear Algebra Appl 343–344:419–450
16. Eidelman Y, Gohberg I (2005) On generators of quasiseparable finite block matrices. Calcolo 42:187–214
17. Eidelman Y, Gohberg I, Olshevsky V (2005) Eigenstructure of Order-One-Quasiseparable Matrices. Three-term and two-term Recurrence Relations. Linear Algebra Appl 405:1–40
18. Geronimus LY (1954) Polynomials orthogonal on a circle and their applications. Amer Math Transl 3:1–78 (Russian original 1948)
19. Gragg WB (1982) Positive definite Toeplitz matrices, the Arnoldi process for isometric operators, and Gaussian quadrature on the unit circle (in Russian). In : Nikolaev ES (ed) Numerical methods in Linear Algebra. pp. 16–32, Moskow University Press, 1982; English translation: (1993) J Comput Appl Math 46:183–198
20. Grenader U, Szegö G (1958) Toeplitz forms and Applications. University of California Press, Berkeley
21. Kailath T, Porat B (1983) State-space generators for orthogonal polynomials. Prediction theory and harmonic analysis, pp 131–163, North-Holland, Amsterdam-New York
22. Lev-Ari H, Kailath T (1984) Lattice filter parameterization and modeling of nonstationary processes. IEEE Trans Inf Theory 30:2–16
23. Lev-Ari H, Kailath T (1986) Triangular factorization of structured Hermitian matrices. In: Gohberg I (ed) Operator Theory: Advances and Applications, vol. 18. Birkhäuser, Boston, pp 301–324
24. Maroulas J, Barnett S (1979) Polynomials with respect to a general basis. I. Theory. J Math Anal Appl 72:177–194
25. Olshevsky V (1998) Eigenvector computation for almost unitary Hessenberg matrices and inversion of Szego-Vandermonde matrices via discrete transmission lines. Linear Algebra Appl 285:37–67
26. Olshevsky V (2001) Associated polynomials, unitary Hessenberg matrices and fast generalized Parker-Traub and Bjorck-Pereyra algorithms for Szego-Vandermonde matrices. In: Bini D, Tyrtyshnikov E, Yalamov P (eds) Structured Matrices: Recent Developments in Theory and Computation. NOVA Science Publishers, USA, pp 67–78
27. Regalia PA (1995) Adaptive IIR filtering in signal processing and control. Marcel Dekker, New York
28. Schur I (1917) Uber potenzreihen, die in Innern des Einheitskrises Beschrnkt Sind. J Reine Angew Math 147:205–232; English translation: Schur I (1986) Methods in Operator Theory and Signal Processing, Gohberg I (ed) Birkhuser, pp 31–89
29. Simon B (2005) Orthogonal polynomials on the unit circle. Part 2. Spectral theory. American Mathematical Society Colloquium Publications, 54, Parts 1 & 2. American Mathematical Society, Providence, RI
30. Stoer J, Bulirsch R (1992) Introduction to numerical analysis. Springer-Verlag, New York, pp 277–301
31. Teplyaev AV (1992) The pure point spectrum of random orthogonal polynomials on the circle. Soviet Math Dokl 44:407–411

Chapter 3
Partial Stabilization of Descriptor Systems Using Spectral Projectors

Peter Benner

Abstract We consider the stabilization problem for large-scale linear descriptor systems in continuous- and discrete-time. We suggest a partial stabilization algorithm which preserves stable poles of the system while the unstable ones are moved to the left half plane using state feedback. Our algorithm involves the matrix pencil disk function method to separate the finite from the infinite generalized eigenvalues and the stable from the unstable eigenvalues. In order to stabilize the unstable poles, either the generalized Bass algorithm or an algebraic Bernoulli equation can be used. Some numerical examples demonstrate the behavior of our algorithm.

3.1 Introduction

We consider linear descriptor systems

$$E(\mathcal{D}x(t)) = Ax(t) + Bu(t), \quad t > 0, \quad x(0) = x_0, \tag{3.1}$$

where $\mathcal{D}x(t) = \frac{d}{dt}x(t), t \in \mathbb{R}$, for continuous-time systems and $\mathcal{D}x(t) = x(t+1)$, $t \in \mathbb{N}$, for discrete-time systems. Here, $A, E \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$. We assume the matrix pencil $A - \lambda E$ to be regular, but make no assumption on its index. For continuous-time systems, (3.1) is a first-order differential-algebraic equation (DAE) if E is singular and an ordinary differential equation (ODE) if E is nonsingular. We are particularly interested in the DAE case. In this case, $x(t) \in \mathbb{R}^n$ is called a

P. Benner (✉)
Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106, Magdeburg, Germany
e-mail: benner@mpi-magdeburg.mpg.de

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_3,

descriptor vector which is in general not a state vector as some components may be chosen freely under certain conditions, see [30, Chap. 4]. The components of the vector $u(t) \in \mathbb{R}^m$ are considered to be forcing functions or controls. For further properties of DAEs and descriptor systems see, e.g., [16, 19, 30] and references therein.

Throughout this article we assume that the generalized eigenvalues of $A - \lambda E$ (the *poles* of the system corresponding to (3.1)) are given by

$$\Lambda(A, E) = \Lambda_1 \cup \Lambda_2 \cup \{\infty\}$$

with $\Lambda_j \subset \Gamma_j$, $j = 1, 2$. Here, $\Gamma_1 = \mathbb{C}^-, \Gamma_2 = \mathbb{C}^+$ for continuous-time systems (with $\mathbb{C}^\pm$ denoting the open left and right half planes) and $\Gamma_1 = \{|z| < 1\}, \Gamma_2 = \{|z| > 1\}$ (the interior/exterior of the unit disk) for discrete-time systems. The case that eigenvalues are on the boundary of the stability region Γ_1 can be treated as well, see Remark 2, but the dichotomy assumption with respect to $\partial\Gamma$ simplifies the presentation for now.

Descriptor systems arise in various applications including circuit simulation, multibody dynamics, (semi-)discretization of the Stokes and Oseen equations (linearizations of the instationary Navier–Stokes equations) or the Euler equations, and in various other areas of applied mathematics and computational engineering, see, e.g., [30, 33] and references therein.

The *stabilization problem* for (3.1) can be formulated as follows: choose $u \in L_2(0, \infty; \mathbb{R}^m)$ such that the dynamical system (3.1) is asymptotically stable, i.e., solution trajectories satisfy $\lim_{t\to\infty} x(t) = 0$. Large-scale applications include active vibration damping for large flexible space structures, like, for example, the International Space Station [18], or initializing Newton's method for large-scale algebraic Riccati equations (AREs) [37]. But also many other procedures for controller and observer design make use of stabilization procedures [43].

For nonsingular E, it is well known (e.g., [20, 21]) that stabilization can be achieved by *state feedback* $u = Fx$, where $F \in \mathbb{R}^{m\times n}$ is chosen such that the *closed-loop system* $E(\mathcal{D}x(t)) = (A - BF)x$ is asymptotically stable, i.e., $\Lambda(A - BF, E) \subset \Gamma_1$ iff the matrix pair $(E^{-1}A, E^{-1}B)$ is stabilizable, i.e., $\operatorname{rank}([A - \lambda E, B]) = n$ for all $\lambda \in \mathbb{C} \setminus \Gamma_1$. (In the following, we will call $\Lambda(A, E)$ the *open-loop eigenvalues* and $\Lambda(A - BF, E)$ the *closed-loop eigenvalues*.)

For singular E, the situation is slightly more complicated. Varga [43] distinguishes *S*- and *R*-stabilization problems. Both require the computation of a state feedback matrix F such that the closed-loop matrix pencil $A - BF - \lambda E$ is regular. *S*-stabilization asks for F such that $\Lambda(A - BF, E)$ contains exactly $r = \operatorname{rank}(E)$ stable poles while for *R*-stabilization, all finite poles are requested to be stable. Both problems have a solution under suitable conditions (see [43] and references therein). Here, we will treat the *R*-stabilization problem only for which the assumption of stabilizability of the matrix triple (E, A, B), i.e., $\operatorname{rank}([A - \lambda E, B]) = n$ for all finite $\lambda \in \mathbb{C} \setminus \Gamma_1$ guarantees solvability [19]. Our procedure can be useful when solving the *S*-stabilization problem as well: the *R*-stabilization is needed

after a preprocessing step in the procedure for S-stabilization suggested in [43]. Also note that for the S-stabilization problem to be solvable, one needs to assume strong stabilizability of the descriptor system (3.1) (see [43]), which is rarely encountered in practice. E.g., none of the examples considered in this paper has this property.

In the standard ODE case (E nonsingular), a stabilizing feedback matrix F can be computed in several ways. One popular approach is to use pole assignment (see, e.g., [20, Sect. 10.4]) which allows to pre-assign the spectrum of the closed-loop system. There are several difficulties associated with this approach [25]. Also, for fairly large-scale problems, the usual algorithms are quite involved [20, Chap. 11]. In many applications, it is not necessary to fix the poles of the closed-loop system, in particular if the stabilization is only used to initialize a control algorithm or Newton's method for AREs. For this situation, a standard approach is based on the solution of a particular Lyapunov equation (see Proposition 2 and Eq. (3.3)). Solving this Lyapunov equation, the stabilizing feedback for standard state-space systems (i.e., $E = I_n$) is $F := B^T X^\dagger$ for continuous-time systems and $F := B^T(EXE^T + BB^T)^\dagger A$ for discrete-time systems, where X is the solution of the (continuous or discrete) Lyapunov equation and $M^\dagger$ is the pseudo-inverse of M, see, e.g., [37]. This approach is called the *Bass algorithm* [2, 3]. For a detailed discussion of properties and (dis-)advantages of the described stabilization procedure see [20, Sect. 10.2] and [37]. Both approaches, i.e., pole assignment and the Bass algorithm are generalized in [43] in order to solve the R-stabilization problem. Here, we will assume that achieving closed-loop stability is sufficient and poles need not be at specific locations, and therefore we will not discuss pole placement any further.

For large-scale systems, often, the number of unstable poles is small (e.g., 5%). Hence it is often more efficient and reliable to first separate the stable and unstable poles and then to apply the stabilization procedure only to the unstable poles. For standard systems, such a method is described in [26] while for generalized state-space systems with invertible E, procedures from [43] and [8] can be employed. The approach used in [8] is based on the disk function method and only computes a block-triangularization of the matrix pencil $A - \lambda E$ which has the advantage of avoiding the unnecessary and sometimes ill-conditioned separation of all eigenvalues required in the QZ algorithm for computing the generalized Schur form $A - \lambda E$ which is employed in [43]. The main contribution of this paper is thus to show how the disk function method can be used in order to generalize the method from [8] to descriptor systems with singular E.

In the next section, we will give some necessary results about descriptor systems and stabilization of generalized state-space systems with nonsingular matrix E. Section 3.3 then provides a review of spectral projection methods and in particular of the disk and sign function methods. A partial stabilization algorithm based on the disk function method is then proposed in Sect. 3.4. In Sect. 3.5, we give some numerical examples demonstrating the effectiveness of the suggested approach. We end with some conclusions and an outlook.

3.2 Theoretical Background

As we will only discuss the R-stabilization problem, we will need the following condition.

Definition 1 Let (E, A, B) be a matrix triple as in (3.1). Then the descriptor system (3.1) is *stabilizable if*

$$rank([A - \lambda E, B]) = n \quad \text{for all} \quad \lambda \in \mathbb{C} \setminus \Gamma_1.$$

We say that the system is c-stabilizable if $\Gamma_1 = \mathbb{C}^-$ and d-stabilizable if $\Gamma_1 = \{z \in \mathbb{C} \mid |z| < 1\}$.

The following result from [19] guarantees solvability of the R-stabilization problem.

Proposition 1 *Let the descriptor system* (3.1) *be stabilizable, then there exists a feedback matrix* $F \in \mathbb{R}^{m\times n}$ *such that*

$$\Lambda(A - BF, E) \subset \Gamma_1 \cup \{\infty\}.$$

In the following, we will speak of (partial) stabilization and always mean R-stabilization.

If E is nonsingular, any method for stabilizing standard state-space systems can be generalized to this situation. Here, we will make use of generalizations of the Bass algorithm [2, 3] and an approach based on the (generalized) algebraic Bernoulli equation (ABE)

$$A^T XE + E^T XA - E^T XBB^T XE = 0. \tag{3.2}$$

The first approach is based on the following result which can be found, e.g., in [8, 43].

Proposition 2 *Let* (E, A, B) *as in* (3.1) *be stabilizable with E nonsingular. If*

$$F := B^T E^{-T} X_c^{\dagger} \quad \text{or} \quad F := B^T (EX_d E^T + BB^T)^{\dagger} A \tag{3.3}$$

for continuous- or discrete-time systems, respectively, where X_c *and* X_d *are the unique solutions of the generalized (continuous or discrete) Lyapunov equations*

$$(A + \beta_c E)XE^T + EX(A + \beta_c E)^T = 2BB^T \quad \text{or} \quad AXA^T - \beta_d^2 EXE^T = 2BB^T, \tag{3.4}$$

respectively, for $\beta_c > \max_{\lambda\in\Lambda(A,E)}\{-\mathrm{Re}(\lambda)\}, 0 < \beta_d < \min_{\lambda\in\Lambda(A,E)\setminus\{0\}}\{|\lambda|\}$, *then* $A - BF - \lambda E$ *is stable.*

An often used, but conservative upper bound for the parameter β_c in the continuous Lyapunov equations above is $\|E^{-1}A\|$ for any matrix norm. As we will apply Proposition 2 in our partial stabilization procedure only to a matrix pencil that is completely unstable, i.e., $\Lambda(A, E) \subset \Gamma_2$, we can set $\beta_c = 0$ and $\beta_d = 1$. Usually, it turns out to be more effective to set $\beta_c > 0$ as this yields a better stability margin of the closed-loop poles. Note that β_c (or β_d in the discrete-time case) serves as a spectral shift so that the stabilized poles are to the left of $-\beta_c$ (or inside a circle with radius β_d).

Stabilization using the ABE (3.2) can be used for continuous-time systems and is based on the following result [6].

Proposition 3 *If* (E, A, B) *is as in Proposition 2 and* $\Lambda(A, E) \cap \jmath\mathbb{R} = \emptyset$, *then the ABE* (3.2) has a unique c-stabilizing *positive semidefinite solution* X_+, *i.e.*, $\Lambda(A - BB^T X_+ E, E) \subset \mathbb{C}^-$.

Moreover, $\operatorname{rank}(X^+) = \mu$, *where* μ *is the number of eigenvalues of* $A - \lambda E$ *in* $\mathbb{C}^+$ *and*

$$\Lambda(A - BB^T X_+ E, E) = (\Lambda(A, E) \cap \mathbb{C}^-) \cup \big(-(\Lambda(A, E) \cap \mathbb{C}^+)\big).$$

Another possibility for stabilization of standard discrete-time systems (*E invertible in* (3.1)) is discussed in [22]. An extension of this method to the case of singular E would allow the stabilization of large-scale, sparse discrete-time descriptor systems. This is under current investigation.

3.3 Spectral Projection Methods

In this section we provide the necessary background on spectral projectors and methods to compute them. These will be the major computational steps required in the partial stabilization method described in Sect. 3.4.

3.3.1 Spectral Projectors

First, we give some fundamental definitions and properties of projection matrices.

Definition 2 A matrix $P \in \mathbb{R}^{n\times n}$ is a *projector* (*onto a subspace* $\mathcal{S} \subset \mathbb{R}^n$) if $\operatorname{range}(P) = \mathcal{S}$ and $P^2 = P$.

Definition 3 Let $Z, Y \in \mathbb{R}^{n\times n}$ be a regular matrix pencil with $\Lambda(Z, Y) = \Lambda_1 \cup \Lambda_2$, $\Lambda_1 \cap \Lambda_2 = \emptyset$, and let $\mathcal{S}_1$ be the (right) deflating subspace of the matrix pencil $Z - \lambda Y$ corresponding to Λ_1. Then a projector onto $\mathcal{S}_1$ is called a *spectral projector*.

From this definition we obtain the following properties of spectral projectors.

Lemma 1 *Let* $Z - \lambda Y$ *be as in Definition 3, and let* $P \in \mathbb{R}^{n\times n}$ *be a spectral projector onto the right deflating subspace of* $Z - \lambda Y$ *corresponding to* Λ_1. *Then*

a. $\operatorname{rank}(P) = |\Lambda_1| := k$,
b. $\ker(P) = \operatorname{range}(I - P)$, $\operatorname{range}(P) = \ker(I - P)$,
c. $I - P$ *is a spectral projector onto the right deflating subspace of* $Z - \lambda Y$ *corresponding to* Λ_2.

Given a spectral projector P we can compute an orthogonal basis for the corresponding deflating subspace $\mathcal{S}_1$ and a spectral or block decomposition of $Z - \lambda Y$ in the following way: let

$$P = VR\Pi^T,\ R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \searrow & \square \\ & \end{bmatrix},\ R_{11} \in \mathbb{R}^{k\times k},$$

be a QR decomposition QR decomposition with column pivoting (or a rank-revealing QR decomposition (*RRQR*)) [24], where Π is a permutation matrix. Then the first k columns of V form an orthonormal basis for $\mathcal{S}_1$. If we also know an orthonormal basis of the corresponding left deflating subspace and extend this to a full orthogonal matrix $U \in \mathbb{R}^{n\times n}$, we can transform Z, Y to *block-triangular form*

$$\tilde{Z} - \lambda\tilde{Y} := U^T(Z - \lambda Y)V = \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix} - \lambda, \begin{bmatrix} Y_{11} & Y_{12} \\ 0 & Y_{22} \end{bmatrix}, \tag{3.5}$$

where $\Lambda(Z_{11}, Y_{11}) = \Lambda_1, \Lambda(Z_{22}, Y_{22}) = \Lambda_2$.

Once V is known, the orthogonal matrix U can be computed with little effort based on the following observation [42].

Proposition 4 *Let $Z - \lambda Y \in \mathbb{C}^{n\times n}$ be a regular matrix pencil with no eigenvalues on the boundary $\partial\Gamma_1$ of the stability region Γ_1. If the columns of $V_1 \in \mathbb{C}^{n\times n_1}$ form an orthonormal basis of the stable* right *deflating subspace of $Z - \lambda Y$, i.e., the deflating subspace corresponding to $\Lambda(Z, Y) \cap \Gamma_1$, then the first n_1 columns of the orthogonal matrix U in the QR decomposition with column pivoting,*

$$UR\Pi^T = U\begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}\Pi^T = [\,ZV_1 \quad YV_1\,], \tag{3.6}$$

form an orthonormal basis of the stable left deflating subspace of $Z - \lambda Y$.

The matrix U from (3.6) can then be used for the block-triangularization in (3.5).

The block decomposition given in (3.5) will prove extremely useful in what follows. Besides spectral projectors onto the deflating subspaces corresponding to the finite and infinite parts of the spectrum of a matrix pencil, we will also need those related to the stability regions in continuous- and discrete-time.

Definition 4 Let $Z, Y \in \mathbb{R}^{n\times n}$ with $\Lambda(Z, Y) = \Lambda_1 \cup \Lambda_2, \Lambda_1 \cap \Lambda_2 = \emptyset$, and let $\mathcal{S}_1$ be the (right) deflating subspace of the matrix pencil $Z - \lambda Y$ corresponding to Λ_1. Then $\mathcal{S}_1$ is called

a. *c-stable if* $\Lambda_1 \subset \mathbb{C}^-$ and *c-unstable* if $\Lambda_1 \subset \mathbb{C}^+$;
b. *d-stable* if $\Lambda_1 \subset \{|z| < 1\}$ and *d-unstable* if $\Lambda_1 \subset \{|z| > 1\}$.

3.3.2 The Matrix Sign Function

The sign function method was first introduced by Roberts [36] to solve algebraic Riccati equations. The *sign function* of a matrix $Z \in \mathbb{R}^{n\times n}$ with no eigenvalues on

the imaginary axis can be defined as follows: Let $Z = S\begin{bmatrix} J^- & 0 \\ 0 & J^+ \end{bmatrix} S^{-1}$ denote the Jordan decomposition of Z where the Jordan blocks corresponding to the, say, k eigenvalues in the open left half plane are collected in J^- and the Jordan blocks corresponding to the remaining $n-k$ eigenvalues in the open right half plane are collected in J^+. Then

$$\operatorname{sign}(Z) := S\begin{bmatrix} -I_k & 0 \\ 0 & I_{n-k} \end{bmatrix} S^{-1}.$$

The sign function provides projectors onto certain subspaces of the matrix Z: $\mathcal{P}^- := \frac{1}{2}(I_n - \operatorname{sign}(Z))$ defines the oblique projection onto the c-stable Z-invariant subspace along the c-unstable Z-invariant subspace whereas $\mathcal{P}^+ := \frac{1}{2}(I_n + \operatorname{sign}(Z))$ defines the oblique projection onto the c-unstable Z-invariant subspace along the c-stable Z-invariant subspace. Therefore, the sign function provides a tool for computing a spectral decomposition with respect to the imaginary axis. As we will see in the following subsection, the sign function method can be applied implicitly to $Y^{-1}Z$ so that also the corresponding spectral projectors for matrix pencils $Z - \lambda Y$ with invertible Y can be computed. This can be used for continuous-time stabilization problems once the infinite eigenvalues of $A - \lambda E$ have been deflated.

3.3.3 Computation of the Sign Function

The sign function can be computed via the Newton iteration for the equation $Z^2 = I_n$ where the starting point is chosen as Z, i.e.,

$$Z_0 \leftarrow Z, \quad Z_{j+1} \leftarrow \frac{1}{2}(Z_j + Z_j^{-1}), \quad j = 0, 1, \ldots \tag{3.7}$$

Under the given assumptions, the sequence $\{Z_j\}_{j=0}^{\infty}$ converges to $\operatorname{sign}(Z) = \lim_{j\to\infty} Z_j$ [36] with an ultimately quadratic convergence rate. As the initial convergence may be slow, the use of acceleration techniques is recommended; e.g., *determinantal scaling* [17] adds the following step to (3.7):

$$Z_j \leftarrow \frac{1}{c_j} Z_j, \quad c_j = |\det(Z_j)|^{\frac{1}{n}}, \tag{3.8}$$

where $\det(Z_j)$ denotes the determinant of Z_j. For a summary of different strategies for accelerating the convergence of the Newton iteration, see [28]. It should be noted that eigenvalues close to the imaginary axis may defer convergence considerably with stagnation in the limiting case of eigenvalues on the imaginary axis.

Algorithm 1 Sign function method.

INPUT: A matrix pencil $Z-\lambda Y$, $Z, Y \in \mathbb{R}^{n\times n}$ with no eigenvalues on the imaginary axis.

OUTPUT: *Oblique* spectral projectors $\mathcal{P}^-$ and $\mathcal{P}^+$ onto the c-stable and c-unstable, respectively, deflating subspaces of $Z-\lambda Y$.

Set $Z_0 = Z$, $Y_0 = Y$.
{*Newton iteration*}
for $j = 0, 1, \ldots$ until convergence **do**
  $Z_j = \Pi^T L U$
  {*LU factorization: L/U lower/upper triangular, Π permutation matrix*},
  $c_j = \prod_{k=1}^{n} |u_{kk}|^{\frac{1}{n}}$,
  Solve $LW = \Pi Y$ by forward substitution,
  Solve $UX = W$ by backward substitution,
  $Z_{j+1} = \frac{1}{2c_j} Z_j + \frac{c_j}{2} YX$,
  $s = j+1$.
end for
Set $\mathcal{P}^- := Z_\infty - Y$, $\mathcal{P}^+ := Z_\infty + Y$.

In our case, we will have to apply the sign function method to a matrix pencil rather than a single matrix. Therefore, we employ a generalization of the matrix sign function method to a matrix pencil $Z-\lambda Y$ given in [23]. Assuming that Z and Y are nonsingular, the *generalized Newton iteration* for the matrix sign function is given by

$$Z_0 \leftarrow Z, \quad Z_{j+1} \leftarrow \frac{1}{2c_j}(Z_j + c_j^2 Y Z_j^{-1} Y), \quad j = 0, 1, \ldots, \tag{3.9}$$

with the scaling now defined as $c_j \leftarrow \left(\frac{|\det(Z_j)|}{|\det(Y)|}\right)^{\frac{1}{n}}$. This iteration is equivalent to computing the sign function of the matrix $Y^{-1}Z$ via the Newton iteration as given in (3.7). If $\lim_{j\to\infty} Z_j = Z_\infty$, then $Z_\infty - Y$ defines the oblique projection onto the c-stable right deflating subspace of $Z-\lambda Y$ along the c-unstable deflating subspace, and $Z_\infty + Y$ defines the oblique projection onto the c-unstable right deflating subspace of $Z-\lambda Y$ along the c-stable deflating subspace.

As a basis for the c-stable invariant subspace of a matrix Z or the c-stable deflating subspace of a matrix pencil $Z-\lambda Y$ is given by the range of any projector onto this subspace, it can be computed by a RRQR factorization of the corresponding projectors $\mathcal{P}^-$ or $Z_\infty - Y$, respectively.

A formal description of the suggested algorithm for computing spectral projectors onto the c-stable and c-unstable deflating subspaces of matrix pencils is given in Algorithm 1. The necessary matrix inversion is realized by LU decomposition with partial pivoting and forward/backward solves. Note that no explicit multiplication with the permutation matrix Π is necessary—this is realized by

swapping rows of X or columns of Y. Convergence of the iteration is usually based on relative changes in the iterates Z_j.

For block-triangularization related to spectral division with respect to the imaginary axis, matrices U, V as in (3.5) can be obtained from a RRQR factorization of $\mathcal{P}^-$ and Proposition 4. An explicit algorithm for this purpose is provided in [42]. As we will see later, for our purposes it is not necessary to form U explicitly, just the last $n - n_1$ columns of U need to be accumulated. We will come back to this issue in the context of the disk function method in more detail; see Eq. (3.10) and the discussion given there.

Much of the appeal of the matrix sign function approach comes from the high parallelism of the matrix kernels that compose the Newton-type iterations [34]. Efficient parallelization of this type of iterations for the matrix sign function has been reported, e.g., in [4, 27]. An approach that basically reduces the cost of the generalized Newton iteration to that of the Newton iteration is described in [41]. An inverse-free version of the generalized sign function method which is about 1.5 times as expensive as Algorithm 1 is discussed in [10].

Unfortunately, the matrix sign function is not directly applicable to descriptor systems. If Y ($= E$ in our case) is singular, then convergence of the generalized Newton iteration (3.9) is still possible if the index is less than or equal to 2 [41], but convergence will only be linear. Moreover, as we do not want to restrict ourselves to descriptor systems with low index, we will need another spectral projection method that can be computed by a quadratically convergent algorithm without restrictions on the index. The disk function method described in the following subsection will satisfy these demands.

3.3.4 The Matrix Disk Function

The *right matrix pencil disk function* can be defined for a regular matrix pencil $Z - \lambda Y, Z, Y \in \mathbb{R}^{n\times n}$, as follows: Let

$$Z - \lambda Y = S \begin{bmatrix} J^0 - \lambda I_k & 0 \\ 0 & J^\infty - \lambda N \end{bmatrix} T^{-1}$$

denote the Kronecker (Weierstrass) canonical form of the matrix pencil (see, e.g., [30] and references therein), where $J^0 \in \mathbb{C}^{k\times k}, J^\infty \in \mathbb{C}^{(n-k)\times(n-k)}$ contain, respectively, the Jordan blocks corresponding to the eigenvalues of $Z - \lambda Y$ inside and outside the unit circle. Then, the matrix (pencil) disk function is defined as

$$\operatorname{disk}(Z, Y) := T \left(\begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 0 & 0 \\ 0 & I_{n-k} \end{bmatrix} \right) T^{-1} =: \mathcal{P}^0 - \lambda \mathcal{P}^\infty.$$

Alternative definitions of the disk function are given in [9].

The matrix pencil disk function can be used to compute projectors onto deflating subspaces of a matrix pencil as $\mathcal{P}^0$ is an oblique projection onto the right d-stable deflating subspace of $Z - \lambda Y$ along the d-unstable one, and $\mathcal{P}^\infty$ defines an oblique projection onto the right d-unstable deflating subspace of $Z - \lambda Y$ along the d-stable one. Thus, the disk function provides a tool for spectral decomposition along the unit circle. Splittings with respect to other curves in the complex plane can be computed by applying a suitable conformal mapping to $Z - \lambda Y$ [5]. As we want to use the disk function in order to compute a projector onto the deflating subspace corresponding to the infinite or finite eigenvalues, we will need a curve enclosing all finite eigenvalues. We will come back to this issue in Sect. 3.4.

In the next subsection we discuss how the disk function can be computed iteratively without having to invert any of the iterates.

3.3.5 Computing the Disk Function

The algorithm discussed here is taken from [5], and is based on earlier work by Malyshev [31]. This algorithm is generally referred to as *inverse-free iteration*. We also make use of improvements suggested in [42] to reduce its cost.

Given a regular matrix pencil $Z - \lambda Y$ having no eigenvalues on the unit circle, Algorithm 2 provides an implementation of the inverse-free iteration which computes an approximation to the right deflating subspace corresponding to the eigenvalues inside the unit circle. It is based on a generalized power iteration (see [7, 42] for more details) and the fact that (see [7, 31])

$$\lim_{j\to\infty} (Z_j + Y_j)^{-1} Y_j = \mathcal{P}^0, \quad \lim_{j\to\infty} (Z_j + Y_j)^{-1} Z_j = \mathcal{P}^\infty.$$

Convergence of the algorithm is usually checked based on the relative change in R_j. Note that the QR decomposition in Step 1 is unique if we choose positive diagonal elements as $\begin{bmatrix} Y_j^T, & -Z_j^T \end{bmatrix}^T$ has full rank in all steps [24].

The convergence of the inverse free iteration can be shown to be globally quadratic [5] with deferred convergence in the presence of eigenvalues very close to the unit circle and stagnation in the limiting case of eigenvalues on the unit circle. Also, the method is proven to be numerically backward stable in [5]. Again, accuracy problems are related to eigenvalues close to the unit circle due to the fact that the spectral decomposition problem becomes ill-conditioned in this case.

The price paid for avoiding matrix inversions during the iteration is that every iteration step is about twice as expensive as one step of the Newton iteration (3.9) in the matrix pencil case and three times as expensive as the Newton iteration (3.7) in the matrix case. On the other hand, the inverse free iteration can be implemented very efficiently on a parallel distributed-memory architecture like the sign function method since it is based on matrix multiplications and QR factorizations which are

Algorithm 2 Inverse Free Method.

INPUT: A matrix pencil $Z - \lambda Y$, $Z, Y \in \mathbb{R}^{n\times n}$ with no eigenvalues on the unit circle.

OUTPUT: The matrix pencil disk function of $Z - \lambda Y$.

1: Set $Z_0 = Z$, $Y_0 = Y$. {*Inverse free iteration*}
2: **for** $j = 0, 1, \ldots$ until convergence **do**
3: $\begin{bmatrix} Y_j \\ -Z_j \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} R_j \\ 0 \end{bmatrix}$ (QR factorization),
4: $Z_{j+1} = U_{12}^T Z_j$,
5: $Y_{j+1} = U_{22}^T Y_j$,
6: $s = j + 1$.
7: **end for**
8: Set disk $(Z, Y) := (Z_s + Y_s)^{-1}(Y_s - \lambda Z_s)$.

well studied problems in parallel computing; see, e.g., [24, 35] and the references given therein. Also, a recently proposed version yields a reduced cost per iteration step which leads to the conclusion that usually, the inverse-free iteration is faster than the QZ algorithm if less than 30 iterations are required—usually, convergence can be observed after only 10–20 iterations. See [32] for details.

It should be noted that for our purposes, neither the disk function nor the projectors $\mathcal{P}^0$ nor $\mathcal{P}^\infty$ need to be computed explicitly. All we need are the related matrices Q, Z from (5). This requires orthogonal bases for the range and nullspace of these projectors. These can be obtained using a clever subspace extraction technique proposed in [42]. The main idea is here that a basis for the range of $\mathcal{P}^0$ can be obtained from the kernel of $\mathcal{P}^\infty$. This can be computed from Z_s directly, i.e., $Z_s + Y_s$ is never inverted, neither explicitly nor implicitly. The left deflating subspace is then computed based on Proposition 4. The complete subspace extraction technique yielding the matrices U, V as in (3.5) with $\Lambda(Z_{22}, Y_{22})$ being the d-stable part of the spectrum of $Z - \lambda Y$ can be found in Algorithm 3.

Note that the triangular factors and permutation matrices in Algorithm 3 are not needed and can be overwritten. For our purposes, we can save some more workspace and computational cost. It is actually sufficient to store V (which is later on needed to recover the feedback matrix of the original system) and to compute

$$Z_{22} = U_2^T Z V_2, \quad Y_{22} = U_2^T Y V_2. \tag{3.10}$$

This can be exploited if an efficient implementation of the QR decomposition like the one in LAPACK [1] is available. Accumulation of U_2 only is possible there, so that only $\frac{4}{3}nr(n-r)$ flops are needed rather than $\frac{4}{3}n^3$ when accumulating the full matrix U. Moreover, instead of $8n^3$ flops as needed for the four matrix products in (3.5), the computations in (3.10) require only $4nr(n+r)$ flops.

Algorithm 3 Subspace Extraction for Disk Function Method.

INPUT: A matrix pencil $Z-\lambda Y$ $Z, Y \in \mathbb{R}^{n\times n}$ with no eigenvalues on the unit circle, Z_s as computed by Algorithm 2, and a tolerance τ for rank detection.

OUTPUT: Matrices U, V upper triangularizing $Z - \lambda Y$ so that with

$$U^T(Z-\lambda Y)V = \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix} - \lambda, \begin{bmatrix} Y_{11} & Y_{12} \\ 0 & Y_{22} \end{bmatrix},$$

we have $\Lambda(Z_{11}, Y_{11}) \subset \{z \in \mathbb{C} \mid |z| < 1\}$, $\Lambda(Z_{22}, Y_{22}) \subset \{z \in \mathbb{C} \mid |z| > 1\}$.

1: {*Compute range of* $\mathcal{P}^0$*, i.e., nullspace of* Z_s*.*}
Compute the RRQR

$$\begin{bmatrix} V_2 & V_1 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \Pi_V^T = Z_s^T,$$

where the partitioning is determined with respect to the numerical rank $r = r(\tau)$ of Z_s^T based on τ and the columns of $V_1 \in \mathbb{R}^{n\times n-r}$ form an orthonormal basis for $\ker Z_s$.

2: Set $V := \begin{bmatrix} V_1 & V_2 \end{bmatrix}$.

3: {*Compute basis for the left deflating subspaces.*}
Compute the QR decomposition with column pivoting

$$UT\Pi_U^T \equiv \begin{bmatrix} U_1 & U_2 \end{bmatrix} T\Pi_U^T = \begin{bmatrix} ZV_1 & YV_1 \end{bmatrix},$$

with the same partitioning with respect to $r(\tau)$ as above.

3.4 Partial Stabilization Using Spectral Projection

For the derivation of our algorithm, we will assume that the descriptor system (3.1) has n_f finite and n_∞ infinite poles, i.e., $\Lambda(A,E) = \Lambda_f \dot\cup \{\infty\}$ with $|\Lambda_f| = n_f$. Moreover, we assume that there are n_1 stable and n_2 unstable poles, i.e., $\Lambda(A,E) = \Lambda_1 \dot\cup \Lambda_2$, where $\Lambda_1 \subset \Gamma_1$, $\Lambda_2 \subset \overline{\mathbb{C}} \setminus \Gamma_1$, and $|\Lambda_1| = n_1, |\Lambda_2| = n_2 + n_\infty$. This yields the relations $n = n_f + n_\infty = n_1 + n_2 + n_\infty$. Our partial stabilization algorithm based on the disk function method will consist of the following steps:

1. Deflate the infinite poles of the system using a spectral projector onto the corresponding deflating subspace of $A - \lambda E$ computed by the disk function method. We call the resulting $n_f \times n_f$ matrix pencil $A_1 - \lambda E_1$. Note that now, E_1 is nonsingular.
   Transform B accordingly, yielding B_1.
2. Deflate the stable finite poles of the system using a spectral projector onto the deflating subspace of $A_1 - \lambda E_1$. This can be computed by the disk (sign) function method applied to (A_1, E_1) in the discrete-time (continuous-time) case or to the Cayley-transformed matrix pair $(A_1 + E_1, A_1 - E_1)$ in the continuous-time (discrete-time) case. We call the resulting $n_2 \times n_2$ matrix pencil $A_2 - \lambda E_2$. Note that now, $A_2 - \lambda E_2$ has only unstable eigenvalues.
   Transform B_1 accordingly, yielding B_2.

3. Solve the stabilization problem for the generalized state-space system

$$E_2(\mathcal{D}x_2(t)) = A_2x(t) + B_2u(t), \quad t > 0, \quad x_2(0) = x_{2,0}.$$

4. Assemble the stabilizing feedback for the original system (1) using either Propositions 2 or 3.

A possible implementation, based only on the disk function method, is summarized in Algorithm 4. An analogous implementation using the sign function method in Step 2 can easily be derived replacing Algorithm 2 by Algorithm 1 there (and changing the order of the if-else conditions).

Algorithm 4 Partial Stabilization Using the Disk Function Method.

INPUT: A stabilizable descriptor system as in (1) with $A-\lambda E$ regular; α so that a circle around the origin with radius $1/\alpha$ encloses all finite eigenvalues of $A-\lambda E$.

OUTPUT: A stabilizing feedback matrix $F \in \mathbb{R}^{m\times n}$.

1: Apply the disk function method to $(E, \alpha A)$ to obtain a spectral projector onto the deflating subspace corresponding to the finite eigenvalues of $A - \lambda E$ and use Algorithm 3 to compute an orthogonal equivalence transformation to block-triangular form such that $\Lambda(A, E)$ is divided into

$$Q_1(A - \lambda E)Z_1 = \begin{bmatrix} A_\infty & A_{12} \\ 0 & A_1 \end{bmatrix} - \lambda \begin{bmatrix} E_\infty & E_{12} \\ 0 & E_1 \end{bmatrix},$$

where $\Lambda(A_\infty, E_\infty) = \{\infty\}$, $\Lambda(A_1, E_1)$ is finite, E_1 nonsingular, and partition $Q_1B =: \begin{bmatrix} B_\infty \\ B_1 \end{bmatrix}$ accordingly.

2: **if** the system is continuous-time,
apply Algorithm 2 to $(A_1 + E_1, A_1 - E_1)$
else
apply Algorithm 2 to (A_1, E_1)
endif
in order to compute an orthogonal equivalence transformation to block-triangular form such that $\Lambda(A_1, E_1)$ is divided according to the boundary of the stability region, i.e.,

$$Q_2(A_1 - \lambda E_1)Z_2 = \begin{bmatrix} A_{\text{stab}} & * \\ 0 & A_2 \end{bmatrix} - \lambda \begin{bmatrix} E_{\text{stab}} & * \\ 0 & E_2 \end{bmatrix},$$

where $\Lambda(A_{\text{stab}}, E_{\text{stab}}) \subset \Gamma_1$, $\Lambda(A_2, E_2) \subset \Gamma_2$, and partition $Q_2B_1 =: \begin{bmatrix} B_{\text{stab}} \\ B_2 \end{bmatrix}$ accordingly.

3: Compute F_2 using either Proposition 2 applied to (A_2, E_2, B_2) or Proposition 3 applied to (A_2, E_2, B_2) in the continuous-time case and to (A_2+E_2, A_2-E_2, B_2) in the discrete-time case.

4: Set $F := \begin{bmatrix} 0, & [\,0 \; F_2]Z_2^T \end{bmatrix} Z_1^T$.

The determination of the input parameter α in Algorithm 4 is in itself a non-trivial task. Often, knowledge about the spectral distribution can be obtained a priori from the physical modeling. It is an open problem how to determine α directly from the entries of the matrices A, E. In principle, the generalized Gershgorin theory derived in [38] provides computable regions in the closed complex plane containing the finite eigenvalues. In practice, though, these regions often extend to infinity and thus provide no useful information. Recent developments in [29] give rise to the hope that computable finite bounds can be obtained; this will be further explored in the future.

The solution of the matrix equations needed in Step 3 of Algorithm 4 (either one of the Lyapunov equations from (3.4) or the ABE (3.2)) can be obtained in several ways, see, e.g., [20, 37] for a discussion of different Lyapunov solvers. As we base our partial stabilization algorithm on spectral projection methods like the sign and disk function methods under the assumption that these are particularly efficient on current computer architectures, it is quite natural to also use the sign function methods for the generalized Lyapunov and Bernoulli equations in (3.4) and (3.2). Efficient algorithms for this purpose are derived and discussed in detail in [6, 13, 15]. Note that the matrix pencils defining the Lyapunov operators in (3.4) have all their eigenvalues in Γ_2. Thus the methods from [13, 15] can be applied to these equations. Our numerical results in Sect. 3.5 are all computed using these sign function based matrix equation solvers.

Remark 1 Due to the usual ill-conditioning of the stabilization problem, it turns out that sometimes the computed closed-loop poles are not all stable. In that case, we suggest to apply Steps 2–3 of Algorithm 4 again to $(A_2 - B_2F_2, E_2)$, resulting in a feedback matrix

$$F := \begin{bmatrix} 0, & \begin{bmatrix} 0, & F_2 + \begin{bmatrix} 0, & F_3 \end{bmatrix} Z_3^T \end{bmatrix} Z_2^T \end{bmatrix} Z_1^T.$$

This is often sufficient to completely stabilize the system numerically, see Example 2. Otherwise, Steps 2–3 should be repeated until stabilization is achieved.

Remark 2 So far we have assumed spectral dichotomy with respect to the boundary curve $\partial\Gamma$ of the stability region. For Step 1 of Algorithm 4, this is not necessary. It becomes an issue only in the following steps, but can easily be resolved. For the spectral decomposition with respect to the boundary of the stability region performed in Step 2, we can simply shift/scale so that the eigenvalues on $\partial\Gamma$ are moved to Γ_2. This may also be advisable if stable eigenvalues are close to the boundary of the stability region and should be made "more stable". This basic idea is implemented, for instance, in the Descriptor System and Rational Matrix Manipulation Toolbox [44]. For Algorithm 4 this means that in Step 2, the spectral decomposition is computed for $(A_1 + \alpha_c E_1, E_1)$ $(\alpha_c > 0)$ in the continuous-time case and for $(A_1, \alpha_d E_1)$ $(0 < \alpha_d < 1)$ in the discrete-time case. The resulting matrix pencil (A_2, E_2) will then again have eigenvalues on $\partial\Gamma$ (or stable ones close to it). This can be treated in Step 3 again with shifting/scaling: when

applying Proposition 2, simply set $\beta_c > 0$ or $\beta_d < 1$ (as already advised in Sect. 3.2), while if Proposition 3 is to be used, it is applied to $(A_2 + \alpha_c E_2, E_2, B_2)$ with $\alpha_c > 0$. (Note that Proposition 3 applies only in the continuous-time case.)

In the following section, we will test the suggested partial stabilization method as presented in Algorithm 4 for several problems, in particular for stabilization problems for linear(ized) flow problems.

3.5 Numerical Examples

In order to demonstrate the effect of the partial stabilization method on the poles of the system (3.1), we implemented the generalized Bass and Bernoulli versions of Algorithm 4 as MATLAB functions. For the solution of the Lyapunov and Bernoulli equations in (3.4) and (3.2) we employ MATLAB implementations of the sign function based solvers described in detail in [6, 13, 15].

Note that there is no software for partial stabilization to compare to as even the MATLAB function `gstab` from the Descriptor System and Rational Matrix Manipulation Toolbox[1] (Descriptor Toolbox for short) [44] only treats systems with invertible E. Nevertheless, we compare our results with those obtained by `gstab` applied to the projected problem resulting from Step 1 of Algorithm 4. Moreover, we tried to apply the pole placement function `gplace` from the Descriptor Toolbox on each level, i.e., for the full descriptor system and the projected systems after Steps 1 and 2 of Algorithm 4. All computations were performed using MATLAB release R2006a with IEEE double precision arithmetic on Windows XP notebooks with Pentium M CPU and running either at 1.13 GHz with 512 Mb of main memory or 2.13 GHz with 1 Gb of main memory.

Example 1 In order to be able to distinguish all eigenvalues in the plots, we first use a small scale example with $n = 20, m = 3$. Therefor, a diagonal matrix pencil with ten infinite eigenvalues and finite spectrum $\{-4.5, -3.5, \ldots, -0.5, 5.5, \ldots 9.5\}$ is generated. Thus, five unstable poles have to be stabilized. Our algorithm computes a feedback matrix with moderate norm: $\|F\|_2 \approx 148$ using the generalized Bass stabilization and $\|F\|_2 \approx 232$ when using the algebraic Bernoulli equation. Thus, the norm of the gain is quite reduced in the generalized Bass case as $\|F_1\|_2 \approx 4,490$ if applied to the system described by (A_1, B_1, E_1). Note that in the Bernoulli case, there is no reduction in the norm of the gain matrix which can be expected as in an appropriate basis, the solution X_1 for the larger problem is zero except for the diagonal block in the lower right corner which is the Bernoulli solution of the small fully unstable system. The full-scale feedback matrix results from applying orthogonal transformations only to the small-size feedback so that the 2-norm is not changed. The function `gplace` from the Descriptor System and Rational Matrix

[1] Available Version: 1.05, 1 October 2005.

Manipulation Toolbox works in this example as well: if we aim at placing the poles at the location of the Bernoulli stabilized ones, then the corresponding feedback gain matrix has norm ≈204. The computed closed-loop poles are slightly less accurate than in the Bernoulli approach, though, with relative errors in the computed poles roughly four times larger.

The open- and closed-loop poles are displayed in Fig. 3.1. As usual for the Bass algorithm, the stabilized poles are placed on a vertical line in the left half plane while the stable open-loop poles are preserved. As expected from Proposition 3, the Bernoulli approach reflects the unstable poles with respect to the imaginary axis.

Example 2 Consider the instationary Stokes equation describing the flow of an incompressible fluid at low Reynolds numbers:

$$\begin{aligned} \frac{\partial v}{\partial t} &= \triangle v - \nabla \rho + f, && (\xi, t) \in \Omega \times (0, t_f), \\ 0 &= div\ v, && (\xi, t) \in \Omega \times (0, t_f) \end{aligned} \tag{3.11}$$

with appropriate initial and boundary conditions. Here, $v(\xi, t) \in \mathbb{R}^2$ is the velocity vector, $\rho(\xi, t) \in \mathbb{R}$ is the pressure, $f(\xi, t) \in \mathbb{R}^2$ is a vector of external forces, $\Omega \subset \mathbb{R}^2$ is a bounded open domain and $t_f > 0$ is the endpoint of the considered time interval. The spatial discretization of the Stokes equation (3.11) by a finite volume method on a uniform staggered grid leads to a descriptor system, where the matrix coefficients are sparse and have a special block structure given by

$$E = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathrm{A} = \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{12}^{\mathrm{T}} & 0 \end{bmatrix},$$

see [40] and references therein. Here, $A_{11} \in \mathbb{R}^{n_v \times n_v}$ corresponds to the discretized Laplace operator while A_{12} and A_{12}^T represent discretizations of the gradient and divergence operators in (3.11).

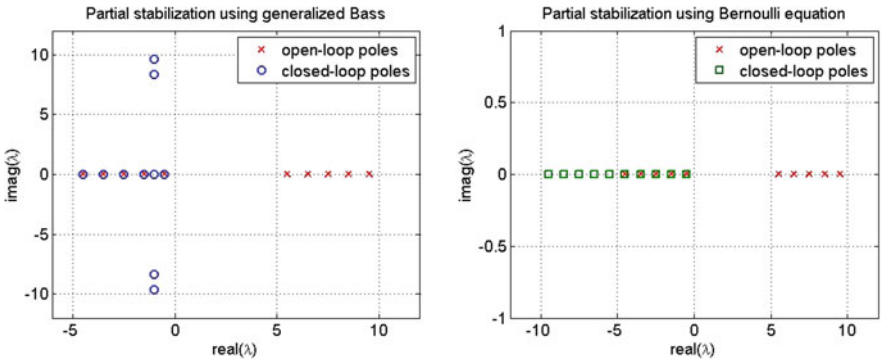


Fig. 3.1 Example 1. Open-loop and closed-loop poles computed using generalized Lyapunov and Bernoulli equations

The input matrix B can be obtained from several sources. Assuming volume forces, we can write $f(\xi,t) = b(\xi)u(t)$ and the matrix B is obtained from the discretization of $b(\xi)$. Another possibility is boundary control.

The descriptor system obtained in this way is stable and of index 2 [40]. We de-stabilize the system by adding αI_{n_v} to A_{11} so that the index is preserved. Such a term arises, e.g., when the volume forces are also proportional to the velocity field, $f(\xi,t) = \tilde{\alpha}v(\xi,t) + b(\xi)u(t)$ (where we do not claim physical relevance of this situation—α is then obtained from $\tilde{\alpha}$ by scaling related to the mesh size).

In all of the following computations, we consider a coarse 16×16 grid, resulting in $n_v = 480$ velocity and $n_p = 255$ pressure variables so that $n = 735$. This results in $n_\infty = 510$ infinite poles and 225 finite ones.

In the first setting, we mimic a situation where the volume forces are given by the superposition of two different sources. Thus, $m = 2$, where we choose the two columns of B at random. Choosing $\alpha = 100$, 3 poles become unstable. Altogether, the stabilization problem turns out to be ill-conditioned enough to make `gplace` fail to stabilize the descriptor system as well as the system projected on the subspaces corresponding to the finite eigenvalues (resulting from Step 1 of Algorithm 4) with all pole assignments we tried (random assignment, equally distributed from -1 to $-n_f$). Note that `gplace` returns with an error message when applied to the full descriptor system while for the projected system, non-stabilizing feedbacks are returned without a warning. The Descriptor Toolbox function `gstab` stabilized the projected system with $\|F\|_2 \approx 85.5$. Both our approaches based on the generalized Bass algorithm and on the algebraic Bernoulli equation succeed in stabilizing the system, where $\|F_{\text{Bass}}\|_2 \approx 170$ and $\|F_{\text{ABE}}\|_2 \approx 233$. It appears that here, the stabilization of the fully unstable, small system requires more effort than the stabilization of the larger projected system. Open- and closed-loop poles for both approaches are shown in Fig. 3.2.

From the close-up (bottom row in Fig. 3.2) we can again see that the Bernoulli approach reflects the unstable poles with respect to the origin. It should be noted, though, that some of the computed closed-loop poles do not possess the desirable property to come out as real eigenvalues (all finite poles are real in this example), but the imaginary parts (which are zero in exact arithmetic) are rather small (of order 10^{-12}).

In our second set of tests we use an input matrix related to boundary control similar to the version used in [39], i.e., we have $m = 64$. To make the stabilization problem more difficult, we set $\alpha = 1,000$ for the de-stabilization, resulting in 105 unstable poles. Neither `gplace` (with assignment of random poles, mirrored unstable poles, or $\{-\ell, -\ell+1, \ldots, -1\}$ where $\ell = n_1 + n_2$ or $\ell = n_2$) nor `gstab` were able to stabilize any of the systems: both fail for the full descriptor system with error messages, they compute feedback matrices for the generalized state-space systems resulting from projecting onto the deflating subspaces corresponding to finite eigenvalues and unstable poles, respectively. The best result using `gplace` was obtained when applied to the projected fully unstable system with poles assigned to $\{-105, -104, \ldots, -1\}$. In this case, only 14 computed

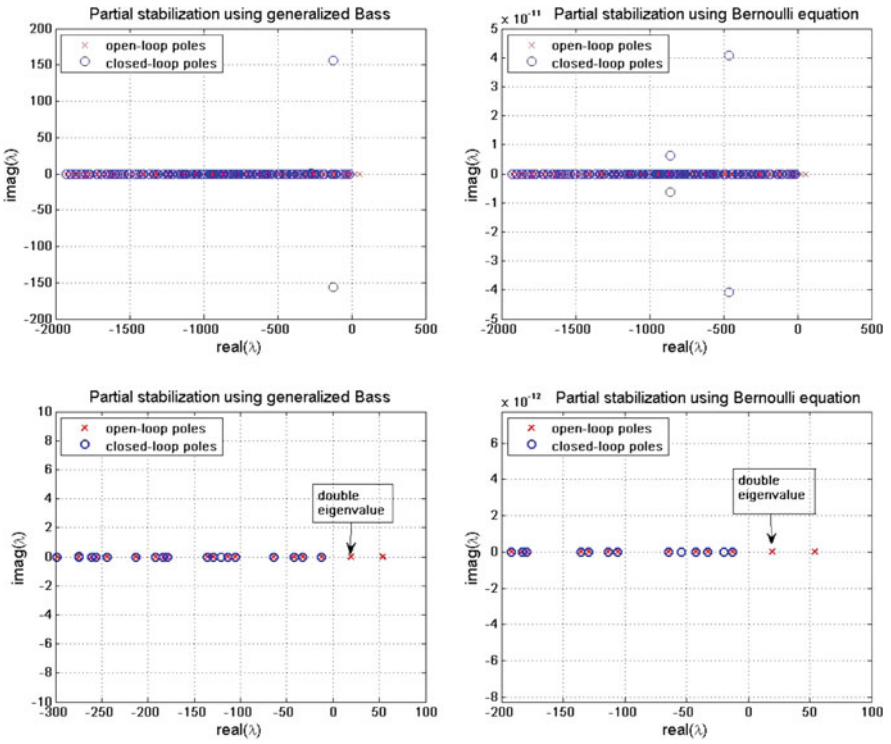


Fig. 3.2 Example 2 with B related to volume force ($m = 2$). Open-loop and closed-loop poles computed using generalized Lyapunov and Bernoulli equations (*top row*) with close-up around the origin (*bottom row*)

closed-loop poles remained unstable. But it should be observed that none of the computed closed-loop poles is close to the assigned ones! In all other attempts and also for `gstab`, the number of unstable computed closed-loop poles was much larger.

On the other hand, both versions of our partial stabilization algorithms based on the generalized Bass algorithm and the algebraic Bernoulli equation were able to stabilize this system. A second stabilization step as described in Remark 1 was necessary, though. For the generalized Bass algorithm with $\beta = 1$, eight closed-loop poles remain unstable after the first stabilization step while only two unstable poles had to be treated in the second stabilization step when using the Bernoulli approach. The resulting gain matrices show that a lot of effort is needed to stabilize the system: $\|F_{\text{Bass}}\|_2 \approx 1.4 \cdot 10^8$ and $\|F_{\text{ABE}}\|_2 \approx 4.2 \cdot 10^8$. The plotted pole distributions shown in Fig. 3.3 demonstrate that the slightly higher effort of the Bernoulli approach is worthwhile as there are no highly undamped closed-poles (i.e., poles with relatively large imaginary parts compared to their real parts). The close-ups in the bottom row of this figure also show again that unstable poles are reflected with respect to the imaginary axis in the Bernoulli approach. It must be

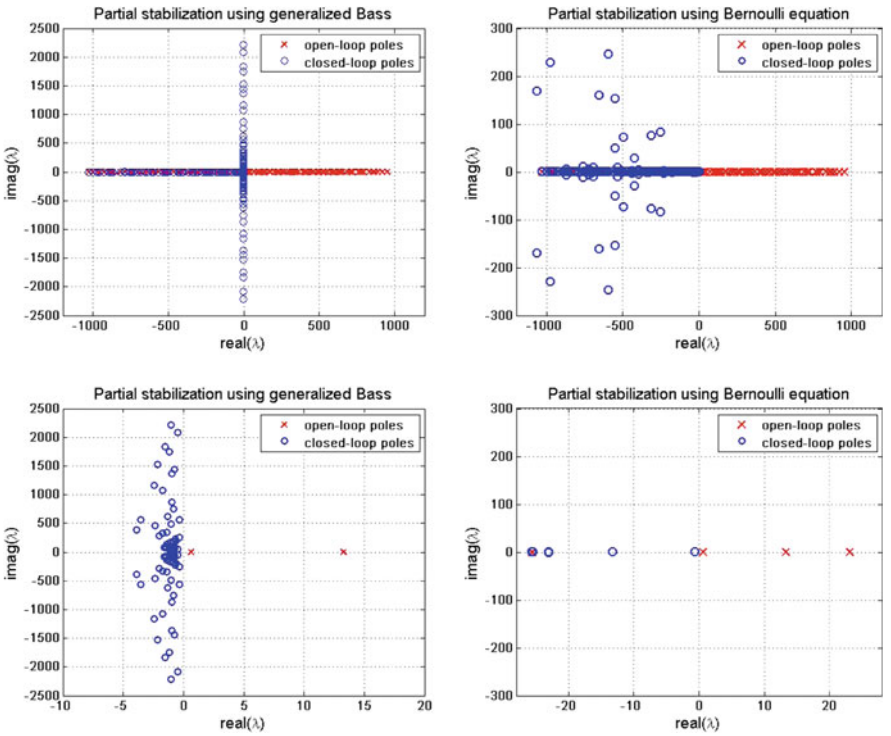


Fig. 3.3 Example 2 with B related to boundary control ($m = 64$). Open-loop and closed-loop poles computed using generalized Lyapunov and Bernoulli equations (*top row*) with close-up around the origin (*bottom row*)

noticed, though, that all closed-loop poles in the Bernoulli approach should theoretically be real which is obviously not true for the computed ones. This is due to the fact that the closed-loop matrix pencil represents a highly non-normal eigenvalue problem and small perturbations may lead to large deviations in the eigenvalues. Fortunately, the closed-loop poles with nonzero imaginary parts are far enough from the imaginary axis so that in practice, the closed-loop pole distribution computed using the Bernoulli approach can be considered reasonable.

3.6 Conclusions and Outlook

The partial stabilization methods suggested in this paper use the disk function method to first separate finite and infinite, and then the disk or sign function method to separate stable and unstable poles of a stabilizable descriptor system. The stable poles are preserved while the unstable ones are stabilized by state feedback. The feedback gain matrix is computed using either the generalized Bass

algorithm as described in [8] (similar to [43]) or an approach based on the algebraic Bernoulli equation. In the latter case, the stabilized poles are the mirror images of the unstable open-loop poles. Due to the ill-conditioning of the stabilization problem, closed-loop poles may not be stable in contrast to expectation raised by the theory. Often, applying the partial stabilization algorithm again to the closed-loop system resolves this problem. The numerical examples demonstrate that our algorithm can solve stabilization problems so far not treatable with available software.

The disk function based stabilization method can be applied to fairly large systems with up to several thousand state-space variables as it can be implemented very efficiently on modern computer architectures. A parallel implementation of the algorithm based on the matrix disk function and partial stabilization methods for standard systems in generalized state-space form [11, 12] as implemented in the Parallel Library in Control[2] (PLiC, see [14]) is straightforward and is planned for the future.

References

1. Anderson E, Bai Z, Bischof C, Demmel J, Dongarra J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Sorensen D (1999) LAPACK users' guide. 3rd edn. SIAM, Philadelphia
2. Armstrong E (1975) An extension of Bass' algorithm for stabilizing linear continuous constant systems. IEEE Trans Automat Control AC 20:153–154
3. Armstrong E, Rublein GT (1976) A stabilization algorithm for linear discrete constant systems. IEEE Trans Automat Control AC 21:629–631
4. Bai Z, Demmel J, Dongarra J, Petitet A, Robinson H, Stanley K (1997) The spectral decomposition of nonsymmetric matrices on distributed memory parallel computers. SIAM J Sci Comput 18:1446–1461
5. Bai Z, Demmel J, Gu M (1997) An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. Numer Math 76(3):279–308
6. Barrachina S, Benner P, Quintana-Ortí E (2007) Efficient algorithms for generalized algebraic Bernoulli equations based on the matrix sign function. Numer Algorithms 46(4): 351–368
7. Benner P (1997) Contributions to the numerical solution of algebraic Riccati equations and related Eigenvalue problems. Logos–Verlag, Berlin, Germany, *Also* : Dissertation, Fakultät für Mathematik, TU Chemnitz–Zwickau
8. Benner P (2003) Partial stabilization of generalized state-space systems using the disk function method. Proc Appl Math Mech 2(1):479–480
9. Benner P, Byers R (1997) Disk functions and their relationship to the matrix sign function. In: Proceedings of the European Control Conference ECC 97, Paper 936. BELWARE Information Technology, Waterloo, Belgium, CD-ROM
10. Benner P, Byers R (2006) An arithmetic for matrix pencils: Theory and new algorithms. Numer Math 103(4):539–573
11. Benner P, Castillo M, Hernández V, Quintana-Ortí E (2000) Parallel partial stabilizing algorithms for large linear control systems. J Supercomput 15:193–206

[2] See http://www.pscom.uji.es/plic.

12. Benner P, Castillo M, Quintana-Ortí E (2005) Partial stabilization of large-scale discrete-time linear control systems. Int J Comp Sci Eng 1(1):15–21
13. Benner P, Quintana-Ortí E (1999) Solving stable generalized Lyapunov equations with the matrix sign function. Numer Algorithms 20(1):75–100
14. Benner P, Quintana-Ortí E, Quintana-Ortí G (1999) A portable subroutine library for solving linear control problems on distributed memory computers. In: Cooperman G, Jessen E, Michler G (eds) Workshop on wide area networks and high performance computing, Essen (Germany), September 1998 Lecture Notes in Control and Information. Springer, Heidelberg pp 61–88
15. Benner P, Quintana-Ortí E, Quintana-Ortí G (2002) Numerical solution of discrete stable linear matrix equations on multicomputers. Parallel Algorithms Appl 17(1):127–146
16. Brenan K, Campbell S, Petzold L (1989) Numerical solution of initial-value problems in differential–algebraic equations. Elsevier Science, North-Holland
17. Byers R (1987) Solving the algebraic Riccati equation with the matrix sign function. Linear Algebra Appl 85:267–279
18. Chu P, Wie B, Gretz B, Plescia C (1990) Approach to large space structure control system design using traditional tools. AIAA J Guidance Control Dynam 13:874–880
19. Dai L (1989) Singular control systems. Number 118 in lecture notes in control and information sciences. Springer, Berlin
20. Datta B (2004) Numerical methods for linear control systems. Elsevier Academic Press, San Diego/London
21. Dragan V, Halanay A (1997) Stabilization of linear systems. Birkhäuser, Basel, Switzerland
22. Gallivan K, Rao X, Van Dooren P (2006) Singular Riccati equations stabilizing large-scale systems. Linear Algebra Appl 415(2–3):359–372
23. Gardiner J, Laub A (1986) A generalization of the matrix-sign-function solution for algebraic Riccati equations. Internat J Control 44:823–832
24. Golub G, Van Loan C (1996) Matrix computations 3rd edn. Johns Hopkins University Press, Baltimore
25. He C, Laub A, Mehrmann V (1995) Placing plenty of poles is pretty preposterous. Preprint SPC 95_17, DFG–Forschergruppe "SPC", Fakultät für Mathematik, TU Chemnitz–Zwickau, 09107 Chemnitz, Germany, May 1995. Available http://www.tu-chemnitz.de/sfb393/spc95pr.html
26. He C, Mehrmann V (1994) Stabilization of large linear systems. In: Kulhavá L, Kárný M, ∼Warwick K (eds) Preprints of the European IEEE Workshop CMP'94, Prague, September 1994, pp 91–100
27. Huss S, Quintana-Ortí E, Sun X, Wu J (2000) Parallel spectral division using the matrix sign function for the generalized eigenproblem. Int J High Speed Comput 11(1):1–14
28. Kenney C, Laub A (1995) The matrix sign function. IEEE Trans Automat Control 40(8):1330–1348
29. Kostić V (2008) Eigenvalue localization for matrix pencils. In Applied Linear Algebra—in Honor of Ivo Marek, April 28–30, 2008, Novi Sad, Serbia (Unpublished talk)
30. Kunkel P, Mehrmann V (2006) Differential-algebraic equations: analysis and numerical solution. EMS Publishing House, Zurich
31. Malyshev A (1993) Parallel algorithm for solving some spectral problems of linear algebra. Linear Algebra Appl 188/189:489–520
32. Marqués M, Quintana-Ortí E, Quintana-Ortí G (2007) Specialized spectral division algorithms for generalized eigenproblems via the inverse-free iteration. In: Kågström B, Elmroth E, Dongarra J, Waśniesky J (eds) PARA'06 Applied computing: State-of-the-art in scientific computing, volume 4699 of Lecture Notes in Computer Science. Springer, Heidelberg, pp 157–166
33. Mehrmann V, Stykel T (2006) Descriptor systems: a general mathematical framework for modelling, simulation and control. at-Automatisierungstechnik 54(8):405–415
34. Quintana-Ortí E, Quintana-Ortí G, Sun X, van de Geijn R (2001) A note on parallel matrix inversion. SIAM J Sci Comput 22:1762–1771

35. Quintana-Orti G, Sun X, Bischof C (1998) A BLAS-3 version of the QR factorization with column pivoting. SIAM J Sci Comput 19:1486–1494
36. Roberts J (1971) Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. Int J Control 32:677–687, 1980. (Reprint of Technical Report No. TR-13, CUED/B-Control, Cambridge University, Engineering Department)
37. Sima V (1996) Algorithms for linear-quadratic optimization, volume 200 of pure and applied mathematics. Marcel Dekker, Inc, New York
38. Stewart G (1975) Gershgorin theory for the generalized eigenvalue problem $Ax = \lambda Bx$. Math Comp 29:600–606
39. Stykel T (2002) Analysis and numerical solution of generalized Lyapunov equations. Dissertation, TU Berlin
40. Stykel T (2006) Balanced truncation model reduction for semidiscretized Stokes equation. Linear Algebra Appl 415(2–3):262–289
41. Sun X, Quintana-Ortí E (2002) The generalized Newton iteration for the matrix sign function. SIAM J Sci Comput 24(2):669–683
42. Sun X, Quintana-Ortí E (2004) Spectral division methods for block generalized Schur decompositions. Math Comp 73:1827–1847
43. Varga A (1995) On stabilization methods of descriptor systems. Sys Control Lett 24:133–138
44. Varga A (2000) A descriptor systems toolbox for MATLAB. In Proceedings of the 2000 IEEE International Symposium on CACSD, Anchorage. Alaska, USA, 25–27 Sept 2000, pp 150–155. IEEE Press, Piscataway, NJ

Chapter 4
Comparing Two Matrices by Means of Isometric Projections

T. P. Cason, P.-A. Absil and P. Van Dooren

Abstract In this paper, we go over a number of optimization problems defined on a manifold in order to compare two matrices, possibly of different order. We consider several variants and show how these problems relate to various specific problems from the literature.

4.1 Introduction

When comparing two matrices A and B it is often natural to allow for a class of transformations acting on these matrices. For instance, when comparing adjacency matrices A and B of two graphs with an equal number of nodes, one can allow symmetric permutations $P^T AP$ on one matrix in order to compare it to B, since this is merely a relabelling of the nodes of A. The so-called comparison then consists in finding the best match between A and B under this class of transformations.

A more general class of transformations would be that of unitary similarity transformations $Q^* AQ$, where Q is a unitary matrix. This leaves the eigenvalues of

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

T. P. Cason (✉) · P.-A. Absil · P. Van Dooren
Department of Mathematical Engineering, Université catholique de Louvain, Bâtiment Euler, avenue Georges Lemaître 4, 1348 Louvain-la-Neuve, Belgium
e-mail: thomas.cason@uclouvain.be
URL: http://www.inma.ucl.ac.be/∼cason

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_4,

A unchanged but rotates its eigenvectors, which will of course play a role in the comparison between A and B. If A and B are of different order, say m and n, one may want to consider their restriction on a lower dimensional subspace:

$$U^*AU \quad \text{and} \quad V^*BV, \tag{4.1}$$

with U and V belonging to $\mathrm{St}(k,m)$ and $\mathrm{St}(k,n)$ respectively, and where $\mathrm{St}(k,m) = \left\{U \in \mathbb{C}^{m\times k} : U^*U = I_k\right\}$ denotes the *compact Stiefel manifold*. This yields two square matrices of equal dimension $k \leq \min(m,n)$, which can again be compared.

But one still needs to define a measure of comparison between these restrictions of A and B which clearly depends on U and V. Fraikin et al. [1] propose in this context to maximize the inner product between the *isometric projections*, U^*AU and V^*BV, namely:

$$\arg\max_{\substack{U^*U=I_k\\V^*V=I_k}} \langle U^*AU, V^*BV\rangle := \Re\,\mathrm{tr}((U^*AU)^*(V^*BV)),$$

where $\Re$ denotes the real part of a complex number. They show this is also equivalent to

$$\arg\max_{\substack{X=VU^*\\U^*U=I_k\\V^*V=I_k}} \langle XA, BX\rangle = \Re\,\mathrm{tr}(A^*X^*BX),$$

and eventually show how this problem is linked to the notion of graph similarity introduced by Blondel et al. [2]. The graph similarity matrix S introduced in that paper also proposes a way of comparing two matrices A and B via the fixed point of a particular iteration. But it is shown in [3] that this is equivalent to the optimization problem

$$\arg\max_{\|S\|_F=1} \langle SA, BS\rangle = \Re\,\mathrm{tr}((SA)^*BS)$$

or also

$$\arg\max_{\|S\|_F=1} \langle S, BSA^*\rangle = \Re\,\mathrm{tr}((SA)^*BS).$$

Notice that S also belongs to a Stiefel manifold, since $\mathrm{vec}(S) \in \mathrm{St}(1, mn)$.

In this paper, we use a distance measure rather than an inner product to compare two matrices. As squared distance measure between two matrices M and N, we will use

$$\mathrm{dist}^2(M,N) = \|M-N\|_F^2 = \mathrm{tr}((M-N)^*(M-N)).$$

We will analyze distance minimization problems that are essentially the counterparts of the similarity measures defined above. These are

$$\arg\min_{\substack{U^*U=I_k\\V^*V=I_k}} \mathrm{dist}^2(U^*AU, V^*BV),$$

$$\arg \min_{\substack{X=VU^* \\ U^*U=I_k \\ V^*V=I_k}} \operatorname{dist}^2(XA, BX),$$

and

$$\arg \min_{\substack{X=VU^* \\ U^*U=I_k \\ V^*V=I_k}} \operatorname{dist}^2(X, BXA^*),$$

for the problems involving two isometries U and V. Notice that these three distance problems are not equivalent although the corresponding inner product problems are equivalent.

Similarly, we will analyze the two problems

$$\arg \min_{\|S\|_F=1} \operatorname{dist}^2(SA, BS) = \operatorname{tr}((SA - BS)^*(SA - BS))$$

and

$$\arg \min_{\|S\|_F=1} \operatorname{dist}^2(S, BSA^*) = \operatorname{tr}((S - BSA^*)^*(S - BSA^*)),$$

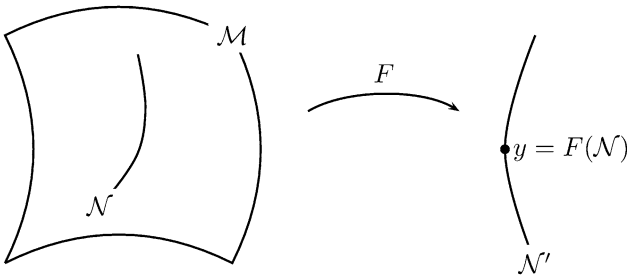
for the problems involving a single matrix S. Again, these are not equivalent in their distance formulation although the corresponding inner product problems are equivalent.

We will develop optimality conditions for those matrix comparison problems, indicate their relations with existing problems from the literature and give an analytic solution for particular matrices A and B.

4.2 Preliminaries on Riemannian Optimization

All those problems are defined on feasible sets that have a *manifold* structure. Roughly speaking, this means that the feasible set is locally smoothly identified with $\mathbb{R}^d$, where d is the dimension of the manifold. Optimization on a manifold generalizes optimization in $\mathbb{R}^d$ while retaining the concept of smoothness. In this section, we recall the essential background on optimization on manifolds, and refer the reader to [4] for details.

A well known and largely used class of manifolds is the class of embedded submanifolds. The submersion theorem gives a useful sufficient condition to prove that a subset of a manifold $\mathcal{M}$ is an embedded submanifold of $\mathcal{M}$. If there exists a smooth mapping $F : \mathcal{M} \to \mathcal{N}'$ between two manifolds of dimension d_m and $d'_n (< d_m)$ and $y \in \mathcal{N}'$ such that the rank of F is equal to d'_n at each point of $\mathcal{N} := F^{-1}(y)$, then $\mathcal{N}$ is a embedded submanifold of $\mathcal{M}$ and the dimension of $\mathcal{N}$ is $d_m - d'_n$.



Example The unitary group $\mathrm{U}(n) = \{Q \in \mathbb{C}^{n\times n} : Q^*Q = I_n\}$ is an embedded submanifold of $\mathbb{C}^{n\times n}$. Indeed, consider the function

$$F : \mathbb{C}^{n\times n} \to \mathcal{S}_{\mathrm{Her}}(n) : Q \mapsto Q^*Q - I_n$$

where $\mathcal{S}_{\mathrm{Her}}(n)$ denotes the set of Hermitian matrices of order n. Clearly, $\mathrm{U}(n) = F^{-1}(0_n)$. It remains to show for all $\hat{H} \in \mathcal{S}_{\mathrm{Her}}(k)$, there exists an $H \in \mathbb{C}^{n\times n}$ such that $\mathrm{D}F(Q) \cdot H = Q^*H + H^*Q = \hat{H}$. It is easy to see that $\mathrm{D}F(Q) \cdot (Q\hat{H}/2) = \hat{H}$, and according to the submersion theorem, it follows that $\mathrm{U}(n)$ is an embedded submanifold of $\mathbb{C}^{n\times n}$. The dimension of $\mathbb{C}^{n\times n}$ and $\mathcal{S}_{\mathrm{Her}}(n)$ are $2n^2$ and n^2 respectively. Hence $\mathrm{U}(n)$ is of dimension n^2.

In our problems, embedding spaces are matrix-Euclidean spaces $\mathbb{C}^{m\times k} \times \mathbb{C}^{n\times k}$ and $\mathbb{C}^{n\times m}$ which have a trivial manifold structure since $\mathbb{C}^{m\times k} \times \mathbb{C}^{n\times k} \simeq \mathbb{R}^{2mnk^2}$ and $\mathbb{C}^{n\times m} \simeq \mathbb{R}^{2mn}$. For each problem, we further analyze whether or not the feasible set is an embedded submanifold of their embedding space.

When working with a function on a manifold $\mathcal{M}$, one may be interested in having a local linear approximation of that function. Let M be an element of $\mathcal{M}$ and $\mathfrak{F}_M(\mathcal{M})$ denote the set of smooth real-valued functions defined on a neighborhood of M.

Definition 1.1 A tangent vector ξ_M to a manifold $\mathcal{M}$ at a point M is a mapping from $\mathfrak{F}_M(\mathcal{M})$ to $\mathbb{R}$ such that there exists a curve γ on $\mathcal{M}$ with $\gamma(0) = M$, satisfying

$$\xi_M f = \left.\frac{\mathrm{d}f(\gamma(t))}{\mathrm{d}t}\right|_{t=0}, \quad \forall f \in \mathfrak{F}_M(\mathcal{M}).$$

Such a curve γ is said to realize the tangent vector ξ_x.

So, the only thing we need to know about a curve γ in order to compute the first-order variation of a real-value function f at $\gamma(0)$ along γ is the tangent vector ξ_x realized by γ. The tangent space to $\mathcal{M}$ at M, denoted by $T_M\mathcal{M}$, is the set of all tangent vectors to $\mathcal{M}$ at M and it admits a structure of vector space over $\mathbb{R}$. When considering an embedded submanifold in a Euclidean space $\mathcal{E}$, any tangent vector ξ_M of the manifold is equivalent to a vector E of the

Euclidean space. Indeed, let $\hat{f}$ be any a differentiable continuous extension of f on $\mathcal{E}$, we have

$$\xi_M f := \left.\frac{\mathrm{d}f(\gamma(t))}{\mathrm{d}t}\right|_{t=0} = \mathrm{D}\hat{f}(M)\cdot E, \tag{4.2}$$

where E is $\dot{\gamma}(0)$ and D is the directional derivative operator

$$\mathrm{D}\hat{f}(M)\cdot E = \lim_{t\to 0}\frac{\hat{f}(M+tE)-\hat{f}(M)}{t}.$$

The tangent space reduces to a linear subspace of the original space $\mathcal{E}$.

Example Let $\gamma(t)$ be a curve on the unitary group $\mathrm{U}(n)$ passing through Q at $t=0$, i.e. $\gamma(t)^*\gamma(t) = I_n$ and $\gamma(0) = Q$. Differentiating with respect to t yields

$$\dot{\gamma}(0)^* Q + Q^*\dot{\gamma}(0) = 0_n.$$

One can see from Eq. 4.2 that the tangent space to $\mathrm{U}(n)$ at Q is contained in

$$\{E\in\mathbb{C}^{n\times n} : E^*Q + Q^*E = 0_n\} = \{Q\Omega\in\mathbb{C}^{n\times n} : \Omega^* + \Omega = 0_n\}. \tag{4.3}$$

Moreover, this set is a vector space over $\mathbb{R}$ of dimension n^2, and hence is the tangent space itself.

Let g_M be an inner product defined on the tangent plane $T_M\mathcal{M}$. The gradient of f at M, denoted $\mathrm{grad}f(M)$, is defined as the unique element of the tangent plane $T_M\mathcal{M}$, that satisfies

$$\xi_M f = g_M(\mathrm{grad}f(M), \xi_M), \quad \forall \xi_M \in T_M\mathcal{M}.$$

The gradient, together with the inner product, fully characterizes the local first order approximation of a smooth function defined on the manifold. In the case of an embedded manifold of a Euclidean space $\mathcal{E}$, since $T_M\mathcal{M}$ is a linear subspace of $T_M\mathcal{E}$, an inner product $\hat{g}_M$ on $T_M\mathcal{E}$ generates by restriction an inner product g_M on $T_M\mathcal{M}$. The orthogonal complement of $T_M\mathcal{M}$ with respect to $\hat{g}_M$ is called the normal space to $\mathcal{M}$ at M and denoted by $(T_M\mathcal{M})^\perp$. The gradient of a smooth function $\hat{f}$, defined on the embedding manifold may be decomposed into its orthogonal projection on the tangent and normal space, respectively

$$\mathrm{P}_M\mathrm{grad}\hat{f}(M) \quad \text{and} \quad \mathrm{P}_M^\perp\mathrm{grad}\hat{f}(M),$$

and it follows that the gradient of f (the restriction of $\hat{f}$ on $\mathcal{M}$) is the projection on the tangent space of the gradient of $\hat{f}$

$$\mathrm{grad}f(M) = \mathrm{P}_M\mathrm{grad}\hat{f}(M).$$

If $T_M\mathcal{M}$ is endowed with an inner product g_M for all $M\in\mathcal{M}$ and g_M varies smoothly with M, the $\mathcal{M}$ is termed a *Riemannian manifold*.

Example Let A and B, two Hermitian matrices. We define

$$\hat{f} : \mathbb{C}^{n\times n} \to \mathbb{R} : Q \mapsto \Re \operatorname{tr}(Q^* AQB),$$

and f its restriction on the unitary group $\mathrm{U}(n)$. We have

$$\mathrm{D}\hat{f}(Q) \cdot E = 2\Re \operatorname{tr}(E^* AQB).$$

We endow the tangent space $T_Q\mathbb{C}^{n\times n}$ with an inner product

$$\hat{g} : T_Q\mathbb{C}^{n\times n} \times T_Q\mathbb{C}^{n\times n} \to \mathbb{R} : E, F \mapsto \Re \operatorname{tr}(E^* F),$$

and the gradient of $\hat{f}$ at Q is then given by $\operatorname{grad}\hat{f}(Q) = 2AQB$. One can further define an orthogonal projection on $T_Q\mathrm{U}(n)$

$$\mathrm{P}_Q E := E - Q\operatorname{Her}(Q^* E),$$

and the gradient of f at Q is given by $\operatorname{grad} f(Q) = \mathrm{P}_Q \operatorname{grad}\hat{f}(Q)$.

Those relations are useful when one wishes to analyze optimization problems, and will hence be further developed for the problems we are interested in.

4.3 The Matrix Comparison Problems and Their Geometry

Below we look at the various problems introduced earlier and focus on the first problem to make these ideas more explicit.

Problem 1 Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, let

$$\hat{f} : \mathbb{C}^{m\times k} \times \mathbb{C}^{n\times k} \to \mathbb{C} : (U, V) \mapsto \hat{f}(U, V) = \operatorname{dist}(U^* AU, V^* BV),$$

find the minimizer of

$$f : \mathrm{St}(k, m) \times \mathrm{St}(k, n) \to \mathbb{C} : (U, V) \mapsto f(U, V) = \hat{f}(U, V),$$

where

$$\mathrm{St}(k, m) = \left\{U \in \mathbb{C}^{m\times k} : U^* U = I_k\right\}$$

denotes the compact Stiefel manifold.

Let $A = (A_1, A_2)$ and $B = (B_1, B_2)$ be pairs of matrices. We define the following useful operations:

- an entrywise product, $A \diamond B = (A_1 B_1, A_2 B_2)$,
- a contraction product, $A \bigstar B = A_1 B_1 + A_2 B_2$, and
- a conjugate-transpose operation, $A^* = (A_1^*, A_2^*)$.

The definitions of the binary operations, $\diamond$ and $\bigstar$, are (for readability) extended to single matrices when one has to deal with pairs of identical matrices. Let, for instance, $A = (A_1, A_2)$ be a pair of matrices and B be a single matrix, we define

$$A \diamond B = (A_1, A_2) \diamond B = (A_1, A_2) \diamond (B, B) = (A_1 B, A_2 B)$$
$$A \bigstar B = (A_1, A_2) \bigstar B = (A_1, A_2) \bigstar (B, B) = A_1 B + A_2 B.$$

The feasible set of Problem 1 is given by the cartesian product of two compact Stiefel manifolds, namely $\mathcal{M} = \mathrm{St}(k, m) \times \mathrm{St}(k, n)$ and is hence a manifold itself (cf. [4]). Moreover, we can prove that $\mathcal{M}$ is an embedded submanifold of

$$\mathcal{E} := \mathbb{C}^{m \times k} \times \mathbb{C}^{n \times k}.$$

Indeed, consider the function

$$F : \mathcal{E} \to \mathcal{S}_{\mathrm{Her}}(k) \times \mathcal{S}_{\mathrm{Her}}(k) : M \mapsto M^* \diamond M - (I_k, I_k)$$

where $\mathcal{S}_{\mathrm{Her}}(k)$ denotes the set of Hermitian matrices of order k. Clearly, $\mathcal{M} = F^{-1}(0_k, 0_k)$. It remains to show that each point $M \in \mathcal{M}$ is a regular value of F which means that F has full rank, i.e. for all $\hat{Z} \in \mathcal{S}_{\mathrm{Her}}(k) \times \mathcal{S}_{\mathrm{Her}}(k)$, there exists $Z \in \mathcal{E}$ such that $\mathrm{D}F(M) \cdot Z = \hat{Z}$. It is easy to see that $\mathrm{D}F(M) \cdot (M \diamond \hat{Z}/2) = \hat{Z}$, and according to the submersion theorem, it follows that $\mathcal{M}$ is an embedded submanifold of $\mathcal{E}$.

The tangent space to $\mathcal{E}$ at a point $M = (U, V) \in \mathcal{E}$ is the embedding space itself (i.e. $T_M\mathcal{E} \simeq \mathcal{E}$), whereas the tangent space to $\mathcal{M}$ at a point $M = (U, V) \in \mathcal{M}$ is given by

$$\begin{aligned} T_M\mathcal{M} &:= \{\dot{\gamma}(0) : \gamma, \text{ differentiable curve on } \mathcal{M} \text{ with } \gamma(0) = M\} \\ &= \{\xi = (\xi_U, \xi_V) : \mathrm{Her}(\xi^* \diamond M) = 0\} \\ &= \left\{ M \diamond \begin{pmatrix} \Omega_U \\ \Omega_V \end{pmatrix} + M_\perp \diamond \begin{pmatrix} K_U \\ K_V \end{pmatrix} : \Omega_U, \Omega_V \in \mathcal{S}_{s-\mathrm{Her}}(k) \right\}, \end{aligned}$$

where $M_\perp = (U_\perp, V_\perp)$ with $U_\perp$ and $V_\perp$ any orthogonal complement of respectively U and V, where $\mathrm{Her}(\cdot)$ stands for

$$\mathrm{Her}(\cdot) : X \mapsto (X + X^*)/2,$$

and where $\mathcal{S}_{s-\mathrm{Her}}(k)$ denotes the set of skew-Hermitian matrices of order k. We endow the tangent space $T_M\mathcal{E}$ with an inner product:

$$\hat{g}_M(\cdot, \cdot) : T_M\mathcal{E} \times T_M\mathcal{E} \to \mathbb{C} : \xi, \zeta \mapsto \hat{g}_M(\xi, \zeta) = \Re \,\mathrm{tr}(\xi^* \bigstar \zeta),$$

and define its restriction on the tangent space $T_M\mathcal{M} (\subset T_M\mathcal{E})$:

$$g_M(\cdot, \cdot) : T_M\mathcal{M} \times T_M\mathcal{M} \to \mathbb{C} : \xi, \zeta \mapsto g_M(\xi, \zeta) = \hat{g}_M(\xi, \zeta).$$

One may now define the normal space to $\mathcal{M}$ at a point $M \in \mathcal{M}$:

$$\begin{aligned} T_M^{\perp}\mathcal{M} :&= \{\xi : \hat{g}_M(\xi, \zeta) = 0, \quad \forall \zeta \in T_M\mathcal{M}\} \\ &= \{M \diamond (H_U, H_V) : H_U, H_V \in S_{\mathrm{Her}}(k)\}, \end{aligned}$$

where $\mathcal{S}_{\mathrm{Her}}(k)$ denotes the set of Hermitian matrices of order k.

Problem 2 Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, let

$$\hat{f} : \mathbb{C}^{n\times m} \to \mathbb{C} : X \mapsto \hat{f}(X) = \mathrm{dist}(XA, BX)$$

find the minimizer of

$$f : \mathcal{M} \to \mathbb{C} : X \mapsto f(X) = \hat{f}(X),$$

where $\mathcal{M} = \{VU^* \in \mathbb{C}^{n\times m} : (U, V) \in \mathrm{St}(k, m) \times \mathrm{St}(k, n)\}$.

$\mathcal{M}$ is a smooth and connected manifold. Indeed, let $\Sigma := \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}$ be an element of $\mathcal{M}$. Every $X \in \mathcal{M}$ is congruent to Σ by the congruence action $((\tilde{U}, \tilde{V}), X) \mapsto \tilde{V}^* X \tilde{U}$, $(\tilde{U}, \tilde{V}) \in \mathrm{U}(m) \times \mathrm{U}(n)$, where $\mathrm{U}(n) = \{U \in \mathbb{C}^{n\times n} : U^*U = I_n\}$ denotes the unitary group of degree n. The set $\mathcal{M}$ is an orbit of this smooth complex algebraic Lie group action of $\mathrm{U}(m) \times \mathrm{U}(n)$ on $\mathbb{C}^{n\times m}$ and therefore a smooth manifold [5, App. C]. $\mathcal{M}$ is the image of the connected subset $\mathrm{U}(m) \times \mathrm{U}(n)$ of the continuous (and in fact smooth) map $\pi : \mathrm{U}(m) \times \mathrm{U}(n) \to \mathbb{C}^{n\times m}, \pi(\tilde{U}, \tilde{V}) = \tilde{V}^* X \tilde{U}$, and hence is also connected.

The tangent space to $\mathcal{M}$ at a point $X = VU^* \in \mathcal{M}$ is

$$\begin{aligned} T_M\mathcal{M} &:= \{\dot{\gamma}(0) : \gamma \text{ curve on } \mathcal{M} \text{ with } \gamma(0) = X\} \\ &= \left\{\xi_v U^* + V\xi_U^* : \mathrm{Her}(V^*\xi_V) = \mathrm{Her}(U^*\xi_U) = 0_k\right\} \\ &= \left\{V\Omega U^* + VK_U^* U_\perp^* + V_\perp K_V U^* : \Omega \in \mathcal{S}_{s-\mathrm{Her}}(k)\right\}. \end{aligned}$$

We endow the tangent space $T_X\mathbb{C}^{n\times m} \simeq \mathbb{C}^{n\times m}$ with an inner product:

$$\hat{g}_X(\cdot, \cdot) : T_X\mathbb{C}^{n\times m} \times T_X\mathbb{C}^{n\times m} \mapsto \mathbb{C} : \xi, \zeta \to \hat{g}_X(\xi, \zeta) = \Re\,\mathrm{tr}(\xi^*\zeta),$$

and define its restriction on the tangent space $T_X\mathcal{M} (\subset T_X\mathcal{E})$:

$$g_X(\cdot, \cdot) : T_X\mathcal{M} \times T_X\mathcal{M} \mapsto \mathbb{C} : \xi, \zeta \to g_X(\xi, \zeta) = \hat{g}_X(\xi, \zeta).$$

One may now define the normal space to $\mathcal{M}$ at a point $X \in \mathcal{M}$:

$$\begin{aligned} T_X^{\perp}\mathcal{M} &:= \{\xi : \hat{g}_X(\xi, \zeta) = 0, \forall \zeta \in T_X\mathcal{M}\} \\ &= \left\{VHU^* + V_\perp KU_\perp^* : H \in \mathcal{S}_{\mathrm{Her}}(k)\right\}. \end{aligned}$$

Problem 3 Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, let

$$\hat{f} : \mathbb{C}^{n\times m} \to \mathbb{C} : X \mapsto \hat{f}(X) = \mathrm{dist}(X, BXA^*)$$

find the minimizer of

$$f : \mathcal{M} \to \mathbb{C} : X \mapsto f(X) = \hat{f}(X),$$

where $\mathcal{M} = \{VU^* \in \mathbb{C}^{n\times m} : (U, V) \in \mathrm{St}(k, m) \times \mathrm{St}(k, n)\}$.

Since they have the same feasible set, developments obtained for Problem 2 hold also for Problem 3.

Problem 4 Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, let

$$\hat{f} : \mathbb{C}^{n\times m} \to \mathbb{C} : S \mapsto \hat{f}(S) = \mathrm{dist}(SA, BS)$$

find the minimizer of

$$f : \mathcal{M} \to \mathbb{C} : X \mapsto f(X) = \hat{f}(X),$$

where $\mathcal{M} = \left\{S \in \mathbb{C}^{n\times m} : \|S\|_F = 1\right\}$.

The tangent space to $\mathcal{M}$ at a point $S \in \mathcal{M}$ is

$$T_S\mathcal{M} = \{\xi : \Re\,\mathrm{tr}(\xi^* S) = 0\}.$$

We endow the tangent space $T_S\mathbb{C}^{n\times m} \simeq \mathbb{C}^{n\times m}$ with an inner product:

$$\hat{g}_S(\cdot, \cdot) : T_S\mathbb{C}^{n\times m} \times T_S\mathbb{C}^{n\times m} \mapsto \mathbb{C} : \xi, \zeta \to \hat{g}_S(\xi, \zeta) = \Re\,\mathrm{tr}(\xi^*\zeta),$$

and define its restriction on the tangent space $T_S\mathcal{M}(\subset T_S\mathcal{E})$:

$$g_S(\cdot, \cdot) : T_S\mathcal{M} \times T_S\mathcal{M} \mapsto \mathbb{C} : \xi, \zeta \to g_S(\xi, \zeta) = \hat{g}_S(\xi, \zeta).$$

One may now define the normal space to $\mathcal{M}$ at a point $S \in \mathcal{M}$:

$$T_S^{\perp}\mathcal{M} := \{\xi : \hat{g}_S(\xi, \zeta) = 0,\ \forall \zeta \in T_S\mathcal{M}\} = \{\alpha S : \alpha \in \mathbb{R}\}$$

Problem 5 Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, let

$$\hat{f} : \mathbb{C}^{n\times m} \to \mathbb{C} : S \mapsto \hat{f}(S) = \mathrm{dist}(S, BSA^*)$$

find the minimizer of

$$f : \mathcal{M} \to \mathbb{C} : X \mapsto f(X) = \hat{f}(X),$$

where $\mathcal{M} = \left\{S \in \mathbb{C}^{n\times m} : \|S\|_F = 1\right\}$.

Since they have the same feasible set, developments obtained for Problem 4 also hold for Problem 5.

4.4 Optimality Conditions

Our problems are optimization problems of smooth functions defined on a compact domain $\mathcal{M}$, and therefore there always exists an optimal solution $M \in \mathcal{M}$ where the first order optimality condition is satisfied,

$$\operatorname{grad} f(M) = 0. \tag{4.4}$$

We study the stationary points of Problem 1 in detail, and we show how the other problems can be tackled.

Problem 1 We first analyze this optimality condition for Problem 1. For any $(W, Z) \in T_M\mathcal{E}$, we have

$$\begin{aligned} \mathrm{D}\hat{f}(U,V)\cdot(W,Z) &= 2\Re\mathrm{tr}\begin{pmatrix} (W^*AU + U^*AW - Z^*BV - V^*BZ)^* \\ (U^*AU - V^*BV) \end{pmatrix} \\ &= \hat{g}_{(U,V)}\left(2\begin{pmatrix} AU\Delta_{AB}^* + A^*U\Delta_{AB} \\ BV\Delta_{BA}^* + B^*V\Delta_{BA} \end{pmatrix}, (W,Z)\right), \end{aligned} \tag{4.5}$$

with $\Delta_{AB} := U^*AU - V^*BV =: -\Delta_{BA}$, and hence the gradient of $\hat{f}$ at a point $(U, V) \in \mathcal{E}$ is

$$\operatorname{grad}\hat{f}(U,V) = 2\begin{pmatrix} AU\Delta_{AB}^* + A^*U\Delta_{AB} \\ BV\Delta_{BA}^* + B^*V\Delta_{BA} \end{pmatrix}. \tag{4.6}$$

Since the normal space $T_M^\perp\mathcal{M}$ is the orthogonal complement of the tangent space $T_M\mathcal{M}$, one can, for any $M \in \mathcal{M}$, decompose any $E \in \mathcal{E}$ into its orthogonal projections on $T_M\mathcal{M}$ and $T_M^\perp\mathcal{M}$:

$$\mathrm{P}_M E := E - \mathrm{P}_M^\perp E \quad \text{and} \quad \mathrm{P}_M^\perp E := M \diamond \mathrm{Her}(M^* \diamond E). \tag{4.7}$$

The gradient of f at a point $(U, V) \in \mathcal{M}$ is

$$\operatorname{grad} f(U,V) = \mathrm{P}_M \operatorname{grad}\hat{f}(M). \tag{4.8}$$

For our problem, the first order optimality condition (4.4) yields, by means of (4.6), (4.7) and (4.8)

$$\begin{pmatrix} AU\Delta_{AB}^* + A^*U\Delta_{AB} \\ BV\Delta_{BA}^* + B^*V\Delta_{BA} \end{pmatrix} = \begin{pmatrix} U \\ V \end{pmatrix} \diamond \mathrm{Her}\begin{pmatrix} U^*AU\Delta_{AB}^* + U^*A^*U\Delta_{AB} \\ V^*BV\Delta_{BA}^* + V^*B^*V\Delta_{BA} \end{pmatrix}. \tag{4.9}$$

Observe that f is constant on the equivalence classes

$$[(U,V)] = \{(U,V) \diamond Q : Q \in \mathrm{U}(k)\},$$

and that any point of $[(U, V)]$ is a stationary point of f whenever (U, V) is.

We consider the special case where U^*AU and V^*BV are simultaneously diagonalizable by a unitary matrix at all stationary points (U, V), i.e. eigendecomposition of U^*AU and V^*BV are respectively WD_AW^* and WD_BW^*, with $W \in \mathrm{U}(k)$ and $D_A = \operatorname{diag}\left(\theta_1^A, \ldots, \theta_k^A\right)$, $D_B = \operatorname{diag}\left(\theta_1^B, \ldots, \theta_k^B\right)$. This happens when A and B are both Hermitian. Indeed, in that case, applying $\begin{pmatrix} U^* \\ V^* \end{pmatrix} \diamond$ on the lest of (4.9) yields

$$U^*AU\,V^*BV = V^*BV\,U^*AU,$$

which implies that U^*AU and V^*BV have the same eigenvectors. The cost function at stationary points simply reduces to $\sum_{i=1}^{k}\left|\theta_i^A - \theta_i^B\right|^2$ and the minimization problem roughly consists in finding the isometric projections U^*AU, V^*BV such that their eigenvalues are pairwise as near as possible.

More precisely, the first optimality condition becomes

$$\left[\begin{pmatrix} A \\ B \end{pmatrix} \diamond \begin{pmatrix} U \\ V \end{pmatrix} \diamond W - \begin{pmatrix} U \\ V \end{pmatrix} \diamond W \diamond \begin{pmatrix} D_A \\ D_B \end{pmatrix}\right] \diamond \begin{pmatrix} D_A - D_B \\ D_B - D_A \end{pmatrix} = 0, \qquad (4.10)$$

that is,

$$\left[\begin{pmatrix} A \\ B \end{pmatrix} \diamond \begin{pmatrix} \bar{U}_i \\ \bar{V}_i \end{pmatrix} - \begin{pmatrix} \bar{U}_i \\ \bar{V}_i \end{pmatrix} \diamond \begin{pmatrix} \theta_i^A \\ \theta_i^B \end{pmatrix}\right] \diamond \begin{pmatrix} \theta_i^A - \theta_i^B \\ \theta_i^B - \theta_i^A \end{pmatrix} = 0, \quad i = 1, \ldots, k \qquad (4.11)$$

where $\bar{U}_i$ and $\bar{V}_i$ denotes the ith column of $\bar{U} = UW$ and $\bar{V} = VW$ respectively. This implies that for all $i = 1, \ldots, k$ either $\theta_i^A = \theta_i^B$ or $\left(\theta_i^A, \bar{U}_i\right)$ and $\left(\theta_i^B, \bar{V}_i\right)$ are eigenpairs of respectively A and B.

Definition 1.2 Let A and A_U be two square matrices respectively of order m and k, where $m \geq k$. A_U is said to be imbeddable in A if there exists a matrix $U \in \mathrm{St}(k, m)$ such that $U^*AU = A_U$.

For Hermitian matrices, [6] gives us the following result.

Theorem 1.3 *Let A and A_U be Hermitian matrices and $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_m$ and $\theta_1^A \leq \theta_2^A \leq \cdots \leq \theta_k^A$ their respective eigenvalues. Then a necessary and sufficient condition for A_U to be imbeddable in A is that*

$$\theta_i^A \in [\alpha_i, \alpha_{i-k+m}], \quad i = 1, \ldots, k.$$

The necessity part of this theorem is well known as the *Cauchy interlacing theorem* [7, p. 202]. The sufficiency part is less easy to prove cf. [6, 8].

Definition 1.4 For S_1 and S_2, two non-empty subsets of a metric space together with the distance d, we define

$$e_d(S_1, S_2) = \inf_{\substack{s_1 \in S_1 \\ s_2 \in S_2}} d(s_1, s_2)$$

When considering two non-empty subsets $[\alpha_1, \alpha_2]$ and $[\beta_1, \beta_2]$ of $\mathbb{R}$, one can easily see that

$$e_d([\alpha_1, \alpha_2], [\beta_1, \beta_2]) = \max(0, \alpha_1 - \beta_2, \beta_1 - \alpha_2).$$

Theorem 1.5 *Let $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_m$ and $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_n$ be the eigenvalues of Hermitian matrices respectively A and B. The solution of Problem 1 is*

$$\sum_{i=1}^{k} \left(e_d\left([\alpha_i, \alpha_{i-k+m}], [\beta_i, \beta_{i-k+n}]\right)\right)^2$$

with d the Euclidean norm.

Proof Recall that when A and B are Hermitian matrices, U^*AU and V^*BV are jointly diagonalizable by a unitary matrix at all stationary points (U, V). Since the distance is invariant by joint unitary transformation, the value of the minimum is not affected if we restrict (U, V) to be such that U^*AU and V^*BV are diagonal. Problem 1 reduces to minimize

$$f(U, V) = \sum_{i=1}^{k} \left|\theta_i^A - \theta_i^B\right|^2,$$

where $U^*AU = \mathrm{diag}(\theta_1^A, \ldots, \theta_k^A)$ and $V^*BV = \mathrm{diag}(\theta_1^B, \ldots, \theta_k^B)$. It follows from Theorem 1.3 that the minimum of Problem 1 is

$$\min_{\theta_i^A, \theta_i^B} \min_{\pi} \sum_{i=1}^{k} \left(\theta_{\pi(i)}^A - \theta_i^B\right)^2 \tag{4.12}$$

such that

$$\theta_1^A \leq \theta_2^A \leq \cdots \leq \theta_k^A, \quad \theta_1^B \leq \theta_2^B \leq \cdots \leq \theta_k^B, \tag{4.13}$$

$$\theta_i^A \in [\alpha_i, \alpha_{i-k+m}], \quad \theta_i^B \in \left[\beta_i, \beta_{i-k+n}\right], \tag{4.14}$$

and $\pi(\cdot)$ is a permutation of $1, \ldots, k$.

Let $\theta_1^A, \ldots, \theta_k^A$, and $\theta_1^B, \ldots, \theta_k^B$ satisfy (4.13). Then, the identity permutation $\pi(i) = i$ is optimal for problem (4.12). Indeed, if π is not the identity, then there exists i and j such that $i < j$ and $\pi(i) > \pi(j)$, and we have

$$\begin{aligned}
&\left(\theta_i^A - \theta_{\pi(i)}^B\right)^2 + \left(\theta_j^A - \theta_{\pi(j)}^B\right)^2 - \left[\left(\theta_j^A - \theta_{\pi(i)}^B\right)^2 + \left(\theta_i^A - \theta_{\pi(j)}^B\right)^2\right] \\
&\quad = 2\left(\theta_j^A - \theta i^A\right)\left(\theta_{\pi(i)}^B - \theta_{\pi(j)}^B\right) \leq 0.
\end{aligned}$$

Since the identity permutation is optimal, our minimization problem simply reduces to

$$\sum_{i=1}^{k} \min_{(4.13)(4.14)} \left(\theta_i^A - \theta_i^B\right)^2.$$

We now show that (4.13) can be relaxed. Indeed, assume there is an optimal solution that does not satisfy the ordering condition, i.e. there exist i and j, $i < j$ such that $\theta_j^A \leq \theta_i^A$. One can see that the following inequalities hold

$$\alpha_i \leq \alpha_j \leq \theta_j^A \leq \theta_i^A \leq \alpha_{i-k+m} \leq \alpha_{j-k+m}.$$

Since θ_i^A belongs to $[\alpha_j, \alpha_{j-k+m}]$ and θ_j^A belongs to $[\alpha_i, \alpha_{i-k+m}]$, one can switch i and j and build an ordered solution that does not change the cost function and hence remains optimal.

It follows that $\sum_{i=1}^{k} \min_{(4.13)(4.14)} \left(\theta_i^A - \theta_i^B\right)^2$ is equal to $\sum_{i=1}^{k} \min_{(4.14)} \left(\theta_i^A - \theta_i^B\right)^2$. This result is precisely what we were looking for. □

Figure 4.1 gives an example of optimal matching.

Problem 2 For all $Y \in T_X\mathbb{C}^{n\times m} \simeq \mathbb{C}^{n\times m}$, we have

$$\mathrm{D}\hat{f}(X) \cdot Y = 2\Re\,\mathrm{tr}(Y^*\,(XAA^* - B^*XA - BXA^* + B^*BX)), \tag{4.15}$$

and hence the gradient of $\hat{f}$ at a point $X \in \mathbb{C}^{n\times m}$ is

$$\mathrm{grad}\hat{f}(X) = 2(XAA^* - B^*XA - BXA^* + B^*BX). \tag{4.16}$$

Since the normal space $T_X^{\perp}\mathcal{M}$ is the orthogonal complement of the tangent space $T_X\mathcal{M}$, one can, for any $X = VU^* \in \mathcal{M}$, decompose any $E \in \mathbb{C}^{n\times m}$ into its orthogonal projections on $T_X\mathcal{M}$ and $T_X^{\perp}\mathcal{M}$:

$$\begin{aligned} \mathrm{P}_X E &= E - V\,\mathrm{Her}(V^*EU)U^* - (I_n - VV^*)E(I_m - UU^*), \text{and} \\ \mathrm{P}_X^{\perp} E &= V\,\mathrm{Her}(V^*EU)U^* + (I_n - VV^*)E(I_m - UU^*). \end{aligned} \tag{4.17}$$

For any $Y \in T_X\mathcal{M}$, (4.15) hence yields

$$\mathrm{D}\hat{f}(X) \cdot Y = \mathrm{D}f(X) \cdot Y = g_X\left(\mathrm{P}_X \mathrm{grad}\hat{f}(X), Y\right),$$

and the gradient of f at a point $X = VU^* \in \mathcal{M}$ is

$$\mathrm{grad}f(X) = \mathrm{P}_X \mathrm{grad}\hat{f}(X). \tag{4.18}$$

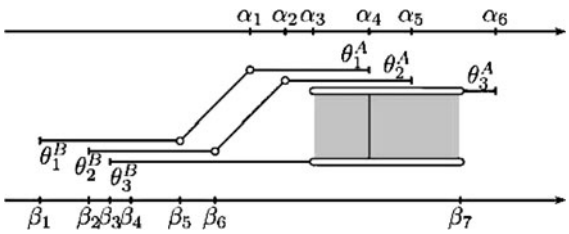


Fig. 4.1 Let the α_i and β_i be the eigenvalues of the Hermitian matrices A and B, and $k = 3$. Problem 1 is then equivalent to $\sum_{i=1}^{3} \min_{\theta_i^A, \theta_i^B} \left(\theta_i^A - \theta_i^B\right)^2$ such that $\theta_i^A \in [\alpha_i, \alpha_{i+3}]$, $\theta_i^B \in [\beta_i, \beta_{i+4}]$. The two first terms of this sum have strictly positive contributions whereas the third one can be reduced to zero within a continuous set of values for θ_3^A and θ_3^B in $[\alpha_3, \beta_7]$

Problem 3 This problem is very similar to Problem 2. We have

$$\mathrm{D}\hat{f}(X) \cdot Y = 2\Re\,\mathrm{tr}(Y^*(X - B^*XA - BXA^* + B^*BXA^*A)), \tag{4.19}$$

for all $Y \in T_X\mathbb{C}^{n\times m} \simeq \mathbb{C}^{n\times m}$, and hence the gradient of $\hat{f}$ at a point $X \in \mathbb{C}^{n\times m}$ is

$$\mathrm{grad}\hat{f}(X) = 2(X - B^*XA - BXA^* + B^*BXA^*A).$$

The feasible set is the same as in Problem 2. Hence the orthogonal decomposition (4.17) holds, and the gradient of f at a point $X = VU^* \in \mathcal{M}$ is

$$\mathrm{grad}f(X) = \mathrm{P}_X\mathrm{grad}\hat{f}(X).$$

Problem 4 For all $T \in T_S\mathbb{C}^{n\times m} \simeq \mathbb{C}^{n\times m}$, we have

$$\mathrm{D}\hat{f}(S) \cdot T = 2\Re\mathrm{tr}(T^*(SAA^* - B^*SA - BSA^* + B^*BS)), \tag{4.20}$$

and hence the gradient of $\hat{f}$ at a point $S \in \mathbb{C}^{n\times m}$ is

$$\mathrm{grad}\hat{f}(S) = 2(SAA^* - B^*SA - BSA^* + B^*BS). \tag{4.21}$$

Since the normal space, $T_S^{\perp}\mathcal{M}$, is the orthogonal complement of the tangent space, $T_S\mathcal{M}$, one can, for any $S \in \mathcal{M}$, decompose any $E \in \mathbb{C}^{n\times m}$ into its orthogonal projections on $T_S\mathcal{M}$ and $T_S^{\perp}\mathcal{M}$:

$$\mathrm{P}_S E = E - S\,\Re\,\mathrm{tr}(S^*E) \quad \text{and} \quad \mathrm{P}_S^{\perp}E = S\Re\,\mathrm{tr}(S^*E). \tag{4.22}$$

For any $T \in T_S\mathcal{M}$, (4.20) then yields

$$\mathrm{D}\hat{f}(S) \cdot T = \mathrm{D}f(S) \cdot T = g_S\big(\mathrm{P}_S\mathrm{grad}\hat{f}(S), T\big),$$

and the gradient of f at a point $S \in \mathcal{M}$ is $\mathrm{grad}f(S) = \mathrm{P}_S\mathrm{grad}\hat{f}(S)$.

For our problem, (4.4) yields, by means of (4.21) and (4.22)

$$\lambda S = (SA - BS)A^* - B^*(SA - BS)$$

where $\lambda = \mathrm{tr}((SA - BS)^*(SA - BS)) \equiv \hat{f}(S)$. Its equivalent vectorized form is

$$\lambda\mathrm{vec}(S) = \big(A^T \otimes I - I \otimes B\big)^*\big(A^T \otimes I - I \otimes B\big)\mathrm{vec}(S).$$

Hence, the stationary points of Problem 4 are given by the eigenvectors of $(A^T \otimes I - I \otimes B)^*(A^T \otimes I - I \otimes B)$. The cost function f simply reduces to the corresponding eigenvalue and the minimal cost is then the smallest eigenvalue.

Problem 5 This problem is very similar to Problem 4. A similar approach yields

$$\lambda S = (S - BSA^*) - B^*(S - BSA^*)A$$

where $\lambda = \mathrm{tr}((S - BSA^*)^*(S - BSA^*))$. Its equivalent vectorized form is

$$\lambda \mathrm{vec}(S) = (I \otimes I - \bar{A} \otimes B)^* (I \otimes I - \bar{A} \otimes B)\mathrm{vec}(S),$$

where $\bar{A}$ denotes the complex conjugate of A.

Hence, the stationary points of Problem 5 are given by the eigenvectors of $(I \otimes I - \bar{A} \otimes B)^* (I \otimes I - \bar{A} \otimes B)$, and the cost function f again simply reduces to the corresponding eigenvalue and the minimal cost is then the smallest eigenvalue.

4.5 Iterative Methods

The solutions of the problems mentioned above may not have a closed form expression or may be very expensive to compute. Iterative optimization methods build a sequence of iterates that hopefully converges as fast as possible towards the optimal solution and eventually gives an estimate of it.

The complexity of classical algorithms can significantly increase according to the number of variables to deal with. An interesting approach when one works on a feasible set that has a manifold structure amounts to use classical methods defined on Euclidean spaces and apply them to the tangent space to the manifold (see [4]). The link between the tangent space and the manifold is naturally done using a so called *retraction* mapping. Let M be a point of a manifold $\mathcal{M}$, the retraction R_M is a smooth function that maps the tangent vector $\xi_M \in T_M\mathcal{M}$ to a point on $\mathcal{M}$ such that

- 0_M (the zero element of $T_M\mathcal{M}$) is mapped onto M, and
- there is no *distortion* around the origin, which means that

$$DR_M(0_M) = \mathrm{id}_{T_M\mathcal{M}}$$

where $\mathrm{id}_{T_M\mathcal{M}}$ denotes the identity mapping on $T_M\mathcal{M}$.

Given a cost function f defined on a manifold $\mathcal{M}$, one can build a *pullback* cost function $\hat{f}_M = f \circ R_M$ on the vector space $T_M\mathcal{M}$. One can now easily generalize methods defined on Euclidean space. A well known class of iterative methods are line-search methods. In $\mathbb{R}^d$, choose a starting point x_0 and proceed through the following iteration

$$x_{k+1} = x_k + t_k z_k,$$

where z_k is a suitable *search direction* and t_k a scalar called the *step size*. This iteration can be generalized as follows on manifolds:

$$M_{k+1} = R_{M_k}(t_k \xi_k),$$

where ξ_k is a tangent vector. For minimization problems, one may choose the opposite of the gradient as search direction

$$\xi_k = -\mathrm{grad} f(M_k).$$

This particular case is known as the *steepest descent* method. The step size can further be set using the so-called *Armijo Back-Tracking* scheme, for example.

Some iterative methods, like Newton's method, use higher-order derivatives of f. We further plan to investigate several of these methods in order to solve the problems mentioned above.

4.6 Relation to the Crawford Number

The field of values of a square matrix A is defined as the set of complex numbers

$$\mathcal{F}(A) := \{x^*Ax : x^*x = 1\},$$

and is known to be a closed convex set [9]. The Crawford number is defined as the distance from that compact set to the origin

$$Cr(A) := \min\{|\lambda| : \lambda \in \mathcal{F}(A)\},$$

and can be computed e.g. with techniques described in [9]. One could define the *generalized Crawford number* of two matrices A and B as the distance between $\mathcal{F}(A)$ and $\mathcal{F}(B)$, i.e.

$$Cr(A, B) := \min\{|\lambda - \mu| : \lambda \in \mathcal{F}(A), \mu \in \mathcal{F}(B)\}.$$

Clearly, $Cr(A, 0) = Cr(A)$ which thus generalizes the concept. Moreover this is a special case of our problem since

$$Cr(A, B) = \min_{U^*U=V^*V=1} \|U^*AU - V^*BV\|.$$

One can say that Problem 1 is a k-dimensional extension of this problem.

Acknowledgments We thank the reviewers for their useful remarks and for mentioning reference [6].

References

1. Fraikin C, Nesterov Y, Van Dooren P (2007) Optimizing the coupling between two isometric projections of matrices. SIAM J Matrix Anal Appl 30(1):324–345
2. Blondel VD, Gajardo A, Heymans M, Senellart P, Van Dooren P (2004) A measure of similarity between graph vertices: applications to synonym extraction and web searching. SIAM Rev 46(4):647–666
3. Fraikin C, Van Dooren P (2007) Graph matching with type constraints on nodes and edges. In: Frommer A, Mahoney MW, Szyld DB (eds) Web information retrieval and linear algebra algorithms, number 07071 in Dagstuhl seminar proceedings, Dagstuhl, Germany, 2007. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI).
4. Absil P-A, Mahony R, Sepulchre R (2008) Optimization algorithms on matrix manifolds. Princeton University Press, Princeton

5. Helmke U, Moore JB (1994) Optimization and dynamical systems. Communications and control engineering series. Springer-Verlag London Ltd, London. With a foreword by R. Brockett
6. Fan K, Pall G (1957) Imbedding conditions for Hermitian and normal matrices. Canad J Math 9:298–304
7. Parlett BN (1980) The symmetric eigenvalue problem. Prentice-Hall Inc, Englewood Cliffs. Prentice-Hall series in computational mathematics
8. Parlett B, Strang G (2008) Matrices with prescribed Ritz values. Linear Algebra Appl 428:1725–1739
9. Horn R, Johnson CR (1991) Topics in matrix analysis. Cambridge University Press, New York

Chapter 5
A Framelet-Based Algorithm for Video Enhancement

Raymond H. Chan, Yiqiu Dong and Zexi Wang

Abstract Video clips are made up of many still frames. Most of the times, the frames are small perturbations of their neighboring frames. Recently, we proposed a framelet-based algorithm to enhance the resolution of any frames in a video clip by solving it as a super-resolution image reconstruction problem. In this paper, we extend the algorithm to video enhancement, where we compose a high-resolution video from a low-resolution one. An experimental result of our algorithm on a real video clip is given to illustrate the performance.

5.1 Introduction

High-resolution images are useful in remote sensing, surveillance, military imaging, and medical imaging, see for instances [4, 12, 14]. However, high-resolution images are more expensive to obtain compared to low-resolution ones which can be obtained from an array of inexpensive low-resolution sensors.

R. H. Chan (✉)
Department of Mathematics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
e-mail: rchan@math.cuhk.edu.hk

Y. Dong
Institute of Mathematics and Scientific Computing, University of Graz, Heinrichstrasse 36, 8010 Graz, Austria
e-mail: yiqiu.dong@uni-graz.at

Z. Wang
Department of Finance and Management Science, Norwegian School of Economics and Business Administration, Helleveien 30, 5045 Bergen, Norway
e-mail: zexi.wang@nhh.no

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_5,

Therefore, there has been much interest in reconstructing high-resolution images from low-resolution ones that are small perturbation of each other, see [9, 11, 13, 15, 17, 20]. One approach is based on the maximum likelihood technique using the expectation maximization algorithm to seek the high resolution image [3]. Another approach is the regularization method in [10] which was based on the total squared error between the observed low-resolution images and the predicted low-resolution images. The predicted images are the results of projecting the high-resolution image estimate through the observation model. The framelet algorithm proposed in [5, 6] is different from these methods. It applies the unitary extension principle in [16] to form a system of tight frame filters. In this approach, there is only matrix-vector multiplication, and no need for solving a minimization problem. Recently, it was shown that this framelet algorithm, which is an iterative algorithm, indeed converges to a minimizer of a variational problem [2].

Video clips are made up of many still frames (about 25–30 frames per second), and the scene usually does not change much from one frame to the next. Thus given a reference frame, its nearby frames can be considered as its small perturbations, and we can make use of them to get a high-resolution image of the reference frame. More precisely, consider a sequence of frames $\{f_k\}_{k=-K}^{K}$ in a video clip, where k increases with the time when the frame f_k is taken. Let f_0 be the reference frame we want to enhance its resolution. The frames $\{f_k\}_{k\neq 0}$ can be considered as small spatial perturbations of f_0. Then, we can use the framelet algorithm proposed in [5] to improve the resolution of f_0. Such a still enhancement algorithm is given in [7].

The goal of this paper is to extend the algorithm in [7] to video enhancement, where high-resolution video streams are constructed from low-resolution ones. The paper is organized as follows. In Sect. 5.2, we introduce the framelet algorithm for the high-resolution image reconstruction given in [5]. In Sect. 5.3, we describe the algorithm proposed in [7] for enhancing video stills. Then in Sect. 5.4, we extend it to video enhancement and apply the resulting algorithm on a real video to enhance the video resolution. Conclusion is given in Sect. 5.5.

In this paper, we use bold-face characters to indicate vectors. If f represents an image $f(x, y)$, $\mathbf{f}$ represents the column vector constructed by raster scanning of f row by row.

5.2 High-Resolution Image Reconstruction

5.2.1 The Model

Here we briefly recall the high-resolution image reconstruction model introduced in [1]. For more details, please refer to the paper. Let h be a piecewise continuous function measuring the intensity of a scene. An image of h at sampling resolution T can be modeled by the integral

$$h(n_1,n_2) \equiv \frac{1}{T^2} \int_{(n_2-\frac{1}{2})T}^{(n_2+\frac{1}{2})T} \int_{(n_1-\frac{1}{2})T}^{(n_1+\frac{1}{2})T} h(x,y)dxdy, \quad n_1,n_2 \in \mathbb{Z}. \tag{5.1}$$

Here (n_1,n_2) are the pixel locations. High-resolution image reconstruction refers to the construction of an image with sampling resolution T by using K^2 low-resolution images of sampling resolution KT, where K is a positive integer. In this paper, we only consider $K = 2$. Larger value of K can be considered similarly, but with more complicated notations.

When $K = 2$, we are given four low-resolution images, $g_{0,0}, g_{0,1}, g_{1,0}, g_{1,1}$ of sampling resolution $2T$, sampling at

$$g_{i,j}(n_1',n_2') = \frac{1}{4T^2} \int_{(2(n_2'-\frac{1}{2})+j)T}^{(2(n_2'+\frac{1}{2})+j)T} \int_{(2(n_1'-\frac{1}{2})+i)T}^{(2(n_1'+\frac{1}{2})+i)T} h(x,y)dxdy, \tag{5.2}$$

where $i, j = 0, 1$. The locations $(0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$ are the sensor positions.

A straightforward way to form an image g of sampling resolution T is to interlace the four low-resolution images, i.e.

$$g(n_1,n_2) = g_{i,j}(n_1',n_2'), \tag{5.3}$$

where $i = n_1 \bmod 2$, $j = n_2 \bmod 2$, $n_1' = \lfloor n_1/2 \rfloor, n_2' = \lfloor n_2/2 \rfloor$, see Fig. 5.1. The function g is called the *observed high-resolution image*.

Note that g isnot equal to the desired image h in (5.1) but is a good approximation of it. If we assume $h(x,y)$ is constant in the sampling region

$$[(n_1-0.5)T,(n_1+0.5)T) \times [(n_2-0.5)T,(n_2+0.5)T),$$

for all $n_1,n_2 \in \mathbb{Z}$, (i.e. $h(x,y) \equiv h(n_1,n_2)$ there), then by (5.1)–(5.3), we can easily prove that

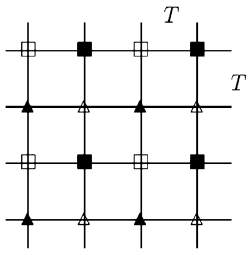


Fig. 5.1 The model of obtaining g by interlacing pixels from $g_{i,j}$ (Open square $g_{0,0}$ pixels, filled square $g_{0,1}$ pixels, filled triangle $g_{1,0}$ pixels, Open triangle $g_{1,1}$ pixels)

$$g(n_1,n_2) = \frac{1}{4}\Big[\frac{1}{4}h(n_1-1,n_2-1) + \frac{1}{2}h(n_1-1,n_2) + \frac{1}{4}h(n_1-1,n_2+1) + \frac{1}{2}h(n_1,n_2-1) + h(n_1,n_2) + \frac{1}{2}h(n_1,n_2+1) + \frac{1}{4}h(n_1+1,n_2-1) + \frac{1}{2}h(n_1+1,n_2) + \frac{1}{4}h(n_1+1,n_2+1)\Big].$$

In matrix form, it is

$$\mathbf{g} = H_{0,0}\mathbf{h}, \tag{5.4}$$

where $H_{0,0} = H_0 \otimes H_0$ with H_0 being the matrix representation of the discrete convolution (i.e. Toeplitz form) with kernel $h_0 = [1/4, 1/2, 1/4]$.

To obtain a better high-resolution image than $\mathbf{g}$, one will have to solve $\mathbf{h}$ from (5.4). It is an ill-posed problem where many methods are available. One approach is the framelet method in [5] that we are going to describe next.

5.2.2 Framelet-Based HR Image Reconstruction

Here we briefly recall the algorithm in [5] and refer the reader to the paper for more details. The convolution kernel h_0 is a low-pass filter. By applying the unitary extension principle in [16], h_0 together with the following high-pass filters form a tight framelet system:

$$h_1 = \left[\frac{\sqrt{2}}{4}, 0, -\frac{\sqrt{2}}{4}\right], \quad h_2 = \left[-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}\right]. \tag{5.5}$$

Define

$$H_{i,j} = H_i \otimes H_j, \quad 0 \le i, \quad j \le 2,$$

where H_i is the discrete convolution matrix with kernel h_i. The perfect reconstruction formula for the framelet systems gives

$$\mathbf{h} = \sum_{i,j=0}^{2} H_{i,j}^{*} H_{i,j}\mathbf{h},$$

see [5]. Based on (5.4) and substituting $H_{0,0}\mathbf{h}$ by $\mathbf{g}$, we have

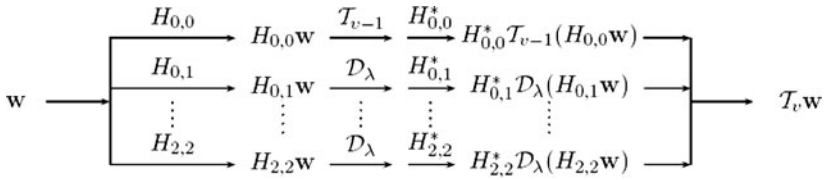


Fig. 5.2 The operator $\mathcal{T}_v$ defined recursively with $\mathcal{T}_0 = I$, the identity

$$\mathbf{h} = H^*_{0,0}\mathbf{g} + \sum_{\substack{i,j=0 \\ (i,j)\neq(0,0)}}^{2} H^*_{i,j}H_{i,j}\mathbf{h}. \tag{5.6}$$

Images are usually contaminated with noise, which are of high-frequency in nature. Since except for $H_{0,0}$ the others filter matrices are all high-pass, the noise is magnified in the second term of (5.6). We can use a threshold operator $\mathcal{D}_\lambda$ to remove the noise. The iteration, in matrix terms, thus becomes

$$\mathbf{h}^{(n+1)} = H^*_{0,0}\mathbf{g} + \sum_{\substack{i,j=0 \\ (i,j)\neq(0,0)}}^{2} H^*_{i,j}\mathcal{D}_\lambda(H_{i,j}\mathbf{h}^{(n)}), \quad n = 0, 1, \ldots,$$

where $\mathbf{h}^{(0)}$ is the initial guess. Here we use Donoho's soft thresholding operator [8]:

$$\mathcal{D}_\lambda(\mathbf{x}) = (t_\lambda(x_1), \ldots, t_\lambda(x_L))^\top,$$

where $t_\lambda(x) = \operatorname{sgn}(x)\max(|x| - \lambda, 0)$, $\lambda = 2\sigma\sqrt{\log L}$, L is the length of the vector $\mathbf{x}$, and σ is the variance of the noise estimated numerically by the method in [8].

However, to avoid too many high-frequency components being removed, we use wavelet packets to further decompose the high-frequency components before doing the thresholding. In essence, we replace the operator $\mathcal{D}_\lambda$ by the recursively-defined $\mathcal{T}_v$ shown in Fig. 5.2. The operator $\mathcal{T}_v$ will first decompose $\mathbf{w}$ until the level v and then threshold all the coefficients except the low-frequency ones on level v. In matrix terms, we have

$$\mathbf{h}^{(n+1)} = H^*_{0,0}\mathbf{g} + \sum_{\substack{i,j=0 \\ (i,j)\neq(0,0)}}^{2} H^*_{i,j}\mathcal{T}_v(H_{i,j}\mathbf{h}^{(n)}), \quad n = 0, 1, \ldots. \tag{5.7}$$

5.2.3 HR Reconstruction with Displacement Errors

Before we can apply the algorithm in Sect. 5.2.2 to improve video quality, there is one problem we have to overcome. In enhancing videos, the frames in the video may not be aligned exactly by length T as in (5.3) or in Fig. 5.1. For example, relative to the reference frame, a nearby frame may have moved in the x-direction by a distance of ℓT where $\ell = n + r$, with $n \in \mathbb{Z}$ and $|r| < 1$. In that case, if we want to apply the algorithm in the last section, we can first shift the frame back by $nT = (n/2)(2T)$ and then consider the shifted frame as a displaced frame of the reference frame with displacement error equals $r/2$. The displacement error can then be corrected by framelet systems as follows. We refer the readers to [5] for more details.

Define the 2D downsampling and upsampling matrices $D_{i,j} = D_j \otimes D_i$ and $U_{i,j} = U_j \otimes U_i$, where $D_i = I_M \otimes \mathbf{e}_i^\top$ and $U_i = I_M \otimes \mathbf{e}_i (i = 0, 1)$, I_M is the identity of size M, $\mathbf{e}_0 = (1, 0)^\top$ and $\mathbf{e}_1 = (0, 1)^\top$. Here M-by-M is the resolution of the low-resolution frame. Then we have

$$\mathbf{g}_{i,j} = D_{i,j}\mathbf{g} \quad \text{and} \quad \mathbf{g} = \sum_{i,j=0}^{1} U_{i,j}\mathbf{g}_{i,j}. \tag{5.8}$$

As mentioned above, in practice, what we obtained is a shifted version of $\mathbf{g}_{i,j}$, i.e. we have $\tilde{\mathbf{g}}_{i,j}(\cdot,\cdot) \equiv \mathbf{g}_{i,j}(\cdot + \epsilon_{i,j}^x, \cdot + \epsilon_{i,j}^y)$, where $0 \le |\epsilon_{i,j}^x| < 0.5$, $0 \le |\epsilon_{i,j}^y| < 0.5$, $0 \le i, j \le 1$. The parameters $\epsilon_{i,j}^x$ and $\epsilon_{i,j}^y$ are called *the displacement errors*. As in (5.4) and (5.8), the observed low-resolution image $\tilde{\mathbf{g}}_{i,j}$ can be considered as the down-sample of $\mathbf{h}$ after it has passed through a filter corresponding to the 2D framelet filter matrix $H = H(\epsilon_{i,j}^y) \otimes H(\epsilon_{i,j}^x)$, where $H(\epsilon)$ denotes the 1D filter $\frac{1}{2}[\frac{1}{2} - \epsilon, 1, \frac{1}{2} + \epsilon]$. More precisely, we have

$$\tilde{\mathbf{g}}_{i,j} = D_{i,j}H\mathbf{h}.$$

Then $\mathbf{g}_{i,j}$ can be obtained from $\tilde{\mathbf{g}}_{i,j}$ via

$$\mathbf{g}_{i,j} = D_{i,j}H_{0,0}\mathbf{h} = D_{i,j}[H - (H - H_{0,0})]\mathbf{h} = \tilde{\mathbf{g}}_{i,j} - D_{i,j}(H - H_{0,0})\mathbf{h}.$$

Hence we have

$$\mathbf{g} = \sum_{i,j=0}^{1} U_{i,j}\mathbf{g}_{i,j} = \sum_{i,j=0}^{1} U_{i,j}[\tilde{\mathbf{g}}_{i,j} - D_{i,j}(H - H_{0,0})\mathbf{h}] = \tilde{\mathbf{g}} - (H - H_{0,0})\mathbf{h}. \tag{5.9}$$

Substituting (5.9) into (5.7), we arrive at the 2D image resolution enhancement formula:

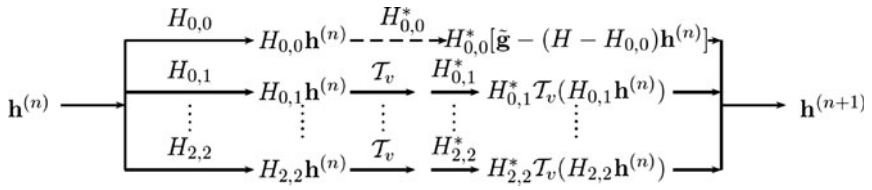


Fig. 5.3 Framelet-based resolution enhancement algorithm for 2D images, see (5.10)

$$\mathbf{h}^{(n+1)} = H^*_{0,0}[\tilde{\mathbf{g}} - (H - H_{0,0})\mathbf{h}^{(n)}] + \sum_{\substack{i,j=0 \\ (i,j)\neq(0,0)}}^{2} H^*_{i,j}\mathcal{T}_v(H_{i,j}\mathbf{h}^{(n)}). \tag{5.10}$$

We depict this algorithm graphically in Fig. 5.3.

5.3 Resolution Enhancement for Video Clips

Video clips consist of many still frames. Each frame can be considered as perturbations of its nearby frames. Therefore, we may generate higher resolution images of any frame in the video by exploiting the high redundancy between the nearby frames. More precisely, consider a sequence of frames $\{f_k\}_{k=-K}^{K}$ in a given video clip, where k increases with the time when the frame f_k is captured. We aim to improve the resolution of the reference frame f_0 by incorporating information from frames $\{f_k\}_{k\neq 0}$. Without loss of generality, we can assume that f_0 is the low-resolution image at the (0,0) sensor position without any displacement error.

An algorithm for video still enhancement is given in [7] which is an adaptation of the algorithm in (5.10). Basically, we have to tackle the following issues:

1. for each frame f_k, we have to estimate its sensor position and displacement errors with respect to f_0, and
2. it may be that not all low-resolution images at all sensor positions are available in the video.

Here we recall the ways we handled these issues in [7].

5.3.1 Estimating the Motion Parameters

For computational efficiency, we assume that the frames $\{f_k\}_{k\neq 0}$ are related to f_0 by an affine transform, i.e.

$$f_k(R_k\mathbf{x} - \mathbf{r}_k) \approx f_0(\mathbf{x}), \quad k \neq 0,$$

where $\mathbf{x}$ are the coordinates of the pixels in the region of interest. Denote

$$R_k\mathbf{x} - \mathbf{r}_k \equiv \begin{bmatrix} c_0^{(k)} & c_1^{(k)} \\ c_3^{(k)} & c_4^{(k)} \end{bmatrix}\mathbf{x} + \begin{bmatrix} c_2^{(k)} \\ c_5^{(k)} \end{bmatrix} = \begin{bmatrix} c_0^{(k)} & c_1^{(k)} & c_2^{(k)} \\ c_3^{(k)} & c_4^{(k)} & c_5^{(k)} \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}. \tag{5.11}$$

Our task is to estimate the parameters $\{c_k\}_{k=0}^5$ that minimize the difference between f_k and f_0, that is,

$$(c_0^{(k)}, c_1^{(k)}, \ldots, c_5^{(k)}) = \text{argmin} \sum_{j\in\mathcal{I}} [f_k(R_k\mathbf{x}_j - \mathbf{r}_k) - f_0(\mathbf{x}_j)]^2,$$

where $\mathcal{I}$ is the index set of pixels in the region of interest, which may be the entire image or part of the image. Many methods can be used to solve this minimization problem, such as the Levenberg-Marquardt iterative nonlinear minimization algorithm [18].

With R_k and $\mathbf{r_k}$, we can compute the sensor position (s_k^x, s_k^y) with $s_k^x, s_k^y \in \{0, 1\}$ and the displacement errors $(\epsilon_k^x, \epsilon_k^y)$ for the frame f_k with respect to f_0. Since $f_k(R_k\mathbf{x} - \mathbf{r}_k) = f_k(R_k(\mathbf{x} - R_k^{-1}\mathbf{r}_k))$, it can be viewed as a translation of f_0 with displacement vector $-R_k^{-1}\mathbf{r}_k$. Our task is to write

$$R_k^{-1}\mathbf{r}_k = \mathbf{u}_k + \frac{1}{2}\begin{bmatrix} s_k^x \\ s_k^y \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \epsilon_k^x \\ \epsilon_k^y \end{bmatrix}. \tag{5.12}$$

Then, $\hat{f}_k(\mathbf{x}) \equiv f_k(R_k(\mathbf{x} - \mathbf{u}_k))$ can be considered as the low-resolution image $g_{s_k^x, s_k^y}$ with displacement errors $(\epsilon_k^x, \epsilon_k^y)$ at sensor position (s_k^x, s_k^y). The algorithm is as follows.

Algorithm 1 $(\hat{f}(\mathbf{x}), s^x, s^y, \epsilon^x, \epsilon^y) \leftarrow (f, f_0)$: locate the frame f against the reference frame f_0.

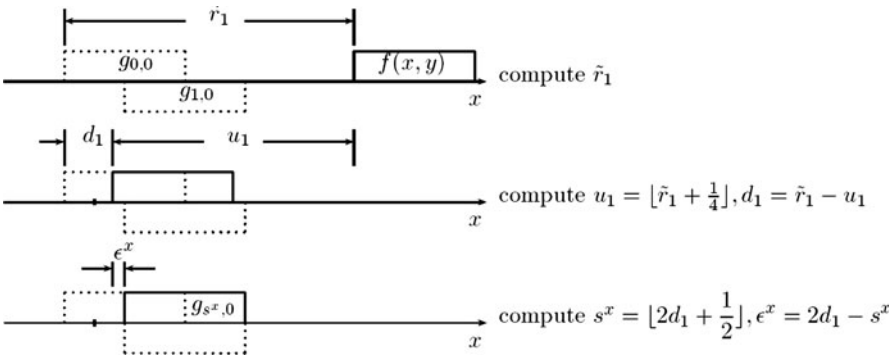


Fig. 5.4 Resolve the horizonal displacement in Algorithm 1. First we compute the total displacement $\tilde{r}_1$. Then write $\tilde{r}_1 = u_1 + d_1$ where $d_1 \in [-1/4, 3/4)$. Then the sensor location $s^x = \lfloor 2d_1 + \frac{1}{2} \rfloor$ and the displacement error $\epsilon^x = 2d_1 - s^x$

1 Compute $[\tilde{r}_1, \tilde{r}_2] = R^{-1}\mathbf{r}$.
2 Let $\mathbf{u} \equiv [\lfloor \tilde{\mathbf{r}}_1 + \frac{1}{4} \rfloor, \lfloor \tilde{\mathbf{r}}_2 + \frac{1}{4} \rfloor]^\top$, then $[d_1, d_2] \equiv [\tilde{r}_1, \tilde{r}_2] - \mathbf{u}^\top$ has entries in $[-\frac{1}{4}, \frac{3}{4})$.
3 Let $[s^x, s^y] \equiv [\lfloor 2d_1 + \frac{1}{2} \rfloor, \lfloor 2d_2 + \frac{1}{2} \rfloor]$, then $s^x, s^y \in \{0, 1\}$.
4 Let $[\epsilon^x, \epsilon^y] \equiv [2d_1 - s^x, 2d_2 - s^y]$, then $|\epsilon^x|, |\epsilon^y| < \frac{1}{2}$ and (5.12) holds.
5 $\hat{f}(\mathbf{x}) \equiv f(R(\mathbf{x} - \mathbf{u}))$.

We use Fig. 5.4 to illustrate how the algorithm resolves the displacement in the horizonal direction. It is similar for the vertical direction.

5.3.2 The Video Still Enhancement Algorithm

After passing a frame f through Algorithm 1, we then have the (s^x, s^y)th low-resolution image with displacement error (ϵ^x, ϵ^y), i.e.

$$\hat{f}(\cdot) = \tilde{g}_{s^x,s^y}(\cdot, \cdot) = g_{s^x,s^y}(\cdot + \epsilon^x, \cdot + \epsilon^y).$$

But in the algorithm for image enhancement (5.10) (see also Fig. 5.3), we assume not one, but a complete set of low-resolution images $\{\tilde{g}_{i,j}\}_{i,j=0}^{1}$ at every sensor position. To compensate for the missing low-resolution images, our idea is to generate them by downsampling the current high-resolution approximation h of f_0 with zero displacement error, i.e.

$$\mathbf{g}_{i,j} = \begin{cases} \hat{\mathbf{f}} - D_{i,j}(H - H_{0,0})\mathbf{h}, & (i,j) = (s^x, s^y), \\ D_{i,j}H_{0,0}\mathbf{h}, & (i,j) \neq (s^x, s^y). \end{cases} \tag{5.13}$$

We use an alternate direction approach to obtain the high-resolution image $\mathbf{h}$. In the m-th outer iteration step, we let $\mathbf{g}_{s^x,s^y} = \hat{\mathbf{f}} - D_{s^x,s^y}(H - H_{0,0})\mathbf{h}_m^{(0)}$, where

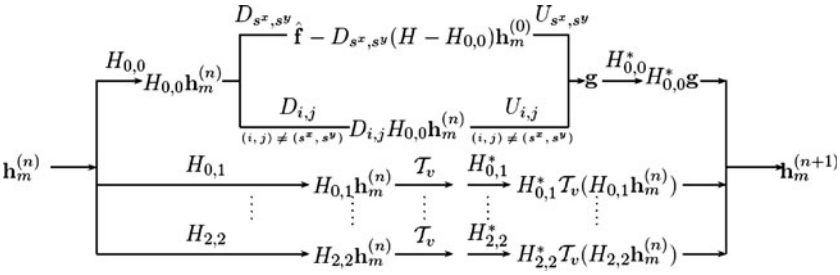


Fig. 5.5 Inner iteration of Algorithm 2. Here, we fix the low-resolution image $\mathbf{g}_{s^x,s^y}$, and compensate the missing ones $\mathbf{g}_{i,j}$ by downsampling $H_{0,0}\mathbf{h}_m^{(n)}$. Then we have $\mathbf{g} = \sum_{i,j=0}^{1} U_{i,j}\mathbf{g}_{i,j}$ as in (5.9) with $\mathbf{g}_{i,j}$ substituted by (5.13). We iterate $\mathbf{h}_m^{(n)}$ w.r.t. n until it converges; then we set $\mathbf{h}_m = \mathbf{h}_m^{(n)}$

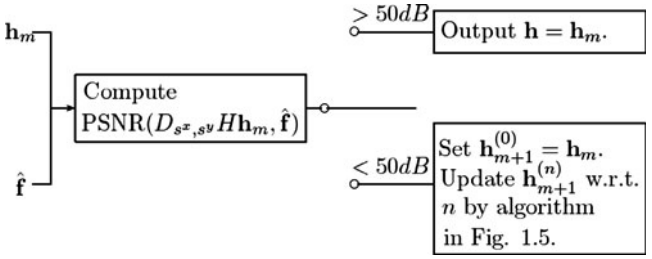


Fig. 5.6 Outer iteration of Algorithm 2. We update the high resolution image $\mathbf{h}$ by a frame $\hat{\mathbf{f}}$ with parameters $(s^x, s^y, \epsilon^x, \epsilon^y)$

$\mathbf{h}_m^{(0)} = \mathbf{h}_{m-1}$. Then, we iterate $\mathbf{h}_m^{(n)}$ with respect to n as shown in Fig. 5.5, which is in fact a modification of Fig. 5.3. When it converges, we set $\mathbf{h}_{m+1}^{(0)} = \mathbf{h}_m$. Once we get an update $\mathbf{h}_m$, we will go into the next outer iteration; see Fig. 5.6. The complete alternate direction algorithm is as follows.

Algorithm 2 $\mathbf{h} \leftarrow$ Update $(\mathbf{h}, \hat{\mathbf{f}}, s^x, s^y, \epsilon^x, \epsilon^y)$: Update the high resolution image $\mathbf{h}$ by a frame $\hat{\mathbf{f}}$ with parameters $(s^x, s^y, \epsilon^x, \epsilon^y)$.

1 Initialize $\mathbf{h}_0 = \mathbf{h}$, and set $m = 0$.

2 If $PSNR(D_{s^x,s^y}H\mathbf{h}_m, \hat{\mathbf{f}}) < 50\,\text{dB}$, set $\mathbf{h}_{\mathbf{m+1}}^{(\mathbf{0})} = \mathbf{h}_m$, and $m = m + 1$; otherwise, output $\mathbf{h} = \mathbf{h}_m$, stop.

3 Iterate $\mathbf{h}_m^{(n)}$ w.r.t. n until convergence (see Fig. 5.5):

(a) $$\mathbf{g}_{i,j} = \begin{cases} \hat{\mathbf{f}} - D_{s^x,s^y}(H - H_{0,0})\mathbf{h}_m^{(0)}, & (i,j) = (s^x, s^y), \\ D_{i,j}H_{0,0}\mathbf{h}_m^{(n)}, & (i,j) \neq (s^x, s^y). \end{cases}$$

(b) $$\mathbf{g} = \sum_{i,j=0}^{1} U_{i,j}\mathbf{g}_{i,j}.$$

(c) $$\mathbf{h}_m^{(n+1)} = H_{0,0}^*\mathbf{g} + \sum_{\substack{i,j=0 \\ (i,j)\neq(0,0)}}^{2} H_{i,j}^*\mathcal{T}_v(H_{i,j}\mathbf{h}_m^{(n)}).$$

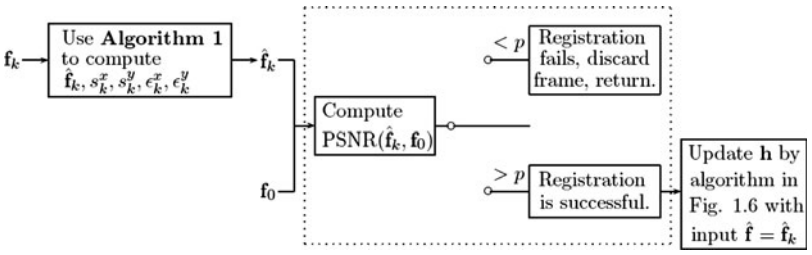


Fig. 5.7 Resolution enhancement for video clips (in the *dotted-lined box* it is used to determine if $\hat{\mathbf{f}}_k$ is good enough to update $\mathbf{h}$)

4 Set $\mathbf{h}_m = \mathbf{h}_m^{(n)}$ after converge, and go back to step 2.

In Fig. 5.7, we give the algorithm for the video still enhancement. Given a reference frame $\mathbf{f}_0$, we use a sequence of $2K$ frames $\{\mathbf{f}_k\}_{k=-K}^{K}$ that are taken just before and after the reference frame $\mathbf{f}_0$. The step in the dotted-lined box is to determine if the shifted frame $\hat{\mathbf{f}}_k$ is close enough to $\mathbf{f}_0$ or else we discard the frame. In the experiments, we set $p = 25$dB. Initially, we estimate $\mathbf{h}$ by bilinear interpolation on $\mathbf{f}_0$, and then use the new information from good frames to update $\mathbf{h}$. The advantage of our algorithm is that based on the rule shown in the dotted-lined box it only chooses the good candidate frames to enhance the resolution, and there is no need to determine the number of frames to be used in advance.

5.4 Video Enhancement Algorithm

Since video streams are made up of frames, and we can now improve the resolution of each frame in a video stream, we can compose these high-resolution frames together to generate a higher resolution video of the given video stream. More precisely, we can apply our algorithm in Fig. 5.7 to the frames $\{f_k\}_{k=-K}^{K}$ to enhance f_0, and then apply the algorithm again to frames $\{f_k\}_{k=-K+1}^{K+1}$ to enhance f_1, etc. Then, by combining the enhanced frames, we obtain a high-resolution video.

In this section, we test this idea for a video clip which is filmed by us by moving our camera over a calendar. The video clip is in .avi format with size 520×480, and can be downloaded at [19]. We first try the video still enhancement algorithm

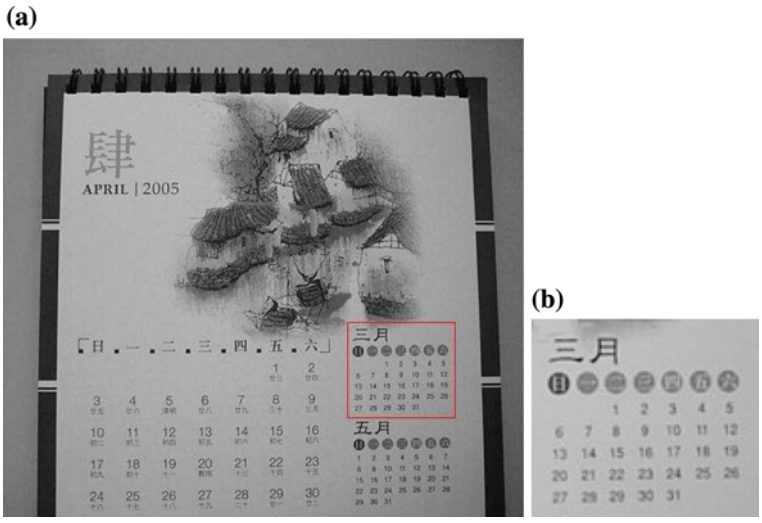


Fig. 5.8 **a** The reference frame f_{60}, and **b** a part of f_{60}

Table 5.1 Alignment results from our algorithm

Frame index	(s^x, s^y)	(ϵ^x, ϵ^y)	$f_0(x) \approx f(R\mathbf{x} + \mathbf{r})$
61	(1,0)	(0.119, 0.036)	Yes
59	(1,0)	(0.246, −0.066)	Yes
62	(0,0)	(0.368, 0.186)	Yes
58	(0,0)	(0.272, −0.126)	Yes
63	(1,0)	(−0.139, 0.086)	Yes
57	(1,0)	(0.334, −0.214)	Yes
64	(1,0)	(0.323, −0.194)	Yes
56	(0,0)	(−0.134, −0.381)	Yes
65	(0,0)	(−0.349, 0.164)	Yes
55	(0,1)	(0.454, 0.448)	Yes
66	(0,0)	(0.421, 0.181)	Yes
54	–	–	No
67	(1,0)	(−0.219, 0.236)	Yes
53	(0,1)	(−0.323, 0.323)	Yes
68	(1,0)	(0.313, 0.232)	Yes
52	–	–	No
69	(0,0)	(0.096, 0.204)	Yes
51	(0,0)	(0.292, −0.500)	Yes
70	(0,0)	(0.485, 0.186)	Yes
50	(0,1)	(0.062, 0.301)	Yes

in [7] to enhance the resolution of a frame in the video. In the seven seconds of the video, we choose the 60th frame f_{60} as our reference frame, see Fig. 5.8a. In Fig. 5.8b, we show a part of the reference frame f_{60} (the area enclosed by the box in Fig. 5.8 (a)) that we try to improve the resolution on.

We let $K = 10$, that is, we use the 50th to the 70th frames to improve the resolution of f_{60}. The alignment parameters for this clip are listed in Table 5.1, which shows that frames f_{52} and f_{54} are discarded. Figure 5.9a gives the first guess of the high-resolution image of f_{60} by the bilinear interpolation. The result from our algorithm (i.e. Fig. 5.7) is shown in Fig. 5.9b. Clearly the calendar by our method is much clearer than that by the bilinear interpolation. Moreover some numbers, such as "16" and "18", which are clearly discernible now, are very difficult to read from the video clip or just by bilinear interpolation.

The results clearly show that the video still enhancement algorithm (Fig. 5.7) is working. Next we extend it to video enhancement. Our aim is to obtain a high-resolution video for the image in Fig. 5.8b. We will use the video still enhancement algorithm (Fig. 5.7) repeatedly to enhance all frames in $\{f_k\}_{k=40}^{63}$. More precisely, frames $\{f_k\}_{k=\ell-10}^{\ell+10}$ will be used to improve the resolution of frames f_ℓ, with $40 \leq \ell \leq 63$. The enhanced stills are put back together to get a higher-resolution video with 24 frames. The original clip and the resulting clips are given in [19]. Because the new one has higher resolution, it is 4 times in size and is much clearer.

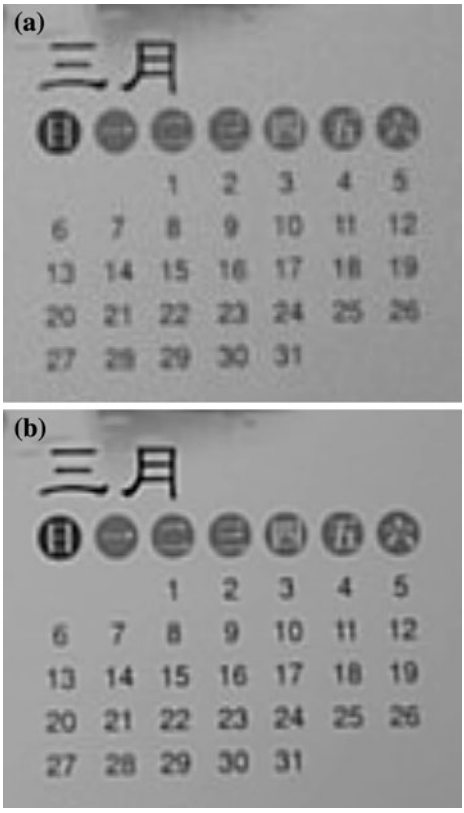


Fig. 5.9 Reconstructed high-resolution images: **a** using bilinear interpolation, **b** using our algorithm in Fig. 5.7

5.5 Conclusion

In this paper, we give a short survey of the framelet algorithm proposed in [7] for high-resolution still enhancement from video clips. We then extend it to video enhancement. Simulation results show that our framelet algorithm can reveal information that is not discernible in the original video clips or by simple interpolation of any particular frame in the video.

By modification of the motion estimation Eq. 5.11, our framelet algorithm can also be extended to more complicated motions. So far, we have not yet make use of the sparsity of the tight-frame coefficients across frames. How to make use of it is an interesting topic for our future work.

References

1. Bose N, Boo K (1998) Highresolution image reconstruction with multisensors. Int J Imag Syst Technol 9:294–304
2. Cai J-F, Chan RH, Shen ZW (2008) A framelet-based image inpainting algorithm. Appl Comput Harmon Anal 24:131–149
3. Cain S, Hardie RC, Armstrong EE (1996) Restoration of aliased video sequences via a maximum-likelihood approach. Infrared Inf Symp (IRIS) Passive Sens 1:377–390
4. Capel D, Zisserman A (2003) Computer vision applied to superresolution. IEEE Signal Process Mag 20:72–86
5. Chan HR, Riemenschneider SD, Shen LX, Shen ZW (2004) Tight frame: the efficient way for highresolution image reconstruction. Appl Comput Harmon Anal 17:91–115
6. Chan HR, Riemenschneider SD, Shen LX, Shen ZW (2004) Highresolution image reconstruction with displacement errors: a framelet approach. Int J Imag Syst Technol 14:91–104
7. Chan HR, Shen ZW, Xia T (2007) A framelet algorithm for enhancing video stills. Appl Comput Harmon Anal 23:153–170
8. Donoho D (1995) De-noising by soft-thresholding. IEEE Trans Inf Theory 41:613–627
9. Eren PE, Sezan MI, Tekalp AM (1997) Robust, object-based high-resolution image reconstruction from low-resolution video. IEEE Trans Image Process 6:1446–1451
10. Hardie RC, Barnard KJ, Bognar JG, Armstrong EE, Watson EA (1998) High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system. Opt Eng 37:247–260
11. Kim SP, Su W-Y (1993) Recursive high-resolution reconstruction of blurred multiframe images. IEEE Trans Image Process 2:534–539
12. Lillesand MT, Kiefer WR, Chipman WJ (2000) Remote sensing and image interpretation. 4th edn. John Wiley & Sons, New York
13. Mateos J, Molina R, Katsaggelos AK (2003) Bayesian high resolution image reconstruction with incomplete multisensor low resolution systems. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing Hong Kong, pp 705–708
14. Mather P (2004) Computer processing of remotely-sensed images: an introduction. 3rd edn. John Wiley & Sons, Chichester
15. Nakazawa Y, Komatsu T, Saito T (1995) High-resolution image acquisition based on temporal integration with hierarchical estimation of image warping. In: Proceedings of IEEE International Conference on Image Processing. Washington, DC, pp 244–247
16. Ron A, Shen ZW (1997) Affine system in $L^2(R^d)$: the analysis of the analysis operator. J Funct Anal 148:408–447
17. Schultz RR, Stevenson RL (1996) Extraction of high-resolution frames from video sequences. IEEE Trans Image Process 5:996–1011
18. Szeliski R (1996) Video mosaics for virtual environments. IEEE Comput Graph Appl 16:22–30
19. Video clip and full results at http://www.math.cuhk.edu.hk/∼rchan/paper/cdw
20. Yang Q, Parvin B (2003) High-resolution reconstruction of sparse data from dense low-resolution spatio-temporal data. IEEE Trans Image Process 12:671–677

Chapter 6
Perturbation Analysis of the Mixed-Type Lyapunov Equation

Mingsong Cheng and Shufang Xu

Abstract This paper concerns the mixed-type Lyapunov equation $X = A^*XB + B^*XA + Q$, where A, B, and Q are $n \times n$ complex matrices and A^* the conjugate transpose of a matrix A. A perturbation bound for the solution to this matrix equation is derived, an explicit expression of the condition number is obtained, and the backward error of an approximate solution is evaluated by using the techniques developed in Sun (Linear Algebra Appl 259:183–208, 1997), Sun and Xu (Linear Algebra Appl 362:211–228, 2003). The results are illustrated by using some numerical examples.

6.1 Introduction

Consider the mixed-type Lyapunov matrix equation

$$X = A^*XB + B^*XA + Q, \tag{6.1}$$

where $A, B, Q \in \mathbf{C}^{n\times n}$. Here $\mathbf{C}^{n\times n}$ denotes the set of all $n \times n$ complex matrices, A^* the conjugate transpose of a matrix A. This kind of equation arises in Newton's method for solving the following matrix equation:

$$X - A^*X^{-2}A = I, \tag{6.2}$$

M. Cheng (✉)
Department of Applied Mathematics, Dalian University of Technology,
116024 Dalian, China
e-mail: mscheng@dlut.edu.cn

S. Xu
LMAM, School of Mathematical Sciences, Peking University,
100871 Beijing, China
e-mail: xsf@pku.edu.cn

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_6,

which are recently studied by Ivanov, Liu, etc. [2, 4, 5]. In Newton's method, we must solve the following equation in each iterative step:

$$X_{k+1} + A^* X_k^{-2} X_{k+1} X_k^{-1} A + A^* X_k^{-1} X_{k+1} X_k^{-2} A = I + 3A^* X_k^{-2} A, \tag{6.3}$$

which is a mixed-type Lyapunov equation (see [1, 9] for more details).

The matrix equation (6.1) can be regarded as a generalization of the discrete and continuous Lyapunov equations. In fact, when $B = \frac{1}{2}A$, the matrix equation (6.1) is just the discrete Lyapunov equation

$$X = A^* X A + Q; \tag{6.4}$$

when $B = I, \widetilde{A} = A - \frac{1}{2}I$, it is just the continuous Lyapunov equation

$$\widetilde{A}^* X + X\widetilde{A} + Q = 0. \tag{6.5}$$

This shows why we call it mixed-type Lyapunov equation. The solvability for this equation has been studied in our former paper [9]. In this paper, our main purpose is threefold. To begin with, we derive a perturbation bound for the solution X. Secondly, we apply the theory of condition developed by Rice [6] to define a condition number of X, and moreover, we use the techniques developed in [8] to derive its explicit expressions. Finally, we use the techniques developed in [7] to evaluate the backward error of an approximate solution.

We start with some notations which we shall use throughout this paper. Define the linear operator

$$\mathbf{L}(X) = X - A^* X B - B^* X A, \quad X \in \mathbf{C}^{n\times n}, \tag{6.6}$$

then the mixed-type Lyapunov equation (6.1) can be written as $\mathbf{L}(X) = Q$. Throughout this paper we always assume that the mixed-type Lyapunov equation (6.1) has a unique solution X, i.e., the linear operator $\mathbf{L}$ is invertible. We use $\mathbf{C}^{n\times n}$ (or $\mathbf{R}^{n\times n}$) to denote the set of complex (or real) $n \times n$ matrices. A^* denotes the conjugate transpose of a matrix A, A^T the transpose of $A, A^\dagger$ the Moore–Penrose inverse of A, and $\bar{A}$ the conjugate of A. The symbols $\|\cdot\|, \|\cdot\|_2$, and $\|\cdot\|_F$ denote a unitary invariant norm, the spectral norm, and the Frobenius norm, respectively. For $A = [a_1, \ldots, a_n] = [a_{ij}] \in \mathbf{C}^{n\times n}$ and a matrix $B, A \otimes B = [a_{ij}B]$ is a Kronecker product, and $\mathrm{vec}(A)$ is a vector defined by $\mathrm{vec}(A) = [a_1^T, \ldots, a_n^T]^T$.

6.2 Perturbation Bound

Let X be the unique solution of the mixed-type Lyapunov equation (6.1), and let the coefficient matrices A, B, and Q be slightly perturbed to

$$\widetilde{A} = A + \Delta A, \quad \widetilde{B} = B + \Delta B, \quad \widetilde{Q} = Q + \Delta Q,$$

respectively, where $\Delta A, \Delta B, \Delta Q \in \mathbf{C}^{n\times n}$. In this section we consider perturbation bounds for the solution X.

Let $\widetilde{X} = X + \Delta X$ with $\Delta X \in \mathbf{C}^{n\times n}$ satisfy the perturbed matrix equation

$$\widetilde{X} = \widetilde{A}^* \widetilde{X} \widetilde{B} + \widetilde{B}^* \widetilde{X} \widetilde{A} + \widetilde{Q}. \tag{6.7}$$

Subtracting (6.1) from (6.7), we have

$$\mathbf{L}(\Delta X) = \Delta Q + h(\Delta A, \Delta B), \tag{6.8}$$

where $\mathbf{L}$ defined by (6.6) and

$$\begin{aligned} h(\Delta A, \Delta B) = {} & (\Delta A)^* \widetilde{X} B + (\Delta B)^* \widetilde{X} A + A^* \widetilde{X} \Delta B + B^* \widetilde{X} \Delta A + (\Delta A)^* \widetilde{X} \Delta B \\ & + (\Delta B)^* \widetilde{X} \Delta A. \end{aligned}$$

Since the linear operator $\mathbf{L}$ is invertible, we can rewrite (6.8) as

$$\Delta X = \mathbf{L}^{-1}(\Delta Q) + \mathbf{L}^{-1}(h(\Delta A, \Delta B)). \tag{6.9}$$

Define

$$\|\mathbf{L}^{-1}\| = \max_{\substack{W\in\mathbf{C}^{n\times n} \\ \|W\|=1}} \|\mathbf{L}^{-1} W\|.$$

It follows that

$$\|\mathbf{L}^{-1} W\| \le \|\mathbf{L}^{-1}\| \|W\|, \quad W \in \mathbf{C}^{n\times n}. \tag{6.10}$$

Now let

$$\alpha = \|A\|, \quad \beta = \|B\|, \quad \delta = \|\Delta Q\|, \quad l = \|\mathbf{L}^{-1}\|^{-1}, \quad \gamma = \|X\|, \tag{6.11}$$

and define

$$\epsilon = \alpha\|\Delta B\| + \beta\|\Delta A\| + \|\Delta A\|\|\Delta B\|. \tag{6.12}$$

Then we can state the main result of this section as follows.

Theorem 1.1 *If $2\epsilon < l$, then the perturbed matrix equation (6.7) has a unique solution $\widetilde{X}$ such that*

$$\|\widetilde{X} - X\| \le \frac{\delta + 2\gamma\epsilon}{l - 2\epsilon} \equiv \delta_*. \tag{6.13}$$

Proof Let

$$f(\Delta X) = \mathbf{L}^{-1}(\Delta Q) + \mathbf{L}^{-1}(h(\Delta A, \Delta B)).$$

Obviously, $f(\Delta X)$ can be regarded as a continuous mapping from $\mathbf{C}^{n\times n}$ to $\mathbf{C}^{n\times n}$. Now define

$$\mathcal{S}_{\delta_*} = \{\Delta X \in \mathbf{C}^{n\times n} \ : \ \|\Delta X\| \le \delta_*\}. \tag{6.14}$$

Then for any $\Delta X \in \mathcal{S}_{\delta_*}$, we have

$$\begin{aligned}\|f(\Delta X)\| &\le \frac{1}{l}\|\Delta Q\| + \frac{1}{l}\|h(\Delta A, \Delta B)\| \\ &\le \frac{\delta}{l} + \frac{2}{l}(\|B\|\|\Delta A\| + \|A\|\|\Delta B\| + \|\Delta A\|\|\Delta B\|)(\|X\| + \|\Delta X\|) \\ &\le \frac{\delta + 2\epsilon\gamma + 2\epsilon\delta_*}{l} \\ &= \delta_*.\end{aligned}$$

Thus we have proved that $f(\mathcal{S}_{\delta_*}) \subset \mathcal{S}_{\delta_*}$. By the Schauder fixed-point theorem, there exists a $\Delta X_* \in \mathcal{S}_{\delta_*}$ such that $f(\Delta X_*) = \Delta X_*$, i.e., there exists a solution ΔX_* to the perturbed equation (6.8) such that

$$\|\Delta X_*\| \le \delta_*. \tag{6.15}$$

Let $\widetilde{X} = X + \Delta X_*$. Then $\widetilde{X}$ is a solution of the perturbed matrix equation (6.7). Next we prove that the perturbed matrix equation (6.7) has a unique solution.

Define two linear operators

$$\mathbf{H}(Y) = (\Delta A)^* Y B + (\Delta B)^* Y A + A^* Y \Delta B + B^* Y \Delta A + (\Delta A)^* Y \Delta B + (\Delta B)^* Y \Delta A,$$

and

$$\widetilde{\mathbf{L}}(Y) = Y - \widetilde{A}^* Y \widetilde{B} - \widetilde{B}^* Y \widetilde{A},$$

respectively, where $Y \in \mathbf{C}^{n\times n}$. Then it is easily seen that

$$\widetilde{\mathbf{L}}(Y) = \mathbf{L}(Y) - \mathbf{H}(Y) = \mathbf{L}(\mathbf{F}(Y)),$$

where

$$\mathbf{F}(Y) = Y - \mathbf{L}^{-1}(\mathbf{H}(Y)).$$

Note that

$$\begin{aligned}\|\mathbf{F}(Y)\| &\ge \|Y\| - \|\mathbf{L}^{-1}(\mathbf{H}(Y))\| \\ &\ge \|Y\|(1 - \|\mathbf{L}^{-1}\|\|\mathbf{H}\|) \\ &\ge \left(1 - \frac{2\epsilon}{l}\right)\|Y\|,\end{aligned}$$

which guarantees that the linear operator $\mathbf{F}$ is invertible under the condition $2\epsilon < l$, and hence, the linear operator $\widetilde{\mathbf{L}}$ is invertible, which implies that the perturbed matrix equation (6.7) has a unique solution $\widetilde{X}$. Thus, the inequality (6.15) implies that the inequality (6.13) holds. The proof is completed.

Remark 2.1 From Theorem 1.1 we get the absolute perturbation bound of first order for the unique solution X as follows:

$$\|\widetilde{X} - X\| \leq \frac{1}{l}\|\Delta Q\| + \frac{2\alpha\gamma}{l}\|\Delta B\| + \frac{2\beta\gamma}{l}\|\Delta A\| + O(\|(\Delta A, \Delta B, \Delta Q)\|^2),$$
$$(\Delta A, \Delta B, \Delta Q) \longrightarrow 0. \tag{6.16}$$

Combining this with (6.9) gives

$$\Delta X = \mathbf{L}^{-1}(\Delta Q) + \mathbf{Q}(\Delta A, \Delta B) + O(\|(\Delta A, \Delta B, \Delta Q)\|^2), \quad (\Delta A, \Delta B, \Delta Q) \longrightarrow 0, \tag{6.17}$$

where

$$\mathbf{Q}(\Delta A, \Delta B) = \mathbf{L}^{-1}((\Delta A)^* XB + (\Delta B)^* XA + A^* X\Delta B + B^* X\Delta A). \tag{6.18}$$

Remark 2.2 Noting that (6.1) implies that

$$\|Q\| \leq (1 + 2\|A\|\|B\|)\|X\|,$$

from (6.16) we immediately obtain the first relative perturbation bound for the solution X by

$$\frac{\|\widetilde{X} - X\|}{\|X\|} \leq \frac{1 + 2\alpha\beta}{l}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta B\|}{\|B\|} + \frac{\|\Delta Q\|}{\|Q\|}\right) + O\left(\|(\Delta A, \Delta B, \Delta Q)\|^2\right),$$
$$(\Delta A, \Delta B, \Delta Q) \longrightarrow 0. \tag{6.19}$$

6.3 Condition Numbers

We now apply the theory of condition developed by Rice [6] to study condition numbers of the unique solution X to the mixed-type Lyapunov equation (6.1).

Suppose that the coefficient matrices A, B, Q are slightly perturbed to $\widetilde{A}, \widetilde{B}, \widetilde{Q} \in \mathbf{C}^{n\times n}$, respectively, and let

$$\Delta A = \widetilde{A} - A, \quad \Delta B = \widetilde{B} - B, \quad \Delta Q = \widetilde{Q} - Q.$$

From Theorem 1.1 and Remark 2.1 we see that if $\|(\Delta A, \Delta B, \Delta Q)\|_F$ is sufficiently small, then the unique solution $\widetilde{X}$ to the perturbed matrix equation (6.7) exists, and

$$\Delta X \equiv \widetilde{X} - X = \mathbf{L}^{-1}(\Delta Q) + \mathbf{Q}(\Delta A, \Delta B) + O(\|(\Delta A, \Delta B, \Delta Q)\|_F^2), \tag{6.20}$$

as $(\Delta A, \Delta B, \Delta Q) \longrightarrow 0$, where $\mathbf{Q}$ is defined by (6.18).

By the theory of condition developed by Rice [6] we define the condition number of the unique solution X by

$$c(X) = \lim_{\delta \to 0} \sup_{\|\frac{\Delta A}{\alpha}, \frac{\Delta B}{\beta}, \frac{\Delta Q}{\rho}\|_F \le \delta} \frac{\|\Delta X\|_F}{\xi \delta}, \tag{6.21}$$

where ξ, α, β and ρ are positive parameters. Taking $\xi = \alpha = \beta = \rho = 1$ in (6.21) gives the absolute condition number $c_{\text{abs}}(X)$, and taking $\xi = \|X\|_F, \alpha = \|A\|_F$, $\beta = \|B\|_F, \rho = \|Q\|_F$ in (6.21) gives the relative condition number $c_{\text{rel}}(X)$.

Substituting (6.20) into (6.21) we get

$$\begin{aligned} c(X) &= \frac{1}{\xi} \max_{\substack{(\frac{\Delta A}{\alpha}, \frac{\Delta B}{\beta}, \frac{\Delta Q}{\rho}) \neq 0 \\ \Delta A, \Delta B, \Delta Q \in \mathbf{C}^{n \times n}}} \frac{\|\mathbf{L}^{-1}(\Delta Q) + \mathbf{Q}(\Delta A, \Delta B)\|_F}{\|\frac{\Delta A}{\alpha}, \frac{\Delta B}{\beta}, \frac{\Delta Q}{\rho}\|_F} \\ &= \frac{1}{\xi} \max_{\substack{(E,M,N) \neq 0 \\ E,M,N \in \mathbf{C}^{n \times n}}} \frac{\|\mathbf{L}^{-1}(\mathbf{G}(E,M,N))\|_F}{\|(E,M,N)\|_F}, \end{aligned} \tag{6.22}$$

where

$$\mathbf{G}(E,M,N) = \rho N + \alpha(E^*XB + B^*XE) + \beta(A^*XM + M^*XA).$$

Let L be the matrix representation of the linear operator $\mathbf{L}$. Then it follows from (6.6) that

$$L = I \otimes I - B^T \otimes A^* - A^T \otimes B^*. \tag{6.23}$$

Let

$$\begin{aligned} L^{-1} &= S + i\Sigma, \\ L^{-1}(I \otimes (B^*X)) &= U_1 + i\Omega_1, \; L^{-1}((XB)^T \otimes I)\Pi = U_2 + i\Omega_2, \\ L^{-1}(I \otimes (A^*X)) &= V_1 + i\Gamma_1, \; L^{-1}((XA)^T \otimes I)\Pi = V_2 + i\Gamma_2, \end{aligned} \tag{6.24}$$

where $U_1, U_2, \Omega_1, \Omega_2, V_1, V_2, \Gamma_1, \Gamma_2 \in \mathbf{R}^{n^2 \times n^2}$, and Π is the vec-permutation matrix (see [3, p. 32–34]), i.e.,

$$\text{vec}(E^T) = \Pi \text{vec}(E).$$

Moreover, let

$$S_c = \begin{bmatrix} S & -\Sigma \\ \Sigma & S, \end{bmatrix}, \; U_c = \begin{bmatrix} U_1 + U_2 & \Omega_2 - \Omega_1 \\ \Omega_1 + \Omega_2 & U_1 - U_2 \end{bmatrix}, V_c = \begin{bmatrix} V_1 + V_2 & \Gamma_2 - \Gamma_1 \\ \Gamma_1 + \Gamma_2 & V_1 - V_2. \end{bmatrix} \tag{6.25}$$

Then the following theorem immediately follows from (6.22).

Theorem 1.2 *The condition number $c(X)$ defined by (6.21) has the explicit expression*

$$c(X) = \frac{1}{\xi} \|(\alpha U_c, \beta V_c, \rho S_c)\|_2, \tag{6.26}$$

where the matrices U_c, V_c and S_c are defined by (6.23)–(6.25).

Remark 3.1 From Theorem 1.2 we have the relative condition number

$$c_{\mathrm{rel}}(X) = \frac{\left\|(\|A\|_F U_c, \|B\|_F V_c, \|Q\|_F S_c)\right\|_2}{\|X\|_F}.$$

6.4 Backward Error

Let $\widetilde{X} \in \mathbf{C}^{n\times n}$ be an approximation to the unique solution X of the mixed-type Lyapunov equation (6.2), and let $\Delta A, \Delta B$, and ΔQ be the corresponding perturbations of the coefficient matrices A, B, and Q in (6.1). A backward error of the approximate solution $\widetilde{X}$ can be defined by

$$\eta(\widetilde{X}) = \min\left\{\left\|\left(\frac{\Delta A}{\alpha}, \frac{\Delta B}{\beta}, \frac{\Delta Q}{\rho}\right)\right\|_F : \Delta A, \Delta B, \Delta Q \in \mathbf{C}^{n\times n},\ \widetilde{X} - (A+\Delta A)^*\widetilde{X}(B+\Delta B) - (B+\Delta B)^*\widetilde{X}(A+\Delta A) = Q + \Delta Q\right\}, \tag{6.27}$$

where α, β and ρ are positive parameters. Taking $\alpha = \beta = \rho = 1$ in (6.27) gives the absolute backward error $\eta_{\mathrm{abs}}(\widetilde{X})$, and taking $\alpha = \|A\|_F, \beta = \|B\|_F, \rho = \|Q\|_F$ in (6.27) gives the relative backward error $\eta_{\mathrm{rel}}(\widetilde{X})$.

Let

$$R = Q - \widetilde{X} + A^*\widetilde{X}B + B^*\widetilde{X}A. \tag{6.28}$$

Then from

$$\widetilde{X} - (A+\Delta A)^*\widetilde{X}(B+\Delta B) - (B+\Delta B)^*\widetilde{X}(A+\Delta A) = Q + \Delta Q,$$

we get

$$\begin{aligned}&(\Delta A)^*\widetilde{X}B + (\Delta B)^*\widetilde{X}A + A^*\widetilde{X}\Delta B + B^*\widetilde{X}\Delta A + \Delta Q\\ &\quad = -R - (\Delta A)^*\widetilde{X}\Delta B - (\Delta B)^*\widetilde{X}\Delta A,\end{aligned} \tag{6.29}$$

which shows that the problem of finding an explicit expression of the backward error $\eta(\widetilde{X})$ defined by (6.27) is an optimal problem subject to a nonlinear constraint. It seems to be difficult to derive an explicit expression for the backward error $\eta(\widetilde{X})$. In this section we only give some estimates for $\eta(\widetilde{X})$.

Define

$$\begin{aligned}&\mathrm{vec}(\Delta A) = x_1 + iy_1, \quad \mathrm{vec}(\Delta B) = x_2 + iy_2, \quad \mathrm{vec}(\Delta Q) = x_3 + iy_3,\\ &\mathrm{vec}(R) = r + is, \quad \mathrm{vec}\big((\Delta A)^*\widetilde{X}\Delta B + (\Delta B)^*\widetilde{X}\Delta A\big) = a + ib,\\ &I \otimes (B^*\widetilde{X}) = U_1 + i\Omega_1, \quad \big((\widetilde{X}B)^T \otimes I\big)\Pi = U_2 + i\Omega_2,\end{aligned}$$

$$
\begin{aligned}
&I \otimes (A^* \widetilde{X}) = V_1 + i\Gamma_1, \quad \big((\widetilde{X}A)^T \otimes I\big)\Pi = V_2 + i\Gamma_2, \\
&U_c = \begin{bmatrix} U_1 + U_2 & \Omega_2 - \Omega_1 \\ \Omega_1 + \Omega_2 & U_1 - U_2 \end{bmatrix}, \quad V_c = \begin{bmatrix} V_1 + V_2 & \Gamma_2 - \Gamma_1 \\ \Gamma_1 + \Gamma_2 & V_1 - V_2 \end{bmatrix}, \\
&T_c = (\alpha U_c, \beta V_c, \rho I_{2n^2}), \\
&g = \left(\frac{x_1^T}{\alpha}, \frac{y_1^T}{\alpha}, \frac{x_2^T}{\beta}, \frac{y_2^T}{\beta}, \frac{x_3^T}{\rho}, \frac{y_3^T}{\rho}\right)^T,
\end{aligned} \tag{6.30}
$$

where Π is the vec-permutation. Using these symbols (6.29) can be rewritten as

$$
T_{cg} = -\begin{pmatrix} r \\ s \end{pmatrix} - \begin{pmatrix} a \\ b \end{pmatrix}. \tag{6.31}
$$

Since $\rho > 0$, the $2n^2 \times 6n^2$ matrix T_c is of full row rank, and hence, $T_c T_c^\dagger = I_{2n^2}$, which implies that every solution to the equation

$$
g = -T_c^\dagger \begin{pmatrix} r \\ s \end{pmatrix} - T_c^\dagger \begin{pmatrix} a \\ b \end{pmatrix} \tag{6.32}
$$

must be a solution to the equation (6.31). Consequently, for any solution g to the Eq. (6.32) we have

$$
\eta(\widetilde{X}) \le \|g\|_2. \tag{6.33}
$$

Let

$$
\gamma = \left\| T_c^\dagger \begin{pmatrix} r \\ s \end{pmatrix} \right\|_2, \quad \tau = \|T_c^\dagger\|_2^{-1}, \quad \mu = \|\widetilde{X}\|_2, \tag{6.34}
$$

and define

$$
\mathcal{L}(g) = -T_c^\dagger \begin{pmatrix} r \\ s \end{pmatrix} - T_c^\dagger \begin{pmatrix} a \\ b \end{pmatrix}. \tag{6.35}
$$

Then we have

$$
\begin{aligned}
\|\mathcal{L}(g)\|_2 &\le \gamma + \frac{1}{\tau}\left\| \begin{pmatrix} a \\ b \end{pmatrix} \right\|_2 = \gamma + \frac{\alpha\beta}{\tau}\left\| \left(\frac{\Delta A}{\alpha}\right)^* \widetilde{X} \frac{\Delta B}{\beta} + \left(\frac{\Delta B}{\beta}\right)^* \widetilde{X} \frac{\Delta A}{\alpha} \right\|_F \\
&\le \gamma + \frac{2\alpha\beta\mu}{\tau}\left\|\frac{\Delta A}{\alpha}\right\|_F \left\|\frac{\Delta B}{\beta}\right\|_F \\
&\le \gamma + \frac{\alpha\beta\mu}{\tau}\left(\left\|\frac{\Delta A}{\alpha}\right\|_F^2 + \left\|\frac{\Delta B}{\beta}\right\|_F^2\right) \\
&\le \gamma + \frac{\alpha\beta\mu}{\tau}\|g\|_2^2.
\end{aligned} \tag{6.36}
$$

Consider the equation

$$\xi = \gamma + \frac{\alpha\beta\mu}{\tau}\xi^2. \tag{6.37}$$

It is easy to verify that if

$$\gamma \le \frac{\tau}{4\alpha\beta\mu}, \tag{6.38}$$

then the Eq. (6.37) has the positive number

$$\xi_1 = \frac{2\gamma\tau}{\tau + \sqrt{\tau^2 - 4\alpha\beta\mu\gamma\tau}} \tag{6.39}$$

as its smallest positive real root. Thus, it follows from (6.36) that

$$\|g\|_2 \le \xi_1 \implies \|\mathcal{L}(g)\|_2 \le \xi_1. \tag{6.40}$$

Therefore, by the Schauder fixed-point theorem, there exists a g_* satisfying $\|g_*\|_2 \le \xi_1$ such that $\mathcal{L}(g_*) = g_*$, which means that g_* is a solution to the Eq. (6.32), and hence it follows from (6.33) that

$$\eta(\widetilde{X}) \le \|g_*\|_2 \le \xi_1, \tag{6.41}$$

i.e., ξ_1 is an upper bound for $\eta(\widetilde{X})$. Next we derive a lower bound for $\eta(\widetilde{X})$. Suppose that $\left(\frac{\Delta A_{\min}}{\alpha}, \frac{\Delta B_{\min}}{\beta}, \frac{\Delta Q_{\min}}{\rho}\right)$ satisfies

$$\eta(\widetilde{X}) = \left\| \left(\frac{\Delta A_{\min}}{\alpha}, \frac{\Delta B_{\min}}{\beta}, \frac{\Delta Q_{\min}}{\rho} \right) \right\|_F. \tag{6.42}$$

Then we have

$$T_c g_{\min} = -\begin{pmatrix} r \\ s \end{pmatrix} - \begin{pmatrix} a_{\min} \\ b_{\min} \end{pmatrix}, \tag{6.43}$$

where

$$\begin{aligned} a_{\min} + ib_{\min} &= \operatorname{vec}\big((\Delta A_{\min})^* \widetilde{X} \Delta B_{\min} + (\Delta B_{\min})^* \widetilde{X} \Delta A_{\min}\big), \\ x_{1,\min} + iy_{1,\min} &= \operatorname{vec}(\Delta A_{\min}), \quad x_{2,\min} + iy_{2,\min} = \operatorname{vec}(\Delta B_{\min}), \\ x_{3,\min} + iy_{3,\min} &= \operatorname{vec}(\Delta Q_{\min}), \\ g_{\min} &= \left(\frac{x_{1,\min}^T}{\alpha}, \frac{y_{1,\min}^T}{\alpha}, \frac{x_{2,\min}^T}{\beta}, \frac{y_{2,\min}^T}{\beta}, \frac{x_{3,\min}^T}{\rho}, \frac{y_{3,\min}^T}{\rho} \right)^T. \end{aligned}$$

Let $T_c = W(D, 0)Z^T$ be a singular value decomposition, where W and Z are orthogonal matrices, $D = \operatorname{diag}(d_1, d_2, \ldots, d_{2n^2})$ with $d_1 \ge \cdots \ge d_{2n^2} > 0$. Substituting this decomposition into (6.43), and letting

$$Z^T g_{\min} = \begin{bmatrix} v \\ * \end{bmatrix} \quad v \in \mathbf{R}^{2n^2},$$

we get

$$v = D^{-1} W^T \left[-\begin{pmatrix} r \\ s \end{pmatrix} - \begin{pmatrix} a_{\min} \\ b_{\min} \end{pmatrix} \right]$$

Then we have

$$\begin{aligned}
\eta(\widetilde{X}) = \|g_{\min}\|_2 &= \left\| \begin{pmatrix} v \\ * \end{pmatrix} \right\|_2 \geq \|v\|_2 \\
&\geq \left\| D^{-1} W^T \begin{pmatrix} r \\ s \end{pmatrix} \right\|_2 - \left\| D^{-1} W^T \begin{pmatrix} a_{\min} \\ b_{\min} \end{pmatrix} \right\|_2 \\
&\geq \left\| T_c^{\dagger} \begin{pmatrix} r \\ s \end{pmatrix} \right\|_2 - \left\| T_c^{\dagger} \begin{pmatrix} a_{\min} \\ b_{\min} \end{pmatrix} \right\|_2 \\
&\geq \gamma - \|T_c^{\dagger}\|_2 \|(\Delta A_{\min})^* \widetilde{X} \Delta B_{\min} + (\Delta B_{\min})^* \widetilde{X} \Delta A_{\min}\|_F \\
&\geq \gamma - \frac{2\alpha\beta\mu}{\tau} \left\| \frac{\Delta A_{\min}}{\alpha} \right\|_F \left\| \frac{\Delta B_{\min}}{\beta} \right\|_F \\
&\geq \gamma - \frac{\alpha\beta\mu}{\tau} \left\| \left(\frac{\Delta A_{\min}}{\alpha}, \frac{\Delta B_{\min}}{\beta} \right) \right\|_F^2 \\
&\geq \gamma - \frac{\alpha\beta\mu}{\tau} \xi_1^2,
\end{aligned} \tag{6.44}$$

in which the last inequality follows from the fact that

$$\left\| \left(\frac{\Delta A_{\min}}{\alpha}, \frac{\Delta B_{\min}}{\beta} \right) \right\|_F \leq \left\| \left(\frac{\Delta A_{\min}}{\alpha}, \frac{\Delta B_{\min}}{\beta}, \frac{\Delta Q_{\min}}{\rho} \right) \right\|_F = \eta(\widetilde{X}) \leq \xi_1.$$

Let now

$$l(\gamma) = \gamma - \frac{\alpha\beta\mu}{\tau} \xi_1^2 = \gamma - \frac{\alpha\beta\mu}{\tau} \left(\frac{2\gamma\tau}{\tau + \sqrt{\tau^2 - 4\alpha\beta\mu\gamma\tau}} \right)^2.$$

If we can prove that $l(\gamma) > 0$, then (6.44) just gives a useful lower bound for $\eta(\widetilde{X})$. Therefore, we now devote to proving that $l(\gamma) > 0$. Since ξ_1 is a solution to the Eq. (6.37), we have

$$\xi_1 = \gamma + \frac{\alpha\beta\mu}{\tau} \xi_1^2,$$

and hence we have

$$l(\gamma) = \gamma - \frac{\alpha\beta\mu}{\tau} \xi_1^2 = 2\gamma - \xi_1 = \frac{2\gamma\sqrt{\tau^2 - 4\alpha\beta\mu\gamma\tau}}{\tau + \sqrt{\tau^2 - 4\alpha\beta\mu\gamma\tau}} > 0.$$

In summary, we have proved the following theorem.

Theorem 1.3 *Let $A, B, Q, \widetilde{X} \in \mathbf{C}^{n\times n}$ be given matrices, $\eta(\widetilde{X})$ be the backward error defined by (6.27), and let the scalars γ, τ, μ be defined by (6.34). If $\gamma < \frac{\tau}{4\alpha\beta\mu}$, then we have*

$$0 < l(\gamma) \le \eta(\widetilde{X}) \le u(\gamma), \tag{6.45}$$

where

$$u(\gamma) = \frac{2\gamma\tau}{\tau + \sqrt{\tau^2 - 4\alpha\beta\mu\gamma\tau}}, \quad l(\gamma) = \gamma - \frac{\alpha\beta\mu}{\tau}u^2(\gamma). \tag{6.46}$$

Remark 4.1 The functions $u(\gamma)$ and $l(\gamma)$ defined by (6.46) have the Taylor expansions

$$u(\gamma) = \gamma + \frac{\alpha\beta\mu}{\tau}\gamma^2 + O(\gamma^3)$$

and

$$l(\gamma) = \gamma - \frac{\alpha\beta\mu}{\tau}\gamma^2 + O(\gamma^3),$$

respectively. Consequently, when γ is sufficiently small, we have

$$\gamma - \frac{\alpha\beta\mu}{\tau}\gamma^2 \lesssim \eta(\widetilde{X}) \lesssim \gamma + \frac{\alpha\beta\mu}{\tau}\gamma^2. \tag{6.47}$$

6.5 Numerical Examples

To illustrate the results of the previous sections, in this section some simple examples are given, which were carried out using MATLAB 6.5 on a PC Pentium IV/1.7G computer, with machine epsilon $\epsilon \approx 2.2 \times 10^{-16}$.

Example 5.1 Consider the mixed-type Lyapunov matrix equation (6.1) with $A = \alpha J, B = \beta J, X = 4I_2$ and $Q = X - A^*XB - B^*XA$, where

$$J = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.$$

Take $\alpha = \beta = 0.5$, and suppose that the perturbations in the coefficient matrices are

$$\Delta A_k = 10^{-k} \times \begin{bmatrix} 0.901 & 0.402 \\ 0.332 & 0.451 \end{bmatrix}, \quad \Delta B_k = 10^{-k} \times \begin{bmatrix} 0.778 & 0.231 \\ -0.343 & 0.225 \end{bmatrix},$$

$$\Delta Q_k = 10^{-k} \times \begin{bmatrix} 0.401 & 0.225 \\ 0.331 & -0.429 \end{bmatrix}.$$

In this case the relative condition number $c_{\text{rel}}(X) = 8.6603$, which is computed by the formula given as in Remark 3.1. By Theorem 1.1, we can compute perturbation bounds $\delta_*^{(k)}$:

$$r^{(k)} \equiv \|X - X^{(k)}\| \le \delta_*^{(k)},$$

where $X^{(k)}$ are the solutions of the mixed-type Lyapunov equation (6.1) with the coefficient matrices $A_k = A + \Delta A_k, B_k = B + \Delta B_k$ and $Q_k = Q + \Delta Q_k$, respectively. Some results are listed in Table 6.1.

On the other hand, take $\alpha = 0.5$ and $\beta = 0.9998$. In this case the relative condition number is $c_{\text{rel}}(X) = 32396.61$. This shows that the solution X is ill-conditioned. However, we can still compute the perturbation bounds, which are shown in Table 6.2.

The results listed in Tables 6.1 and 6.2 show that the perturbation bound given by Theorem 1.1 is relatively sharp.

Example 5.2 Consider the mixed-type Lyapunov equation (6.1) with the coefficient matrices $A = \frac{1}{2}J, B = \beta_k J, X = I_2$ and $Q = X - A^*XB - B^*XA$, where

$$\beta_k = 1 - 10^{-k}, \quad J = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Suppose that the perturbations in the coefficient matrices are

$$\Delta A = 10^{-10} \times \begin{bmatrix} 0.491 & 0.962 \\ 0.342 & 0.471 \end{bmatrix}, \quad \Delta B = 10^{-10} \times \begin{bmatrix} 0.478 & 0.232 \\ 0.413 & -0.535 \end{bmatrix},$$
$$\Delta Q = 10^{-10} \times \begin{bmatrix} 0.128 & 0.625 \\ 0.331 & -0.429 \end{bmatrix}.$$

Some numerical results on the relative perturbation bounds $\delta_*/\|X\|, r_*$ and $c_{\text{rel}}(X)$ are shown in Table 6.3, where δ_* is as in (6.13), $r_* = \|\widetilde{X} - X\|/\|X\|$, and $c_{\text{rel}}(X)$ is given in Remark 3.1. The results listed in Table 6.3 show that the relative perturbation bound $\delta_*/\|X\|$ is fairly sharp, even if in the case the solution X is ill-conditioned.

Table 6.1

k	6	7	8	9	10
$r^{(k)}$	1.854×10^{-5}	1.854×10^{-6}	1.854×10^{-7}	1.854×10^{-8}	1.854×10^{-9}
$\delta_*^{(k)}$	5.313×10^{-5}	5.313×10^{-6}	5.313×10^{-7}	5.313×10^{-8}	5.313×10^{-9}

Table 6.2

k	6	7	8	9	10
$r^{(k)}$	5.261×10^{-2}	5.201×10^{-3}	5.195×10^{-4}	5.194×10^{-5}	5.194×10^{-6}
$\delta_*^{(k)}$	2.154×10^{-1}	2.056×10^{-2}	2.046×10^{-3}	2.045×10^{-4}	2.045×10^{-5}

Table 6.3

k	1	2	3	4	5
$\delta_*/\|\|X\|\|$	2.667×10^{-9}	2.792×10^{-8}	2.805×10^{-7}	2.806×10^{-6}	2.806×10^{-5}
r_*	2.416×10^{-9}	2.619×10^{-8}	2.640×10^{-7}	2.642×10^{-6}	2.642×10^{-5}
$c_{\text{rel}}(X)$	12.77	140.01	1.412×10^{-3}	1.414×10^{-4}	1.414×10^{-5}

Table 6.4

j	$\|\widetilde{X} - X\|_F$	γ	$l(\gamma)$	$u(\gamma)$
1	0.1149	0.0149	0.0144	0.0154
3	0.1149×10^{-2}	0.1525×10^{-3}	0.1524×10^{-3}	0.1525×10^{-3}
5	0.1149×10^{-4}	0.1525×10^{-5}	0.1525×10^{-5}	0.1525×10^{-5}
7	0.1149×10^{-6}	0.1525×10^{-7}	0.1525×10^{-7}	0.1525×10^{-7}
9	0.1149×10^{-8}	0.1525×10^{-9}	0.1525×10^{-9}	0.1525×10^{-9}

Example 5.3 Consider the mixed-type Lyapunov equation (6.1) with the coefficient matrices

$$A = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \end{bmatrix}, \quad Q = X - A^*XB - B^*XA,$$

where $X = \text{diag}(1, 2, 3), B$ is a 3×3 Hilbert matrix. Let now

$$\widetilde{X} = X + 10^{-j} \times \begin{bmatrix} 0.5 & -0.1 & 0.2 \\ -0.1 & 0.3 & 0.6 \\ 0.2 & 0.6 & -0.4 \end{bmatrix}$$

be an approximate solution. Take $\alpha = \|A\|_F, \beta = \|B\|_F$ and $\rho = \|Q\|_F$ in Theorem 1.3. Some numerical results on lower and upper bounds for the backward error $\eta(\widetilde{X})$ are displayed in Table 6.4.

From the results listed in Table 6.4 we see that the backward error of $\widetilde{X}$ decreases as the error $\|\widetilde{X} - X\|_F$ decreases, and moreover, we see that for smaller γ (e.g., $\gamma < 10^{-4}$) we can get a quite better estimate for the backward error $\eta(\widetilde{X})$ by taking γ as an approximation to $u(\gamma)$ or $l(\gamma)$.

6.6 Conclusion

In this paper we first give a perturbation bound for the solution X to the mixed-type Lyapunov equation (6.1). Then we derive an explicit expression of the condition number for the solution X to the mixed-type Lyapunov equation (6.1). Moreover,

we give an upper and lower bounds of the backward error for an approximate solution to the mixed-type Lyapunov equation (6.1). Numerical examples show that our estimate is fairly sharp.

Acknowledgments This research was supported in part by NSFC under grant 10571007.

References

1. Cheng MS (2004) Theory and methods of nonlinear matrix equations $X \pm A^*X^{-2}A = I$. Ph.D. Dissertation, Peking University
2. El-Sayed SM (2001) Two iteration processes for computing positive definite solutions of the equation $X - A^*X^{-n}A = Q$. Computer Math Appl 41:579–588
3. Graham A (1981) Kronecker products and matrix calculus: with applications. Ellis Horwood Limited, Chichester
4. Ivanov IG, Hasanov VI, Minchev BV (2001) On matrix equations $X \pm A^*X^{-2}A = I$. Linear Algebra Appl 326:27–44
5. Liu XG, Gao H (2003) On the positive definite solutions of the matrix equations $X^s \pm A^TX^{-t}A = I_n$. Linear Algebra Appl 368:83–97
6. Rice JR (1966) A theory of condition. J SIAM Numer Anal 3:287–310
7. Sun JG (1997) Backward error for the discrete-time algebraic Riccati equation. Linear Algebra Appl 259:183–208
8. Sun JG, Xu SF (2003) Perturbation analysis of the maximal solution of the matrix equation $X + A^*X^{-1}A = P$. II. Linear Algebra Appl 362:211–228
9. Xu SF, Cheng MS (2006) On the solvability for the mixed-type Lyapunov equation. IMA J Appl Math 71:287–294

Chapter 7
Numerical and Symbolical Methods for the GCD of Several Polynomials

Dimitrios Christou, Nicos Karcanias, Marilena Mitrouli and Dimitrios Triantafyllou

Abstract The computation of the Greatest Common Divisor (GCD) of a set of polynomials is an important issue in computational mathematics and it is linked to Control Theory very strong. In this paper we present different matrix-based methods, which are developed for the efficient computation of the GCD of several polynomials. Some of these methods are naturally developed for dealing with numerical inaccuracies in the input data and produce meaningful approximate results. Therefore, we describe and compare numerically and symbolically methods such as the ERES, the Matrix Pencil and other resultant type methods, with respect to their complexity and effectiveness. The combination of numerical and symbolic operations suggests a new approach in software mathematical computations denoted as *hybrid computations*. This combination offers great advantages, especially when we are interested in finding approximate solutions. Finally the notion of approximate GCD is discussed and a useful criterion estimating the *strength* of a given approximate GCD is also developed.

D. Christou · N. Karcanias
School of Engineering and Mathematical Sciences, Control Engineering Research Centre, City University, Northampton Square, London, EC1V 0HB, UK
e-mail: dchrist@math.uoa.gr

N. Karcanias
e-mail: N.Karcanias@city.ac.uk

M. Mitrouli (✉) · D. Triantafyllou
Department of Mathematics, University of Athens, Panepistemiopolis, 15784 Athens, Greece
e-mail: mmitroul@math.uoa.gr

D. Triantafyllou
e-mail: dtriant@math.uoa.gr

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_7,

7.1 Introduction

The problem of finding the greatest common divisor (GCD) of a polynomial set has been a subject of interest for a very long time and has widespread applications. Since the existence of a nontrivial common divisor of polynomials is a property that holds for specific sets, extra care is needed in the development of efficient numerical algorithms calculating correctly the required GCD. Several numerical methods for the computation of the GCD of a set $P_{m,n}$, of m polynomials of $\Re[s]$ of maximal degree n, have been proposed, [2–4, 11, 20, 24, 26, 28–30, 32, 33, 36] and references therein. These methods can be classified as:

(i) Numerical methods based on Euclid's algorithm and its generalizations.
(ii) Numerical methods based on procedures involving matrices (matrix based methods).

The methods that are based on Euclid's algorithm, are designed for processing two polynomials and they are applied iteratively for sets of more than two polynomials. On the other hand, the matrix-based methods usually perform specific transformations to a matrix formed directly from the coefficients of the polynomials of the entire given set.

The GCD has a significant role in Control Theory [15, 31]. A number of important invariants for Linear Systems rely on the notion of Greatest Common Divisor (GCD) of several polynomials. In fact, it is instrumental in defining system notions such as zeros, decoupling zeros, zeros at infinity or notions of Minimality of system representations. On the other hand, Systems and Control Methods provide concepts and tools, which enable the development of new computational procedures for GCD [16].

The existence of certain types and/or values of invariants and system properties may be classified as *generic* or *nongeneric* on a family of linear models. Numerical computations dealing with the derivation of an approximate value of a property, function, which is nongeneric on a given model set, will be called nongeneric computations (NGC) [16]. Computational procedures aiming at defining the generic value of a property, function on a given model set (if such values exists), will be called generic (GC). On a set of polynomials with coefficients taking values from a certain parameter set, the existence of GCD is nongeneric [14, 35]; numerical procedures that aim to produce an approximate nontrivial value by exploring the numerical properties of the parameter set are typical examples of NGC computations and approximate GCD procedures will be considered subsequently. NG computations refer to both continuous and discrete type system invariants. The various techniques, which have been developed for the computation of approximate solutions of GCD [19, 26] and LCM (Least Common Multiple) [16, 21, 22], are based on methodologies where exact properties of these notions are relaxed and appropriate solutions are sought using a variety of numerical tests.

The development of a methodology for robust computation of nongeneric algebraic invariants, or nongeneric values of generic ones, has as prerequisites:

(a) The development of a numerical linear algebra characterization of the invariants, which may allow the measurement of degree of presence of the property on every point of the parameter set.
(b) The development of special numerical tools, which avoid the introduction of additional errors.
(c) The formulation of appropriate criteria which, allow the termination of algorithms at certain steps and the definition of meaningful approximate solutions to the algebraic computation problem.

It is clear that the formulation of the algebraic problem as an equivalent numerical linear algebra problem, is essential in transforming concepts of algebraic nature to equivalent concepts of analytic character and thus setup up the right framework for approximations.

A major challenge for the control theoretic applications of the GCD is that frequently we have to deal with a very large number of polynomials. It is this requirement that makes the pairwise type approaches for GCD [1, 24, 28, 36] not suitable for such applications [32]. However, because of the use of the entire set of polynomials, matrix-based methods tend to have better performance and quite good numerical stability, especially in the case of large sets of polynomials [2–4, 10, 20, 26].

The study of the invariance properties of the GCD [18] led to the development of the ERES method [26], which performs extensive row operations and shifting on a matrix formed directly from the coefficients of the polynomials. The ERES method has also introduced for the first time a systematic procedure for computing *approximate* GCDs [19] for a set of polynomials and extends the previously defined notion of *almost zeros* [17]. The notion of *almost zeros* is linked to the *approximate GCD* problem [19] and it is based on a relaxation of the exact notion of a zero.

Another algorithm for the GCD computation based on Systems Theory and Matrix Pencils, was introduced in 1994, [20], and a variant using generalized resultant matrices was presented in 2006 [23].

The implementation of matrix-based methods computing the GCD in a programming environment often needs a careful selection of the proper arithmetic system. Most modern mathematical software packages use variable floating-point or exact symbolic arithmetic. If symbolic arithmetic is used, the results are always accurate, but the time of execution of the algorithms can be prohibitively high. In variable precision floating-point arithmetic the internal accuracy of the system can be determined by the user. Variable precision operations are faster and more economical in memory bytes than symbolic operations, but if we increase the number of digits of the system's accuracy, the time and memory requirements will also increase. An alternative approach is to combine symbolic with floating-point operations of enough digits of accuracy in an appropriate way. Such combination

will be referred as *Hybrid Computations*. This technique often improves the performance of the matrix based algorithms.

In the following, we will be mainly concerned with the performance of the ERES, Matrix Pencil, and Resultant ERE methods in a numerical-symbolical computational environment. Also, a useful indicator for the quality of the GCD, known as the *strength of an approximate GCD* [19], is described.

7.2 The ERES, Resultant ERE (RERE) and Modified RERE (MRERE) Methods

In this section, we present the description of the two methods for computing the GCD of several polynomials using *Extended Row Equivalence* (ERE) [16]. Their corresponding algorithms are tested and compared thoroughly and representative examples are given in tables.

Suppose that we have a set of m polynomials:

$$\mathcal{P}_{m,n} = \{a(s), b_i(s) \in \Re[s], i = 1, 2, \ldots, m-1 \text{ with}$$
$$n = \deg\{a(s)\} \text{ and } p = \max_{1 \le i \le m-1} \{\deg\{b_i(s)\}\} \le n\}$$

with the following form:

$$a(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0$$

$$b_i(s) = b_{i,n} s^p + b_{i,n-1} s^{p-1} + \cdots + b_{i,n-p+1} s + b_{i,n-p}$$

$$\text{for } i = 1, 2, \ldots, m-1$$

and suppose that there is at least one i: $b_{i,n} \neq 0$ but $b_{i,j} = 0$ for $j > n - p \ \forall i$, [2].

For any $\mathcal{P}_{m,n}$ set, we define a vector representative (vr) $\underline{p}_m(s)$ and a basis matrix P_m represented as:

$$\underline{p}_m(s) = [a(s), b_1(s), \ldots, b_{m-1}(s)]^t \tag{7.1a}$$

$$= [\underline{p}_0, \underline{p}_1, \ldots, \underline{p}_{n-1}, \underline{p}_n] \cdot \underline{e}_n(s) = P_m \cdot \underline{e}_n(s) \tag{7.1b}$$

where $P_m \in \Re^{m \times (n+1)}$, $\underline{e}_n(s) = [1, s, \ldots, s^{n-1}, s^n]^t$.

The basis matrix P_m is formed directly from the coefficients of the polynomials of the set.

Additionally, for any vector of the form:

$$\underline{r}^t = [0, \ldots, 0, a_k, \ldots, a_d] \in \Re^d, \ a_k \neq 0$$

we define the *Shifting* operation

$$shf: \ shf(\underline{r}^t) = [a_k, \ldots, a_d, 0, \ldots, 0] \in \Re^d$$

In the following, without loss of generality, we suppose that the GCD of a given set of polynomials has no zero roots.

7.2.1 The ERES Method

The ERES method is an iterative matrix based method, which is based on the properties of the GCD as an invariant of the original set of polynomials under extended-row-equivalence and shifting operations [18]. The algorithm of the ERES method [26, 27] is based on stable algebraic processes, such as Gaussian elimination with partial pivoting scaling, normalization and Singular Value Decomposition, which are applied iteratively on a basis matrix formed directly from the coefficients of the polynomials of the original set. Thus, the ERES algorithm works with all the polynomials of a given set simultaneously. The main target of the ERES algorithm is to reduce the number of the rows of the initial matrix and finally to end up to a unity rank matrix, which contains the coefficients of the GCD. The Singular Value Decomposition provides the ERES algorithm with a termination criterion. The performance of the algorithm is better [5] if we perform hybrid computations. The following algorithm corresponds to an implementation of the ERES method in a hybrid computational environment.

7.2.1.1 The ERES Algorithm

– Form the basis matrix $P_m \in \Re^{m\times(n+1)}$.
– Convert the elements of P_m to a rational format:

$$P_m^{(0)} := \text{convert}(P_m, \text{rational})$$

– Initialize k:= -1

Repeat k:= $k + 1$
STEP 1: Let r := the row dimension of $P_m^{(k)}$.
Specify the degree d_i of each polynomial row.
Reorder matrix $P_m^{(k)}: d_{i-1} \le d_i, \ i = 2, \ldots, r$.
If $d_1 = d_2 = \cdots = d_r$ **then**

Convert the elements of $P_m^{(k)}$ to a floating-point format:
$P_m^{(F)} := \text{convert}(P_m^{(k)}, \text{float})$

Normalize the rows of $P_m^{(F)}$ using norm-2:
$P_m^{(N)} := \text{Normalize}(P_m^{(F)})$

Compute the singular value decomposition:
$P_m^{(N)} := V \Sigma W^t, \Sigma = diag\{\sigma_1, \ldots, \sigma_r\}$,
$\sigma_1 > \sigma_2 \geq \cdots \geq \sigma_r$ and $W^t = [\underline{w}_1, \ldots, \underline{w}_{n+1}]^t$
If ε_t-rank$(P_m^{(N)}) = 1$ **then**

Select the GCD vector $\underline{g}$.

($\underline{g} := \underline{w}_1^t$ or $\underline{g} :=$ any row of $P_m^{(k)}$)
quit

STEP 2: Scale properly matrix $P_m^{(k)}$.
Apply Gaussian elimination with partial pivoting to $P_m^{(k)}$.
STEP 3: Apply shifting on every row of $P_m^{(k)}$.
Delete the zero rows and columns.
until $r = 1$

The ERES algorithm produces either a single row-vector or a unity rank matrix. The main advantage of this algorithm is the reduction of the size of the original matrix during the iterations, which leads to fast data processing and low memory consumption.

7.2.1.2 Complexity

For a set of m polynomials the amount of floating point operations performed in the kth iteration of the algorithm depends on the size of the matrix $P_m^{(k)}$. If the size of $P_m^{(k)}$ is $m' \times n'$, the ERES algorithm requires $O(\frac{z^3}{3})$, $z = \min\{m' - 1, n'\}$ operations for the Gaussian elimination, $O(2m'n')$ operations for the normalization and $O(m'n'^2 + n'^3)$ for the SVD process. The first iteration is the most computationally expensive iteration since the initial matrix $P_m^{(0)}$ has larger dimensions than any $P_m^{(k)}$. Unless we know exactly the degree of the GCD of the set we cannot specify from the beginning the number of iterations required by the algorithm. Therefore, we cannot express a general formula for calculating the total number of operations, which are required by the algorithm.

7.2.1.3 Behavior and Stability of the ERES Algorithm

The combination of rational and numerical operations aims at the improvement of the stability of the ERES algorithm and the presence of *good* approximate solutions. The main iterative procedure of the algorithm and especially the process of Gaussian elimination, is entirely performed by using rational operations. With this

technique any additional errors from the Gaussian elimination are avoided. The operations during the Gaussian elimination are always performed accurately and if the input data are exactly known and a GCD exists, the output of the algorithm is produced accurately from any row of the final unity rank matrix. Obviously, rational operations do not reveal the presence of approximate solutions. In cases of sets of polynomials with inexact coefficients, the presence of an approximate solution relies on the proper determination of a numerical ε_t-rank 1 matrix for a specific accuracy ε_t. Therefore, the singular value decomposition together with the normalization process of the matrix $P_m^{(k)}$ are performed by using floating-point operations. The polynomial that comes from the right singular vector that corresponds to the unique singular value of the last unity rank matrix, can be considered as a GCD approximation and represents the numerical output of the ERES algorithm.

The normalization of the rows of any matrix $P_m^{(k)}$ (by the Euclidean norm) does not introduce significant errors and in fact the following result can be proved [26]:

Proposition 7.1 *The normalization $P_m^{(N)}$ of a matrix $P_m^{(k)} \in \Re^{m' \times n'}$, computed by the method in the k^{th} iteration, using floating-point arithmetic with unit round-off u, satisfies the properties*

$$P_m^{(N)} = N \cdot P_m^{(k)} + E_N, \quad \|E_N\|_\infty \leq 3.003 \cdot n' \cdot u$$

where $N \in \Re^{m' \times m'} = \mathrm{diag}(d_1, d_2, \ldots, d_{m'})$, $d_i = \left(\left\| P_m^{(k)}[i, 1, \ldots, n'] \right\|_2 \right)^{-1}$, $i = 1, \ldots, m'$ *the matrix accounting for the performed transformations and* $E_N \in \Re^{m' \times n'}$ *the error matrix.*

It is important to notice that the SVD is actually applied to a numerical copy of the matrix $P_m^{(k)}$ and thus the performed transformations during the SVD procedure do not affect the matrix $P_m^{(k)}$ when returning to the main iterative procedure. For this reason, there is *no accumulation* of numerical errors. The only errors appearing are from the normalization and the singular value decomposition [8, 12] of the last matrix $P_m^{(F)}$ and represent the total numerical error[1] of the ERES algorithm.

The combination of rational-symbolic and floating-point operations ensures the stability of the algorithm and gives to the ERES the characteristics of a hybrid computational method.

[1] The numerical error which occurs from the conversion between rational and floating-point data is close to the software accuracy.

7.2.2 The Resultant ERE (RERE) and Modified ERE (MRERE) Methods

Another method to compute the GCD of several polynomials is to triangularize the generalized Sylvester matrix S [2], which has the following form:

$$S = \begin{bmatrix} S_0 \\ S_1 \\ \vdots \\ S_m \end{bmatrix}$$

where the block $S_i, i = 1, \ldots, m$ represents the Sylvester matrix of the i-th polynomial. In [6] we have modified the huge initial generalized Sylvester matrix in order to take advantage of its special form and modifying the classical procedures (such as SVD, QR and LU factorization) we reduce the required floating point operations from $O(n^4)$ to $O(n^3)$ flops making the algorithms efficient. More specifically the application of Householder or Gaussian transformations to the modified generalized Sylvester matrix requires only $O((n+p)^3(2\log_2(n) - \frac{1}{3}) + (n+p)^2(2m\log_2(n) + p))$ flops and in the worse case where $m = n = p$ the required flops will be equal to $O(16\log_2(n) \cdot n^3)$ or the half flops for the LU factorization respectively. In practice the flops that demand the previous methods are less because of the linear dependent rows which are zeroed and deleted during the triangularization of the matrix.

7.2.2.1 Numerical Stability

In [6] we proved that the final error matrix in the modified QR method is $E = \sum_{i=1}^{\log_2(n)} E_i$ with

$$\| E \|_F \leq \phi(n)\mu\sqrt{n+p}(\| S^* \|_F + \| ((S^*)') \|_F) \tag{7.2}$$

where $((S^*)')$ is the last triangularized sub-matrix, ϕ a slowly growing function of n and μ the machine precision and the final error in the modified LU method is

$$E = \sum_{i=1}^{\log_2(n)} E_i \tag{7.3}$$

with $\| E \|_\infty \leq (n+p)^{[\log_2 n]} pu \| S^* \|_\infty + (n+p)^2 pu \| ((S^*)') \|_\infty)$, where p is the growth factor and u the unit round off.

7.3 The Matrix Pencil Methods

7.3.1 The Standard Matrix Pencil Method (SMP)

The matrix pencil method [20, 23] is a direct method, which is based on system properties of the GCD. The algorithm of the SMP method uses stable processes, such as SVD for computing the right $\mathcal{N}_r$ and left $\mathcal{N}_l$ null spaces of appropriate matrices. The SMP method requires the construction of the observability matrix $Q(\hat{A}, \hat{C}) = [\hat{C}^t, \hat{A}^t\hat{C}^t, \ldots, (\hat{A}^t)^{(d-1)}\hat{C}^t]^t$ of the companion matrix $\hat{A}$ of the polynomial of maximal degree and the left null space $\hat{C}$ of a matrix M_1 as we will see below. It is known that the computation of powers of matrices is not always stable. As it is shown in [23], because of the special form of the companion matrix A and the orthogonality of C, the powers of A and their product with C can fail only if it holds very special formulas between the coefficients of the polynomials ([23] example (8)). An alternative way is this computation to be done symbolically: there will be no rounding off errors and because only an inner product must be computed for the last column for every computation of a power of A (the other columns are the columns of the previous power of A left shifted), the increase of the computational time because of the rational representation of the coefficients and the symbolical computation of the products is not very considerable. In this manner we achieve to compute the observability matrix avoiding one of the main disadvantages of the SMP method (the computation of the powers of A) without significant surcharge of the required time.

Another stable way to compute the null space of Q is first to reduce the pair $(\hat{A}, \hat{C})$ to a block Hessenberg form without computing the observability matrix. From the staircase algorithm [9], we take an orthogonally similar pair $(H, \widetilde{C})$, such that: $Q(\hat{A}, \hat{C}) = P^T[\widetilde{B}, H\widetilde{B}, \ldots, H^{(d-1)}\widetilde{B}]$, where P is an orthogonal matrix such that $P\hat{A}P^T = H$. Because the matrix H has much more elements than the companion matrix A, the computation of the powers of H demands more flops than those of the powers of A and thus in our case it is better to use the first way for the computation of the null space of Q.

The main target of the SMP algorithm is to form the GCD pencil $Z(s)$ and specify any minor of maximal order, which gives the required GCD. This specification can be done symbolically. Let $\mathcal{P}_{m,d}$ be the set of polynomials as defined in Sect. 7.2.

7.3.1.1 The SMP Algorithm

STEP 1: Compute a basis matrix M for the right nullspace $\mathcal{N}_r(P_m)$ using the SVD algorithm.
STEP 2: Construct M_1 by deleting the last row of M.
STEP 3: Compute the matrix $\hat{C}$ such that $\hat{C}M_1 = 0$.

STEP 4: Construct the observability matrix: $Q(\hat{A}, \hat{C}) = [\hat{C}^t, \hat{A}^t\hat{C}^t, \ldots, (\hat{A}^t)^{(d-1)}\hat{C}^t]^t$.
STEP 5: Compute the right nullspace $W = \mathcal{N}_r(Q(\hat{A}, \hat{C}))$ using the SVD algorithm.
STEP 6: Construct the pencil $Z(s) = sW - \hat{A}W$. Any minor of maximal order of $Z(s)$ defines the GCD of set of the polynomials.

7.3.1.2 Complexity

The computation of the right nullspace of P_m requires $O(mn^2 + \frac{11n^3}{2})$ flops, of the of the matrix C demands $O((r-1)n^3 + \frac{11n^3}{2})$ flops, where $r = \rho(P_m)$.

The computation of $\hat{C}$ such that $\hat{C}M_1 = 0$ requires $O(4\mu d^2 + 8d^3)$ flops applying the SVD algorithm to M_1^t, where $\mu = d - r + 1, r = \rho(P_m)$ (*). The computation of any minor of maximal order of $Z(s)$ can be done symbolically using the LU factorization.

Totally the Standard Matrix Pencil method demands: $O(4md^2 + 4d^3(r-1) + 4\mu d^2 + 24d^3 + \frac{3}{2}d^2r)$ flops. If the computation of Q is done symbolically the required flops are diminished by the flops in (*) but the computational time is increased slightly.

7.3.1.3 Numerical Stability

The Standard Matrix Pencil Method requires two SVD calls and the construction of the observability matrix.

The numerical computation of the powers of A $(\hat{A})^{(k)}$ is in practise stable. Of course there are no errors in symbolically implementation of this step. Since the matrix $(\hat{A})^{(k)}$ is computed, the numerical computation of the product $(\hat{A}^t)^{(k)}\hat{C}^t = (\hat{C}(\hat{A})^{(k)})^t$ is stable because the matrix C is orthonormal. For the last matrix multiplication it holds: $fl(\hat{C}(\hat{A})^{(k)}) = \hat{C}(\hat{A})^{(k)} + E$, with $\|E\|_2 \leq d^2u_1\|C\|_2\|A^k\|_2 = d^2u_1\|A^k\|_2$, where $fl(\cdot)$ denotes the computed floating point number and u_1 is of order of unit round off. The computation of the minor of maximal order of $Z(s)$ is done symbolically and so there are no rounding off errors.

7.3.2 The Modified Resultant Matrix Pencil Method (MRMP)

The modified matrix pencil method [23] is a similar with the MP method, which is based to the modified Sylvester matrix S^*, it constructs another GCD pencil $Z(s)$ and specify any minor of maximal order, which gives the required GCD. This specification can also be done symbolically.

7.3.2.1 The MRMP Algorithm

STEP 1: Define a basis $\widetilde{M}$ for the right nullspace of the modified Sylvester matrix S^* using the modified QR factorization in first phase of SVD.

STEP 2: Define the Matrix Pencil $\widetilde{Z}(s) = s\widetilde{M}_1 - \widetilde{M}_2$ for the Resul-tant set, where $\widetilde{M}_1$, $\widetilde{M}_2$ are the matrices obtained from $\widetilde{M}$ by deleting the last and the first row of $\widetilde{M}$ respectively.

STEP 3: Compute any non-zero minor determinant $d(s)$ of $\widetilde{Z}(s)$ and thus obtain GCD $= d(s)$.

7.3.2.2 Complexity

The Modified Resultant Matrix Pencil method requires $O((n+p)^3(2log_2(n) - \frac{1}{3}) + (n+p)^2(2m\log_2(n) + p) + 12k(n+p)^2)$, where k is the number of the calls of the SVD-step.

7.3.2.3 Numerical Stability

The computed GCD is the exact GCD of a slightly disrupted set of the initial polynomials. The final error is $E = E_1 + E_2$, with $\|E_1\|_F \leq \varphi(n)u\|S\|_F$ and $\|E\|_2 \leq (\varphi(n) + c(m,n) + c(m,n) \cdot \varphi(n) \cdot u) \cdot u \cdot \|S\|_F$ where u is the unit round off error, $\|\cdot\|_F$ the Frobenius norm and $\varphi(n)$ is a slowly growing function of n [8] and $c(m,n)$ is a constant depending on m, n.

7.3.3 Another Subspace-Based Method for Computing the GCD of Several Polynomials (SS)

The subspace concept is actually very common among several methods for computing the GCD of many polynomials, including those we described in the previous sections. The SVD procedure applied to a generalized Sylvester matrix is the basic tool for a subspace method. A representative and rather simple algorithm, which approaches the GCD problem from the subspace concept, is presented in [30] and we shall refer to it as the *SS algorithm*.

Given a set of univariate polynomials $\mathcal{P}_{m,n}$, the first two steps of the SS algorithm involves the construction of an $m(n+1) \times (2n+1)$ generalized Sylvester matrix Y from the input polynomials and the computation of the left null space of the transposed Y^t via SVD. If we denote by $U_0 \in \Re^{(2n+1)\times k}$ the basis matrix for the computed left null space of Y^t and C is the $(2n+1) \times (2n+1-k)$ Toeplitz matrix of a degree K polynomial with arbitrary coefficients, then the GCD

vector is actually the unique (up to a scalar) solution of the system $U_0^t C = 0$, [30]. Obviously, the degree of the GCD is $k = \text{colspan}\{U_0\}$.

For the approximate GCD problem, an equivalent and more appropriate way to compute the GCD vector with the SS algorithm is to construct k Hankel matrices $\tilde{U}_i \in \Re^{(k+1)\times(2n+1-k)}$, $i = 1, \ldots, k$ from the columns of U_0, form the matrix $\tilde{U} = [\tilde{U}_1, \ldots, \tilde{U}_k] \in \Re^{(k+1)\times k(2n+1-k)}$ and compute a basis matrix V_0 for the left null space of $\tilde{U}$ by using the SVD procedure. The last column of V_0, which corresponds to the smallest singular value (expected to be zero), contains the $k+1$ coefficients of the GCD. The yielded GCD can be considered as an approximate ε-GCD for a tolerance ε equal to the machine's numerical precision. However, for a different tolerance ε, we can select a singular value σ_j from the singular value decomposition of Y^t such that $\sigma_j > \varepsilon \cdot f(h,n)$ and $\sigma_{j+1} \leq \varepsilon$, [7], and compute an ε-GCD of degree $k' = 2n + 1 - j \neq k$.

Although it is not mentioned in [30], the computational cost of the SS algorithm is dominated by the SVD of the generalized Sylvester matrix Y^t, which requires $O(2m^2n^3 + 5m^2n^2)$ flops, [12]. However, the stability and the effectiveness of the algorithm in large sets of polynomials is not well documented in [30] and additionally there not any reference about the total numerical error of the algorithm. Practically, the performance of the SS algorithm is good when using floating-point operations of medium-high accuracy but becomes very slow in hybrid computations.

7.4 Approximate Solutions

It is well known that, when working with inexact data in a computational environment with limited numerical accuracy, the outcome of a numerical algorithm is usually an approximation of the expected exact solution due to the accumulation of numerical errors. In the case of GCD algorithms, the solution produced can be considered either as an approximate solution of the original set of polynomials, within a tolerance ε, or as the exact solution of a perturbed set of polynomials. The following definition is typical for the approximate GCD.

Definition 7.1 Let $\mathcal{P}_{m,n} = \{a(s), b_i(s),\ i = 1, \ldots, m-1\}$ a set of univariate polynomials as defined in (1) and $\varepsilon > 0$ a fixed numerical accuracy. An almost common divisor (ε-divisor) of the polynomials of the set $\mathcal{P}_{m,n}$ is an exact common divisor of a perturbed set of polynomials $\mathcal{P}'_{m,n} = \{a(s) + \Delta a(s), b_i(s) + \Delta b_i(s), i = 1, \ldots, m-1\}$, where the polynomial perturbations satisfy $\deg\{\Delta a(s)\} \leq \deg\{a(s)\}$, $\deg\{\Delta b_i(s)\} \leq \deg\{b_i(s)\}$ and

$$\|\Delta a(s)\|^2 + \sum_{i=1}^{m-1} \|\Delta b_i(s)\|^2 < \varepsilon \tag{7.4}$$

An approximate GCD (ε-GCD) of the set $\mathcal{P}_{m,n}$ is an ε-divisor of maximum degree.

The computation of the GCD of a set polynomials is nongeneric. Generally, in most GCD problems the degree of the GCD is unknown and thus a numerical algorithm can easily produce misleading results. An approach to this problem is to certify and fix a maximum degree according to appropriate theorems and techniques [11, 32] and proceed with the computation of a common divisor of this particular degree. The evaluation of the strength of a given approximation is another important issue here.

The definition of the *approximate GCD* as the exact GCD of a perturbed set has led to the development of a general approach for defining the approximate GCD, evaluating the strength of approximation and finally defining the notion of the optimal *approximate GCD* as a distance problem [19]. In fact, recent results on the representation of the GCD [13, 19], using Toeplitz matrices and generalized resultants (Sylvester matrices), allow the reduction of the approximate GCD computation to an equivalent *approximate factorization* of generalized resultants. Specifically, for a given set of polynomials $\mathcal{P}_{m,n}$ and its GCD of degree k:

$$g(s) = s^k + \lambda_1 s^{k-1} + \cdots + \lambda_k, \ \lambda_k \neq 0$$

it holds that [13]:

$$S_{\mathcal{P}} = [O_k | S_{\mathcal{P}^c}] \cdot \Phi_g \tag{7.5}$$

where $S_{\mathcal{P}}$ is the $(mn+p) \times (n+p)$ Sylvester matrix of the set $\mathcal{P}_{m,n}$, O_k is the $(mn+p) \times k$ zero matrix, $S_{\mathcal{P}^c}$ is the $(mn+p) \times (n+p-k)$ Sylvester matrix of the set $\mathcal{P}^c_{m,n-k}$ of coprime polynomials, obtained from the original set $\mathcal{P}_{m,n}$ after dividing its elements by the GCD $g(s)$ and, finally, Φ_g is the $(n+p) \times (n+p)$ lower triangular Toeplitz-like matrix of the polynomial $g(s)$.

We now define the strength of an r-order approximate common divisor of a polynomial set $\mathcal{P}_{m,n}$ [19]:

Definition 7.2 Let $\mathcal{P}_{m,n}$ and $v(s) \in \Re[s], \deg\{v(s)\} = r \leq p$. The polynomial $v(s)$ is an r-order approximate common divisor of $\mathcal{P}_{m,n}$ and its *strength* is defined as a solution of the following minimization problem:

$$f(\mathcal{P}, \mathcal{P}^c) = \min_{\forall \mathcal{P}^c} \{ \| S_{\mathcal{P}} - [O_r | S_{\mathcal{P}^c}] \cdot \Phi_v \|_F \} \tag{7.6}$$

Furthermore, $v(s)$ is an r-order approximate GCD of $\mathcal{P}_{m,n}$ if the minimum corresponds to a coprime set $\mathcal{P}^c_{m,n-r}$, or to a full rank $S_{\mathcal{P}^c}$.

We prefer to use as a metric the Frobenius matrix norm [8] denoted by $\| \cdot \|_F$, which relates in a direct way to the set of polynomials. However, the minimization problem in Definition 7.2 cannot be solved easily, because it may involve too many arbitrary parameters.

Let us have a polynomial $v(s) \in \Re[s]$ of degree r given as a solution by a GCD algorithm. We consider it as an exact GCD of a perturbed set of polynomials $\mathcal{P}'_{m,n}$ of the form:

$$\mathcal{P}'_{m,n} \triangleq \mathcal{P}_{m,n} - \mathcal{Q}_{m,n} \tag{7.7a}$$

$$= \{p'_i(s) = p_i(s) - q_i(s) : \deg\{q_i(s)\} \leq \deg\{p_i(s)\}, i = 1, \ldots, m\} \tag{7.7b}$$

where $\mathcal{Q}_{m,n}$ denotes the set of polynomial perturbations [13]. The polynomials of the set $\mathcal{Q}_{m,n}$ have arbitrary coefficients, which pass to the polynomials of the set $\mathcal{P}'_{m,n}$. If we use now the respective generalized resultants (Sylvester matrices) for each set in Eq. (7.7a), the following relation appears:

$$S_{\mathcal{P}'} = S_{\mathcal{P}} - S_{\mathcal{Q}} \tag{7.8}$$

It is clear that the exact GCD of a set of polynomials yields $S_{\mathcal{Q}} = O \Rightarrow \|S_{\mathcal{Q}}\|_F = 0$. Therefore, we may consider a polynomial as a good approximation of the exact GCD, if $\|S_{\mathcal{Q}}\|_F$ is close enough to zero. In the following, our intention is to find some bounds for $\|S_{\mathcal{Q}}\|_F$.

If we use the factorization of generalized resultants as described in (6), we will have:

$$\begin{aligned} S_{\mathcal{Q}} &= S_{\mathcal{P}} - [O_r | S_{\mathcal{P}'^c}] \cdot \Phi_v \;\Leftrightarrow \\ S_{\mathcal{Q}} \cdot \Phi_v^{-1} &= S_{\mathcal{P}} \cdot \Phi_v^{-1} - [O_r | S_{\mathcal{P}'^c}] \end{aligned} \tag{7.9}$$

where Φ_v^{-1} is the inverse of Φ_v. It is important to notice here that $\mathcal{P}'^c$ contains arbitrary parameters. We can select specific values for these parameters such as:

$$S_{\mathcal{P}} \cdot \Phi_v^{-1} - [O_r | S_{\mathcal{P}'^c}] = [S^{(r)} | O_{n+p-r}] \equiv \widehat{S} \tag{7.10}$$

Therefore, from Eqs. (7.9) and (7.10) it follows:

$$\begin{aligned} S_{\mathcal{Q}} \cdot \Phi_v^{-1} &= \widehat{S} \\ S_{\mathcal{Q}} &= \widehat{S} \cdot \Phi_v \end{aligned}$$

and, since $\text{Cond}(\Phi_v) = \|\Phi_v\|_F \|\Phi_v^{-1}\|_F \geq n + p$, [8], we conclude with the following inequality:

$$\frac{\|\widehat{S}\|_F}{\|\Phi_v^{-1}\|_F} \leq \|S_{\mathcal{Q}}\|_F \leq \|\widehat{S}\|_F \|\Phi_v\|_F \tag{7.11}$$

If $v(s)$ has the same degree as the exact GCD of the set, the properties

$$\underline{\mathcal{S}_v} = \frac{\|\widehat{S}\|_F}{\|\Phi_v^{-1}\|_F} \quad \text{and} \quad \overline{\mathcal{S}_v} = \|\widehat{S}\|_F \|\Phi_v\|_F \tag{7.12}$$

characterizes the quality of the proximity of $v(s)$ to the exact GCD of the set $\mathcal{P}_{m,n}$ and we shall refer to them as the *minimum and maximum strength numbers* of $v(s)$ respectively.

These strength numbers are useful indicators for the evaluation of the strength of a given approximate GCD. More particularly, if $\underline{\mathcal{S}_v} \geq 1$, then the strength of the given approximation is bad and the opposite holds if $\overline{\mathcal{S}_v} < 1$. Normally, we prefer solutions with maximum strength number $\overline{\mathcal{S}_v} << 1$ or better close to the numerical software accuracy of the system. Otherwise, we have to solve the minimization problem (7) to find the actual strength. The advantage is that the computation of the strength numbers is straightforward and can give us information about the strength of a GCD before we go to an optimization method.

7.4.1 Computational Examples

The previous methods have been applied to many sets of polynomials. The final results using variable floating point, symbolic and hybrid operations are presented in Tables 7.1, 7.2, 7.3, and 7.4. The following notation is used in the tables.

m:	the number of polynomials
n:	the maximum degree of the polynomials
p:	the second maximum degree of the polynomials
d:	the degree of the GCD
Tol :	numerical accuracy ε
Rel:	the numerical relative error

Table 7.1 Comparison of algorithms: $Tol = 10^{-16}$

Example		ERES	RERE (LU)		MRERE (LU)		RERE (QR)	
		Hybrid	Num	Sym	Num	Sym	Num	Sym
I	Dig	16	32	–	32	–	35	–
	Rel	0	0.50×10^{-24}	0	0.50×10^{-24}	0	0.13×10^{-24}	0
	Strength	0	0.20×10^{-22}	0	0.20×10^{-22}	0	0.37×10^{-23}	0
	Time	1.842	0.020	0.120	0.081	0	0.060	0.270
	Flops	162,140	14,400	–	9,790	–	32,400	–
II	Dig	16	24	–	25	–	25	–
	Rel	0	0.90×10^{-21}	0	0.12×10^{-19}	0	0.40×10^{-21}	0
	Strength	0	0.21×10^{-20}	0	0.18×10^{-19}	0	0.24×10^{-20}	0
	Time	1.342	0.260	2.190	0.161	1.432	0.881	4.513
	Flops	112,437	176,000	–	87,529	–	362,667	–
III	Dig	16	18	–	18	–	19	–
	Rel	0	0.68×10^{-17}	0	0.68×10^{-17}	0	0.13×10^{-17}	0
	Strength	0	0.32×10^{-16}	0	0.32×10^{-16}	0	0.66×10^{-17}	0
	Time	2.794	0.010	0.340	0.007	0.234	0.030	0.793
	Flops	162,140	6,912	–	4,759	–	16,128	–

Example I: $m = 2$, $n = 16$, $p = 14$, $d = 4$; Example II: $m = 11$, $n = 17$, $p = 17$, $d = 3$; Example III: $m = 2$, $n = 12$, $p = 12$, $d = 6$

Table 7.2 Comparison of algorithms: $Tol = 10^{-16}$

Example		MRERE(QR)		MRMP	MP	SS
		Num	Sym	Hybrid	Hybrid	Num
I	Dig	35	–	25	27	25
	Rel	0.13×10^{-24}	0	0.26×10^{-13}	0.19×10^{-13}	2.9×10^{-12}
	Strength	0.37×10^{-23}	0	0.31×10^{-6}	0.96×10^{-11}	0.2×10^{-1}
	Time	0.050	0.169	0.050	0.060	0.985
	Flops	19,580	–	267,080	121,314	91,898,532
II	Dig	25	–	20	21	20
	Rel	0.95×10^{-21}	0	0.25×10^{-17}	0.80×10^{-17}	6.0×10^{-19}
	Strength	0.15×10^{-20}	0	0.56×10^{-16}	0.33×10^{-16}	5.51×10^{-14}
	Time	0.381	2.178	3.395	0.861	5.360
	Flops	175,058	–	761,725	56,664	4.1×10^{9}
III	Dig	19	–	19	21	20
	Rel	0.13×10^{-17}	0	0.65×10^{-12}	0.59×10^{-17}	6.9×10^{-19}
	Strength	0.66×10^{-17}	0	0.38×10^{-11}	0.28×10^{-16}	3.35×10^{-18}
	Time	0.020	0.468	2.835	0.290	0.579
	Flops	9,517	–	126,720	50,832	24,082,500

Example I: $m = 2$, $n = 16$, $p = 14$, $d = 4$; Example II: $m = 11$, $n = 17$, $p = 17$, $d = 3$; Example III: $m = 2$, $n = 12$, $p = 12$, $d = 6$

Strength: the strength of the GCD
Dig: the digits of software accuracy
Time: the required time in seconds
Flops: the required flops
Num: numerical implementation
Sym: symbolical implementation
Hybrid: Hybrid implementation

In Tables 7.1 and 7.2 the results of the following example sets are presented:

Example I: Two polynomials of degree 16 and 14 with integer coefficients of 8 digits and GCD degree 4, [2].

Example II: Eleven polynomials of degree 17 with integer coefficients of 2 digits and GCD degree 3, [27].

Example III: Two polynomials of degree 12 and GCD degree 6, (example 1, [36]). The roots of the polynomials spread on the circles of radius 0.5 and 1.5.

Comment: In Tables 7.1 and 7.2 the tolerance (*Tol*) is fixed and the digits are variable. In Tables 7.3 and 7.4 both tolerance and digits are fixed.

7.5 Numerical, Symbolical and Hybrid Behavior of the Methods

All the sets of polynomials have been tested numerically and symbolically. Executing the programs symbolically, there are no floating point errors during the processes and the final result is the exact GCD of the polynomials. But the time

Table 7.3 Comparison of algorithms: $Tol = 10^{-16}$, Dig = 32

		ERES Hybrid	RERE(LU) Numerical	MRERE(LU) Numerical	RERE(QR) Numerical	MRERE(QR) Numerical	MRMP Hybrid	MP Hybrid	SS Numerical
A	Rel	3.31×10^{-31}	3.82×10^{-30}	2.94×10^{-31}	1.35×10^{-29}	3.36×10^{-30}	2.08×10^{-31}	2.49×10^{-31}	2.15×10^{-31}
	Strength	1.25×10^{-27}	6.11×10^{-26}	7.70×10^{-27}	3.30×10^{-25}	5.56×10^{-26}	2.60×10^{-26}	3.28×10^{-26}	8.95×10^{-27}
	Time	0.094	0.016	0.016	0.031	0.015	0.125	0.063	0.203
	Flops	471	2,584	1,447	5,167	2,097	9,167	2,480	5,445,000
B	Rel	1.30×10^{-31}	4.56×10^{-26}	2.01×10^{-28}	5.56×10^{-27}	3.80×10^{-26}	1.82×10^{-26}	1.53×10^{-27}	6.12×10^{-31}
	Strength	4.26×10^{-27}	1.49×10^{-23}	6.42×10^{-24}	1.16×10^{-21}	1.44×10^{-21}	1.66×10^{-21}	1.01×10^{-22}	4.37×10^{-26}
	Time	0.141	0.047	0.031	0.172	0.063	0.937	0.156	1.0
	Flops	900	20,667	11,894	41,334	20,603	73,334	2,480	1.2×10^{9}
C	Rel	3.82×10^{-32}	2.43×10^{-28}	1.78×10^{-29}	1.98×10^{-23}	5.11×10^{-23}	3.16×10^{-24}	2.14×10^{-25}	2.02×10^{-31}
	Strength	3.03×10^{-26}	1.99×10^{-23}	9.96×10^{-25}	1.47×10^{-18}	3.34×10^{-18}	5.59×10^{-19}	1.85×10^{-20}	7.07×10^{-26}
	Time	0.110	0.094	0.032	0.375	0.172	1.516	0.203	2.375
	Flops	1,476	40,896	19,355	81,792	30,334	126,720	4,575	5.7×10^{9}

Example A: $m = 10$, $n = 5$, $p = 5$, $d = 2$; Example B: $m = 10$, $n = 10$, $p = 10$, $d = 2$; Example C: $m = 10$, $n = 15$, $p = 10$, $d = 3$

Table 7.4 Comparison of algorithms: $Tol = 10^{-16}$, Dig = 32

		ERES Hybrid	RERE(LU) Numerical	MRERE(LU) Numerical	RERE(QR) Numerical	MRERE(QR) Numerical	MRMP Hybrid	MP Hybrid	SS Numerical
D	Rel	4.37×10^{-32}	0.927	0.928	0.927	0.928	3.84×10^{-27}	1.51×10^{-18}	1.01×10^{-30}
	Strength	6.10×10^{-17}	4.53×10^{-10}	1.71×10^{-9}	2.93×10^{-9}	3.79×10^{-10}	5.026×10^{-12}	3.36×10^{-12}	1.48×10^{-16}
	Time	5.203	1.328	0.562	2.719	1.578	18.048	4.078	18.063
	Flops	3993	479167	204917	958334	372389	1145834	73684	3.1×10^{12}
E	Rel	4.31×10^{-31}	0.14×10^{-26}	0.96×10^{-30}	0.13×10^{-28}	0.29×10^{-28}	0.95×10^{-28}	0.23×10^{-28}	×
	Strength	1.51×10^{-27}	0.13×10^{-26}	0.16×10^{-26}	0.41×10^{-26}	0.42×10^{-27}	0.55×10^{-23}	0.33×10^{-25}	×
	Time	4.390	39.81	2.75	190.19	11.64	32.92	12.62	×
	Flops	37266	10125000	1125291	20331000	2250582	8933088	19551	×
F	Rel	4.39×10^{-32}	0.14×10^{-26}	0.96×10^{-30}	0.13×10^{-28}	0.29×10^{-28}	0.93×10^{-25}	0.55×10^{-25}	×
	Strength	4.67×10^{-27}	0.13×10^{-26}	0.16×10^{-26}	0.41×10^{-26}	0.42×10^{-27}	0.72×10^{-22}	0.48×10^{-21}	×
	Time	4.225	96.20	3.22	198.64	11.39	51.35	24.91	×
	Flops	358875	9703225	485534	19488301	971068	7433284	1125376	×

Example D: $m = 15$, $n = 25$, $p = 25$, $d = 5$; Example E: $m = 50$, $n = 40$, $p = 40$, $d = 5$; Example F: $m = 50$, $n = 40$, $p = 40$, $d = 30$

required for symbolical computations is considerable. On the other hand, the floating-point operations and the accumulation of rounding errors force us to use accuracies on which the GCD is dependent. Different accuracies may lead to different GCDs. This is the main disadvantage of numerical methods. However, the required time is less than the corresponding time of symbolical methods. The use of the partial SVD [25, 34] can reduce the execution time of ERES, MRMP and SMP methods.

Having tested thoroughly, the ERES, RERE, MRERE, SMP methods computing the GCD of several sets of polynomials, we have reached the following conclusions about the behavior of the methods:

(a) The ERES method behaves very well in Hybrid mode, since it produces accurate results fast enough. The method becomes slower in the case of small sets of polynomials of high degree.
(b) The MRERE methods behave very well in numerical mode for various kinds of sets of polynomials and especially in large sets of polynomials of high degree, but lose their speed, when using symbolic type of operations.
(c) The RERE methods demand significantly more flops in comparison with MRERE methods and their complexity makes them inefficient methods.
(d) The MRMP method demands much more time and flops than the SMP method without producing better results.
(e) It seems that for large sets of linearly depended polynomials the Hybrid ERES and the Numerical MRERE yield better results in acceptable time limits, number of digits, with negligible relative error and strength number.

A subtle point in the numerical calculations is that it is not always easy to specify the exact accuracy to get numerically the correct GCD. This can be overcome in symbolical implementation but the more polynomials and the higher degrees we have, the more time we need to compute the GCD. Thus, a combination of the above implementations can lead to an improvement in the overall performance of the algorithms. Of course the hybrid implementation depends on the nature of the algorithm. Not all algorithms can be benefit from a hybrid environment like ERES [5]. The conversion of the data to an appropriate type often leads to more accurate computations and thus less numerical errors.

Table 7.5 Comparison of algorithms: computation of the GCD of polynomials

Method	m	n	Stability	Decision
ERES	Two	High	Stable	Not proposed
RERE	Two	High	Stable	Proposed
MRERE	Two	High	Stable	Proposed
SMP	Two	High	Stable	Not proposed
MRMP	Two	High	Stable	Not proposed
ERES	Several	High	Stable	Proposed
RERE	Several	High	Stable	Not proposed
MRERE	Several	High	Stable	Proposed
SMP	Several	High	Stable	Proposed
MRMP	Several	High	Stable	Not proposed

In the following table we compare the algorithms computing the GCD of polynomials. We propose the most suitable algorithm for the computation of the GCD of two or several polynomials according to its stability, complexity and required computational time. In the following table, m denotes the number of polynomials and n the maximum degree (Table 7.5).

7.6 Conclusions

According to the comparison of algorithms that we made, we conclude with the following:

1. Matrix-based methods can handle many polynomials simultaneously, without resorting to the successive two at a time computations of the Euclidean or other pairwise based approaches. Although the computation of the GCD of a pair of polynomials, by using a Euclid-based algorithm, can be stable, it is not quite clear if such method can be generalised to a set of several polynomials and how this generalisation affects the complexity of the algorithm and the accuracy of the produced results. The sequential computation of the GCD in pairs often leads to an excessive accumulation of numerical errors especially in the case of large sets of polynomials and obviously create erroneous results. On the other hand, matrix-based methods tend to have better performance and numerical stability in the case of large sets of polynomials.
2. The development of robust computational procedures for engineering type models always has to take into account that the models have certain accuracy and that it is meaningless to continue computations beyond the accuracy of the original data set. Therefore, it is necessary to develop proper numerical termination criteria that allow the derivation of *approximate* solutions to the GCD computation problem [20, 26]. In [19] the definition of the approximate GCD is considered as a distance problem in a projective space. The new distance framework given for the approximate GCD provides the means for computing optimal solutions, as well as evaluating the strength of ad-hoc approximations derived from different algorithms.
3. The combination of symbolic-numeric operations performed effectively in a mixture of numerical and symbolical steps can increase the performance of a matrix-based GCD algorithm. Generally, symbolic processing is used to improve on the conditioning of the input data, or to handle a numerically ill-conditioned subproblem, and numeric tools are used in accelerating certain parts of an algorithm, or in computing approximate outputs. The effective combination of symbolic and numerical operations depends on the nature of an algebraic method and the proper handling of the input data either as rational or floating-point numbers. Symbolic-numeric implementation is possible in software programming environments with symbolic-numeric arithmetic capabilities such as Maple, Mathematica, Matlab and others, which involve the efficient combination of exact (rational-symbolic) and numerical (floating-point)

operations. This combination gives a different perspective in the way to implement an algorithm and introduces the notion of *hybrid computations*. The nature of the ERES method allows the implementation of a programming algorithm that combines in an optimal setup the symbolical application of rows transformations and shifting, and the numerical computation of an appropriate termination criterion, which can provide the required approximate solutions. This combination highlights the hybridity of the ERES method and makes it the most suitable method for the computation of approximate GCDs.

4. Most of the methods and algorithms, which were described in the previous section, perform singular value decomposition (svd). The necessary information that we need from the svd often has to do with the smallest or the greatest singular value (ERES, MP). Therefore a partial singular value decomposition algorithm [34] can be applied in order to speed up the whole process.

The paper is focused on the development of matrix-based algorithms for the GCD problem of sets of several real univariate polynomials, which is considered a part of the fundamental problem of computing nongeneric invariants.

References

1. Blankiship B (1963) A new version of Euclid algorithm. Amer Math Mon 70:742–744
2. Barnett S (1971) Greatest common divisor of several polynomials. Proc Cambridge Phil Soc 70:263–268
3. Barnett S (1980) Greatest common divisor from generalized Sylvester matrices. Proc Cambridge Phil Soc 8:271–279
4. Chin P, Corless RM, Corliss GF (1998) Optimization strategies for the approximate GCD problem. Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC'98), pp 228–235
5. Christou D, Mitrouli M (2005) Estimation of the greatest common divisor of many polynomials using hybrid computations performed by the ERES method. Appl Num Anal Comp Math 2(3):293–305
6. Christou D, Karcanias N, Mitrouli M, Triantafyllou D (2007) Numerical and symbolical comparison of resultant type, ERES and MP methods computing the greatest common divisor of several polynomials. Proceedings of European Control Conference ECC'07, Kos, Greece, pp 504–511
7. Corless RM, Gianni PM, Trager BM, Watt SM (1995) The singular value decomposition for polynomial systems. Proceedings of ISSAC'95, Quebec, Canada, 195–207
8. Datta BN (2010) Numerical linear algebra and applications, 2nd edn. SIAM Publications, Philadelphia
9. Datta BN (2004) Numerical methods for linear control systems. Elsevier Academic Press, San Diego
10. Diaz-Toca GM, Gonzalez-Vega L (2006) Computing greatest common divisors and squarefree decompositions through matrix methods: the parametric and approximate cases. Linear Algebra Appl 412:222–246
11. Emiris IZ, Galligo A, Lombardi H (1997) Certified approximate univariate GCDs. J Pure Applied Algebra 117–118:229–251
13. Fatouros S, Karcanias N (2003) Resultant properties of gcd of many polynomials and a factorization representation of gcd. Int J Control 76:1666–1683

12. Golub GH, Van Loan CF (1989) Matrix computations, 2nd edn. The John Hopkins University Press, Baltimore, London
14. Hirsch MW, Smale S (1974) Differential equations, dynamic systems and linear algebra. Academic Press, New York
15. Kailath T (1980) Linear systems. Prentice Hall, Inc, Englewood Cliffs
16. Karcanias N, Mitrouli M (1999) Approximate algebraic computations of algebraic invariants. Symbolic methods in control systems analysis and design. In: IEE Control Engine Series, vol 56, pp 162–168
17. Karcanias N, Giannakopoulos C, Hubard M (1983) Almost zeros of a set of polynomials of $\Re[s]$. Int J Control 38:1213–1238
18. Karcanias N (1987) Invariance properties and characterisation of the greatest common divisor of a set of polynomials. Int J Control 46:1751–1760
19. Karcanias N, Fatouros S, Mitrouli M, Halikias GH (2006) Approximate greatest common divisor of many polynomials, generalised resultants, and strength of approximation. Comput Math Appl 51(12):1817–1830
20. Karcanias N, Mitrouli M (1994) A matrix pencil based numerical method for the computation of the GCD of polynomials. IEEE Trans Autom Cont 39:977–981
21. Karcanias N, Mitrouli M (2000) Numerical computation of the least common multiple of a set of solynomials. Reliab Comput 6(4):439–457
22. Karcanias N, Mitrouli M (2004) System theoretic based characterisation and computation of the least common multiple of a set of polynomials. Linear Algebra Appl 381:1–23
23. Karcanias N, Mitrouli M, Triantafyllou D (2006) Matrix pencil methodologies for computing the greatest common divisor of polynomials: hybrid algorithms and their performance. Int J Control 79(11):1447–1461
24. Karmarkar N, Lakshman YN (1996) Approximate polynomial greatest common divisors and nearest singular polynomials. ISSAC'96, ACM Press, pp 35–39
25. Marco A, Martinez JJ (2004) A new source of structured singular value decomposition problems. Electronic Trans Num Anal 18:188–197
26. Mitrouli M, Karcanias N (1993) Computation of the GCD of polynomials using Gaussian transformations and shifting. Int J Control 58:211–228
27. Mitrouli M, Karcanias N, Koukouvinos C (1996) Further numerical aspects of the ERES algorithm for the computation of the greatest common divisor of polynomials and comparison with other existing methodologies. Utilitas Mathematica 50:65–84
28. Noda M, Sasaki T (1991) Approximate GCD and its applications to ill-conditioned algebraic equations. J Comp Appl Math 38:335–351
29. Pace IS, Barnett S (1973) Comparison of algorithms for calculation of gcd of polynomials. Int J Syst Sci 4(2):211–226
30. Qiu W, Hua Y, Abed-Meraim K (1997) A subspace method for the computation of the GCD of polynomials. Automatica 33(4):741–743
31. Rosenbrock H (1970) State space and multivariable theory. Nelson, London
32. Rupprecht D (1999) An algorithm for computing certified approximate GCD of n univariate polynomials. J Pure Applied Algebra 139:255–284
33. Triantafyllou D, Mitrouli M (2005) Two resultant based methods computing the greatest common divisor of two polynomials. Lect Notes Comput Sci 3401:519–526
34. Van Huffel S, Vandewalle J (1987) An efficient and reliable algorithm for computing the singular subspace of a matrix, associated with its smallest singular values. J Comp Applied Math 19:313–330
35. Wonham W (1984) Linear multivariable control: a geometric approach, 2nd edn. Springer, New York
36. Zeng Z (2004) The approximate GCD of inexact polynomials, Part I: a univariate algorithm. In: Proceedings 2004 international symposium on symbolic and algebraic computation. ACM Press, New York, pp 320–327

Chapter 8
Numerical Computation of the Fixed Poles in Disturbance Decoupling for Descriptor Systems

Delin Chu and Y. S. Hung

Abstract In this paper the algebraic characterizations for the fixed poles in the disturbance decoupling problem for descriptor systems are derived. These algebraic characterizations lead to a numerically reliable algorithm for computing the fixed poles. The algorithm can be implemented directly using existing numerical linear algebra tools such as LAPACK and Matlab.

8.1 Introduction

Consider descriptor systems of the form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t) + Gq(t); \quad x(0) = x_0, \quad t \geq 0 \\ y(t) &= Cx(t), \end{aligned} \tag{8.1}$$

where $E, A \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{n\times m}, G \in \mathbf{R}^{n\times p}, C \in \mathbf{R}^{q\times n}$, and E is singular. The term $q(t)$ represents a disturbance, which may represent modelling or measuring errors, noise or higher order terms in linearization. The existence and uniqueness of (classical) solutions to system (8.1) for sufficiently smooth input functions and consistent initial values is guaranteed if (E, A) is *regular*, i.e., if $\det(\alpha E - \beta A) \neq 0$

D. Chu (✉)
Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543, Singapore
e-mail: matchudl@nus.edu.sg

Y. S. Hung
Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, Hong Kong
e-mail: yshung@hkueee.hku.hk

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_8,

for some $(\alpha, \beta) \in \mathbf{C}^2$. The system (8.1) is said to have *index at most one* if the dimension of the largest nilpotent block in the Kronecker canonical form of (E, A) is at most one, see Gantmacher [1]. It is well-known that systems that are regular and of index at most one can be separated into purely dynamical and purely algebraic parts (fast and slow modes), and in theory the algebraic part can be eliminated to give a reduced-order standard linear time-invariant system. The reduction process, however, may be ill-conditioned with respect to numerical computation. For this reason it is preferable to use descriptor system models rather than turning the system into a standard linear time-invariant model. For descriptor systems, most numerical simulation methods work well for systems of index at most one, and the usual class of piecewise continuous input functions can be used. Also classical techniques for important control applications like stabilization, pole assignment or linear quadratic control can be applied, see e.g., Bunse-Gerstner et al. [2], and Dai [3]. If the index is larger than one, however, then impulses arise in the response of the system if the control is not sufficiently smooth. This restricts the set of admissible input functions and also impulses can arise due to the presence of modelling, measurement, linearization and roundoff errors in the real system. The usual way to deal with higher index systems in the context of control systems is to choose an appropriate feedback control to ensure that the closed-loop system is regular and of index at most one, whenever this is possible. Techniques for the construction of such feedbacks have been developed in Bunse-Gerstner et al. [2] based on orthogonal transformations, which can be implemented as numerically stable algorithms.

The disturbance decoupling problem for standard linear time-invariant systems (i.e., systems of the form (8.1) with $E = I$) is well studied, see Syrmos [4], Willems and Commault [5], and Wonham [6]. The disturbance decoupling problem for descriptor systems has been studied in Chu and Mehrmann [7], Ailon [8], Banaszuk et al. [9], and Fletcher and Asaraai [10]. When a state feedback of the form $u = Fx$ is applied to system (8.1), then the closed-loop system becomes

$$\begin{aligned} E\dot{x} &= (A + BF)x + Gd, \\ z &= Cx. \end{aligned} \tag{8.2}$$

Hence, the disturbance decoupling problem for system (8.1) can be stated as follows:

Definition 1.1 Given descriptor system (8.1).

(i) Disturbance decoupling problem (DDP) is solvable if there exists a matrix $F \in \mathbf{R}^{m\times n}$ such that $(E, A + BF)$ is regular and

$$C(sE - A - BF)^{-1}G = 0. \tag{8.3}$$

(ii) Disturbance decoupling problem with index requirement (DDPI) is solvable if there exists a matrix $F \in \mathbf{R}^{m\times n}$ such that $(E, A + BF)$ is regular and of index at most one, and (8.3) holds.

For standard linear time-invariant systems, it was shown in Basile and Marro [11], Chu [12], and Malabre and Martinez Garcia [13] that when the disturbance decoupling problem is solvable, the closed-loop system is bound to have some fixed poles after applying any disturbance decoupling state feedback. We would expect this property to extend to the case of descriptor systems. However, for descriptor systems, the fixed poles in the disturbance decoupling problem have not been characterized yet in the literature.

The fixed poles are very important in the system design because they cannot be shifted by any disturbance decoupling feedback and therefore they are related directly to the stability of the closed-loop system. From a system design point of view, all poles of the designed system except these fixed poles should be located into the given stability region. Hence, the fixed poles are of high interest in systems and control. In this paper we will show that when the DDP or DDPI is solvable, there also exist fixed poles that are the closed-loop system poles, after applying any disturbance decoupling state feedback. Now we give the definition of the fixed poles in the DDP and DDPI.

Definition 1.2 Given descriptor system (8.1).

(i) Assume that the DDP is solvable. The set of the fixed poles for DDP is defined as

$$\sigma_f := \cap_{F\in\mathcal{F}}\sigma(E, A + BF),$$

where

$$\mathcal{F} = \{F \in \mathbf{R}^{m\times n} |\ F \text{ solves the DDP}\},$$

and $\sigma(E, A + BF)$ denotes the set of the finite eigenvalues of the pencil $(E, A + BF)$.

(ii) Assume that the DDPI is solvable. The set of the fixed poles for DDPI is defined as

$$\sigma_{fi} := \cap_{F\in\mathcal{FI}}\sigma(E, A + BF),$$

where

$$\mathcal{FI} = \{F \in \mathbf{R}^{m\times n} |\ F \text{ solves the DDPI}\}.$$

We will extend the work in Chu [12] to descriptor systems and present algebraic characterizations for σ_f and σ_{fi}. These algebraic characterizations are obtained using orthogonal transformations, which give a numerically reliable algorithm to compute σ_f and σ_{fi}. This algorithm can be implemented using existing numerical linear algebra tools such as LAPACK and Matlab. Furthermore, as a direct consequence of the algebraic characterizations, the solvability conditions for the DDP and DDPI with stability are obtained. To our knowledge, the present

paper is the first one to get algebraic characterizations and develop a numerically reliable algorithm for σ_f and σ_{fi}.

In this paper we use the following notation:

- $S_\infty(M)$ denotes a matrix with orthogonal columns spanning the right nullspace of a matrix M;
- $\text{rank}_g[\cdot](s)$ denotes the generic rank of a rational matrix function and $\text{rank}(M)$ denotes the standard rank of a matrix M;
- For any $M, N \in \mathbf{R}^{n\times n}$, $\sigma(M,N)$ denotes the set of the finite eigenvalues of the pencil (M,N).

8.2 Preliminaries

In this section we give some supporting results. The following lemma characterizes the regular pencils with index at most one.

Lemma 1.3 [3] *Given $E, A \in \mathbf{R}^{n\times n}$. Then the following are equivalent:*

(i) *The pencil (E,A) is regular and of index at most one;*
(ii) $\text{rank}([\,E \quad AS_\infty(E)\,]) = n$;
(iii) $\deg(\det(sE - A)) = \text{rank}(E)$.

If the system is not regular or not of index at most one then feedback can be used to make the system regular (and of index at most one). Necessary and sufficient conditions, when this is possible, are given in the following Lemma.

Lemma 1.4 [3] *Given $E, A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$.*

(i) *There exists $F \in \mathbf{R}^{m\times n}$ such that $(E, A + BF)$ is regular if and only if*

$$\text{rank}_g[\,sE - A \quad B\,] = n.$$

(ii) *There exists $F \in \mathbf{R}^{m\times n}$ such that $(E, A + BF)$ is regular and of index at most one if and only if*

$$\text{rank}[\,E \quad AS_\infty(E) \quad B\,] = n.$$

Let us now consider pole placement. The following lemma provides conditions under which the set of the finite poles of the closed-loop system can be arbitrarily placed.

Lemma 1.5 [3] *Given $E, A \in \mathbf{R}^{n\times n}$ and $B \in \mathbf{R}^{n\times m}$.*

(i) *If*

$$\text{rank}[\,sE - A \quad B\,] = n, \quad \forall s \in \mathbf{C}, \tag{8.4}$$

then there is an integer $k \geq 0$ *such that for any conjugate set* $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$, *there exists a* $F \in \mathbf{R}^{m\times n}$ *satisfying that* $(E, A + BF)$ *is regular and*

$$\sigma(E, A + BF) = \Lambda. \tag{8.5}$$

(ii) *If* (8.4) *is true and*

$$\operatorname{rank}[E \quad AS_\infty(E) \quad B] = n, \tag{8.6}$$

then for any conjugate set $\Lambda = \{\lambda_1, \ldots, \lambda_{\operatorname{rank}(E)}\}$ *there exists a* $F \in \mathbf{R}^{m\times n}$ *such that* $(E, A + BF)$ *is regular and of index at most one, and* (8.5) *holds.*

The next lemma is a simple generalization of Lemma 1.5, so its proof is omitted.

Lemma 1.6 *Given matrices of the forms*

$$sE - A = \begin{bmatrix} \overset{\mu}{} \\ sE_1 - A_1 \\ -A_2 \end{bmatrix} \begin{matrix} \} l_1 \\ \} l_2 \end{matrix}, \quad B = \begin{bmatrix} \overset{\nu}{} \\ B_1 \\ B_2 \end{bmatrix} \begin{matrix} \} l_1 \\ \} l_2 \end{matrix}$$

with $l_1 \leq \mu$ *and* B_2 *being of full row rank.*

(i) *If*

$$\operatorname{rank}\begin{bmatrix} sE_1 - A_1 & B_1 \\ -A_2 & B_2 \end{bmatrix} = l_1 + l_2, \quad \forall s \in \mathbf{C}, \tag{8.7}$$

then there is an integer $k \geq 0$ *such that that for any conjugate set* $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ *there exist a* $F \in \mathbf{R}^{\nu\times\mu}$ *and a nonsingular matrix* $Z \in \mathbf{R}^{\mu\times\mu}$ *such that*

$$(sE - A - BF)Z = \begin{bmatrix} \overset{l_1}{} & \overset{\mu - l_1}{} \\ s\mathcal{E}_{11} - \mathcal{A}_{11} & -\mathcal{A}_{12} \\ 0 & 0 \end{bmatrix} \begin{matrix} \} l_1 \\ \} l_2 \end{matrix} \tag{8.8}$$

with $(\mathcal{E}_{11}, \mathcal{A}_{11})$ *regular and* $\sigma(\mathcal{E}_{11}, \mathcal{A}_{11}) = \Lambda$.

(ii) *If* (8.7) *is true and*

$$\operatorname{rank}\begin{bmatrix} E_1 & A_1S_\infty(E_1) & B_1 \\ 0 & A_2S_\infty(E_1) & B_2 \end{bmatrix} = l_1 + l_2, \tag{8.9}$$

then for any conjugate set $\Lambda = \{\lambda_1, \ldots, \lambda_{\operatorname{rank}(E_1)}\}$ *there exist a* $F \in \mathbf{R}^{\nu\times\mu}$ *and a nonsingular matrix* $Z \in \mathbf{R}^{\mu\times\mu}$ *such that* (8.8) *holds,* $(\mathcal{E}_{11}, \mathcal{A}_{11})$ *is regular and of index at most one, and* $\sigma(\mathcal{E}_{11}, \mathcal{A}_{11}) = \Lambda$.

The following two theorems, which characterize the solvability of the DDP and DDPI, are slight modifications of Theorems 6, 11 and 13 in Chu and Mehrmann [7], where their proofs can be found.

Theorem 1.7 *Given descriptor system* (8.1). *Then there exist orthogonal matrices* $U, V \in \mathbf{R}^{n\times n}$ *such that*

$$U(sE-A)V=\begin{array}{c}\begin{matrix} n_1 & n_2 & n_3 & n_4 \end{matrix}\\ \begin{bmatrix} sE_{11}-A_{11} & sE_{12}-A_{12} & sE_{13}-A_{13} & sE_{14}-A_{14} \\ sE_{21}-A_{21} & sE_{22}-A_{22} & sE_{23}-A_{23} & sE_{24}-A_{24} \\ -A_{31} & sE_{32}-A_{32} & sE_{33}-A_{33} & sE_{34}-A_{34} \\ 0 & sE_{42}-A_{42} & sE_{43}-A_{43} & sE_{44}-A_{44} \\ 0 & 0 & sE_{53}-A_{53} & sE_{54}-A_{54} \\ 0 & 0 & 0 & sE_{64}-A_{64} \end{bmatrix}\end{array}\begin{matrix}\}\tilde{n}_1\\ \}\tilde{n}_2\\ \}\tilde{n}_3\\ \}\tilde{n}_4\\ \}n_3\\ \}\tilde{n}_6\end{matrix},$$

$$UB=\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ 0 \\ 0 \\ 0 \end{bmatrix}\begin{matrix}\}\tilde{n}_1\\ \}\tilde{n}_2\\ \}\tilde{n}_3\\ \}\tilde{n}_4\\ \}n_3\\ \}\tilde{n}_6\end{matrix},\quad UG=\begin{bmatrix} G_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\begin{matrix}\}\tilde{n}_1\\ \}\tilde{n}_2\\ \}\tilde{n}_3\\ \}\tilde{n}_4\\ \}n_3\\ \}\tilde{n}_6\end{matrix}, \tag{8.10}$$

$$CV=\begin{array}{c}\begin{matrix} n_1 & n_2 & n_3 & n_4 \end{matrix}\\ \begin{bmatrix} 0 & C_2 & C_3 & C_4 \end{bmatrix}\end{array},$$

where

$$\sum_{i=1}^{4} n_i=\sum_{i=1}^{4}\tilde{n}_i+n_3+\tilde{n}_6=n,$$

G_1, E_{21}, B_3 and E_{42} are of full row rank, E_{53} is nonsingular, and furthermore

$$\operatorname{rank}\begin{bmatrix} sE_{11}-A_{11} & B_1 & G_1 \\ sE_{21}-A_{21} & B_2 & 0 \\ -A_{31} & B_3 & 0 \end{bmatrix}=\tilde{n}_1+\tilde{n}_2+\tilde{n}_3,\quad \forall s\in\mathbf{C}, \tag{8.11}$$

$$\operatorname{rank}(sE_{42}-A_{42})=\tilde{n}_4,\quad \operatorname{rank}(sE_{64}-A_{64})=n_4,\quad \forall s\in\mathbf{C}, \tag{8.12}$$

$$\operatorname{rank}_g\begin{bmatrix} sE_{42}-A_{42} & sE_{43}-A_{43} & sE_{44}-A_{44} \\ 0 & sE_{53}-A_{53} & sE_{54}-A_{54} \\ 0 & 0 & sE_{64}-A_{64} \\ C_2 & C_3 & C_4 \end{bmatrix}=n_2+n_3+n_4. \tag{8.13}$$

Theorem 1.8 *Given descriptor system* (8.1). *Assume that orthogonal matrices U and V have been determined such that $(U(sE-A)V, UB, UG, CV)$ are in the condensed form* (8.10).

(i) *The DDP is solvable if and only if the following three conditions hold:*

$$\tilde{n}_6=n_4, \tag{8.14}$$

$$\tilde{n}_1+\tilde{n}_2\le n_1, \tag{8.15}$$

$$\operatorname{rank}_g \begin{bmatrix} sE_{11} - A_{11} & B_1 \\ sE_{21} - A_{21} & B_2 \\ -A_{31} & B_3 \end{bmatrix} = \tilde{n}_1 + \tilde{n}_2 + \tilde{n}_3. \tag{8.16}$$

(ii) *The DDPI is solvable if and only if the conditions* (8.14, 8.15) *and the following two conditions hold:*

$$\operatorname{rank} \begin{bmatrix} E_{11} \\ E_{21} \end{bmatrix} + \operatorname{rank} \begin{bmatrix} E_{32} \\ E_{42} \end{bmatrix} + n_3 = \operatorname{rank}(E), \tag{8.17}$$

$$\operatorname{rank} \begin{bmatrix} E_{11} & A_{11}\mathcal{S} & B_1 \\ E_{21} & A_{21}\mathcal{S} & B_2 \\ 0 & A_{31}\mathcal{S} & B_3 \end{bmatrix} = \tilde{n}_1 + \tilde{n}_2 + \tilde{n}_3, \tag{8.18}$$

here, $\mathcal{S} = S_\infty\left(\begin{bmatrix} E_{11} \\ E_{21} \end{bmatrix}\right)$.

The main feature of the form (8.10) is that it is based on orthogonal transformations, which can be implemented as numerically stable algorithms, thus guaranteeing robust computation of the desired quantities, if this is possible. Furthermore, the conditions (8.14, 8.15, 8.17, 8.18) can be verified very easily. Now we show that the verification of the condition (8.16) can be done using the following condensed form (8.19).

Theorem 1.9 *Given descriptor system* (8.1). *Suppose orthogonal matrices U and V have been determined such that* $U(sE - A)V, UB, UG$ *and* CV *are in the form* (8.10).

(i) *Then there exist orthogonal matrices P and Q such that*

$$P \begin{bmatrix} sE_{11} - A_{11} \\ sE_{21} - A_{21} \\ -A_{31} \end{bmatrix} Q = \begin{array}{c} \begin{matrix} \tau_1 & \tau_2 & \tau_3 \end{matrix} \\ \begin{bmatrix} s\Theta_{11} - \Phi_{11} & s\Theta_{12} - \Phi_{12} & s\Theta_{13} - \Phi_{13} \\ s\Theta_{21} - \Phi_{21} & s\Theta_{22} - \Phi_{22} & s\Theta_{23} - \Phi_{23} \\ 0 & s\Theta_{32} - \Phi_{32} & s\Theta_{33} - \Phi_{33} \\ 0 & 0 & s\Theta_{43} - \Phi_{43} \end{bmatrix} \begin{matrix} \}\tilde{\tau}_1 \\ \}\tilde{\tau}_2 \\ \}\tau_2 \\ \}\tilde{\tau}_4 \end{matrix} \end{array},$$

$$P \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} \Psi_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} \}\tilde{\tau}_1 \\ \}\tilde{\tau}_2 \\ \}\tau_2 \\ \}\tilde{\tau}_4 \end{matrix}, \tag{8.19}$$

where

$$\sum_{i=1}^{3} \tau_i = n_1, \quad \tilde{\tau}_1 + \tilde{\tau}_2 + \tau_2 + \tilde{\tau}_4 = \sum_{i=1}^{3} \tilde{n}_i,$$

Ψ_1 *and* Θ_{21} *are of full row rank,* Θ_{32} *is nonsingular, and*

$$\operatorname{rank}(s\Theta_{21} - \Phi_{21}) = \tilde{\tau}_2, \quad \operatorname{rank}(s\Theta_{43} - \Phi_{43}) = \tau_3, \quad \forall s \in \mathbf{C}. \tag{8.20}$$

(ii) *Furthermore, the condition* (8.16) *holds if and only if* $\tilde{\tau}_4 = \tau_3$.

Proof The form (8.19) is constructed in Appendix A, and (ii) follows directly from that

$$\begin{aligned} &\tilde{n}_1 + \tilde{n}_2 + \tilde{n}_3 = \tilde{\tau}_1 + \tilde{\tau}_2 + \tau_2 + \tilde{\tau}_4, \\ &\operatorname{rank}_g \begin{bmatrix} sE_{11} - A_{11} & B_1 \\ sE_{21} - A_{21} & B_2 \\ -A_{31} & B_3 \end{bmatrix} = \tilde{\tau}_1 + \tilde{\tau}_2 + \tau_2 + \tau_3. \end{aligned}$$

□

In Sect. 8.3 we will characterize σ_f and σ_{fi} using $\sigma(\Theta_{32}, \Phi_{32})$ and $\sigma(E_{53}, A_{53})$ in the forms (8.10) and (8.19). In order to prove these characterizations, we need to refine the form (8.19) by non-orthogonal transformations given in the following lemma 8.10. However, we note here that the condensed form in Lemma 1.10 is only required for analytically derivation of σ_f and σ_{fi} in Theorem 1.11 in the next section. When it comes to the numerical computation of σ_f and σ_{fi}, there is no need to determine the condensed form of Lemma 1.10. In other words, the use of non-orthogonal transformations in Lemma 1.10 is purely for the purpose of exposition, and has no implication on numerical reliability of the algorithm to be developed in the next section.

Lemma 1.10 *Assume that we have already had the condensed forms* (8.10) and (8.19) *and in* (8.19) $\tilde{\tau}_4 = \tau_3$ *(i.e., the condition* (8.16) *holds). Then there exist nonsingular matrices X and Y such that*

$$XP \begin{bmatrix} sE_{11} - A_{11} \\ sE_{21} - A_{21} \\ -A_{31} \end{bmatrix} \quad QY = \begin{array}{c} \begin{matrix} \tau_2 & \hat{\tau}_1 & \hat{\tau}_3 \end{matrix} \\ \begin{bmatrix} s\Theta_{32} - \Phi_{32} & 0 & 0 \\ -\hat{\Phi}_{12} & sI - \hat{\Phi}_{11} & -\hat{\Phi}_{13} \\ -\hat{\Phi}_{22} & -\hat{\Phi}_{21} & -\hat{\Phi}_{23} \\ -\hat{\Phi}_{42} & -\hat{\Phi}_{41} & -\hat{\Phi}_{43} \end{bmatrix} \begin{matrix} \}\tau_2 \\ \}\hat{\tau}_1 \\ \}\xi_2 \\ \}\xi_4 \end{matrix} \end{array},$$

$$XP \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{\Psi}_1 \\ \hat{\Psi}_2 \\ \hat{\Psi}_4 \end{bmatrix} \begin{matrix} \}\tau_2 \\ \}\hat{\tau}_1 \\ \}\xi_2 \\ \}\xi_4 \end{matrix}, \quad XP \begin{bmatrix} G_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{G}_3 \\ \hat{G}_1 \\ \hat{G}_2 \\ 0 \end{bmatrix} \begin{matrix} \}\tau_2 \\ \}\hat{\tau}_1 \\ \}\xi_2 \\ \}\xi_4 \end{matrix}, \tag{8.21}$$

where

$$\tau_2 + \hat{\tau}_1 + \hat{\tau}_3 = n_1, \quad \tau_2 + \hat{\tau}_1 + \xi_2 + \xi_4 = \sum_{i=1}^{3} \tilde{n}_i,$$

$\hat{G}_2$ *and* $\hat{\Psi}_4$ *are of full row rank, and*

$$\operatorname{rank}\begin{bmatrix} sI-\hat{\Phi}_{11} & -\hat{\Phi}_{13} & \hat{\Psi}_1 \\ -\hat{\Phi}_{21} & -\hat{\Phi}_{23} & \hat{\Psi}_2 \\ -\hat{\Phi}_{41} & -\hat{\Phi}_{43} & \hat{\Psi}_4 \end{bmatrix} = \hat{\tau}_1 + \xi_2 + \xi_4, \quad \forall s \in \mathbf{C}. \tag{8.22}$$

Proof See Appendix B. □

8.3 Main Results

We are ready to give the algebraic characterizations of σ_f and σ_{fi}.

Theorem 1.11 *Given descriptor system* (8.1). *Suppose orthogonal matrices* U, V, P *and* Q *and the condensed forms* (8.10) *and* (8.19) *have been determined.*

(i) *Assume that DDP is solvable. Then*

$$\sigma_f = \sigma(\Theta_{32}, \Phi_{32}) \cup \sigma(E_{53}, A_{53}). \tag{8.23}$$

(ii) *Assume that DDPI is solvable. Then*

$$\sigma_{fi} = \sigma_f = \sigma(\Theta_{32}, \Phi_{32}) \cup \sigma(E_{53}, A_{53}). \tag{8.24}$$

Proof (i) Since DDP is solvable, the conditions (8.14, 8.15, 8.16) hold. In the following we first show that $\sigma(\Theta_{32}, \Phi_{32}) \cup \sigma(E_{53}, A_{53}) \subset \sigma_f$.

Let $F \in \mathbf{R}^{m\times n}$ be any matrix such that $(E, A+BF)$ is regular and $C(sE-A-BF)^{-1}G = 0$. Denote

$$FV = \begin{matrix} n_1 & n_2 & n_3 & n_4 \end{matrix} \\ \begin{bmatrix} F_1 & F_2 & F_3 & F_4. \end{bmatrix}$$

Then, using the form (8.10), we have that

$$\begin{aligned} n_1+n_2+n_3+n_4 = n &= \operatorname{rank}_g(sE-A-BF) + \operatorname{rank}_g(C(sE-A-BF)^{-1}G) \\ &= \operatorname{rank}_g\begin{bmatrix} sE-A-BF & G \\ C & 0 \end{bmatrix} \\ &= n_2+n_3+n_4+\tilde{n}_1 + \operatorname{rank}_g\begin{bmatrix} sE_{21}-A_{21}-B_2F_1 \\ -A_{31}-B_3F_1 \end{bmatrix}, \end{aligned}$$

i.e.,

$$n_1 = \tilde{n}_1 + \operatorname{rank}_g\begin{bmatrix} sE_{21}-A_{21}-B_2F_1 \\ -A_{31}-B_3F_1 \end{bmatrix}. \tag{8.25}$$

Compute the generalized upper triangular form of $\begin{bmatrix} sE_{21} - A_{21} - B_2F_1 \\ -A_{31} - B_3F_1 \end{bmatrix}$ (see Demmel and Kågström [14]) to get orthogonal matrices $\mathcal{U}$ and $\mathcal{V}$ such that

$$\mathcal{U}\begin{bmatrix} sE_{21} - A_{21} - B_2F_1 \\ -A_{31} - B_3F_1 \end{bmatrix}\mathcal{V} = \begin{bmatrix} \hat{n}_1 & n_1 - \hat{n}_1 \\ s\hat{E}_{21} - \hat{A}_{21} & s\tilde{E}_{21} - \tilde{A}_{21} \\ 0 & s\tilde{E}_{31} - \tilde{A}_{31} \end{bmatrix} \begin{matrix} \}\mu_2 \\ \}\tilde{n}_2 + \tilde{n}_3 - \mu_2 \end{matrix},$$

where $\hat{E}_{21}$ is of full row rank and $s\tilde{E}_{31} - \tilde{A}_{31}$ is of full column rank for any $s \in \mathbf{C}$. Set

$$(sE_{11} - A_{11} - B_1F_1)\mathcal{V} = \begin{bmatrix} \hat{n}_1 & n_1 - \hat{n}_1 \\ s\hat{E}_{11} - \hat{A}_{11} & s\tilde{E}_{11} - \tilde{A}_{11}, \end{bmatrix}$$

$$\mathcal{U}\begin{bmatrix} B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} \tilde{B}_2 \\ \tilde{B}_3 \end{bmatrix} \begin{matrix} \}\mu_2 \\ \}\tilde{n}_2 + \tilde{n}_3 - \mu_2 \end{matrix}.$$

From (8.25) we obtain that

$$n_1 = \tilde{n}_1 + \mu_2 + (n_1 - \hat{n}_1),$$

or equivalently,

$$\hat{n}_1 = \tilde{n}_1 + \mu_2.$$

Thus, $\begin{bmatrix} s\hat{E}_{11} - \hat{A}_{11} \\ s\hat{E}_{21} - \hat{A}_{21} \end{bmatrix}$ is square. From Theorem 1.8, $\tilde{n}_6 = n_4$, and so we have

$$\sigma\left(\begin{bmatrix} \hat{E}_{11} \\ \hat{E}_{21} \end{bmatrix}, \begin{bmatrix} \hat{A}_{11} \\ \hat{A}_{21} \end{bmatrix}\right) \cup \sigma(E_{53}, A_{53}) \subset (E, A + BF). \tag{8.26}$$

Now we derive the relationship between $\sigma(\Theta_{32}, \Phi_{32})$ and $\sigma\left(\begin{bmatrix} \hat{E}_{11} \\ \hat{E}_{21} \end{bmatrix}\begin{bmatrix} \hat{A}_{11} \\ \hat{A}_{21} \end{bmatrix}\right)$. Using the form (8.19) we know for a $s_0 \in \mathbf{C}$ that

$$\operatorname{rank}\begin{bmatrix} s_0E_{11} - A_{11} & B_1 \\ s_0E_{21} - A_{21} & B_2 \\ -A_{31} & B_3 \end{bmatrix} < \operatorname{rank}_g\begin{bmatrix} sE_{11} - A_{11} & B_1 \\ sE_{21} - A_{21} & B_2 \\ -A_{31} & B_3 \end{bmatrix} \tag{8.27}$$

if and only if $s_0 \in \sigma(\Theta_{32}, \Phi_{32})$.

But, the property (8.11) gives that

$$\operatorname{rank}\begin{bmatrix} s\tilde{E}_{31} - \tilde{A}_{31} & \tilde{B}_3 \end{bmatrix} = \tilde{n}_2 + \tilde{n}_3 - \mu_2, \quad \forall s \in \mathbf{C},$$

which yields that if $s_0 \notin \sigma\left(\begin{bmatrix} \hat{E}_{11} \\ \hat{E}_{21} \end{bmatrix}, \begin{bmatrix} \hat{A}_{11} \\ \hat{A}_{21} \end{bmatrix}\right)$ then

$$\begin{aligned}
&\operatorname{rank}\begin{bmatrix} s_0E_{11}-A_{11} & B_1 \\ s_0E_{21}-A_{21} & B_2 \\ -A_{31} & B_3 \end{bmatrix} \\
&= \operatorname{rank}\begin{bmatrix} s_0E_{11}-A_{11}-B_1F_1 & B_1 \\ s_0E_{21}-A_{21}-B_2F_1 & B_2 \\ -A_{31}-B_3F_1 & B_3 \end{bmatrix} \\
&= \operatorname{rank}\begin{bmatrix} s_0\hat{E}_{11}-\hat{A}_{11} & s_0\tilde{E}_{11}-\tilde{A}_{11} & B_1 \\ s_0\hat{E}_{21}-\hat{A}_{21} & s_0\tilde{A}_{21}-\tilde{A}_{21} & \tilde{B}_2 \\ 0 & s_0\tilde{E}_{31}-\tilde{A}_{31} & \tilde{B}_3 \end{bmatrix} \\
&= (\tilde{n}_1+\mu_2)+(\tilde{n}_2+\tilde{n}_3-\mu_2) \\
&= \tilde{n}_1+\tilde{n}_2+\tilde{n}_3 = \operatorname{rank}_g\begin{bmatrix} sE_{11}-A_{11} & B_1 \\ sE_{21}-A_{21} & B_2 \\ -A_{31} & B_3 \end{bmatrix}. \qquad (8.28)
\end{aligned}$$

So, (8.27) and (8.28) imply that $\sigma(\Theta_{32},\Phi_{32}) \subset \sigma\left(\begin{bmatrix}\hat{E}_{11}\\ \hat{E}_{21}\end{bmatrix}, \begin{bmatrix}\hat{A}_{11}\\ \hat{A}_{21}\end{bmatrix}\right)$. Therefore, (8.26) gives that $\sigma(\Theta_{32},\Phi_{32}) \cup \sigma(E_{53},A_{53}) \subset \sigma(E, A+BF)$, consequently, we have

$$\sigma(\Theta_{32},\Phi_{32}) \cup \sigma(E_{53},A_{53}) \subset \sigma_f. \qquad (8.29)$$

Next, let's consider the form (8.21) and denote

$$XP\begin{bmatrix} sE_{12}-A_{12} \\ sE_{22}-A_{22} \\ sE_{32}-A_{32} \end{bmatrix} = \begin{bmatrix} s\hat{E}_{32}-\hat{A}_{32} \\ s\hat{E}_{12}-\hat{A}_{12} \\ s\hat{E}_{22}-\hat{A}_{22} \\ s\hat{E}_{42}-\hat{A}_{42} \end{bmatrix} \begin{matrix} \}\tau_2 \\ \}\hat{\tau}_1 \\ \}\xi_2 \\ \}\xi_4 \end{matrix}.$$

Since $\tau_2+\hat{\tau}_1+\hat{\tau}_3 = n_1$, and

$$\tilde{n}_1+\tilde{n}_2 = \operatorname{rank}\begin{bmatrix} E_{11} & G_1 \\ E_{21} & 0 \end{bmatrix} = \operatorname{rank}\begin{bmatrix} \Theta_{32} & 0 & \hat{G}_3 \\ 0 & I & \hat{G}_1 \\ 0 & 0 & \hat{G}_2 \end{bmatrix} = \tau_2+\hat{\tau}_1+\xi_2,$$

so, the condition (8.15) is equivalent to that $\xi_2 \le \hat{\tau}_3$. Note that (8.22) holds, by Lemma 1.6(i), there is an integer $k \ge 0$ satisfying that for any conjugate set $\Lambda_1 = \{\lambda_1,\ldots,\lambda_k\}$ there exist matrices $\hat{F}_1, \hat{F}_3$ and a nonsingular matrix Z such that

$$\begin{bmatrix} sI-\hat{\Phi}_{11}-\hat{\Psi}_1\hat{F}_1 & -\hat{\Phi}_{13}-\hat{\Psi}_1\hat{F}_3 \\ -\hat{\Phi}_{21}-\hat{\Psi}_2\hat{F}_1 & -\hat{\Phi}_{23}-\hat{\Psi}_2\hat{F}_3 \\ -\hat{\Phi}_{41}-\hat{\Psi}_4\hat{F}_1 & -\hat{\Phi}_{42}-\hat{\Psi}_4\hat{F}_3 \end{bmatrix} Z = \begin{bmatrix} \hat{\tau}_1 & \xi_2 & \hat{\tau}_3-\xi_2 \\ sI-\tilde{\Phi}_{11} & -\tilde{\Phi}^{(1)}_{13} & -\tilde{\Phi}^{(2)}_{13} \\ -\tilde{\Phi}_{21} & -\tilde{\Phi}^{(1)}_{23} & -\tilde{\Phi}^{(2)}_{23} \\ 0 & 0 & 0 \end{bmatrix} \begin{matrix} \\ \}\hat{\tau}_1 \\ \}\xi_2 \\ \}\xi_4 \end{matrix}, \qquad (8.30)$$

where $\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} \tilde{\Phi}_{11} & \tilde{\Phi}_{13}^{(1)} \\ \tilde{\Phi}_{21} & \tilde{\Phi}_{23}^{(1)} \end{bmatrix}\right)$ is regular and

$$\sigma\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} \tilde{\Phi}_{11} & \tilde{\Phi}_{13}^{(1)} \\ \tilde{\Phi}_{21} & \tilde{\Phi}_{23}^{(1)} \end{bmatrix}\right) = \Lambda_1. \tag{8.31}$$

Additionally, $\hat{\Psi}_4$ and E_{42} are of full row rank and the condition (8.14) gives that $\xi_4 + \tilde{n}_4 = (\tilde{\tau}_3 - \xi_2) + n_2$, so by Lemma 1.4, for any conjugate set $\Lambda_2 = \{\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{\tilde{k}}\}$ with $\tilde{k} = \operatorname{rank}\begin{bmatrix} \hat{E}_{42} \\ E_{42} \end{bmatrix}$, there exist matrices $\tilde{F}_3, F_2$ such that $\left(\begin{bmatrix} 0 & \hat{E}_{42} \\ 0 & E_{42} \end{bmatrix}, \begin{bmatrix} \hat{\Psi}_4\tilde{F}_3 & \hat{A}_{42} + \hat{\Psi}_4 F_2 \\ 0 & A_{42} \end{bmatrix}\right)$ is regular, of index at most one, and

$$\sigma\left(\begin{bmatrix} 0 & \hat{E}_{42} \\ 0 & E_{42} \end{bmatrix}, \begin{bmatrix} \hat{\Psi}_4\tilde{F}_3 & \hat{A}_{42} + \hat{\Psi}_4 F_2 \\ 0 & A_{42} \end{bmatrix}\right) = \Lambda_2.$$

Let $\hat{F}_2$ satisfy $\hat{\Phi}_{42} + \hat{\Psi}_4\hat{F}_2 = 0$. Set

$$F_1 = (\begin{bmatrix} \hat{F}_2 & \hat{F}_1 & \hat{F}_3 \end{bmatrix} + \begin{bmatrix} 0 & \tilde{F}_3 \end{bmatrix})Z^{-1}, \quad F = [F_1 \quad F_2 \quad 0 \quad 0]V^T. \tag{8.32}$$

Then, we have

$$\begin{bmatrix} XP & 0 \\ 0 & I \end{bmatrix} U(sE - A - BF)V \begin{bmatrix} QY & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I_{\tau_2} & 0 & 0 \\ 0 & Z & 0 \\ 0 & 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} s\Theta_{32} - \Phi_{32} & 0 & 0 & * & * & * & * \\ \star & sI - \tilde{\Phi}_{11} & -\tilde{\Phi}_{13}^{(1)} & * & * & * & * \\ \star & -\tilde{\Phi}_{21} & -\tilde{\Phi}_{23}^{(1)} & * & * & * & * \\ 0 & 0 & 0 & -\hat{\Psi}_4\tilde{F}_3 & s\hat{E}_{42} - \hat{A}_{42} - \hat{\Psi}_4 F_2 & * & * \\ 0 & 0 & 0 & 0 & sE_{42} - A_{42} & * & * \\ 0 & 0 & 0 & 0 & 0 & sE_{53} - A_{53} & * \\ 0 & 0 & 0 & 0 & 0 & 0 & sE_{64} - A_{64} \end{bmatrix}. \tag{8.33}$$

A simple calculation yields that $(E, A + BF)$ is regular,

$$\begin{aligned} &\sigma(E, A + BF) = \sigma(\Theta_{32}, \Phi_{32}) \cup \Lambda_1 \cup \Lambda_2 \cup \sigma(E_{53}, A_{53}), \\ &C(sE - A - BF)^{-1}G = 0. \end{aligned}$$

This means that

$$\sigma_f \subset \sigma(\Theta_{32}, \Phi_{32}) \cup \sigma(E_{53}, A_{53}). \tag{8.34}$$

Therefore,

(i) follows directly from (8.29) and (8.34).

(ii) Assume DDPI is solvable. Thus, the conditions (8.14, 8.15, 8.17, 8.18) hold. Note that if $F \in \mathbf{R}^{m\times n}$ solves the DDPI then it also solves the DDP, so,

$$\sigma(\Theta_{32}, \Phi_{32}) \cup \sigma(E_{53}, A_{53}) = \sigma_f \subset \sigma_{fi}. \tag{8.35}$$

Moreover,

- the condition (8.18) gives that $\begin{bmatrix} \hat{\Phi}_{23} & \hat{\Psi}_2 \\ \hat{\Phi}_{43} & \hat{\Psi}_4 \end{bmatrix} = \xi_2 + \xi_4$, thus, by Lemma 1.6 (ii), $\hat{F}_1, \hat{F}_3$ in (8.31) can be chosen such that $\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} \tilde{\Phi}_{11} & \tilde{\Phi}_{13}^{(1)} \\ \tilde{\Phi}_{21} & \tilde{\Phi}_{23}^{(1)} \end{bmatrix} \right)$ is regular, of index at most one, and (8.31) is true;
- it follows from the form (8.21) that $\operatorname{rank} \begin{bmatrix} E_{11} \\ E_{21} \end{bmatrix} = \tau_2 + \hat{\tau}_1$;
- because

$$\operatorname{rank} \begin{bmatrix} E_{11} & E_{12} & G_1 \\ E_{21} & E_{22} & 0 \\ 0 & E_{32} & 0 \\ 0 & E_{42} & 0 \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \Theta_{32} & 0 & \hat{E}_{32} & \hat{G}_3 \\ 0 & I & \hat{E}_{12} & \hat{G}_1 \\ 0 & 0 & \hat{E}_{22} & \hat{G}_2 \\ 0 & 0 & \hat{E}_{42} & 0 \\ 0 & 0 & E_{42} & 0 \end{bmatrix},$$

$$\operatorname{rank} \begin{bmatrix} E_{11} & G_1 \\ E_{21} & 0 \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \Theta_{32} & 0 & \hat{G}_3 \\ 0 & I & \hat{G}_1 \\ 0 & 0 & \hat{G}_2 \end{bmatrix},$$

and $G_1, E_{21}, \hat{G}_2$ are of full row rank, Θ_{32} is nonsingular, we have

$$\operatorname{rank} \begin{bmatrix} E_{32} \\ E_{42} \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \hat{E}_{42} \\ E_{42} \end{bmatrix}.$$

Hence, the F in (8.32) satisfies that

$$\begin{aligned} \deg(\det(sE - A - BF)) &= \deg(\det(s\Theta_{32} - \Phi_{32})) + \deg\left(\det\left(\begin{bmatrix} sI - \tilde{\Phi}_{11} & -\tilde{\Phi}_{13}^{(1)} \\ -\tilde{\Phi}_{21} & -\tilde{\Phi}_{23}^{(1)} \end{bmatrix} \right) \right) \\ &\quad + \deg\left(\det \begin{bmatrix} -\hat{\Psi}_4 \tilde{F}_3 & s\hat{E}_{42} - \hat{A}_{42} - \hat{\Psi}_4 F_2 \\ 0 & sE_{42} - A_{42} \end{bmatrix} \right) \\ &\quad + \deg(\det(sE_{53} - A_{53})) + \deg(\det(sE_{64} - A_{64})) \\ &= \tau_2 + \hat{\tau}_1 + \operatorname{rank} \begin{bmatrix} \hat{E}_{42} \\ E_{42} \end{bmatrix} + n_3 \\ &= \operatorname{rank} \begin{bmatrix} E_{11} \\ E_{21} \end{bmatrix} + \operatorname{rank} \begin{bmatrix} E_{32} \\ E_{42} \end{bmatrix} + n_3 \\ &= \operatorname{rank}(E). \end{aligned}$$

Hence, $(E, A + BF)$ is additionally of index at most one. This means that F solves the DDPI, which and (8.33) give that

$$\sigma_{fi} \subset \sigma(\Theta_{32}, \Phi_{32}) \cup \sigma(E_{53}, A_{53}). \tag{8.36}$$

Therefore, (ii) follows from (8.35) and (8.36). □

Theorem 1.11 implies that when the DDPI is solvable, the fixed poles in the DDP and DDPI are the same, although the index requirement in the DDP is not imposed.

As a direct consequence of the proof of Theorem 1.11 we can obtain solvability conditions for the DDP and DDPI with stability.

Corollary 1.12 *Given descriptor system* (8.1). *Assume that orthogonal matrices* U, V, P *and* Q *and the condensed forms* (8.10) *and* (8.19) *have been determined. Let* $\mathbf{C}^-$ *denote the open left half complex plane.*

(i) *The DDP with stability is solvable, i.e., there exists a* $F \in \mathbf{R}^{m\times n}$ *such that* $(E, A + BF)$ *is regular, stable and* (8.3) *holds if and only if the conditions* (8.14, 8.15, 8.16) *hold and* $\sigma_f \subset \mathbf{C}^-$.

(ii) *The DDPI with stability is solvable, i.e., there exists a* $F \in \mathbf{R}^m \times n$ *such that* $(E, A + BF)$ *is regular, of index at most one, stable and* (8.3) *holds if and only if the conditions* (8.14, 8.15, 8.17, 8.18) *hold and* $\sigma_{fi} \subset \mathbf{C}^-$.

Based on Theorems 1.8 and 1.11, we can compute σ_f and σ_{fi} via the following algorithm.

Algorithm 3

Input: $A \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{n\times m}, C \in \mathbf{R}^{q\times n}, G \in \mathbf{R}^{n\times p}$.

Output: σ_f or σ_{fi} (if possible).

Step 1 Compute the condensed form (8.10) (see Chu and Mehrmann [7]). If (8.14) or (8.15) is not true, print "DDP and DDPI are not solvable" and stop. Otherwise, continue.

Step 2 Check (8.17) and (8.18). If (8.17) or (8.18) is not true, print "DDPI is not solvable".

Step 3 Perform Algorithm 4 in Appendix A to compute the condensed form (8.19). If $\tilde{\tau}_4 \neq \tau_3$, print DDP and DDPI are not solvable" and stop. Otherwise, continue.

Step 4 Compute $\sigma(\Theta_{32}, \Phi_{32})$ and $\sigma(E_{53}, A_{53})$.

Step 5 If DDPI is solvable, set $\sigma_f = \sigma_{fi} = \sigma(\Theta_{32}, \Phi_{32}) \cup \sigma(E_{53}, A_{53})$, and output σ_f and σ_{fi}. If only DDP is solvable, set $\sigma_f = \sigma(\Theta_{32}, \Phi_{32}) \cup \sigma(E_{53}, A_{53})$, and output σ_f.

Note that Algorithm 3 is only based on orthogonal transformations, hence, it can be implemented in a numerically reliable manner via the existing numerical algebra tools such as LAPACK and Matlab.

Now we present an example to illustrate the use of Algorithm 3. All calculations were carried out in MATLAB 5.0 with four decimal places display on a HP

712/80 workstation with IEEE standard (machine accuracy $\epsilon \cong 10^{-16}$). Since we are only interested in the solvability conditions (8.14–8.18) and the eigenvalues of the pencils $s\Theta_{32} - \Phi_{32}$ and $sE_{53} - A_{53}$, so we do not store orthogonal matrices U, V, P and Q in the forms (8.10) and (8.19).

Example 1.13 Given descriptor system (8.1) with

$$E = \begin{bmatrix} -0.7863 & 0.4021 & 0.2314 & -0.2600 & -0.7066 & -0.1875 & -1.1030 \\ -0.1538 & -0.2767 & 0.1409 & -0.0918 & 0.0615 & 0.1701 & -0.2873 \\ 2.0846 & -1.7034 & -0.5839 & 0.1624 & 0.8925 & 0.1831 & 0.5748 \\ 1.0141 & -1.2203 & -0.0836 & -0.0557 & 0.5309 & 0.5997 & 0.1480 \\ -0.0925 & -0.0113 & 0.0298 & 0.0278 & 0.1035 & -0.0039 & 0.0718 \\ 1.0492 & -1.6596 & -0.2188 & 0.1829 & 1.2403 & 0.1666 & 0.4429 \\ -0.0987 & 0.1941 & 0.0184 & 0.0202 & -0.0431 & 0.0044 & 0.1207 \end{bmatrix},$$

$$A = \begin{bmatrix} 6.3555 & -4.5744 & -1.8617 & 2.0120 & 6.1926 & 1.8761 & 7.2305 \\ 0.56320 & 2.0114 & -1.2514 & 1.2150 & 0.1732 & -0.60900 & 2.4890 \\ -14.319 & 11.934 & 3.8888 & -1.5240 & -6.5104 & -1.7185 & -3.8026 \\ -7.2305 & 8.1554 & 0.68420 & -0.13180 & -3.7048 & -4.2523 & -1.2279 \\ 0.97910 & -0.12010 & -0.09360 & -0.36710 & -0.84600 & 0.04080 & -0.45740 \\ -77576 & 12.035 & 1.1888 & -1.5432 & -8.9678 & -1.0801 & -2.8889 \\ 0.25280 & -0.61880 & -0.17930 & 0.15880 & 0.17470 & 0.51260 & -0.40620 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.4937 & -0.2152 \\ 0.1761 & -0.2545 \\ -0.3581 & -0.0736 \\ -0.1451 & -0.2297 \\ 0.0116 & -0.0491 \\ -0.0759 & -0.5923 \\ -0.0460 & 0.1003 \end{bmatrix}, \quad G = \begin{bmatrix} 0.0655 \\ 0.1149 \\ -0.0266 \\ -0.0809 \\ 0.0932 \\ -0.0935 \\ 0.0549 \end{bmatrix},$$

$$C = \begin{bmatrix} -0.2196 & -0.5108 & 0.0312 & 0.3180 & 0.2294 & 0.1753 & -0.2892 \\ 0.0506 & 0.0161 & -0.1879 & 0.4957 & -0.3728 & 0.2024 & -0.2401 \end{bmatrix}.$$

Our purpose is to check if DDP and DDPI are solvable, and if so, to compute σ_f and σ_{fi}.

First, we compute the condensed form (8.10). The computed (UEV, UAV, UB, UG, CV) and the parameters $n_i, \tilde{n}_i, i = 1, \ldots, 4$, and $\tilde{n}_6$ are as follows:

$$UEV = \left[\begin{array}{ccc|cc|cc} -1.2184 & -0.6681 & 0.2716 & -0.2559 & -0.8668 & -1.6301 & -0.8093 \\ \hline -0.6934 & -0.3802 & 0.1545 & -0.1456 & -0.4933 & -0.9277 & -0.4606 \\ -1.4498 & -0.7950 & 0.3231 & -0.3045 & -1.0315 & -1.9398 & -0.9630 \\ \hline 0 & 0 & 0 & 0.6435 & 0.9601 & 0.5165 & 0.6178 \\ \hline 0 & 0 & 0 & 0.3200 & 0 & 0.6178 & 0.9218 \\ \hline 0 & 0 & 0 & 0 & 0 & 0.7266 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right],$$

$$UAV = \left[\begin{array}{ccc|cc|cc} 9.5451 & 4.9248 & -1.7622 & 2.5184 & 6.5125 & 11.584 & 5.6769 \\ \hline 4.5201 & 2.1315 & -1.2537 & 1.3285 & 4.1477 & 7.4736 & 4.1181 \\ 10.597 & 5.4639 & -1.9665 & 2.9700 & 7.8418 & 13.850 & 6.9401 \\ \hline 0.30280 & 0.59360 & 0.34200 & -3.9364 & -5.9259 & -3.3632 & -4.0259 \\ \hline 0 & 0 & 0 & -1.8696 & 0.95680 & -3.4489 & -5.7912 \\ \hline 0 & 0 & 0 & 0 & 0 & -4.3489 & 0.28440 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0.46920 \end{array}\right],$$

$$UB = \left[\begin{array}{cc} 0.3740 & 0.1578 \\ \hline 0.1918 & 0.0809 \\ 0.4143 & 0.1748 \\ \hline 0.2895 & -0.6860 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \end{array}\right], \quad UG = \left[\begin{array}{c} 0.2126 \\ \hline 0 \\ 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \end{array}\right],$$

$$CV = \left[\begin{array}{ccc|cc|c|c} 0 & 0 & 0 & 0.6457 & 0.4017 & 0 & 0 \\ 0 & 0 & 0 & 0.5361 & -0.4837 & 0 & 0 \end{array}\right],$$

$$n_1 = 3,\ n_2 = 2,\ n_3 = n_4 = 1,\ \tilde{n}_1 = 1,\ \tilde{n}_2 = 2,\ \tilde{n}_3 = \tilde{n}_4 = \tilde{n}_6 = 1.$$

A simple calculation gives that the conditions (8.14, 8.15, 8.17, 8.18) hold. Hence, by Theorem 1.8, the DDPI is solvable (and therefore DDP is also solvable).

Then we compute the condensed form (8.19) as follows:

$$P \left[\begin{array}{c} sE_{11} - A_{11} \\ sE_{21} - A_{21} \\ -A_{21} \end{array}\right] Q = \left[\begin{array}{c|c|c} 0.4364 & 1.4954s + 10.748 & -1.4780s - 11.184 \\ 0.4338 & 0.7735s + 5.6018 & -0.7249s - 4.6822 \\ \hline 0 & 0.0648s + 0.02070 & -0.0640s + 0.31240 \\ \hline 0 & 0 & -0.5298 \end{array}\right],$$

$$P \left[\begin{array}{c} B_1 \\ B_2 \\ B_3 \end{array}\right] = P \left[\begin{array}{cc} 0.3740 & 0.1578 \\ 0.1918 & 0.0809 \\ 0.4143 & 0.1748 \\ 0.2895 & -0.6860 \end{array}\right] = \left[\begin{array}{cc} -0.6573 & 0.0785 \\ 0 & -0.7256 \\ \hline 0 & 0 \\ \hline 0 & 0 \end{array}\right],$$

$$\tau_1 = \tau_2 = \tau_3 = 1,\ \tilde{\tau}_1 = 2,\ \tilde{\tau}_2 = 0,\ \tilde{\tau}_4 = 1.$$

Finally,

$$s\Theta_{32} - \Phi_{32} = 0.0648s + 0.02070, \quad sE_{53} - A_{53} = 0.7266s + 4.3489,$$

by Theorem 1.11, we have

$$\sigma_{fi} = \sigma_f = \sigma(\Theta_{32}, \Phi_{32}) \cup \sigma(E_{53}, A_{53}) = \{-0.31944, -5.9853\} \subset \mathbf{C}^-.$$

Hence, both DDP and DDPI with stability are solvable.

8.4 Conclusions

In this paper we have studied the fixed poles in disturbance decoupling for descriptor systems. Algebraic characterizations for these fixed poles are derived based on two condensed forms under orthogonal transformations. These algebraic characterizations lead to a numerically reliable algorithm for computing the fixed poles. This algorithm can be implemented directly using existing numerical linear algebra tools such as LAPACK and Matlab.

Appendix A: Proof of Theorem 1.9(i)

Proof We prove Theorem (i) constructively by means of the following algorithm.

Algorithm 4

Input: Input $E_{11}, E_{21}, A_{11}, A_{21}, A_{31}, B_1, B_2$ and B_3 in the form (8.10).
Output: Orthogonal matrices P and Q and the form (8.19).

Step 1 Perform a QR factorization of $\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$ to get orthogonal matrix P_1 such that

$$P_1 \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} =: \begin{bmatrix} \Psi_1 \\ 0 \end{bmatrix} \begin{matrix} \}\tilde{\tau}_1 \\ \}\tilde{\tau}_2^{(1)} \end{matrix},$$

where Ψ_1 being of full row rank. Set

$$P_1 \begin{bmatrix} sE_{11} - A_{11} \\ sE_{21} - A_{21} \\ -A_{31} \end{bmatrix} =: \begin{bmatrix} s\Theta_{11}^{(1)} - \Phi_{11}^{(1)} \\ s\Theta_{21}^{(1)} - \Phi_{21}^{(1)} \end{bmatrix} \begin{matrix} \}\tilde{\tau}_1 \\ \}\tilde{\tau}_2^{(1)} \end{matrix}.$$

Step 2 Compute the generalized upper triangular form of $s\Theta_{21}^{(1)} - \Phi_{21}^{(1)}$ (see Demmel and Kågström [14]) to get orthogonal matrices P_2 and Q such that

$$P_2\left(s\Theta_{21}^{(1)} - \Phi_{21}^{(1)}\right)Q =: \begin{matrix} \\ \\ \\ \end{matrix} \begin{matrix} \tau_1 & \tau_2 & \tau_3 \end{matrix} \begin{bmatrix} s\Theta_{21} - \Phi_{21} & s\Theta_{22} - \Phi_{22} & s\Theta_{23} - \Phi_{23} \\ 0 & s\Theta_{32} - \Phi_{32} & s\Theta_{33} - \Phi_{33} \\ 0 & 0 & s\Theta_{43} - \Phi_{43} \end{bmatrix} \begin{matrix} \}\tilde{\tau}_2 \\ \}\tau_2 \\ \}\tilde{\tau}_4 \end{matrix},$$

where Θ_{21} is of full row rank, Θ_{32} is nonsingular, and the property (8.20) holds. Set

$$(s\Theta_{11}^{(1)} - \Phi_{11}^{(1)})Q =: \begin{matrix}\tau_1 & \tau_2 & \tau_3\end{matrix}\left[\begin{matrix} s\Theta_{11} - \Phi_{11} & s\Theta_{12} - \Phi_{12} & s\Theta_{13} - \Phi_{13}\end{matrix}\right],$$

and

$$P := \begin{bmatrix} I & \\ & P_2 \end{bmatrix} P_1.$$

Then P and Q give the form (8.19).

Appendix B: Proof of Lemma 1.10

Proof Since Θ_{32} is nonsingular and $\operatorname{rank}(s\Theta_{43} - \Phi_{43}) = \tau_3$ for any $s \in \mathbf{C}$, by matrix pencil theory (see Gantmacher [1]), there exist matrices $\mathcal{X}_{34}$ and $\mathcal{Y}_{23}$ such that

$$s\Theta_{33} - \Phi_{33} + (s\Theta_{32} - \Phi_{32})\mathcal{Y}_{23} + \mathcal{X}_{34}(s\Theta_{43} - \Phi_{43}) = 0.$$

Note that $\tau_3 = \tilde{\tau}_4$, so if we let

$$X_1 = \begin{bmatrix} 0 & 0 & I_{\tau_2} & 0 \\ I_{\tilde{\tau}_1} & 0 & 0 & 0 \\ 0 & I_{\tilde{\tau}_2} & 0 & 0 \\ 0 & 0 & 0 & I_{\tau_3} \end{bmatrix} \begin{bmatrix} I_{\tilde{\tau}_1} & 0 & -\Theta_{12}\Theta_{32}^{-1} & 0 \\ 0 & I_{\tilde{\tau}_2} & -\Theta_{22}\Theta_{32}^{-1} & 0 \\ 0 & 0 & I_{\tau_2} & \mathcal{X}_{34} \\ 0 & 0 & 0 & I_{\tau_3} \end{bmatrix},$$

and

$$Y_1 = \begin{bmatrix} I_{\tau_1} & 0 & 0 \\ 0 & I_{\tau_2} & \mathcal{Y}_{23} \\ 0 & 0 & I_{\tau_3} \end{bmatrix} \begin{bmatrix} 0 & I_{\tau_1} & 0 \\ I_{\tau_2} & 0 & 0 \\ 0 & 0 & I_{\tau_3} \end{bmatrix},$$

then we have

$$X_1 P \begin{bmatrix} sE_{11} - A_{11} \\ sE_{21} - A_{21} \\ -A_{31} \end{bmatrix} \quad QY_1 = \begin{matrix}\tau_2 & \tau_1 & \tau_3\end{matrix}\left[\begin{matrix} s\Theta_{32} - \Phi_{32} & 0 & 0 \\ \tilde{\Phi}_{12} & s\Theta_{11} - \Phi_{11} & s\tilde{\Theta}_{13} - \tilde{\Phi}_{13} \\ \tilde{\Phi}_{22} & s\Theta_{21} - \Phi_{21} & s\tilde{\Theta}_{23} - \tilde{\Phi}_{23} \\ 0 & 0 & s\Theta_{43} - \Phi_{43} \end{matrix}\right] \begin{matrix}\}\tau_2 \\ \}\tilde{\tau}_1 \\ \}\tilde{\tau}_2 \\ \}\tau_3\end{matrix},$$

$$X_1 P \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} 0 \\ \Psi_1 \\ 0 \\ 0 \end{bmatrix} \begin{matrix}\}\tau_2 \\ \}\tilde{\tau}_1 \\ \}\tilde{\tau}_2 \\ \}\tau_3\end{matrix}, \quad X_1 P \begin{bmatrix} G_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{G}_3 \\ \tilde{G}_1 \\ \tilde{G}_2 \\ \tilde{G}_4 \end{bmatrix} \begin{matrix}\}\tau_2 \\ \}\tilde{\tau}_1 \\ \}\tilde{\tau}_2 \\ \}\tau_3\end{matrix}.$$

Because $\begin{bmatrix} E_{11} & B_1 & G_1 \\ E_{21} & B_2 & 0 \\ 0 & B_3 & 0 \end{bmatrix}$ is of full row rank, so, $\begin{bmatrix} \Theta_{11} & \tilde{\Theta}_{13} & \Psi_1 & \tilde{G}_1 \\ \Theta_{21} & \tilde{\Theta}_{23} & 0 & \tilde{G}_2 \\ 0 & \Theta_{43} & 0 & \tilde{G}_4 \end{bmatrix}$ is also of full row rank. Thus, there exist nonsingular matrices X_{22} and Y_{22} such that

$$X_{22}\begin{bmatrix} s\Theta_{11}-\Phi_{11} & s\tilde{\Theta}_{13}-\tilde{\Phi}_{13} \\ s\Theta_{21}-\Phi_{21} & s\tilde{\Theta}_{23}-\tilde{\Phi}_{23} \\ 0 & s\Theta_{43}-\Phi_{43} \end{bmatrix} \quad Y_{22}=\begin{matrix} \\ \\ \\ \end{matrix}\begin{matrix} \hat{\tau}_1 & \hat{\tau}_3 \\ \end{matrix}\begin{bmatrix} sI-\hat{\Phi}_{11} & -\hat{\Phi}_{13} \\ -\hat{\Phi}_{21} & -\hat{\Phi}_{23} \\ -\hat{\Phi}_{41} & -\hat{\Phi}_{43} \end{bmatrix}\begin{matrix} \}\hat{\tau}_1 \\ \}\xi_2\,, \\ \}\xi_4 \end{matrix}$$

$$X_{22}\begin{bmatrix} \Psi_1 \\ 0 \\ 0 \end{bmatrix}=\begin{bmatrix} \hat{\Psi}_1 \\ \hat{\Psi}_2 \\ \hat{\Psi}_4 \end{bmatrix}\begin{matrix} \}\hat{\tau}_1 \\ \}\xi_2\,, \\ \}\xi_4 \end{matrix} \quad X_{22}\begin{bmatrix} \tilde{G}_1 \\ \tilde{G}_2 \\ \tilde{G}_4 \end{bmatrix}=\begin{bmatrix} \hat{G}_1 \\ \hat{G}_2 \\ 0 \end{bmatrix}\begin{matrix} \}\hat{\tau}_1 \\ \}\xi_2\,, \\ \}\xi_4 \end{matrix}$$

where $\hat{G}_2$ and $\hat{\Psi}_4$ are of full row rank. Considering that $\begin{bmatrix} s\Theta_{11}-\Phi_{11} & \Psi_1 \\ s\Theta_{21}-\Phi_{21} & 0 \end{bmatrix}$ and $s\Theta_{43}-\Phi_{43}$ are of full row rank for any $s \in \mathbf{C}$ (note that $\tau_3=\tilde{\tau}_4$), so (8.22) holds. Therefore, Lemma 1.10 is proved with

$$X=\begin{bmatrix} I & 0 \\ 0 & X_{22} \end{bmatrix}X_1, \quad Y=Y_1\begin{bmatrix} I & 0 \\ 0 & Y_{22} \end{bmatrix}, \quad X_{22}\begin{bmatrix} \tilde{\Phi}_{12} \\ \tilde{\Phi}_{22} \\ 0 \end{bmatrix}=\begin{bmatrix} \hat{\Phi}_{12} \\ \hat{\Phi}_{22} \\ \hat{\Phi}_{42} \end{bmatrix}\begin{matrix} \}\hat{\tau}_1 \\ \}\xi_2\,. \\ \}\xi_4 \end{matrix}$$

References

1. Gantmacher FR (1959) Theory of matrices, vol I. Chelsea, New York
2. Bunse-Gerstner A, Mehrmann V, Nichols NK (1992) Regularization of descriptor systems by derivative and proportional state feedback. SIAM J Matrix Anal Appl 13:46–67
3. Dai L (1989) Singular control systems, vol 118 of lecture notes in control and information sciences. Springer, Berlin
4. Syrmos VL (1994) Disturbance decoupling using constrained Sylvester equations. IEEE Trans Automat Control AC-39:797–803
5. Willems JC, Commault C (1981) On disturbance decoupling by measurement feedback with stability or pole placement. SIAM J Control Optim 19(4):490–504
6. Wonham WM (1985) Linear multivariable control: a geometric approach, 2nd edn. Springer, New York
7. Chu D, Mehrmann V (2000) Disturbance decoupling for descriptor systems by state feedback. SIAM J Control Optim 38:1830–1858
8. Ailon A (1993) A solution to the disturbance decoupling problem in singular systems via analogy with state-space systems. Automatica 29:1541–1545
9. Banaszuk M, Kociecki M, Przyluski KM (1990) The disturbance decoupling problem for implicit linear discrete-time systems. SIAM J Control Optim 28:1270–1293
10. Fletcher LR, Asaraai A (1989) On disturbance decoupling in descriptor systems. SIAM J Control Optim 27:1319–1332
11. Basile G, Marro G (1992) Controlled and conditioned invariants in linear system theory. Prentice Hall, Englewood Cliffs

12. Chu D (2003) The fixed poles of the disturbance decoupling problem and almost stability subspace $\mathcal{V}^{\star}_{b,g}f(\mathrm{Ker}(C))$. Numer Math 96:221–252
13. Malabre M, Martinez Garcia JC (1997) On the fixed poles for disturbance rejection. Automatica 33:1209–1211
14. Demmel JW, Kågström B (1993) The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: robust software with error bounds and applications. Part I: theory and algorithms. ACM Trans Math Softw 19(2):160–174

Bibliography

15. Anderson E, Bai Z, Bischof C, Demmel JW, Dongarra J, Croz JD, Greenbaum A, Hammarling S, McKenney A, Ostrouchov S, Sorensen D (1992) LAPACK users' guide. Society for Industrial and Applied Mathematics, Philadelphia
16. Golub GH, Van Loan CF (1996) Matrix computations, 3rd edn. The Johns Hopkins University Press, Baltimore

Chapter 9
Robust Control of Discrete Linear Repetitive Processes with Parameter Varying Uncertainty

Błażej Cichy, Krzysztof Gałkowski, Eric Rogers and Anton Kummert

Abstract Repetitive processes propagate information in two independent directions where the duration of one of these is infinite.

They pose control problems that cannot be solved by application of results for other classes of 2D systems. This paper develops robust controller design algorithms for discrete linear processes based on the poly-quadratic stability that produce less conservative results than currently available alternatives.

9.1 Introduction

Repetitive processes are a distinct class of 2D systems of both system theoretic and applications interest whose unique characteristic is a series of sweeps, termed passes, through a set of dynamics defined over a fixed finite duration known as the pass length. On each pass an output, termed the pass profile, is produced which acts as a

B. Cichy (✉) · K. Gałkowski
Institute of Control and Computation Engineering, University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Gora, Poland
e-mail: B.Cichy@issi.uz.zgora.pl

K. Gałkowski
e-mail: K.Galkowski@issi.uz.zgora.pl

E. Rogers
School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK
e-mail: etar@ecs.soton.ac.uk

A. Kummert
Faculty of Electrical, Information and Media Engineering, University of Wuppertal, Rainer-Gruenter-Strasse 21, 42119 Wuppertal, Germany
e-mail: kummert@uni-wuppertal.de

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_9,

forcing function on, and hence contributes to, the dynamics of the next pass profile. This, in turn, leads to the unique control problem in that the output sequence of pass profiles can contain oscillations whose amplitude in the pass-to-pass direction.

To introduce a formal definition, let the integer $\alpha < +\infty$ denote the number of samples resulting from sampling the assumed constant pass length at a constant rate. Then in a repetitive process the pass profile $y_k(p), 0 \leq p \leq \alpha - 1$, generated on pass k acts as a forcing function on, and hence contributes to, the dynamics of the next pass profile $y_{k+1}(p), 0 \leq p \leq \alpha - 1, k \geq 0$.

Physical examples of these processes include long-wall coal cutting and metal rolling operations [17]. Also in recent years applications have arisen where adopting a repetitive process setting for analysis has distinct advantages over alternatives. An example of these algorithmic applications is iterative algorithms for solving nonlinear dynamic optimal stabilization problems based on the maximum principle [16] where use of the repetitive process setting provides the basis for the development of highly reliable and efficient iterative solution algorithms. A second example is iterative learning control schemes which form one approach to controlling systems operating in a repetitive (or pass-to-pass) mode with the requirement that a reference trajectory $r(p)$ defined over a finite interval $0 \leq p \leq \alpha - 1$ is followed to a high precision—see, for example, [11]. In this case a repetitive process setting for analysis provides a stability theory which, unlike many alternatives, allows for design to meet pass-to-pass error convergence and control of the along the pass dynamics. Also iterative learning control laws designed in this setting have been experimentally verified [10] on a gantry robot with very good correlation between simulation and actually measured performance.

Attempts to analyze repetitive processes using standard (or 1D) systems theory/algorithms fail (except in a few very restrictive special cases) precisely because such an approach ignores their inherent 2D systems structure, i.e. information propagation occurs from pass-to-pass and along a given pass. Also the initial conditions are reset before the start of each new pass and the structure of these can be somewhat complex. For example, if the pass state initial vector is an explicit function of the pass profile vector at points along the previous pass then this alone can destroy the most basic performance specification of stability. In seeking a rigorous foundation on which to develop a control/estimation/filtering theory for these processes, it is natural to attempt to exploit structural links which exist with other classes of 2D linear systems.

The case of 2D discrete linear systems recursive in the positive quadrant $(i,j) : i \geq 0, j \geq 0$ (where i and j denote the directions of information propagation) has been the subject of much research effort over the years using, in the main, the well known Roesser and Fornasini Marchesini state–space models (for details on these see, for example, the references given in [17]). More recently, productive research has been reported on $\mathcal{H}_\infty$ and $\mathcal{H}_2$ approaches to filtering and control law design—see, for example, [3, 19]. (Filtering of this general form is, of course, well established in 1D linear systems theory, see, for example, [8, 13, 18]).

As noted above, the structure of the boundary conditions for linear repetitive processes can cause problems which have no Roesser or Fornasini Marchesini

state–space model counterparts. Moreover, there are key systems theoretic properties for repetitive processes that have no interpretation in terms of these (and other) 2D systems models. An example here is pass profile controllability [17] that is the physically motivated requirement that there exists a control input sequence which will force the process to produce a pre-specified pass profile on a given pass. This means that the systems theory for other classes of 2D discrete linear systems is very often not applicable to repetitive processes.

As noted above, material, or metal, rolling is one of a number of physically based problems which can be modeled as a linear repetitive process [17]. In this paper, we use a material rolling process as a basis to illustrate the solution we develop to a currently open robust stability and stabilization problem for discrete linear repetitive processes. The design itself can be completed using Linear Matrix Inequalities (LMIs) [4, 12].

In physical application terms, the system or process parameters are most often not known exactly and only some nominal values or admissible intervals are available. Hence, although the nominal process is most often time invariant, the uncertain process can be time-varying. This will be the case here and to solve the problem we generalize previously reported LMI based design algorithms for uncertain discrete linear repetitive processes [5] and other classes of systems [2, 9, 15] with polytopic uncertainty. These results are based on sufficient, but not necessary, stability conditions.

The use of sufficient but not necessary stability conditions obviously creates a potentially serious problem since the results obtained can be very conservative in the sense that the range over which the admissible parameters can vary is very small. Here we develop substantial new results on how this problem can be overcome. The essential mechanism used is to allow a control law where the entries in the defining matrices explicitly depend on the pass number k, and the along the pass variable $p, k \geq 0$ and $0 \leq p \leq \alpha - 1$. This is a form of adaptation which allows the control law to follow or track the evolution of any uncertainty present in the model used for control law design and evaluation. In addition to reducing the conservativeness of the design, this will also allow the admissible uncertainty set over which a satisfactory solution exists to be enlarged.

Throughout this chapter, the null matrix and the identity matrix with appropriate dimensions are denoted by 0 and I, respectively. Moreover, a real symmetric positive definite matrix, say N, is denoted by $N \succ 0$. Next we describe the modeling of the material rolling process used in this paper.

9.2 Material Rolling as a Linear Repetitive Process

Material rolling is an extremely common industrial process where, in essence, deformation of the workpiece takes place between two rolls with parallel axes revolving in opposite directions. Consider also the following differential equation model of the metal rolling process shown schematically in Fig. 9.1 whose

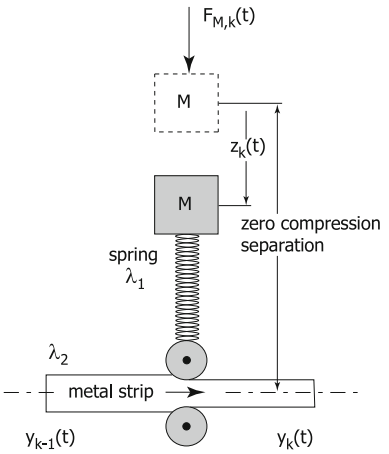


Fig. 9.1 Schematic of material rolling

derivation is explained fully in [1] (see also [7] for other repetitive process based analysis of this model)

$$\ddot{y}_k(t) + a_0 y_k(t) + b_2 \ddot{y}_{k-1}(t) + b_0 y_{k-1}(t) = c_0 u_k(t) \tag{9.1}$$

where

$$a_0 = \frac{\lambda_1 \lambda_2}{M(\lambda_1 + \lambda_2)}, \quad b_2 = \frac{-\lambda_2}{\lambda_1 + \lambda_2}, \quad b_0 = \frac{-\lambda_1 \lambda_2}{M(\lambda_1 + \lambda_2)}, \quad c_0 = \frac{-\lambda_1}{M(\lambda_1 + \lambda_2)}$$

and $u_k(t) = F_{M,k}(t)$. Here $y_{k+1}(t)$ and $y_k(t)$ denote the thickness of the material on two successive passes, M is the lumped mass of the roll-gap adjusting mechanism, λ_1 is the stiffness of the adjustment mechanism springs and λ_2 is the hardness of the material strip, and $F_{M,k}(t)$ is the force developed by the motor.

Applying the backward Euler discretization method with sampling period T to (9.1) leads to the following state–space model which is a special case of that for discrete linear repetitive processes

$$\begin{aligned} x_{k+1}(p+1) &= Ax_{k+1}(p) + Bu_{k+1}(p) + B_0 y_k(p) \\ y_{k+1}(p) &= Cx_{k+1}(p) + Du_{k+1}(p) + D_0 y_k(p) \end{aligned} \tag{9.2}$$

where

$$A = \frac{1}{1 + a_0 T^2} \begin{bmatrix} 1 & T \\ -a_0 T & 1 \end{bmatrix}, \quad B = \frac{c_0 T}{1 + a_0 T^2} \begin{bmatrix} T \\ 1 \end{bmatrix}$$

$$B_0 = \frac{(-b_0 + a_0 b_2)T}{1 + a_0 T^2} \begin{bmatrix} T \\ 1 \end{bmatrix}, \quad C = \frac{1}{1 + a_0 T^2} [1 \quad T]$$

$$D = \frac{c_0 T^2}{1 + a_0 T^2}, \qquad D_0 = \frac{-b_2 - b_0 T^2}{1 + a_0 T^2}$$

and

$$x_k(p) = \begin{bmatrix} y_{k+1}(p-1) + b_2 y_k(p-1) \\ (y_{k+1}(p-1) + b_2 y_k(p-1) - y_{k+1}(p-2) - b_2 y_k(p-2))T^{-1} \end{bmatrix}$$

In the general case on pass $k, x_{k+1}(p)$ is the $n \times 1$ current pass state vector $y_k(p)$ is the $m \times 1$ pass profile vector and $u_k(p)$ is the $l \times 1$ current pass input vector.

To complete the process description it is necessary to specify the initial, or boundary, conditions, that is, the pass state initial vector sequence and the initial pass profile. Here these are taken to be of the form

$$\begin{aligned} x_{k+1}(0) &= d_{k+1}, \quad k \geq 0 \\ y_0(p) &= f(p), \quad 0 \leq p \leq \alpha - 1 \end{aligned} \tag{9.3}$$

where d_{k+1} is an $n \times 1$ vector with constant entries and $f(p)$ is an $m \times 1$ vector whose entries are known functions of p.

9.3 Stability and Stabilization of Discrete Linear Repetitive Processes

The stability theory [17] for linear repetitive processes is based on an abstract model in a Banach space setting that includes a wide range of such processes as special cases, including those described by (9.2) and (9.3). In terms of their dynamics it is the pass-to-pass coupling (noting again their unique feature) which is critical. This is of the form $y_{k+1} = L_\alpha y_k$, where $y_k \in E_\alpha$ (E_α a Banach space with norm $||\cdot||$) and L_α is a bounded linear operator mapping E_α into itself. (In the case of processes described by (9.2) and (9.3), L_α is a discrete linear systems convolution operator.)

Asymptotic stability, i.e. bounded-input bounded-output (BIBO) stability over the fixed finite pass length $\alpha > 0$, requires the existence of finite real scalars $M_\alpha > 0$ and $\lambda_\alpha \in (0, 1)$ such that $||L_\alpha^k|| \leq M_\alpha \lambda_\alpha^k, k \geq 0$, where $||\cdot||$ also denotes the induced operator norm. For the processes described by (9.2) and (9.3) it has been shown elsewhere (see, for example, Chap. 3 of Rogers et al. [17]) that this property holds if, and only if, all eigenvalues of the matrix D_0 have modulus strictly less than unity—written here as $r(D_0) < 1$ where $r(\cdot)$ denotes the spectral radius of its matrix argument.

Suppose that $r(D_0) < 1$ and the input sequence applied $\{u_{k+1}\}_k$ converges strongly as $k \to \infty$ (i.e. in the sense of the norm on the underlying function space) to u_∞. Then the strong limit $y_\infty := \lim_{k \to \infty} y_k$ is termed the limit profile corresponding to this input sequence and its dynamics are described by

$$\begin{aligned} x_\infty(p+1) &= (A + B_0(I - D_0)^{-1}C)x_\infty(p) \\ &\quad + (B + B_0(I - D_0)^{-1}D)u_\infty(p) \\ y_\infty(p) &= (I - D_0)^{-1}Cx_\infty(p) \\ &\quad + (I - D_0)^{-1}Du_\infty(p) \\ x_\infty(0) &= d_\infty, \end{aligned} \tag{9.4}$$

where (again a strong limit) $d_\infty := \lim_{k\to\infty} d_k$. In physical terms, this result states that under asymptotic stability the repetitive dynamics can, after a "sufficiently large" number of passes have elapsed, be replaced by those of a 1D discrete linear system. In particular, this property demands that the amplifying properties of the coupling between successive pass profiles are completely damped out after a sufficiently large number of passes have elapsed. This fact has clear implications in terms of the control of these processes—see [17] for a detailed treatment of this point.

The finite pass length means that the limit profile can have unacceptable along the pass dynamics and, in particular, be unstable in the 1D linear systems sense. A simple example here is the case when $A = -0.5, B = 1, B_0 = 0.5 + \beta$, $C = 1, D = D_0 = 0$, where $\beta > 0$ is a real scalar. Hence if $|\beta| \geq 1$ the limit profile is unstable.

If we wish to avoid cases such as this example from arising, one route is to demand the BIBO stability property for any possible value of the pass length (mathematically this can be analyzed by letting $\alpha \to \infty$). This is the stability along the pass property which requires the existence of finite real scalars $M_\infty > 0$ and $\lambda_\infty \in (0, 1)$ that are independent of α and are such that $||L_\alpha^k|| \leq M_\infty \lambda_\infty^k, k \geq 0$.

Numerous sets of necessary and sufficient conditions for stability along the pass of (9.2) and (9.3) are known but here it is the following result which is the starting point.

Theorem 1.1 [6]. *A discrete linear repetitive process described by* (9.2) *and* (9.3) *is stable along the pass if there exist matrices* $W_1 \succ 0$ *and* $W_2 \succ 0$ *such that*

$$\widehat{A}^T W \widehat{A} - W \prec 0 \tag{9.5}$$

where $W = \text{diag}\{W_1, W_2\} \succ 0$, *and*

$$\widehat{A} = \begin{bmatrix} A & B_0 \\ C & D_0 \end{bmatrix} \tag{9.6}$$

Even though this condition is sufficient but not necessary it forms a basis of control law design via a Lyapunov function interpretation. In particular, follow [14] and introduce the candidate Lyapunov function as

$$\mathcal{V}(k, p) = x_{k+1}^T(p) W_1 x_{k+1}(p) + y_k^T(p) W_2 y_k(p) \tag{9.7}$$

where $W_1 \succ 0, W_2 \succ 0$, with associated increment

$$\Delta \mathcal{V}(k,p) = x_{k+1}^T(p+1)W_1 x_{k+1}(p+1) + y_{k+1}^T(p)W_2 y_{k+1}(p) - x_{k+1}^T(p)W_1 x_{k+1}(p) - y_k^T(p)W_2 y_k(p) \tag{9.8}$$

Then it is easy to show that

$$\Delta \mathcal{V}(k,p) < 0 \tag{9.9}$$

is equivalent to (9.5).

An extensively analyzed control law for processes described by (9.2) and (9.3) (see, for example, [14]) has the following form over $0 \leq p \leq \alpha - 1, k \geq 0$

$$u_{k+1}(p) = [K_1 \quad K_2] \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} \tag{9.10}$$

where K_1 and K_2 are appropriately dimensioned matrices to be designed. In effect, this control law is composed of a weighted sum of current pass state feedback and feedforward of the previous pass profile.

The LMI of (9.5) extends in a natural manner to the design of (9.10) for stability along the pass but here we will use the approach based on [2, 9, 15] and first adopted for repetitive processes in [6]. This will prove to be of particular use in the analysis of the case when there is uncertainty associated with the process state–space model.

Theorem 1.2 *Suppose that a control law of the form* (9.10) *is applied to a discrete linear repetitive process described by* (9.2) *and* (9.3). *Then the resulting process is stable along the pass if there exist matrices* $W = \text{diag}\{W_1, W_2\}, W_1 \succ 0, W_2 \succ 0$, G, *and*

$$N = \begin{bmatrix} \overline{N}_1 & \overline{N}_2 \\ \overline{N}_1 & \overline{N}_2 \end{bmatrix} \tag{9.11}$$

such that

$$\begin{bmatrix} -G - G^T + W & (\widehat{A}G + \widehat{B}N)^T \\ \widehat{A}G + \widehat{B}N & -W \end{bmatrix} \prec 0 \tag{9.12}$$

If this condition holds, stabilizing K_1 *and* K_2 *in the control law* (9.10) *are given by*

$$K = NG^{-1} \tag{9.13}$$

where matrices K *and* $\widehat{B}$ *are given by*

$$K = \begin{bmatrix} K_1 & K_2 \\ K_1 & K_2 \end{bmatrix}, \quad \widehat{B} = \begin{bmatrix} B & 0 \\ 0 & D \end{bmatrix} \tag{9.14}$$

In implementation terms, this control law requires that all elements of the current pass state vector are available for measurement. If this is not true then an observer will required.

Remark 1 The LMIs here are very similar to those known from the literature (see, for example, [2, 9, 15]) for 1D linear systems. This does not, of course, mean that repetitive processes can be analyzed by direct application of existing 1D linear systems theory, merely that in some cases recourse can be made to tools from this latter area. Even then, there are two major differences. The first of these is that the decision matrices must be block-diagonal where the diagonal entries are of dimensions $n \times n$ and $m \times m$ respectively (the first corresponds to the current pass state vector and the second the previous pass profile—see the form of the Lyapunov function). Secondly, even with no uncertainty, the Lyapunov based stability analysis for linear repetitive processes only gives a sufficient condition and hence there is always some conservativeness present.

9.4 Robust Stability and Stabilization of Discrete Linear Repetitive Processes

In addition to Theorem 1.2 of the previous section, the design of control laws for discrete linear repetitive processes has been the subject of much research effort—see, for example, [5, 6, 7,14]. Here, we continue the development of this general area by establishing new results related to the practical case where there is possibly large uncertainty associated with the process (state–space model) description. In particular, we consider the case when the model matrices $\widehat{A}$ and $\widehat{B}$ defined by (9.6) and (9.14) respectively are not precisely known, but belong to a convex bounded (polytope type) uncertain domain denoted here by $\mathcal{D}$. This, in turn, means that any uncertain matrix can be written as a convex combination of the vertices of $\mathcal{D}$ as follows

$$\mathcal{D} = \left\{ \left[\widehat{A}(\xi(k,p)), \widehat{B}(\xi(k,p))\right] : \left[\widehat{A}(\xi(k,p)), \widehat{B}(\xi(k,p))\right] = \sum_{i=1}^{v} \xi_i(k,p)\left[\widehat{A}_i, \widehat{B}_i\right], \right.$$
$$\left. \sum_{i=1}^{v} \xi_i(k,p) = 1, \xi_i(k,p) \geq 0, k \geq 0, 0 \leq p \leq \alpha - 1 \right\} \tag{9.15}$$

where v denotes the number of vertices. Note also that the uncertainty here is variable in both independent directions of information propagation, i.e. along the pass (p direction) and pass-to-pass (k direction).

At this stage, we can write the following linear parameter dependent state–space model describing the process dynamics

$$\begin{aligned} x_{k+1}(p+1) &= A(\xi(k,p))x_{k+1}(p) + B(\xi(k,p))u_{k+1}(p) + B_0(\xi(k,p))y_k(p) \\ y_{k+1}(p) &= C(\xi(k,p))x_{k+1}(p) + D(\xi(k,p))u_{k+1}(p) + D_0(\xi(k,p))y_k(p) \end{aligned} \tag{9.16}$$

Consider also the parameterized candidate Lyapunov function

$$\mathcal{V}(k,p,\xi(k,p)) = x_{k+1}^T(p)W_1(\xi(k,p))x_{k+1}(p) + y_k^T(p)W_2(\xi(k,p))y_k(p) \tag{9.17}$$

with

$$\begin{aligned} W_1(\xi(k,p)) &= \sum_{i=1}^{v} \xi_i(k,p)W_{i1} \\ W_2(\xi(k,p)) &= \sum_{i=1}^{v} \xi_i(k,p)W_{i2} \end{aligned} \tag{9.18}$$

and $\mathcal{V}(0,0,\xi(0,0)) < \infty$, and associated increment

$$\begin{aligned} \Delta\mathcal{V}(k,p,\xi(k,p)) = {} & x_{k+1}^T(p+1)W_1(\xi(k,p+1))x_{k+1}(p+1) \\ & + y_{k+1}^T(p)W_2(\xi(k+1,p))y_{k+1}(p) - x_{k+1}^T(p)W_1(\xi(k,p))x_{k+1}(p) \\ & - y_k^T(p)W_2(\xi(k,p))y_k(p) \end{aligned} \tag{9.19}$$

Then we can define so-called poly-quadratic stability (see [2, 9, 15] for the 1D systems case) as follows.

Definition 1.3 A discrete linear repetitive process described by (9.2) and (9.3) with uncertainty defined by (9.15) and Lyapunov function (9.17) and (9.18) is said to be poly-quadratically stable provided

$$\Delta\mathcal{V}(k,p,\xi(k,p)) < 0 \tag{9.20}$$

for all $k \geq 0, 0 \leq p \leq \alpha - 1$.

The requirement of (9.20) can be written as

$$\widehat{A}(\xi(k,p))^T\mathcal{W}^+\widehat{A}(\xi(k,p)) - \mathcal{W} \prec 0 \tag{9.21}$$

where $\mathcal{W} = \text{diag}(\{W_1(\xi(k,p))), W_2(\xi(k,p))\}$ is defined by (9.18)

$$\begin{aligned} \mathcal{W}^+ &= \text{diag}\{W_1(\xi(k,p+1)), W_2(\xi(k+1,p))\} \\ &= \sum_{i=1}^{v} \text{diag}\{\xi_i(k,p+1)W_{i1}, \xi_i(k+1,p)W_{i2}\} \\ &= \sum_{i=1}^{v} \zeta_i(k,p)\text{diag}\{W_{i1}, W_{i2}\} \end{aligned}$$

with $\sum_{i=1}^{v} \zeta_i(k,p) = 1; \zeta_i(k,p) \geq 0; k \geq 0; 0 \leq p \leq \alpha - 1$. We also require that $\xi_i(k,p+1) = \xi_i(k+1,p) = \zeta_i(k,p)$ and the matrix $\widehat{A}$ of (9.6) in this case becomes

$$\widehat{A}(\xi(k,p)) = \begin{bmatrix} A(\xi(k,p)) & B_0(\xi(k,p)) \\ C(\xi(k,p)) & D_0(\xi(k,p)) \end{bmatrix} = \sum_{i=1}^{v} \xi_i(k,p)\widehat{A}_i \tag{9.22}$$

where $\widehat{A}_i$ are the polytope vertices (see 9.15).

Remark 2 When

$$\begin{aligned} &\text{diag}\{W_1(\xi(k,p+1)), \\ &\quad W_2(\xi(k+1,p))\} = \text{diag}\{W_1(\xi(k,p)), W_2(\xi(k,p))\} = W \end{aligned}$$

and hence $\xi_i(k,p) = \zeta_i(k,p), i = 1,\ldots,v$, poly-quadratic stability reduces to stability along the pass (Theorem 1.1).

The following result (drawing on the work in [2]) aims to minimize the conservativeness present from the use of a sufficient, but not necessary, stability condition.

Theorem 1.4 *A discrete linear repetitive process described by* (9.2) *and* (9.3) *with uncertainty defined by* (9.15) *is poly-quadratically stable if there exists block-diagonal matrices* $S_i \succ 0, i = 1,\ldots,v$, *i.e.* $S_i = \text{diag}\{S_{i1}, S_{i2}\}$, *and a matrix* G *such that*

$$\begin{bmatrix} G + G^T - S_i & G^T\widehat{A}_i^T \\ \widehat{A}_i G & S_j \end{bmatrix} \succ 0 \tag{9.23}$$

for all $i,j = 1,\ldots,v$.

Proof Assume that (9.23) is feasible for all $i,j = 1,\ldots,v$. Then

$$G + G^T - S_i \succ 0$$

and, since G is full rank and $S_i \succ 0$,

$$(S_i - G)^T S_i^{-1}(S_i - G) \succeq 0$$

or, equivalently,

$$G^T S_i^{-1} G \succeq G^T + G - S_i$$

Hence if (9.23) holds

$$\begin{bmatrix} G^T S_i^{-1} G & G^T\widehat{A}_i^T \\ \widehat{A}_i G & S_j \end{bmatrix} \succ 0$$

or equivalently,

$$\begin{bmatrix} G^T & 0 \\ 0 & S_j \end{bmatrix} \begin{bmatrix} S_i^{-1} & \widehat{A}_i^T S_j^{-1} \\ S_j^{-1}\widehat{A}_i & S_j^{-1} \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & S_j \end{bmatrix} \succ 0$$

Next, introduce the substitutions $W_i = S_i^{-1}$ and $W_j = S_j^{-1}$ to obtain

$$\begin{bmatrix} W_i & \widehat{A}_i^T W_j \\ W_j \widehat{A}_i & W_j \end{bmatrix} \succ 0$$

for all $i, j = 1, \ldots, v$. Further, for each i, multiply the corresponding inequalities for $j = 1, \ldots, v$ by $\zeta_j(k,p)$ and sum over j to obtain

$$\begin{bmatrix} W_i & \widehat{A}_i^T \left(\sum_{j=1}^{v} \zeta_j(k,p) W_j \right) \\ \left(\sum_{j=1}^{v} \zeta_j(k,p) W_j \right) \widehat{A}_i & \sum_{j=1}^{v} \zeta_j(k,p) W_j \end{bmatrix} \succ 0$$

Also $\sum_{j=1}^{v} \zeta_j(k,p) W_j = \mathcal{W}^+$ and multiplying the resulting inequalities by $\xi_i(k,p)$ for $i = 1, \ldots, v$, and summing over i gives

$$\begin{bmatrix} \sum_{i=1}^{v} \xi_i(k,p) W_i & \left(\sum_{i=1}^{v} \xi_i(k,p) \widehat{A}_i^T \right) \mathcal{W}^+ \\ \mathcal{W}^+ \left(\sum_{i=1}^{v} \xi_i(k,p) \widehat{A}_i \right) & \mathcal{W}^+ \end{bmatrix} \succ 0$$

or, equivalently,

$$\begin{bmatrix} \mathcal{W} & \widehat{A}^T(\xi(k,p)) \mathcal{W}^+ \\ \mathcal{W}^+ \widehat{A}(\xi(k,p)) & \mathcal{W}^+ \end{bmatrix} \succ 0 \tag{9.24}$$

Finally, applying Definition 1.3, followed by an obvious application of the Schur's complement formula, to (9.24) gives

$$\mathcal{W} - \widehat{A}(\xi(k,p))^T \mathcal{W}^+ \widehat{A}(\xi(k,p)) \succ 0 \tag{9.25}$$

which is equivalent to (9.21) and the proof is complete. □

Remark 3 Recall that the diagonal structure of S_i in this last result arises directly from the stability theory for discrete linear repetitive processes. As noted previously, this leads to only sufficient conditions and hence some conservativeness can be present which is, however, lower than if the matrices G_i were also taken as block-diagonal (these matrices only relate to the corresponding LMI construction).

With the control law (9.10) applied, (9.23) becomes

$$\begin{bmatrix} G + G^T - S_i & G^T (\widehat{A}_i + \widehat{B}_i K)^T \\ (\widehat{A}_i + \widehat{B}_i K) G & S_j \end{bmatrix} \succ 0 \tag{9.26}$$

where K is defined in (9.14), and the following result now gives a sufficient condition for the existence of a poly-quadratically stabilizing control law of the form (9.10).

Theorem 1.5 *Suppose that a control law of the form* (9.10) *is applied to a discrete linear repetitive process described by* (9.2) *and* (9.3) *with uncertainty defined by* (9.15). *Then the resulting process is poly-quadratically stabilizable if there exist matrices* $S_i \succ 0, i = 1, \ldots, v$, *i.e.* $S_i = \text{diag}\{S_{i1}, S_{i2}\}$, G, *and* N (defined by (9.11)) *such that*

$$\begin{bmatrix} G + G^T - S_i & G^T \widehat{A}_i^T + N^T \widehat{B}_i^T \\ \widehat{A}_i G + \widehat{B}_i N & S_j \end{bmatrix} \succ 0 \tag{9.27}$$

for all $i, j = 1, \ldots, v$. *If this condition holds then stabilizing* K_1 *and* K_2 *in the control law are given by* (9.10) *with*

$$K = NG^{-1} \tag{9.28}$$

where K *was defined in* (9.14).

Proof This result follows immediately from (9.26), on setting $KG = N$. □

In the next section we consider parameter variable control laws in an attempt to reduce the level of conservativeness associated with the results so far.

9.5 A Parameter Variable Control Law

To enlarge the admissible uncertainty range for which stabilization is still possible (and hence reduce conservativeness), this section considers a parameter variable control law of the form

$$\begin{aligned} u_{k+1}(p) &= K(\xi(k,p)) \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} = [K_1(\xi(k,p)) \quad K_2(\xi(k,p))] \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} \\ &= \sum_{i=1}^{v} \xi_i(k,p) [K_{i1} \quad K_{i2}] \begin{bmatrix} x_{k+1}(p) \\ y_k(p) \end{bmatrix} \end{aligned} \tag{9.29}$$

over $0 \leq p \leq \alpha - 1, k \geq 0$ where the designed matrix $K(\xi(k,p))$ contains uncertainty as defined in (9.15). In effect, this control law is composed of the weighted sum of current pass state feedback and feedforward of the previous pass profile and we have the following result (which can be interpreted as the generalization to the repetitive process case of a well known result [2]).

Theorem 1.6 *A discrete linear repetitive process described by* (9.2) *and* (9.3) *with uncertainty structure of the form* (9.15) *is poly-quadratically stable, if there exist matrices* $S_i \succ 0$, *i.e.* $S_i = \text{diag}\{S_{i1}, S_{i2}\} \succ 0$, *and* $G_i, i = 1, \ldots v$, *such that*

$$\begin{bmatrix} G_i + G_i^T - S_i & G_i^T \widehat{A}_i^T \\ \widehat{A}_i G_i & S_j \end{bmatrix} \succ 0 \tag{9.30}$$

for all $i, j = 1, \ldots, v$.

Proof Assume that (9.30) is feasible for all $i,j = 1,\ldots,v$. Then

$$G_i + G_i^T - S_i \succ 0$$

and, since G_i is full rank and $S_i \succ 0$

$$(S_i - G)^T S_i^{-1}(S_i - G_i) \succeq 0$$

or, equivalently,

$$G_i^T S_i^{-1} G_i \succeq G_i^T + G_i - S_i$$

Hence if (9.30) holds

$$\begin{bmatrix} G_i^T S_i^{-1} G_i & G_i^T \widehat{A}_i^T \\ \widehat{A}_i G_i & S_j \end{bmatrix} \succ 0$$

or equivalently,

$$\begin{bmatrix} G_i{}^T & 0 \\ 0 & S_j \end{bmatrix} \begin{bmatrix} S_i^{-1} & \widehat{A}_i^T S_j^{-1} \\ S_j^{-1}\widehat{A}_i & S_j^{-1} \end{bmatrix} \begin{bmatrix} G_i & 0 \\ 0 & S_j \end{bmatrix} \succ 0$$

Next, introduce the substitutions $W_i = S_i^{-1}$ and $W_j = S_j^{-1}$ to obtain

$$\begin{bmatrix} W_i & \widehat{A}_i^T W_j \\ W_j \widehat{A}_i & W_j \end{bmatrix} \succ 0$$

for all $i,j = 1,\ldots,v$. Further, for each i, multiply the corresponding inequalities for $j = 1,\ldots,v$ by $\zeta_j(k,p)$ and sum over j to obtain

$$\begin{bmatrix} W_i & \widehat{A}_i^T \left(\sum_{j=1}^v \zeta_j(k,p) W_j\right) \\ \left(\sum_{j=1}^v \zeta_j(k,p) W_j\right) \widehat{A}_i & \sum_{j=1}^v \zeta_j(k,p) W_j \end{bmatrix} \succ 0$$

Also $\sum_{j=1}^v \zeta_j(k,p) W_j = \mathcal{W}^+$ and multiplying the resulting inequalities by $\xi_i(k,p)$ for $i = 1,\ldots,v$, and summing over i gives

$$\begin{bmatrix} \sum_{i=1}^v \xi_i(k,p) W_i & \left(\sum_{i=1}^v \xi_i(k,p) \widehat{A}_i^T\right) \mathcal{W}^+ \\ \mathcal{W}^+ \left(\sum_{i=1}^v \xi_i(k,p) \widehat{A}_i\right) & \mathcal{W}^+ \end{bmatrix} \succ 0$$

or, equivalently,

$$\begin{bmatrix} \mathcal{W} & \widehat{A}^T(\xi(k,p)) \mathcal{W}^+ \\ \mathcal{W}^+ \widehat{A}(\xi(k,p)) & \mathcal{W}^+ \end{bmatrix} \succ 0 \tag{9.31}$$

Applying Definition 1.3, followed by an obvious application of the Schur's complement formula, to (9.31) now gives

$$\mathcal{W} - \widehat{A}(\xi(k,p))^T \mathcal{W}^+ \widehat{A}(\xi(k,p)) \succ 0 \tag{9.32}$$

which is equivalent to (9.21) and the proof is complete. □

Applying the control law (9.29) to (9.30) gives

$$\begin{bmatrix} G_i + G_i^T - S_i & G_i^T(\widehat{A}_i + \widehat{B}_i K_i)^T \\ (\widehat{A}_i + \widehat{B}_i K_i) G_i & S_j \end{bmatrix} \succ 0 \tag{9.33}$$

where the matrix $K_i, i = 1, \ldots, v$, is given by

$$K_i = \begin{bmatrix} K_{i1} & K_{i2} \\ K_{i1} & K_{i2} \end{bmatrix} \tag{9.34}$$

The following result now gives a condition sufficient for the existence of a poly-quadratically stabilizing control law of the form (9.29) together with a design algorithm.

Theorem 1.7 *Suppose that a control law of the form* (9.29) *is applied to a discrete linear repetitive process described by* (9.2) *with uncertainty of the form* (9.15). *Then the resulting process is poly-quadratically stable if there exist block-diagonal matrices* $S_i \succ 0$, *i.e.* $S_i = \text{diag}\{S_{i1}, S_{i2}\} \succ 0, G_i$, *and*

$$N_i = \begin{bmatrix} N_{i1} & N_{i2} \\ N_{i1} & N_{i2} \end{bmatrix} \tag{9.35}$$

$i = 1, \ldots, v$, *such that*

$$\begin{bmatrix} G_i + G_i^T - S_i & G_i^T \widehat{A}_i^T + N_i^T \widehat{B}_i^T \\ \widehat{A}_i G_i + \widehat{B}_i N_i & S_j \end{bmatrix} \succ 0 \tag{9.36}$$

for all $i, j = 1, \ldots, v$. *If these conditions hold, the vertex matrices* K_{i1} *and* K_{i2} *in the stabilizing control law of* (9.29) *are given by*

$$K_i = N_i G_i^{-1} \tag{9.37}$$

where K_i *is defined in* (9.34).

Proof This follows immediately from (9.33) on setting $K_i G_i = N_i$. □

Theorem 1.7 and (9.29) provides the setting for the construction of the control law considered to stabilize the process with much wider uncertainty, or variability

domains, than alternatives. However, the final control law is still not available at this stage as only the vertex values of the required matrices are known. To complete the task, we need to know the exact parameters $\xi_i(k,p),\ k \geq 0,\ 0 \leq p \leq \alpha - 1$, and $i = 1, \ldots, v$, which can be recovered from the process dynamics by using the Matlab function "fmincon" (or any equivalent algorithm) to solve the following problem:

Determine $\xi_i(k,p) \in \mathbf{R}^+, i = 1, \ldots, v$ such that

$$\sum_{i=1}^{v} \xi_i(k,p) V_i = P(k,p) \tag{9.38}$$

where

$$\sum_{i=1}^{v} \xi_i(k,p) = 1, \xi_i(k,p) \geq 0, k \geq 0, 0 \leq p \leq \alpha - 1$$

and

$$P(k,p) = \{A(k,p), B(k,p), B_0(k,p), C(k,p), D(k,p), D_0(k,p)\} \tag{9.39}$$

where V_i denotes the convex domain vertices and $P(k,p)$ the process state–space model matrix which is assumed to lie in the polytope constructed from them. (This construction will be explained in detail for the material rolling example considered in the next section).

Given $\xi_i(k,p), k \geq 0, 0 \leq p \leq \alpha - 1, i = 1, \ldots, v$, and the vertex matrices of the control law (9.29) from (9.37), we can complete the design using

$$K(\xi(k,p)) = [\, K_1(\xi(k,p)) \quad K_2(\xi(k,p)) \,] = \sum_{i=1}^{v} \xi_i(k,p) [\, K_{i1} \quad K_{i2} \,] \tag{9.40}$$

9.6 Application to Material Rolling

In this section we illustrate Theorem 1.7 by application to the material rolling model of Sect. 9.2 when the model parameters λ_1, λ_2 are uncertain and the rest of parameters (i.e. T and M) are constant. Also we take $T = 0.2$ s, $M = 100$ kg, and assume that the model parameters λ_1 and λ_2 satisfy

$$\lambda_1 \in [\underline{\lambda_1}, \overline{\lambda_1}] = [0.216, 0.984], \quad \lambda_2 \in [\underline{\lambda_2}, \overline{\lambda_2}] = [0.72, 3.28] \tag{9.41}$$

Note first that a control law of the form of Theorem 1.5 can be computed in this case but requires that λ_1 and λ_2 satisfy

$$\lambda_1 \in [\underline{\lambda_1}, \overline{\lambda_1}] = [0.228, 0.978], \quad \lambda_2 \in [\underline{\lambda_2}, \overline{\lambda_2}] = [0.76, 3.26]$$

i.e. this design comes at the price of a more restrictive range for the values of λ_1 and λ_2.

The fact that this problem has two variables that can vary means that there are four uncertainty domain vertices. Also the uncertainty domain for the process matrices is easily verified as convex with vertices as follows
vertex 1

$$A = \begin{bmatrix} 0.9377 & 187.5361 \\ -3.116 \cdot 10^{-4} & 0.9377 \end{bmatrix}, \quad B = \begin{bmatrix} -0.0866 \\ -4.3278 \cdot 10^{-4} \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0.0144 \\ 7.1907 \cdot 10^{-5} \end{bmatrix}$$
$$C = [0.9377 \quad 187.5361], \quad D = -0.0866, \quad D_0 = 0.7836$$

vertex 2

$$A = \begin{bmatrix} 0.7676 & 153.5191 \\ -1.162 \cdot 10^{-3} & 0.7676 \end{bmatrix}, \quad B = \begin{bmatrix} -0.0709 \\ -3.5427 \cdot 10^{-4} \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0.0536 \\ 2.6816 \cdot 10^{-4} \end{bmatrix}$$
$$C = [0.7676 \quad 153.5191], \quad D = -0.0709, \quad D_0 = 0.8229$$

vertex 3

$$A = \begin{bmatrix} 0.8574 & 171.481 \\ -7.1297 \cdot 10^{-4} & 0.8574 \end{bmatrix}, \quad B = \begin{bmatrix} -0.198 \\ -9.9024 \cdot 10^{-4} \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0.0823 \\ 4.1172 \cdot 10^{-4} \end{bmatrix}$$
$$C = [0.8574 \quad 171.481], \quad D = -0.198, \quad D_0 = 0.5049$$

vertex 4

$$A = \begin{bmatrix} 0.925 & 185.0033 \\ -3.7492 \cdot 10^{-4} & 0.925 \end{bmatrix}, \quad B = \begin{bmatrix} -0.0229 \\ -1.143 \cdot 10^{-4} \end{bmatrix}, \quad B_0 = \begin{bmatrix} 4.6328 \cdot 10^{-3} \\ 2.32 \cdot 10^{-5} \end{bmatrix}$$
$$C = [0.925 \quad 185.0033], \quad D = -0.0229, \quad D_0 = 0.9428$$

The parameters λ_1, λ_2 vary with k and p and, since they are different on each pass, we denote them by $\lambda_1(k,p)$ and $\lambda_2(k,p)$, respectively. Also they are assumed to lie in the fixed intervals given by (9.41) as shown in Fig. 9.2.

Theorem 1.7 in this case can provide a variety of possible control laws but it is very difficult to select the one which will also satisfy the performance specifications. To overcome this problem, we use the following corollary of Theorem 1.7.

Corollary 1.8 *Suppose that a control law of the form* (9.29) *is applied to a discrete linear repetitive process described by* (9.2) *and* (9.3) *with uncertainty of the form* (9.15). *Then the resulting process is poly-quadratically stable if there exist block-diagonal matrices* S_i, *i.e.* $S_i = \text{diag}\{S_{i1}, S_{i2}\} \succ 0$, *and matrices* G_i *and* $N_i, i = 1, \ldots, v$, (*where* N_i *is defined in* (9.35)), *such that the following convex optimization problem has a solution*

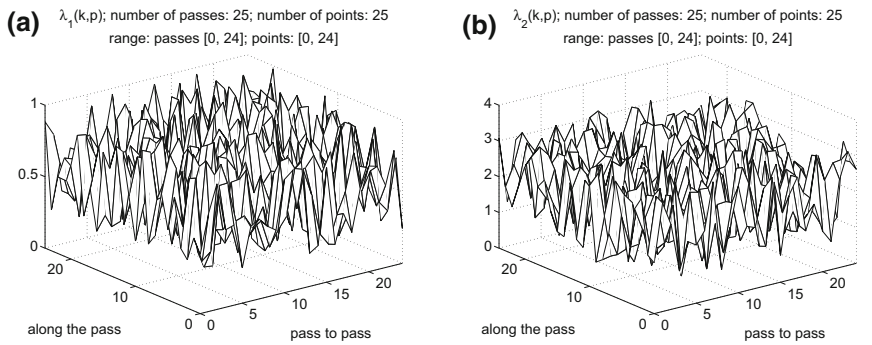


Fig. 9.2 **a** Values for $\lambda_1(k,p)$; **b** values for $\lambda_2(k,p)$

$$\begin{aligned} &\text{maximize} \quad F = \sum_{i=1}^{v} \text{trace}(G_i) \\ &\text{subject to} \\ &\begin{bmatrix} G_i + G_i^T - S_i & G_i^T \widehat{A}_i^T + N_i^T \widehat{B}_i^T \\ \widehat{A}_i G_i + \widehat{B}_i N_i & S_j \end{bmatrix} \succ 0 \end{aligned} \tag{9.42}$$

Also we require that the matrices G_i and S_i are diagonal for all $i = 1, \ldots, v$. The control law matrix vertices are obtained as in Theorem 1.7.

For the case considered here, this last result gives the following control law matrix vertices

$$\begin{aligned} K_{11} &= [3.4741 \quad 2166.7], & K_{12} &= 0.0873 \\ K_{21} &= [4.4876 \quad 2166.7], & K_{22} &= 0.1504 \\ K_{31} &= [2.0452 \quad 865.8534], & K_{32} &= 0.0818 \\ K_{41} &= [10.6 \quad 8092.6], & K_{42} &= 0.215 \end{aligned}$$

Now we are in a position to develop the procedure by which the variable control law matrices for a given k and point p are derived. Consider the matrix of (9.39) for given $T, M, \lambda_1(k,p)$ and $\lambda_2(k,p)$, where $A(k,p), \ldots, D_0(k,p)$ denote the system matrices $A, \ldots, D_0$ computed for $T = t, M = m$, and variable parameters $\lambda_1 = \lambda_1(k,p)$ and $\lambda_2 = \lambda_2(k,p)$ from (9.1)–(9.2) at given k, p. Having obtained $P(k,p)$ for k, p we can now recover from (9.38) (by using an algorithm based on the Matlab function "fmincon") the underling parameters $\xi_i(k,p)$, $k \geq 0, 0 \leq p \leq \alpha - 1$, and $i = 1, \ldots, v$. Then, since the matrix vertices at (k,p) are known, it is a simple task to calculate the variable control law matrices at (k,p) using (9.40). Next, we go to $p+1$ if $p < \alpha - 1$ with k unchanged or if $p = \alpha - 1$ we go to $k+1$ and set $p = 0$, and so on.

Without control action, the process here is unstable along the pass. Suppose also that the task is to reduce the thickness of the workpiece by one unit in the case when $\alpha = 25$. Then, since the process dynamics are assumed to be linear, we can take the boundary conditions to be $x_{k+1}(p) = 0, k \geq 0$, and $y_0(p) = 1, 0 \leq p \leq 24$.

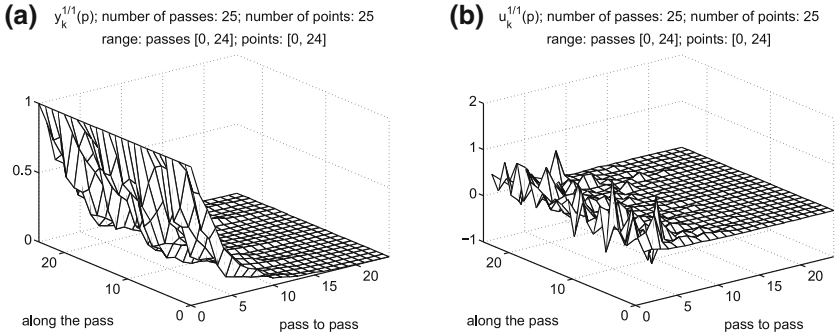


Fig. 9.3 **a** The process pass profile sequence for the controlled process; **b** the control input sequence

Hence as k increases the sequence of pass profiles should approach zero. This is confirmed by the plot of Fig. 9.3a and there are two other major issues which need to be considered. These are the transient behavior in both p and k and the magnitude of the control input signal respectively. If the transient performance is not acceptable, the only option is to return and attempt to tune the design. For the second, Fig. 9.3b shows the control input sequence required to apply the control law in this case. If this is unacceptable, then further development is required and again this is left as a subject for further work—the result here shows that the control action is bounded and hence baseline acceptable.

Finally, note that the control action energy required (maximum absolute value of the control signal) is much lower than that arising when Corollary 1.8 is not applied. Now, however, more passes must be completed before the control objectives are met.

9.7 Conclusions

This chapter has focused on the control of discrete linear repetitive processes in the presence of uncertainty in the state–space model used for control law design. A review of previous results in this area leads to the conclusion that the design of practically relevant control laws is possible, but the uncertainty model which has been used to obtain such results may be restrictive in the sense that there is little margin for tuning the control law matrices to obtain stability plus desired performance. The main objective in the work reported in this chapter is to develop control laws which vary in both the pass number and along the pass variables. This has been achieved by allowing the control law matrices to vary with both the pass number and the along the pass variable. The result is an algorithm for basic selection of the control law matrices without destroying the static nature of the control law.

References

1. Agathoklis P, Foda S (1989) Stability and the matrix Lyapunov equation for delay differential systems. Int J Control 49(2):417–432
2. Daafouz J, Bernussou J (2001) Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties. Syst Control Lett 43:355–359
3. Du C, Xie L (2002) H_∞ Control and Filtering of Two-dimensional Systems, volume 278 of Lecture notes in control and information sciences. Springer, Berlin
4. Gahinet P, Nemirowski A, Laub AJ, Chilali M (1995) LMI Control Toolbox for use with MATLAB. The Mathworks Partner Series. The MathWorks Inc
5. Gałkowski K, Rogers E, Xu S, Lam J, Owens DH (2002) LMIs—a fundamental tool in analysis and controller design for discrete linear repetitive processes. IEEE Trans Circuits Syst I: Fundam Theory Appl 49(6):768–778
6. Gałkowski K, Lam J, Rogers E, Xu S, Sulikowski B, Paszke W, Owens DH (2003) LMI based stability analisys and robust controller design for discrete linear repetitive processes. Int J Robust Nonlinear Control 13:1195–1211
7. Gałkowski K, Rogers E, Paszke W, Owens DH (2003) Linear repetitive process control theory applied to a physical example. Appl Math Comp Sci 13(1):87–99
8. Geromel JC, de Oliveira MC (2001) H_2 and H_∞ robust filtering for convex bounded uncertain systems. IEEE Trans Autom Control 46(1):100–107
9. Geromel JC, de Oliveira MC, Hsu L (1998) LMI characterization of structural and robust stability. Linear Algebra Appl 285:69–80
10. Hladowski L, Gałkowski K, Cai Z, Rogers E, Freeman CT, Lewin PL (2008) A 2D systems approach to iterative learning control with experimental verification. In: Proceedings of 17th IFAC World Congress, pp 2832–2837
11. Moore KL, Chen YQ, Bahl V (2005) Monotonically convergent iterative learning control for linear discrete-time systems. Automatica 41(9):1529–1537
12. Nesterov Y, Nemirovskii A (1994) Interior-point polynomial algorithms in convex programing, vol 13 of SIAM studies in applied mathematics. SIAM, Philadelphia
13. de Oliveira MC, Bernussou J, Geromel J (1999) A new discrete-time robust stability condition. Syst Control Lett 37:261–265
14. Paszke W, Gałkowski K, Rogers E, Owens DH (2006) H_∞ and guaranteed cost control of discrete linear repetitive processes. Linear Algebra Appl 412:93–131
15. Peaucelle D, Arzelier D, Bachelier O, Bernussou J (2000) A new robust D-stability condition for polytopic uncertainty. Syst Cont Lett 40:21–30
16. Roberts PD (2002) Two-dimensional analysis of an iterative nonlinear optimal control problem. IEEE Trans Circuits Syst I: Fundam Theory Appl 48(6):872–878
17. Rogers E, Gałkowski K, Owens DH (2007) Control systems theory and applications for linear repetitive processes, volume 349 of Lecture Notes in Control and Information Sciences. Springer, Heidelberg
18. de Souza CE, Trofino A (1999) Advances in linear matrix inequality methods in control: advances in design and control. In: Recent advances on linear matrix inequalities in control. Philadelphia. SIAM, pp 175–185
19. Tuan HD, Apkarian P, Nguyen TQ (2002) Robust mixed H_2/H_∞ filtering of 2-D systems. IEEE Trans Signal Process 50(7):1759–1771

Chapter 10
Unique Full-Rank Solution of the Sylvester-Observer Equation and Its Application to State Estimation in Control Design

Karabi Datta and Mohan Thapa

Abstract Needs to be found, is a classical equation. There has been much study, both from theoretical and computational view points, on this equation. The results of existence and uniqueness are well-known and numerically effective algorithms have been developed in recent years (see, Datta [2]), to compute the solution.

10.1 Introduction

The Sylvester matrix equation

$$AX - XF = R \tag{10.1}$$

where A, F and R are given and X needs to be found, is a classical equation. There has been much study, both from theoretical and computational view points, on this equation. The results of existence and uniqueness are well-known and numerically effective algorithms have been developed in recent years (see, Datta [2]), to compute the solution X. A variation of this equation called the Sylvester-observer

Dedicated to **Biswa Datta** for his contribution to numerical aspects of control theory, in particular to Sylvester-observer equation.

K. Datta (✉) · M. Thapa
Department of Mathematical Sciences, Northern Illinois University,
DeKalb, IL 60115, USA
e-mail: dattak@math.niu.edu

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_10,

equation (Luenberger [3], Datta [2]) arises in the design of Luenberger observer in Control Theory. In this variation, the matrix A and partial informations of F and R are known, the matrix X and the unknown parts of F and R are to be found. The matrix F is required to have a pre-assigned spectrum and the matrix X must be a full-rank matrix.

Several computationally viable algorithms have been developed in recent years for the solution of the Sylvester-observer equation. These includes (i) the observer-Hessenberg method of Van Dooren [4], (ii) generalization of the Van Dooran method to the block case by Carvalho and Datta [5], (iii) a new block algorithm by Carvalho et al. [6], (iv) the SVD-based algorithm by Datta and Sarkissian [7], (v) the Arnoldi-based method for large-scaled solution by Datta and Saad [8], Calvetti et al. [9] and a parallel and high performance algorithm by Bischof et al. [10].

All the above methods implicitly construct a full-rank solution assuming that such a solution exists. But, no systematic study has been done yet. The purpose of this paper is to study the existence and uniqueness of a full-rank solution to the Sylvester-observer equation, when the spectrum F is prescribed in advance. The results are new and applicable in a computational setting to determine if such a solution exists, when the matrix A and partial information on F and R are given.

10.2 Full-Rank Solution of the Sylvester Equation

Consider the Sylvester equation

$$AX - XF = R, \tag{10.2}$$

where A, F, R are respectively, of order $n \times n$, $s \times s$, and $n \times s$. Let X be a unique solution of Eq. (10.2); that is, $\Omega(A) \cap \Omega(F) = \phi$. The following result on the existence of a full-rank solution of (10.2) was proved by de Souza and Bhattacharyya [11].

Theorem 1 [11] *Necessary conditions for the unique solution X of* (10.2) *to be of full-rank are that* (A, R) *is controllable and* (R, F) *is observable. The condition is also sufficient if R has rank 1.*

A necessary and sufficient condition in the general case was obtained in Datta et al. [12].

Theorem 2 [12] *Let* A, F *and* R *be, receptively, the* $n \times n$, $s \times s$ *and* $n \times s$ *matrices. Then*

$$AX - XF = R \tag{10.3}$$

has unique a full-rank solution if and only if the matrix

$$S_k(R) = \left[R\ AR \cdots A^k R\right]_{n,(k+1)s} \begin{pmatrix} I & a_1 I & \cdots & & a_k I \\ 0 & I & \cdots & & a_{k-1} I \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & I & a_1 I \\ 0 & \cdots & \cdots & 0 & I \end{pmatrix}_{(k+1)s,(k+1)s} \begin{pmatrix} F^k \\ \vdots \\ I \end{pmatrix}_{(k+1)s,s}$$

has full-rank s, where $0 < k+1 \leq s$ *is the degree of the minimal or the characteristic polynomial of* F*, and* a_i*'s are the coefficients of the characteristic (minimal) polynomial of* F*.*

Forming the matrix $S_k(R)$ and checking its rank numerically is a computationally prohibitive task. Below we now show how these computations can be more effective and practical by taking advantage of results from control theory and certain results exploiting the structures of the associated matrices.

it is well known (see, Datta [2]) that a controllable pair (A, R) can be transformed to a controller-Hessenberg form $(H, \tilde{R})$ by an orthogonal similarity, where H is a block upper-Hessenberg matrix and $\tilde{R}$ has the special structure, as given below. We will assume that (A, R) has been given in that form; that is, $A \equiv H$, and $R \equiv \tilde{R}$: thus

$$A \equiv H = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1p} \\ H_{21} & H_{22} & \cdots & H_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & & H_{pp-1} & H_{pp} \end{pmatrix}_{n \times n},$$

$$R = \tilde{R} = \begin{pmatrix} (R_1)_{n_1 \times n_1} & \cdots & (0)_{n_1 \times n_r} \\ (0)_{n_2 \times n_1} & \cdots & (0)_{n_2 \times n_r} \\ \vdots & & \vdots \\ (0)_{n_p \times n_1} & & (0)_{n_p \times n_r} \end{pmatrix}_{n \times s}$$

where $r \leq p,\ s \leq n,\ (n_1 + n_2 + \cdots + n_r) = s,\ (n_1 + n_2 + \cdots + n_p) = n$, and (H_{ij}) is $n_i \times n_j$ where $i = 1 :\ n_p;\ j = 1 : n_p$. Let rank $(R) = n_1$. Also, without any loss of generality, let us assume that the matrix F has the block lower Hessenberg form:

$$F = \begin{pmatrix} (F_{11})_{n_1 \times n_1} & F_{12} & 0 & \cdots & (0)_{n_1 \times n_r} \\ (F_{21})_{n_1 \times n_2} & (F_{22})_{n_2 \times n_2} & F_{23} & & \vdots \\ * & & \ddots & \ddots & 0 \\ \vdots & & & F_{r-1,r-1} & F_{rr-1} \\ * & \cdots & \cdots & * & (F_{rr})_{n_r \times n_r} \end{pmatrix}_{s \times s}$$

It is easy to see that the qth powers of H and F are given by $(q < p)$

$$H^q = (H_{ij}^q) = \begin{pmatrix} H_{11}^{(q)} & H_{12}^{(q)} & \cdots & \cdots & \cdots & H_{1p}^{(q)} \\ H_{21}^{(q)} & * & \cdots & & \cdots & * \\ \vdots & \ddots & & & & \vdots \\ H_{q+1,1}^{(q)} & * & \cdots & \cdots & \cdots & \vdots \\ 0 & H_{q+2,2}^{(q)} & \ddots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & H_{p,p-q}^{(q)} & \cdots & H_{p,p-1}^{(q)} & H_{p,p}^{(q)} \end{pmatrix}$$

where $H_{q+1,1}^{(q)} = (H_{q+1,q} \cdot H_{q\cdot q-1} \cdots H_{32} H_{21})_{n_{q+1},n_1}$, and

$$F^q = \left(F_{ij}^{(q)}\right) = \begin{pmatrix} F_{11}^{(q)} & F_{12}^{(q)} & F_{13}^{(q)} & \cdots & F_{1q+1}^{(q)} & 0 & \cdots & \cdots & 0 \\ * & F_{22}^{(q)} & F_{23}^{(q)} & \cdots & & F_{2q+2}^{(q)} & \ddots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \cdots & \ddots & \vdots & \\ \vdots & \ddots & \ddots & & & \vdots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & & \vdots & \cdots & \cdots & F_{r-q,r}^{(q)} \\ \vdots & & & \ddots & \ddots & \vdots & \cdots & \cdots & \vdots \\ \vdots & & & & \ddots & \ddots & \cdots & \ddots & \vdots \\ \vdots & & & & & \ddots & \ddots & \ddots & * \\ * & \cdots & * & \cdots & * & \cdots & \cdots & * & F_{rr}^{(q)} \end{pmatrix}$$

where $F_{1,q+1}^q = (F_{12} F_{23} \cdots F_{qq+1})_{n_1,n_{q+1}}$.

In the next theorem, we show that the full-rankness of the unique solution X of the Sylvester equation

$$HX - XF = R$$

can be checked only by knowing powers of certain block matrices of H and F.

Theorem 3 *Let*

$$HX - XF = R, \tag{10.4}$$

where (H, F) and R are as given above. Assume that (H, R) is controllable, (R, F) is observable, R has full rank n_1, and H and F do not have a common eigenvalue.

Then Eq. (10.4) *has a unique full-rank solution if and only if*

$$\operatorname{rank}\begin{pmatrix} R_1F_{1k+1}^{(k)} & * & \cdots & * \\ 0 & H_{21}R_1F_{1k}^{k-1} & \vdots & \\ \vdots & \ddots & & \vdots \\ \vdots & & H_{k,1}^{(k-1)}R_1F_{12} & H_{k,1}^{(k-1)}R_1(F_{11}+a_1I) \\ & & X & +H_{k,1}^{k}R_1 \\ 0 & & & H_{k+1,1}^{k}R_1 \\ 0 & & & 0 \end{pmatrix}_{n,s} = s \tag{10.5}$$

where $(k+1)$ *is the degree of the minimal or the characteristic polynomial of* F *and* a_i*'s are the coefficient of the characteristic (minimal) polynomial of* F.

Proof Define $\tilde{H} = (R, \quad HR, \ \cdots H^kR)$ and

$$\tilde{F} = \begin{pmatrix} I & a_1I & \cdots & a_kI \\ 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_1I \\ 0 & \cdots & 0 & I \end{pmatrix} \begin{pmatrix} F^k \\ F^{k-1} \\ \vdots \\ I \end{pmatrix}$$

Theorem 3 will then follow from Theorem 2 if we can show that rank $(\tilde{H} \cdot \tilde{F}) = s$.

By direct computations, we have

$$\tilde{H} = \begin{pmatrix} R_1 & 0 & \cdots & H_{11}R_1 & 0 & \cdots & H_{11}^kR_1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & H_{21}R_1 & 0 & \cdots & \vdots & \vdots & & \vdots \\ \vdots & \cdots & & \cdots & \cdots & \cdots & \cdots & \vdots & & \vdots \\ \vdots & \cdots & & \cdots & 0 & \cdots & H_{k+1,1}^kR_1 & 0 & & 0 \\ \vdots & & & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \vdots & & & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & & \cdots & & \cdots & \cdots & 0 & \cdots & 0 \end{pmatrix}_{n\times(k+1)s}$$

and

$$\tilde{F} = \begin{pmatrix} I & a_1 I & a_2 I & \cdots & a_k I \\ 0 & I & a_1 I & \cdots & a_{k-1} I \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & 0 & I \end{pmatrix} F_1 \qquad (10.6)$$

$$_{(k+1)s,(k+1)s}$$

where

$$F_1 = \begin{pmatrix} F_{11}^k & F_{12}^k & \cdots & \cdots & \cdots & F_{1k+1}^k \\ * & F_{22}^k & \cdots & \cdots & \cdots & F_{2k+1}^k \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ * & * & \cdots & \cdots & \cdots & F_{rr}^k \\ F_{11}^{k-1} & F_{12}^{k-1} & \cdots & \cdots & \cdots & 0 \\ * & F_{22}^{k-1} & \cdots & \cdots & \cdots & F_{2k+1}^{k-1} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ * & * & \cdots & \cdots & \cdots & F_{rr}^{k-1} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ F_{11} & F_{12} & 0 & \cdots & \cdots & 0 \\ * & F_{22} & F_{23} & \cdots & 0 & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & 0 \\ * & \cdots & \cdots & * & F_{r-1r-1} & F_{rr-1} \\ * & \cdots & \cdots & \cdots & * & F_{rr} \\ I & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & I & \cdots & \cdots & \cdots & 0 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & I \end{pmatrix} \quad .$$

$$_{s(k+1),s}$$

By direct multiplication of the two matrices $\tilde{F}$ and $\tilde{H}$ and deleting zero rows and zero columns, we can write

$$\operatorname{rank}(\tilde{H}\cdot\tilde{F}) = \operatorname{rank}\left[\begin{pmatrix} R_1 & H_{11}R_1 & \cdots & H_{11}^k R_1 \\ 0 & H_{21}R_1 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & H_{k+1k}^k R_1 \\ 0 & 0 & \cdots & 0 \end{pmatrix}_{n,(k+1)n_1}\right.$$

$$\left.\times\begin{pmatrix} \sum_{i=0}^{k} F_{11}^{k-i}a_i & \sum_{i=0}^{k-1} F_{12}^{k-i}a_i & \cdots & a_1F_{1k}^{k-1} & & F_{1k+1}^{(k)} \\ \vdots & \cdots & \cdots & & \ddots & \vdots \\ \vdots & \cdots & \cdots & \ddots & & \vdots \\ \vdots & F_{12}^2 + a_1F_{12} & F_{13}^2 & & & \vdots \\ F_{11}+a_1I & F_{12} & 0 & \cdots & \cdots & \vdots \\ I & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right]_{n_{1(k+1)},s} \tag{10.7}$$

The 2nd matrix of Eq. (10.7) after column permutations can be written as

$$\begin{pmatrix} F_{1k+1}^{(k)} & F_{1k}^{k-1} + a_1F_{1k}^{k-1} & \cdots & \cdots & \sum_{i=0}^{k} F_{11}^{k-i}a_i \\ 0 & F_{1k}^{(k-1)} & \cdots & \cdots & \sum_{i=0}^{k-1} F_{11}^{(k-i+1)}a_i \\ \vdots & \ddots & \ddots & \cdots & \vdots \\ \vdots & & \ddots & F_{12} & F_{11}+a_1I \\ 0 & \cdots & \cdots & 0 & I \end{pmatrix}_{(k+1)n_1,s} \tag{10.8}$$

Since rank of a matrix is not altered by multiplication with a permutation matrix, from above, we then have

$$\operatorname{rank}(\tilde{H},\tilde{F}) = \operatorname{rank}\begin{pmatrix} R_1F_{1k+1}^{(k)} & * & \cdots & & * \\ 0 & H_{21}R_1F_{1k}^{k-1} & & \vdots & \\ \vdots & \ddots & & & \vdots \\ \vdots & & H_k^{k-1}R_1F_{12} & & H_{k,1}^{(k-1)}R_1(F_{11}+a_1I) \\ & & & & +H_{k,1}^{k}R_1 \\ \vdots & & & \ddots & \\ 0 & & & & H_{k+1,1}^{k}R_1 \\ \vdots & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix} \tag{10.9}$$

The result of Theorem 3 now follows from (10.9).

10.3 Sylvester-Observer Equation

As stated in Introduction that a variation of the Sylvester equation (10.1) known as the Sylvester-observer equation, arises in control theory in the context of constructing Lenenbuger observer. The Sylvester-observer equation has the form:

$$XA - FX = GC,$$

where the matrix C is the output matrix of the linear control system:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y = Cx(t)$$

and given in advance, and the matrix X, F and G need to be computed under certain assumption. In this section we consider the dual of this equation; namely

$$AX - XF = CG \tag{10.10}$$

If G is chosen to be $G = [I \cdots 0]$ then Eq. (10.10) becomes

$$AX - XF = R = (C, 0, \ldots, 0).$$

Since a necessary condition for the solution X to have full-rank is that (A, R) is controllable, we can assume that (A, R) has been transformed into a controller-Hessenberg form $(H, \tilde{R})$ where $H = QAQ^T$ and $\tilde{R} = QR$ have the forms as before. We therefore, concentrate on finding a full-rank solution of

$$H\tilde{X} - \tilde{X}F = \begin{pmatrix} R_1 & 0 & \cdots & 0 \\ 0 & & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & & \cdots & 0 \end{pmatrix}_{n\times s} = \tilde{R}, \tag{10.11}$$

where $\tilde{X} = QX$.

If F is chosen as a block bidiagonal as given below

$$F = \begin{pmatrix} F_{11} & F_{12} & 0 & \cdots & 0 \\ \vdots & F_{22} & F_{23} & \ddots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & F_{k,k} & F_{k,k+1} \\ 0 & \cdots & \cdots & 0 & F_{k+1,k+1} \end{pmatrix}_{s\times s}$$

where $F_{i,i+1}, i = 1, \ldots, k$ has full rank, and $k+1$ is the degree of the minimal (characteristic) polynomial of F such that

$$\begin{aligned} \Big[\text{rank}\Big(H_{k+1,1}^{(k)} R_1\Big) + \ \text{rank}\Big(H_{k,1}^{(k-1)} R_1 F_{12}\Big) + \cdots \\ + \ \text{rank}\Big(H_{21} R_1 F_{1k}^{(k-1)}\Big) + \ \text{rank}\Big(R_1 F_{1k+1}^{(k)}\Big)\Big] = s, \end{aligned}$$

Then it follows from Theorem 3 that

Theorem 4 *The Sylvester Observer equation* (10.11)

$$H\tilde{X} - \tilde{X}F = \tilde{R}$$

has full-rank solution $\tilde{X}$ *if and only if*

$$\begin{aligned} \Big[\text{rank}(H_{k+1,1}^{(k)} R_1) + \ \text{rank}(H_{k,1}^{(k-1)} R_1 F_{12}) + \cdots \\ + \text{rank}(H_{21} R_1 F_{1k}^{k-1}) + \text{rank}(R_1 F_{1k+1}^{(k)})\Big] = s. \end{aligned} \tag{10.12}$$

Derivation of the Result by de Souza and Bhattacharyya. If $k = p - 1$, i.e. $\text{rank}(\tilde{R}) = 1$, then the necessary and sufficient condition of de Souza and Bhattacharyya [11], where $\tilde{R}$ has rank 1 of (10.11) follows immediately as a special case of Theorem 4.

Corollary 1 [11] *Let* $(H, \tilde{R})$ *be a controllable pair and rank*$(\tilde{R}) = 1$. *Then*

$$HX - XF = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \vdots & \vdots \\ 0 & 0 & 0 \end{pmatrix} \tag{10.13}$$

has a unique full-rank solution if and only if rank $(\tilde{H}) = s$.

Proof

$$\begin{aligned}\operatorname{rank}(\tilde{H}\ \tilde{F}) &= \operatorname{rank}\left[\begin{pmatrix} 1 & h_{11} & \cdots & h_{11}^{k} \\ 0 & h_{21} & \cdots & \vdots \\ \vdots & \cdots & \cdots & \vdots \\ 0 & \cdots & 0 & h_{k+1}^{k} \end{pmatrix}_{n,s} \begin{pmatrix} 1 & * & \cdots & \cdots \\ 0 & 1 & * & \cdots \\ \vdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & 1 \end{pmatrix}_{s\times s}\right] \\ &= \operatorname{rank}(\tilde{H}) \end{aligned} \tag{10.14}$$

Since $(H, \tilde{R})$ is controllable, there exists a k such that $\operatorname{rank}(\tilde{H}) = s$. Thus by theorem 3, Eq. (10.13) has a full-rank solution if and only if $(H, \tilde{R})$ is controllable. □

Corollary 2 *If n_1 and k are such that $n_1(k+1) = s$ and rank$(\tilde{H}) = s$, then Eq. (10.11) has a unique full-rank solution if and only if $\tilde{F}$ is nonsingular.*

Proof Proof follows from Eq. (10.9).

Now observe that rank $H_{k+1,1}^{k} R_1 =$ rank$(H_{k+1,k} \cdot H_{k,k-1} \cdots H_{32} H_{21} \cdot R_1) =$ min(n_{k+1}, n_1).

If $n_{k+1} \geq n_1$ for $k = 1, \ldots, p-1$, then we have the following result. □

Corollary 3 *If $n_{k+1} \geq n_1 = s, k = 1, 2, \ldots, p-1$, then there always exists a unique full-rank solution to the following equation*

$$HX - XF = \begin{pmatrix} R_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \tilde{R} \tag{10.15}$$

Proof From Theorem 3 we know that X is a unique full-rank solution if and only if rank$(\tilde{H}, \tilde{F}) = s$.

In this case,

$$\operatorname{rank}(\tilde{H},\tilde{F}) = \operatorname{rank}\left[(\tilde{H}_{n,(k+1)n_1}) \times \begin{pmatrix} F^k + a_1F^{k-1} + \cdots + a_kI \\ F^{k-1} + a_1F^{k-2} + \cdots + a_{k-1}I \\ \vdots \\ I \end{pmatrix}_{n_1(k+1),n_1}\right]$$

$$= \operatorname{rank}\begin{pmatrix} R_1F^k + (a_1I + H_{11}^{(1)})R_1F^{k-1} + \cdots + \left(\sum_{i=0}^{k} a_{k-i}H_{11}^{(i)}\right)R_1 \\ H_{21}^{(1)}R_1F^{k-1} + (a_1H_{21}^{(1)} + H_{21}^{(2)})R_1F^{k-2} + \\ \cdots + \left(\sum_{i=0}^{k-1} a_{k-1-i}H_{21}\right)R_1 \\ \vdots \\ \vdots \\ H_{k,1}^{(k-1)}R_1F + \left(a_1H_{k,1}^{(k-1)} + H_{k,1}^{(k)}\right)R_1 \\ H_{k+1,1}^{(k)}R_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{n,n_1} \tag{10.16}$$

where $n_1 + n_2 + \cdots + n_p = n$ and $a_0 = 1,\ 1 \le k+1$ is the degree of the characteristic or minimal polynomial of F, and rank $(\tilde{H}) \ge s = n_1$, rank $(H_{k+1,1}^{(k)} \cdot R_1) =$ rank $(H_{k+1,k} \cdot H_{k,k-1} \cdots H_{32} \cdot H_{21} \cdot R_1) = \min(n_{k+1}, n_1)$.

From (10.16) it is clear that if $n_{k+1} \ge n_1$, then there always exists a unique full-rank solution. □

10.3.1 Derivation of the Van Dooren Result on Sylvester-Observer Equation

Van Dooren [4] proposed an algorithm for constructing a full-rank solution of a Sylvester-observer equation using observer-Hessenberg form.

Using our result of Corollary 3 we will now explain why his algorithm always yielded a full-rank solution.

Van Dooren considered the Sylvester-observer equation in the form:

$$XA - FX = GC \tag{10.17}$$

Using the observer-Hessenberg form of (A, C), the Eq. (10.17) reduces to

$$\tilde{X}H - F\tilde{X} = G(0, \ldots, R_1) = (0, \ldots, R_1), \tag{10.18}$$

where R_1 is an upper triangular matrix with full-rank n_k, $G = (I)_{n_k \times n_k}$, H is in block upper Hessenberg form and F is a block lower triangular form as described below:

$$H = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1k} \\ H_{21} & H_{22} & \cdots & H_{2k} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & H_{kk-1} & H_{kk} \end{pmatrix}_{n \times n} \quad \text{and}$$

$$F = \begin{pmatrix} F_{11} & 0 & \cdots & \cdots & 0 \\ F_{21} & F_{22} & 0 & \cdots & 0 \\ * & \cdots & F_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & * & F_{rr} \end{pmatrix}_{n_k \times n_k}$$

In order to have a full-rank solution X, Van Dooren's assumption was rank $(R_1) =$ rank$(H_{k,k-1}) = n_k$ and $n_i \leq n_{i-1}$ for $i = 2, \ldots, k$. To put Eq. (10.18) in our setting we first take the transpose of (10.18)

$$H^T\tilde{X}^T - \tilde{X}^T \begin{pmatrix} F_{11}^T & F_{21}^T & * & * & * \\ 0 & F_{22}^T & * & * & * \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & * \\ 0 & 0 & & & F_{rr}^T \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ R_1^T \end{pmatrix} \tag{10.19}$$

and then multiply both sides by a suitable permutation matrix to obtain the following:

$$\begin{pmatrix} H_{kk}^T & H_{k-1k}^T & \cdots & H_{1k}^T \\ H_{kk-1}^T & H_{k-1k-1}^T & \cdots & * \\ \vdots & \ddots & \ddots & \vdots \\ 0 & & H_{21}^T & H_{11}^T \end{pmatrix} \tilde{\tilde{X}}^T - \tilde{\tilde{X}}^T \begin{pmatrix} F_{11}^T & F_{21}^T & * & \cdots & * \\ 0 & F_{22}^T & F_{23}^T & * & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & F_{rr}^T \end{pmatrix}_{n_k \times n_k}$$

$$= \begin{pmatrix} \tilde{R}_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{n \times n_k} \tag{10.20}$$

According to the result of Corollary 3, this Eq. (10.20) has a full-rank solution if and only if $n_k \leq n_1$. Thus in particular if $n_k = n_1$, Eq. (10.17) always has a full-rank solution.

10.4 Algorithm and Numerical Examples

Based on our above discussions, we now give an algorithm for choosing "k" and the diagonal blocks of F so that the Sylvester observer equation

$$AX - XF = (C\ 0, \ldots, 0) \tag{10.21}$$

has a full-rank solution.

Algorithm 1 Constructing a full-rank solution X and F

Input $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{(n,n_1)}$, $s =$ size of F

Output A full-rank solution $X \in \mathbb{R}^{n\times s}$ and an upper block bi-diagonal matrix $F_{s,s}$.

Assumptions (A, C) is controllable, rank $(C) = n_1$.

Step 1 Transform (A, C) to the controller-Hessenberg form $(H, \hat{R})$, where (i) the number of subdiagonal blocks of $H = \hat{p}$
(ii) Size of diagonal block $H_i = n_i$ for $i = 1, \ldots, (\hat{p} + 1)$,
(iii) $\hat{R} = \begin{bmatrix} R_1 \\ 0 \\ \cdots \\ 0 \end{bmatrix}_{(n,n_1)}$.

Step 2 Form $\tilde{R} = \begin{bmatrix} \hat{R} \ldots 0 \end{bmatrix}_{(n,s)}$

Step 3 Compute k:
set $\hat{s} = \text{rank}(R_1) = n_1$
if $\hat{s} = s$ then $k = 0$, stop
for $j = 1$ to $\hat{p}$
$s_j = \min(n_{j+1}, n_1)$
$\hat{s} = \hat{s} + s_j$
If $\hat{s} \geq s$, then $k = j$, stop
end

Step 4 Construct the upper bidiagonal matrix F as described in Sect. 10.3 having $(k + 1)$ diagonal blocks such that the degree of the minimal (characteristic) polynomial of F is $k + 1$.

Step 5 Solve $HY - YF = \tilde{R}$ using a standard Sylvester equation solver such as MATLAB function LYAP, based on the well-known Hessenberg-Schur algorithm of Golub et al. [13].

Step 6 Construct $X = QY$, where Q is the orthogonal matrix used in Step 1 to transform (A, C) to $(H, \hat{R})$

10.4.1 An Illustrative Numerical Example

We take illustrate our algorithm by taking a random matrix of order 20×20 and C a random matrix of order $20 \times 3, s = 9$

Step 1

$$H = \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{19} \\ H_{21} & H_{22} & \cdots & \cdots \\ 0 & \ddots & \ddots & \\ & & H_{87} & H_{88} \end{bmatrix}$$

where block size of $(H_{i+1,i}) = (3,3), i = 1,\ 6,$ block size $H(8,7) = (2,3)$ and the number of subdiagonal blocks of $H = \hat{p} = 7$.

Step 2 $\tilde{R} = \begin{pmatrix} R_1 & 0 \\ 0 & 0 \end{pmatrix}_{20\times 9}$
where

$$R_1 = \begin{pmatrix} 1.3064 & -0.2114 & -0.0485 \\ 0 & -2.9505 & -2.4689 \\ 0 & 0 & -2.0061 \end{pmatrix}$$

Step 3 It is easy to see when $k = 2, s_j = 9 = s$.

Step 4 Choose F with degree of minimal polynomial $= 3$

$$F = \begin{bmatrix} 2I & I & 0 \\ 0 & 3I & I \\ 0 & 0 & 4I \end{bmatrix}_{9\times 9}$$

where $\lambda = 2, 3, 4$ are not the eigenvalues of H.

Step 5 Solve for $X : HY - YF = \tilde{R}$. Verify rank$(Y) = 9$.

10.4.2 Results on Numerical Experiment

We now present results on our numerical experiment on a Benchmark Example taken from Higham [14]. Here A is a pentadiagonal Toeplitz matrix, C is randomly chosen and the eigenvalues of F are chosen such that they are disjoint from those of A. Define $Residual = ||HY - YF - \tilde{R}||_2$.

Example 1 Pentadiagonal Toeplitz matrix (n = 100)

Rank($\tilde{R}$)	$s = n - n_1$	Deg of minimal poly of F	Rank(Y)	Residual
13	87	7	87	2.466913e-014
14	86	7	86	1.740644e-014
15	85	6	85	3.041122e-014

Example 2 Pentadiagonal Toeplitz matrix (n = 300)

Rank($\tilde{R}$)	$s = n - n_1$	Deg of minimal poly of F	Rank(Y)	Residual
38	262	7	262	2.959402e-014
39	261	7	261	2.967265e-014
40	260	7	260	7.285381e-014

Example 3 Pentadiagonal Toeplitz matrix (n = 500)

Rank($\tilde{R}$)	$s = n - n_1$	Deg of minimal poly of F	Rank(Y)	Residual
63	437	7	437	7.452755e-014
64	436	7	436	4.704602e-014
65	435	8	435	7.495685e-014

10.5 Conclusion

A necessary condition for the existence of a full-rank solution to a Sylvester equation, and a necessary and sufficient condition when the right hand side matrix is a rank one matrix, were known for long. However, the characterization of full-rank solution in case the right hand matrix has a arbitrary rank, was an open problem for a long time. In 1997 it was settled by Datta, Hong and Lee [15]. Unfortunately, that condition turned out to be of mostly theoretical interest and is not readily applicable in a practical computational setting.

On the other hand, a variation of the Sylvester equation, called the Sylvester-observer equation, $AX - XF = CG$, arises in practical applications in the context of designing Luenberger observer in control theory, where it is crucial that the matrix X has full-rank.

A criterion for choosing the matrix F, based on the controller-Hessenberg reduction of (A, C), which will guarantee that the solution matrix X has full-rank, is presented in this paper and an associated algorithm is described.

It is hoped that these results will be of some practical value to the control theorists and engineers.

References

1. Bhattacharyya SP, de Souza E (1982) Pole assignment via Sylvester's equation. Syst Control Lett 1:261–283
2. Datta BN (2003) Numerical methods for linear control systems. Elsevier Academic Press
3. Luenberger D (1964) Observing the state of a linear system. IEEE Trans Mil Electron 8: 74–80
4. Van Dooren P (1984) Reduced order observers: a new algorithm and proof. Syst Cont Lett 4:243–251

5. Carvalho J, Datta BN (2001) A new block algorithm for the Sylvester-observer equation arising in state-estimation. In: Proceedings of the IEEE International Conference on Decision and Control, Orlando, pp 3398–3403
6. Carvalho J, Datta K, Hong YP (2003) A new block algorithm for full-rank solution of the Sylvester observer equation. IEEE Trans Autom Control, 48(12):2223–2228
7. Datta BN, Sarkissian D (2000) Block algorithms for state estimation and functional observers. In: Proceedings of the IEEE International Symposium on Computer-Aided Control System Design, Anchorage, pp 19–23
8. Datta BN, Saad Y (1991) Arnoldi methods for large Sylvester-like observer matrix equations, and an associated algorithm for partial spectrum assignment. Lin Alg Appl 154–156:225–244
9. Calvetti D, Lewis B, Reichel L (2001) On the solution of large Sylvester-observer equation. Num Lin Alg Appl 8:435–452
10. Bischof C, Datta BN, Purkayastha A (1996) A parallel algorithm for the Sylvester observer equation. SIAM J Sci Comp 17:686–698
11. de Souza E, Bhattacharyya SP (1981) Controllability, observability and the solution of $AX - XB = C$. Lin Alg Appl 39:167–188
12. Datta K, Hong YP, Lee R (1997) Applications of linear transformations of matrix equations. Lin Alg Appl 267:221–240
13. Golub G, Nash S, Van Loan C (1979) A Hessenberg–Schur method for the problem $AX + XB = C$. IEEE Trans Autom Control AC-24:909–913
14. Higham NJ (1995) The test matrix toolbox for Matlab. Numerical Analysis Report No. 276. Manchester Centre for Computational Mathematics, University of Manchester, UK
15. Datta K (1988) The matrix equation $AX - XB = R$ and its applications. Lin Alg. Appl 109:91–105

Chapter 11
On Symmetric and Skew-Symmetric Solutions to a Procrustes Problem

Yuan-Bei Deng and Daniel Boley

Abstract Using the projection theorem in a Hilbert space, the quotient singular value decomposition (QSVD) and the canonical correlation decomposition (CCD) in matrix theory for efficient tools, we obtained the explicit analytical expressions of the optimal approximation solutions for the symmetric and skew-symmetric least-squares problems of the linear matrix equation $AXB = C$. This can lead to new algorithms to solve such problems.

11.1 Introduction

Certain least-squares problems of linear matrix equations are called Procrustes problems [1, 13]. The unconstrained and constrained least squares problems have been of interest for many applications, including particle physics and geology, inverse Sturm–Liouville problem [11], inverse problems of vibration theory [6], control theory, digital image and signal processing, photogrammetry, finite elements, and multidimensional approximation [8]. Penrose [2, 21] first considered the linear matrix equation

$$AX = B \tag{11.1}$$

Y.-B. Deng (✉)
College of Mathematics and Econometrics, Hunan University Changsha, 410082 Hunan, People's Republic of China
e-mail: ybdeng@hnu.cn

D. Boley
Department of Computer Science and Engineering, University of Minnesota Minneapolis, Minneapolis, MN 55455, USA
e-mail: boley@cs.umn.edu

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_11,

and obtained its general solution and least-squares solution by making use of the Moore–Penrose generalized inverse, then Sun [22] obtained the least-squares solution and the related optimal approximation solution of Eq. 11.1 when X is a real matrix. The least-squares problems of Eq. 11.1 were discussed in 1988 by Higham [13] and Sun [23] when the solution matrix X is constrained to be a real symmetric matrix, and Sun also discussed the related symmetric optimal approximation problem of Eq. 11.1 in [23]. This work was further extended by Andersson and Elfving [1]. For more information about the linear matrix equations, see e.g. [12, 15, 16, 18].

In this paper, the least-squares problems

$$\min_{X\in Q} \|AXB - C\|_F, \tag{11.2}$$

with Q symmetric or skew-symmetric cone or possibly the real matrix space, and the related optimal approximation problems are considered. As it can be seen in the following discussions, we can make sure that the least-squares problem (11.2) always has a solution over the symmetric or skew-symmetric cone. Obviously if the equation $AXB = C$ is consistent, then the least-squares problem (11.2) and the equation $AXB = C$ have the same solution set. The general expressions for the symmetric solution of the equation $AXB = C$ were obtained by using the generalized singular value decomposition of matrices (GSVD) by Chu [4] in 1989 and Hua [5] in 1990 respectively. Fausett and Fulton [8] and Zha [25] considered the least-squares problems (11.2) in the real matrix space, while the symmetric and the skew-symmetric least-squares solutions of (11.2) have been derived by Deng et al. [7]. Liao et al. [17] used the projection method first for finding the best approximate solutions for the matrix equation $AXB + CYD = E$. But it remains unsolved about the optimal approximation solutions for the symmetric and skew-symmetric Procrustes problems of this equation. Therefore in the following, we will consider the optimal approximation solutions of the constrained least squares problems related to (11.2), we obtain the explicit analytical expressions of the optimal approximation solutions for the symmetric and skew-symmetric least-squares problems of the form (11.2). Of course, when the least-squares problem (11.2) has a unique solution, then there is no need to discuss the related optimal approximation problems.

In this paper we always suppose that $R^{m\times n}$ is the set of all $m \times n$ real matrices, $SR^{n\times n}$, $AR^{n\times n}$ and $OR^{n\times n}$ are the sets of all symmetric, skew-symmetric and orthogonal matrices in $R^{n\times n}$, respectively, $A * B$ represents the Hadamard product of A and B (cf. [14]), and $\|Y\|_F$ denotes the Frobenius norm of a real matrix Y, defined as

$$\|Y\|_F^2 = \langle Y, Y\rangle = \sum_{i,j} y_{ij}^2,$$

here the inner product is given by $\langle A, B\rangle = \mathrm{trace}(A^T B)$, and $R^{m\times n}$ becomes a Hilbert space with the inner product.

The specific problems treated in this paper are stated as follows.

Problem I Given matrices $A \in R^{m\times n}$, $B \in R^{n\times p}$, $C \in R^{m\times p}$ and $X_f \in R^{n\times n}$, let

$$S_E = \{X | X \in SR^{n\times n}, \|AXB - C\|_F = \min\}. \tag{11.3}$$

Then find $X_e \in S_E$, such that

$$\|X_e - X_f\|_F = \min_{X\in S_E} \|X - X_f\|_F. \tag{11.4}$$

Problem II Given matrices $A \in R^{m\times n}$, $B \in R^{n\times p}$, $C \in R^{m\times p}$ and $X_f \in R^{n\times n}$, let

$$S_A = \{X | X \in AR^{n\times n}, \|AXB - C\|_F = \min\}. \tag{11.5}$$

Then find $X_a \in S_A$, such that

$$\|X_a - X_f\|_F = \min_{X\in S_A} \|X - X_f\|_F. \tag{11.6}$$

We first introduce some results about the quotient singular value decomposition (QSVD) and the canonical correlation decomposition (CCD) of a pair of matrices, and the projection theorem in a Hilbert space, which are essential tools for the problems to be discussed, and then give a mechanical model as the practical background of the above mentioned problems.

QSVD Theorem [3] *Let $A \in R^{m\times n}$, $B \in R^{n\times p}$. Then there exist orthogonal matrices $U \in OR^{m\times m}, V \in OR^{p\times p}$ and a nonsingular matrix $Y \in R^{n\times n}$ such that*

$$A = U\Sigma_1 Y^{-1}, \quad B^T = V\Sigma_2 Y^{-1}, \tag{11.7}$$

where

$$\Sigma_1 = \begin{pmatrix} I_{r'} & 0 & 0 & 0 \\ 0 & S & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} r' \\ s' \\ m-r'-s' \end{matrix} \tag{11.8}$$

$$\begin{matrix} r' & s' & t' & n-k', \end{matrix}$$

$$\Sigma_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & I_{s'} & 0 & 0 \\ 0 & 0 & I_{t'} & 0 \end{pmatrix} \begin{matrix} r' \\ s' \\ t' \end{matrix} \tag{11.9}$$

$$\begin{matrix} r' & s' & t' & n-k', \end{matrix}$$

with

$$\left.\begin{array}{ll} k' = \operatorname{rank}(A^T, B), & s' = \operatorname{rank}(A) + \operatorname{rank}(B) - k', \\ r' = k' - \operatorname{rank}(B), & S = \operatorname{diag}(\sigma_1, \ldots, \sigma_{s'}), \\ t' = k' - \operatorname{rank}(A), & \sigma_i > 0 \quad (i = 1, \ldots, s'). \end{array}\right\} \tag{11.10}$$

When A and B^T are of full column rank, i.e. $r(B) = r(A) = n$, then $r' = 0, s' = n$, $k' = n$, and

$$\Sigma_1 = \begin{pmatrix} S \\ 0 \\ n \end{pmatrix}_{m-n}^{n}, \quad \Sigma_2 = \begin{pmatrix} 0 \\ I_{s'} \\ n \end{pmatrix}_{n}^{p-n} \tag{11.11}$$

The QSVD as given above differs from the GSVD of (A, B^T) only in the way the second block column of (11.8) and (11.9) are scaled. The diagonal matrices S in (11.8) and identity matrix $I_{s'}$ in (11.9) are scaled in the GSVD to be diagonal matrices $\widetilde{S}$ and $\widetilde{C}$ such that $\widetilde{S}^2 + \widetilde{C}^2 = I$, and the Y is adjusted accordingly. This particular scaling simplifies some of the later formulas in this paper. See details in [9, 20].

The canonical correlations decomposition of the matrix pair (A^T, B) is given by the following theorem.

CCD Theorem [10] *Let $A \in R^{m\times n}, B \in R^{n\times p}$, and assume that $g = \mathrm{rank}(A)$, $h = \mathrm{rank}(B)$, $g \geq h$. Then there exists an orthogonal matrix $Q \in OR^{n\times n}$ and nonsingular matrices $X_A \in R^{m\times m}$, $X_B \in R^{p\times p}$ such that*

$$A^T = Q[\Sigma_A, \mathbf{0}]X_A^{-1}, \quad B = Q[\Sigma_B, \mathbf{0}]X_B^{-1}, \tag{11.12}$$

where $\Sigma_A \in R^{n\times g}$ and $\Sigma_B \in R^{n\times h}$ are of the forms:

$$\Sigma_A = \begin{pmatrix} I_i & 0 & 0 \\ 0 & \Lambda_j & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \Delta_j & 0 \\ 0 & 0 & I_t \\ i & j & t \end{pmatrix} \begin{matrix} i \\ j \\ t \\ n-g-j-t \\ j \\ t \end{matrix}, \quad \Sigma_B = \begin{pmatrix} I_h \\ 0 \\ h \end{pmatrix}_{n-h}^{h}, \tag{11.13}$$

with $i + j + t = g$, Σ_A having partitioned row dimensions $i, j, t, n - g - j - t, j, t$, and Σ_B having partitioned row dimensions $h, n - h$, and

$$\begin{aligned} \Lambda_j &= \mathrm{diag}(\lambda_{i+1}, \ldots, \lambda_{i+j}), \quad 1 > \lambda_{i+1} \geq \cdots \geq \lambda_{i+j} > 0, \\ \Delta_j &= \mathrm{diag}(\delta_{i+1}, \ldots, \delta_{i+j}), \quad 0 < \delta_{i+1} \leq \cdots \leq \delta_{i+j} < 1, \\ \lambda_{i+1}^2 + \delta_{i+1}^2 &= 1, \ldots, \lambda_{i+j}^2 + \delta_{i+j}^2 = 1, \quad i.e., \Lambda_j^2 + \Delta_j^2 = I. \end{aligned}$$

Here,

$$\begin{aligned} i &= \mathrm{rank}(A) + \mathrm{rank}(B) - \mathrm{rank}[A^T, B], \\ j &= \mathrm{rank}[A^T, B] + \mathrm{rank}(AB) - \mathrm{rank}(A) - \mathrm{rank}(B), \\ t &= \mathrm{rank}(A) - \mathrm{rank}(AB). \end{aligned}$$

Notice that the QSVD and the CCD are different. In QSVD, the matrix pair A and B^T have the same column dimensions, the right nonsingular matrices are the same Y in (11.7), the left orthogonal matrices are $U \in OR^{m\times m}$ and $V \in OR^{p\times p}$; while in CCD, the matrix pair A^T and B have same row dimensions, the right nonsingular matrices are X_A^{-1} and X_B^{-1} in (11.12), the left orthogonal matrices are the same Q.

The projection theorem in a Hilbert space is also important for the problems to be solved.

Lemma 1 [19, Theorem 5.14.4, p. 286]. *Let $\mathcal{H}$ be a Hilbert space, and let $\mathcal{M}$ be a closed linear subspace of $\mathcal{H}$. Let $x_0 \in \mathcal{H}$ and define*

$$\delta = \inf\{\|x_0 - y\| : y \in \mathcal{M}\}.$$

Then there is one (and only one) $y_0 \in \mathcal{M}$ such that

$$\|x_0 - y_0\| = \delta.$$

Moreover, $x_0 - y_0 \perp \mathcal{M}$, that is $(x_0 - y_0, y) = 0$ for all $y \in \mathcal{M}$. Furthermore, y_0 is the only point in $\mathcal{M}$ such that $x_0 - y_0 \perp \mathcal{M}$.

11.2 The Solution of Problem I

In this section, the explicit expression for the solution of Problem I is derived. Our approach is based on the projection theorem in a Hilbert space, and also based on QSVD and CCD of matrices. Specifically, it can be essentially divided into three parts. First, we characterize the symmetric solutions X_0 of the least-squares problem (11.2) by using the QSVD; then by utilizing the general form of the solution X_0 and the projection theorem, we transform the least-squares problem into a equation problem; and finally, we find the optimal approximate solution of the matrix equation by making use of CCD.

Without loss of generality, we suppose that $\text{rank}(A) \geq \text{rank}(B)$. Instead of considering the solution of Problem I directly, we will find a matrix C_0, and then transform Problem I into the following equivalent problem.

Problem $\mathbf{I_0}$ Given matrices $A \in R^{m\times n}$, $B \in R^{n\times p}$, $C_0 \in R^{m\times p}$ and $X_f \in R^{n\times n}$, let

$$S_{E_0} = \{X | X \in SR^{n\times n}, AXB = C_0\}. \tag{11.14}$$

Then find $X_e \in S_{E_0}$, such that

$$\|X_e - X_f\|_F = \min_{X\in S_{E_0}} \|X - X_f\|_F. \tag{11.15}$$

We use the projection theorem on $R^{m\times p}$ to prove the two problems are equivalent in the following.

Theorem 1 *Given $A \in R^{m\times n}, B \in R^{n\times p}, C \in R^{m\times p}$, let X_0 be any solution of (11.2) with Q the symmetric cone, and define*

$$C_0 = AX_0B, \tag{11.16}$$

then the matrix equation

$$AXB = C_0, \tag{11.17}$$

is consistent in $SR^{n\times n}$, and the symmetric solution set S_{E_0} of the matrix equation (11.17) is the same as the symmetric solution set S_E of the least-squares problem (11.2).

Proof Let

$$\mathcal{L} = \{Z | Z = AXB,\ X \in SR^{n\times n}\}. \tag{11.18}$$

Then $\mathcal{L}$ is obviously a linear subspace of $R^{m\times p}$. Because X_0 is a symmetric solution of the least-squares problem (11.2), from (11.16) we see that $C_0 \in \mathcal{L}$ and

$$\begin{aligned} \|C_0 - C\|_F &= \|AX_0B - C\|_F \\ &= \min_{X\in SR^{n\times n}} \|AXB - C\|_F \\ &= \min_{Z\in\mathcal{L}} \|Z - C\|_F. \end{aligned}$$

Then by Lemma 1 we have

$$(C_0 - C) \perp \mathcal{L} \quad \text{or} \quad (C_0 - C) \in \mathcal{L}^{\perp}.$$

Next for all $X \in SR^{n\times n}$, $AXB - C_0 \in \mathcal{L}$. It then follows that

$$\begin{aligned} \|AXB - C\|_F^2 &= \|(AXB - C_0) + (C_0 - C)\|_F^2 \\ &= \|AXB - C_0\|_F^2 + \|C_0 - C\|_F^2. \end{aligned}$$

Hence, $S_E = S_{E_0}$, and the conclusion of the theorem is true. □

Now suppose $A \in R^{m\times n}, B \in R^{n\times p}$ and the matrix pair (A, B^T) has the QSVD (11.7), and partition U^TCV into the following block matrix:

$$U^TCV = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix} \begin{matrix} r' \\ s' \\ m-r'-s' \end{matrix}, \tag{11.19}$$

$$\begin{matrix} l' & s' & r' \end{matrix}$$

where $l' = p + r' - k'$, with r', k' defined as in (11.10). Then the expression of C_0 will be shown in the following theorem.

Theorem 2 *Let A, B, C be given in Problem* I, *the matrix pair (A, B^T) have the QSVD (11.7), and U^TCV be partitioned by (11.19), then for any symmetric*

solution X_0 of the least-squares problem (11.2), *the matrix C_0 determined by* (11.16) *can be expressed by the following form:*

$$C_0 = UC^*V^T, \quad C^* = \begin{pmatrix} 0 & C_{12} & C_{13} \\ 0 & S\hat{X}_{22} & C_{23} \\ 0 & 0 & 0 \end{pmatrix} \begin{matrix} r' \\ s' \\ m-r'-s' \end{matrix}, \tag{11.20}$$
$$\begin{matrix} l' & s' & t' \end{matrix}$$

where

$$\hat{X}_{22} = \phi * (C_{22}^T S + SC_{22}),$$
$$\phi = (\varphi_{kl}) \in SR^{s' \times s'}, \quad \varphi_{kl} = \frac{1}{\sigma_k^2 + \sigma_l^2}, \ 1 \le k, l \le s'. \tag{11.21}$$

Proof From Theorem 2.1 in [7] we know that the symmetric solution of the least-squares problem (11.2) can be obtained by using of the QSVD of matrix pair (A, B^T) and the general form of the solution is

$$X_0 = Y \begin{bmatrix} X'_{11} & C_{12} & C_{13} & X'_{14} \\ C_{12}^T & \hat{X}_{22} & S^{-1}C_{23} & X'_{24} \\ C_{13}^T & (S^{-1}C_{23})^T & X'_{33} & X'_{34} \\ X_{14}'^T & X_{24}'^T & X_{34}'^T & X'_{44} \end{bmatrix} Y^T, \tag{11.22}$$

where $\hat{X}_{22}$ is given by (11.21) and $X'_{11} \in SR^{r' \times r'}$, $X'_{33} \in SR^{t' \times t'}$, $X'_{44} \in SR^{(n-k') \times (n-k')}$, $X'_{14} \in R^{r' \times (n-k')}$, $X'_{24} \in R^{s' \times (n-k')}$, $X'_{34} \in R^{t' \times (n-k')}$ are arbitrary matrix blocks.

Substituting (11.7), (11.22) into (11.16), we can easily obtain (11.20). □

Evidently, (11.20) shows that the matrix C_0 in Theorem 2.2 is dependent only on the matrices A, B and C, but is independent of the symmetric solution X of the least-squares problem (11.2). Since C_0 is known, from Theorem 2.1 we know that Problem I is equivalent to Problem I_0. In Problem I_0, since $S_{E_0} \neq \emptyset$, we can derive the general expression of the elements of S_{E_0} in the following theorem. In this theorem, $A \in R^{m \times n}$, $B \in R^{n \times p}$ are given, while C_0 is expressed by (11.20), and assume that $g = \mathrm{rank}(A)$, $h = \mathrm{rank}(B)$, the matrix pair (A^T, B) has CCD (11.12).

We discuss Problem I in two cases.

Case I $g = h$. Suppose $X \in S_{E_0}$, then partition the symmetric matrix $X^* \equiv Q^T XQ$ into the block matrix,

$$X^* = (X_{kl})_{6 \times 6}, \tag{11.23}$$

with the row dimensions (and the related column dimensions) of blocks are $i, j, t, n-g-j-t, j, t$ respectively, and $X_{kl} = X_{lk}^T, k, l = 1, 2, \ldots, 6$. Let $E = X_A^T C_0 X_B$ and also partition E into block matrix,

$$E = (E_{kl})_{4 \times 4}, \tag{11.24}$$

with the row dimensions of blocks are $i, j, t, m-g$ and the column dimensions of blocks are $i, j, t, p-g$ respectively.

Theorem 3 *In Problem* I_0, *the general form of the elements of* S_{E_0} *can be expressed as*

$$X = QX^*Q^T = Q\begin{pmatrix} E_{11} & E_{12} & E_{13} & X_{14} & X_{51}^{(0)T} & E_{31}^T \\ E_{12}^T & X_{22} & X_{23} & X_{24} & X_{52}^{(0)T} & E_{32}^T \\ E_{13}^T & X_{23}^T & X_{33} & X_{34} & X_{53}^{(0)T} & E_{33}^T \\ X_{14}^T & X_{24}^T & X_{34}^T & X_{44} & X_{45} & X_{46} \\ X_{51}^{(0)} & X_{52}^{(0)} & X_{53}^{(0)} & X_{45}^T & X_{55} & X_{56} \\ E_{31} & E_{32} & E_{33} & X_{46}^T & X_{56}^T & X_{66} \end{pmatrix} Q^T \tag{11.25}$$

where $X_{kk} = X_{kk}^T, 2 \le k \le 6$, X_{14}, X_{23}, X_{24}, X_{34}, X_{45}, X_{46} and X_{56} *are arbitrary matrices with the associated sizes, and* $X_{51}^{(0)} = \Delta_j^{-1}(E_{21} - \Lambda_j E_{12}^T), X_{52}^{(0)} = \Delta_j^{-1}(E_{22} - \Lambda_j X_{22}), X_{53}^{(0)} = \Delta_j^{-1}(E_{23} - \Lambda_j X_{23})$.

Proof Suppose $X \in S_{E_0}$, then

$$AXB = C_0. \tag{11.26}$$

Substitute (11.12) into (11.26) to obtain

$$\begin{pmatrix} \Sigma_A^T \\ 0 \end{pmatrix} X^*(\Sigma_B, 0) = E, \tag{11.27}$$

then substitute (11.13), (11.23) and (11.24) into (11.27) to obtain

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & 0 \\ \Lambda_j X_{21} + \Delta_j X_{51} & \Lambda_j X_{22} + \Delta_j X_{52} & \Lambda_j X_{23} + \Delta_j X_{53} & 0 \\ X_{61} & X_{62} & X_{63} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = (E_{kl})_{4\times 4}. \tag{11.28}$$

Because the matrix equation (11.26) is consistent, therefore we can obtain some X_{ij} from (11.28) directly. Comparing with both sides of (11.28), the expression (11.25) can be derived according to the symmetric property of X^*. □

The following lemmas are needed for the main results.

Lemma 2 [24]. *For given* J_1, J_2, J_3 and $J_4 \in R^{m\times n}$,

$$S_a = \mathrm{diag}(a_1, \ldots, a_m) > 0, \quad S_b = \mathrm{diag}(b_1, \ldots, b_m) > 0,$$
$$S_c = \mathrm{diag}(c_1, \ldots, c_m) > 0, \quad S_d = \mathrm{diag}(d_1, \ldots, d_m) > 0,$$

there exists a unique $W \in R^{m\times n}$, *such that*

$$\|S_a W - J_1\|_F^2 + \|S_b W - J_2\|_F^2 + \|S_c W - J_3\|_F^2 + \|S_d W - J_4\|_F^2 = \min$$

and W can be expressed as (*where* $*$ *denotes the Hadamard product*)

$$W = P * (S_a J_1 + S_b J_2 + S_c J_3 + S_d J_4),$$

where

$$P = (p_{kl}) \in R^{m\times n}, \quad p_{kl} = 1/(a_k^2 + b_k^2 + c_k^2 + d_k^2),\ 1 \le k \le m,\ 1 \le l \le n.$$

Lemma 3 *For given* J_1, J_2 and $J_3 \in R^{s\times s}$, $S_a = \mathrm{diag}(a_1, \ldots, a_s) > 0$, $S_b = \mathrm{diag}(b_1, \ldots, b_s) > 0$, $S_c = \mathrm{diag}(c_1, \ldots, c_s) > 0$, *there exists a unique symmetric matrix* $W \in SR^{s\times s}$, *such that*

$$\mu \equiv \|S_a W - J_1\|_F^2 + \|S_b W - J_2\|_F^2 + \|S_c W - J_3\|_F^2 = \min,$$

and W *can be expressed as*

$$W = \psi * (S_a J_1 + J_1^T S_a + S_b J_2 + J_2^T S_b + S_c J_3 + J_3^T S_c), \tag{11.29}$$

where

$$\psi = (\phi_{kl}) \in R^{s\times s}, \quad \phi_{kl} = 1/(a_k^2 + a_l^2 + b_k^2 + b_l^2 + c_k^2 + c_l^2),\ 1 \le k,\ l \le s.$$

Proof For $W \in SR^{s\times s}$, the property $w_{kl} = w_{lk} (1 \le k, l \le s)$ holds, and

$$\begin{aligned}\mu = &\sum_{k=1}^{s} [(a_k w_{kk} - J_{1kk})^2 + (b_k w_{kk} - J_{2kk})^2 + (c_k w_{kk} - J_{3kk})^2] \\ &+ \sum_{1\le k<l\le s} [(a_k w_{kl} - J_{1kl})^2 + (a_l w_{kl} - J_{1lk})^2 + (b_k w_{kl} - J_{2kl})^2 \\ &+ (b_l w_{kl} - J_{2lk})^2 + (c_k w_{kl} - J_{3kl})^2 + (c_l w_{kl} - J_{3lk})^2].\end{aligned}$$

The function μ is a continuous and differentiable quadratic convex function of $\frac{1}{2}s(s+1)$ variables w_{kl}, hence μ obtains its minimum value at $\{w_{kl}\}$ when $\frac{\partial \mu}{\partial w_{kl}} = 0$, i.e.,

$$w_{kl} = \frac{a_k J_{1kl} + a_l J_{1lk} + b_k J_{2kl} + b_l J_{2lk} + c_k J_{3kl} + c_l J_{3lk}}{a_k^2 + a_l^2 + b_k^2 + b_l^2 + c_k^2 + c_l^2}, \quad 1 \le k \le l \le s.$$

Therefore W can be expressed by (11.29). □

Now we state the main theorem, here we still suppose that $\mathrm{rank}(A) = \mathrm{rank}(B)$.

Theorem 4 *Let matrices* A, B, C *and* X_f *be given in Problem* I, *suppose* $\mathrm{rank}(A) = \mathrm{rank}(B)$, *partition the matrix* $Q^T X_f Q$ *into block matrix*

$$Q^T X_f Q = (X_{kl}^{(f)})_{6\times 6}, \tag{11.30}$$

with the same row and column dimensions as X^* *of* (11.23). *Then there is a unique solution* X_e *in Problem* I *and* X_e *can be expressed as*

$$X_e = Q \begin{pmatrix} E_{11} & E_{12} & E_{13} & \{X_{14}^{(f)}\} & X_{51}^{(0)T} & E_{31}^T \\ E_{12}^T & \bar{X}_{22} & \bar{X}_{23} & \{X_{24}^{(f)}\} & \bar{X}_{52}^{(0)T} & E_{32}^T \\ E_{13}^T & \bar{X}_{23}^T & \{X_{33}^{(f)}\} & \{X_{34}^{(f)}\} & \bar{X}_{53}^{(0)T} & E_{33}^T \\ \{X_{41}^{(f)}\} & \{X_{42}^{(f)}\} & \{X_{43}^{(f)}\} & \{X_{44}^{(f)}\} & \{X_{45}^{(f)}\} & \{X_{46}^{(f)}\} \\ X_{51}^{(0)} & \bar{X}_{52}^{(0)} & \bar{X}_{53}^{(0)} & \{X_{54}^{(f)}\} & \{X_{55}^{(f)}\} & \{X_{56}^{(f)}\} \\ E_{31} & E_{32} & E_{33} & \{X_{64}^{(f)}\} & \{X_{65}^{(f)}\} & \{X_{66}^{(f)}\} \end{pmatrix} Q^T \tag{11.31}$$

where (*using the notation* $\{X\}$ *to denote the symmetric part of* X):

$$\{X_{kl}^{(f)}\} \equiv \frac{1}{2}(X_{kl}^{(f)} + X_{lk}^{(f)T}); \tag{11.32}$$

$$\begin{aligned} \bar{X}_{22} = \bar{\Psi} * [&X_{22}^{(f)} + X_{22}^{(f)T} + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}E_{22} - X_{25}^{(f)T}) \\ &+ (\Delta_j^{-1}E_{22} - X_{25}^{(f)T})^T\Lambda_j\Delta_j^{-1} + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}E_{22} - X_{52}^{(f)}) \\ &+ (\Delta_j^{-1}E_{22} - X_{52}^{(f)})^T\Lambda_j\Delta_j^{-1}], \end{aligned} \tag{11.33}$$

$$\bar{\Psi} = (\psi_{kl}) \in R^{j\times j}, \quad \psi_{kl} = \frac{1}{2\left(1 + \left(\frac{\lambda_{i+k}}{\delta_{i+k}}\right)^2 + \left(\frac{\lambda_{i+l}}{\delta_{i+l}}\right)^2\right)}, \quad 1 \le k, l \le j;$$

$$\bar{X}_{23} = G * \left[X_{23}^{(f)} + X_{32}^{(f)T} + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}E_{23} - X_{35}^{(f)T}) + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}E_{23} - X_{53}^{(f)})\right], \tag{11.34}$$

$$G = (g_{kl}) \in R^{i\times t}, \quad g_{kl} = \frac{1}{2}\delta_{i+k}^2, \quad 1 \le k, \le i,\ 1 \le l \le t;$$

and

$$\bar{X}_{52}^{(0)} = \Delta_j^{-1}(E_{22} - \Lambda_j\bar{X}_{22}), \quad \bar{X}_{53}^{(0)} = \Delta_j^{-1}(E_{23} - \Lambda_j\bar{X}_{23}).$$

Proof Suppose $X \in S_E = S_{E_0}$, by using (11.25) and (11.30), we have

$$\begin{aligned} \|X - X_f\|_F^2 &= \|X^* - Q^T X_f Q\|_F^2 \\ &= (\|X_{33} - X_{33}^{(f)}\|_F^2) + (\|X_{44} - X_{44}^{(f)}\|_F^2) + (\|X_{55} - X_{55}^{(f)}\|_F^2) \\ &\quad + (\|X_{66} - X_{66}^{(f)}\|_F^2) + (\|X_{14} - X_{14}^{(f)}\|_F^2 + \|X_{14}^T - X_{41}^{(f)}\|_F^2) \\ &\quad + (\|X_{24} - X_{24}^{(f)}\|_F^2 + \|X_{24}^T - X_{42}^{(f)}\|_F^2) + (\|X_{34} - X_{34}^{(f)}\|_F^2 \\ &\quad + \|X_{34}^T - X_{43}^{(f)}\|_F^2) + (\|X_{45} - X_{45}^{(f)}\|_F^2 + \|X_{45}^T - X_{54}^{(f)}\|_F^2) \\ &\quad + (\|X_{46} - X_{46}^{(f)}\|_F^2 + \|X_{46}^T - X_{64}^{(f)}\|_F^2) + (\|X_{56} - X_{56}^{(f)}\|_F^2 \\ &\quad + \|X_{56}^T - X_{65}^{(f)}\|_F^2) + (\|X_{22} - X_{22}^{(f)}\|_F^2 \end{aligned}$$

$$
\begin{aligned}
&+ \|(\Delta_j^{-1}(E_{22} - \Lambda_j X_{22}))^T - X_{25}^{(f)}\|_F^2 \\
&+ \|\Delta_j^{-1}(E_{22} - \Lambda_j X_{22}) - X_{52}^{(f)}\|_F^2) + (\|X_{23} - X_{23}^{(f)}\|_F^2 \\
&+ \|X_{23}^T - X_{32}^{(f)}\|_F^2 + \|(\Delta_j^{-1}(E_{23} - \Lambda_j X_{23}))^T - X_{35}^{(f)}\|_F^2 \\
&+ \|\Delta_j^{-1}(E_{23} - \Lambda_j X_{23}) - X_{53}^{(f)}\|_F^2) + \alpha_0, \qquad (11.35)
\end{aligned}
$$

where α_0 is a constant.

According to (11.35), $\|X - X_f\|_F^2 = \min$ if and only if each of the brackets in (11.35) takes minimum. Notice that $X_{kk} = X_{kk}^T, k = 3, 4, 5, 6$ and by making use of Lemmas 2 and 3, the results of this theorem can be derived easily. □

Case II In the case of $g > h$, we first partition the symmetric matrix $X^* \equiv Q^T X Q$ into 8×8 block matrix, $X^* = (X_{kl})_{8\times 8}$, with the row dimensions (and the related column dimensions) of blocks are $i, j, t_1 = h - i - j, t_2 = g - h, n - g - j - t_1 - t_2, j, t_1, t_2$ respectively, and $X_{kl} = X_{lk}^T, k, l = 1, 2, \ldots, 8$ and let $E = X_A^T C_0 X_B$, which is partitioned into 5×4 block matrix, $E = (E_{kl})_{5\times 4}$, with the row dimensions of blocks are $i, j, t_1, t_2, m - g$ and the column dimensions of blocks are $i, j, t_1, p - h$ respectively, then by the similar discussion processes, we can obtain the similar results of Theorems 3 and 4, therefore we omit the processes.

According to Theorem 2.4, we can obtain an algorithm for finding the solution X_e of Problem I when $g = h$.

Algorithm 1

1. Input A, B, C and X_f.
2. Make QSVD of matrix pair (A, B^T) by (11.7) and partition the matrix $U^T C V = \{C_{kl}\}_{3\times 3}$ by (11.19).
3. Compute the matrix C_0 by (11.20).
4. Make CCD of the matrix pair (A^T, B) by (11.12) and partition the matrix $X^* = (X_{kl})_{6\times 6}$ and $E = (E_{kl})_{4\times 4}$ by (11.23) and (11.24) respectively.
5. Compute the matrix blocks $\{X_{kl}^{(f)}\}$, $\bar{X}_{22}$ and $\bar{X}_{23}$ by (11.32), (11.33) and (11.34) respectively.
6. Compute the solution X_e by (11.31).

11.3 The Solution of Problem II

The process to solve the Problem II is similar to the process to solve the Problem I. Therefore we only present the main steps and main results when $g \equiv \text{rank}(A) = \text{rank}(B) \equiv h$, while the proofs are omitted.

The first step is to transform Problem II into the following equivalent problem.

Problem II$_0$ Given matrices $A \in R^{m\times n}, B \in R^{n\times p}, C_a \in R^{m\times p}$ and $X_f \in R^{n\times n}$, let

$$S_{A_0} = \{X | X \in AR^{n\times n},\ AXB = C_a\} \tag{11.36}$$

Then find $X_a \in S_{A_0}$, such that

$$\|X_a - X_f\|_F = \min_{X\in S_{A_0}} \|X - X_f\|_F. \tag{11.37}$$

Theorem 5 *Given $A \in R^{m\times n}, B \in R^{n\times p}, C \in R^{m\times p}$, let X_s be any solution of (11.2) with Q the skew-symmetric cone, and define*

$$C_a = AX_sB, \tag{11.38}$$

then the matrix equation

$$AXB = C_a, \tag{11.39}$$

is consistent in $AR^{n\times n}$, and the skew-symmetric solution set S_{A_0} of the matrix equation (11.39) is the same as the skew-symmetric solution set S_A of the the least-squares problem (11.2).

The second step is to find the expression of C_a by using QSVD theorem.

Theorem 6 Let A, B, C *be given in Problem* II, *the matrix pair (A, B^T) has the QSVD (11.7), and U^TCV have partition (11.19), then for any skew-symmetric solutions X_s of the least-squares problem (11.2), the matrix C_a determined by (11.38) can be expressed by the following form:*

$$C_a = UC_*V^T, \quad C_* = \begin{pmatrix} 0 & C_{12} & C_{13} \\ 0 & S\hat{X}_0 & C_{23} \\ 0 & 0 & 0 \end{pmatrix} \begin{matrix} r' \\ s' \\ m-r'-s' \end{matrix}, \tag{11.40}$$

$$\begin{matrix} (p+r'-k') & s' & t' \end{matrix}$$

where

$$\begin{aligned} &\hat{X}_0 = \hat{\phi} * (SC_{22} - C_{22}^T S), \\ &\hat{\phi} = (\hat{\varphi}_{kl}) \in SR^{s'\times s'}, \quad \hat{\varphi}_{kl} = \frac{1}{\sigma_k^2 + \sigma_l^2}, \quad 1 \le k, l \le s'. \end{aligned} \tag{11.41}$$

Now suppose $X \in S_{A_0}$, then partition the skew-symmetric matrix $X_* \equiv Q^TXQ$ into block matrix,

$$X_* = (X_{kl})_{6\times 6}, \tag{11.42}$$

with the row dimensions (and the related column dimensions) of blocks are $i, j, t, n-g-j-t, j, t$ respectively, and $X_{kl} = -X_{lk}^T, k, l = 1, 2, \ldots, 6$. Let $F = X_A^T C_a X_B$ and also partition F into block matrix,

$$F = (F_{kl})_{4\times 4}, \tag{11.43}$$

with the row dimensions of blocks are $i, j, t, m-g$ and the column dimensions of blocks are $i, j, t, p-g$ respectively.

The next step is to present the general form of the elements of S_{A_0} by using CCD theorem.

Theorem 7 *In Problem* II_0, *the general form of* $X \in S_{A_0}$ *can be expressed as*

$$X = Q\begin{pmatrix} F_{11} & F_{12} & F_{13} & X_{14} & -Y_{51}^{(0)T} & -F_{31}^T \\ -F_{12}^T & X_{22} & X_{23} & X_{24} & -Y_{52}^{(0)T} & -F_{32}^T \\ -F_{13}^T & -X_{23}^T & X_{33} & X_{34} & -Y_{53}^{(0)T} & -F_{33}^T \\ -X_{14}^T & -X_{24}^T & -X_{34}^T & X_{44} & X_{45} & X_{46} \\ Y_{51}^{(0)} & Y_{52}^{(0)} & Y_{53}^{(0)} & -X_{45}^T & X_{55} & X_{56} \\ F_{31} & F_{32} & F_{33} & -X_{46}^T & -X_{56}^T & X_{66} \end{pmatrix} Q^T \tag{11.44}$$

where $X_{kk} = -X_{kk}^T, 2 \le k \le 6$, $X_{14}, X_{23}, X_{24}, X_{34}, X_{45}, X_{46}$ and X_{56} *are arbitrary matrices with the associated sizes, and* $Y_{51}^{(0)} = \Delta_j^{-1}(F_{21} + \Lambda_j F_{12}^T), Y_{52}^{(0)} = \Delta_j^{-1}(F_{22} - \Lambda_j X_{22}), Y_{53}^{(0)} = \Delta_j^{-1}(F_{23} - \Lambda_j X_{23}$.

Finally we give a lemma about the skew-symmetric solution of a minimum problem, and present the solution of Problem II.

Lemma 4 *For given matrices* J_1, J_2 *and* $J_3 \in R^{s\times s}, S_a = \mathrm{diag}(a_1, \ldots, a_s) > 0$, $S_b = \mathrm{diag}(b_1, \ldots, b_s) > 0, S_c = \mathrm{diag}(c_1, \ldots, c_s) > 0$, *then there exists a unique skew-symmetric matrix* $W \in AR^{s\times s}$, *such that*

$$\mu \equiv \|S_a W - J_1\|_F^2 + \|S_b W - J_2\|_F^2 + \|S_c W - J_3\|_F^2 = min,$$

and W can be expressed as

$$W = \Phi * (S_a J_1 - J_1^T S_a + S_b J_2 - J_2^T S_b + S_c J_3 - J_3^T S_c), \tag{11.45}$$

where

$$\Phi = (\phi_{kl}) \in R^{s\times s}, \quad \phi_{kl} = 1/(a_k^2 + a_l^2 + b_k^2 + b_l^2 + c_k^2 + c_l^2),\ 1 \le k, l \le s.$$

Theorem 8 *Let matrices* A, B, C *and* X_f *be given in Problem* II, *suppose* $\mathrm{rank}(A) = \mathrm{rank}(B)$, *and partition the matrix* $Q^T X_f Q$ *into block matrix* (11.30). *Then there is a unique solution* X_a *in Problem* II, *and* X_a *can be expressed as*

$$X_a = Q\begin{pmatrix} F_{11} & F_{12} & F_{13} & Y_{14}^{(f)} & -Y_{51}^{(0)T} & -F_{31}^T \\ -F_{12}^T & \hat{Y}_{22} & \hat{Y}_{23} & Y_{24}^{(f)} & -\hat{Y}_{52}^{(0)T} & -F_{32}^T \\ -F_{13}^T & -\hat{Y}_{23}^T & Y_{33}^{(f)} & Y_{34}^{(f)} & -\hat{Y}_{53}^{(0)T} & -F_{33}^T \\ Y_{41}^{(f)} & Y_{42}^{(f)} & Y_{43}^{(f)} & Y_{44}^{(f)} & Y_{45}^{(f)} & Y_{46}^{(f)} \\ Y_{51}^{(0)} & \hat{Y}_{52}^{(0)} & \hat{Y}_{53}^{(0)} & Y_{54}^{(f)} & Y_{55}^{(f)} & Y_{56}^{(f)} \\ F_{31} & F_{32} & F_{33} & Y_{64}^{(f)} & Y_{65}^{(f)} & Y_{66}^{(f)} \end{pmatrix} Q^T \tag{11.46}$$

where

$$Y_{kl}^{(f)} = \frac{1}{2}(X_{kl}^{(f)} - X_{lk}^{(f)T}),$$

$$\begin{aligned}\hat{Y}_{22} = \Psi * [&X_{22}^{(f)} - X_{22}^{(f)T} + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}F_{22} + X_{25}^{(f)T}) \\ &- (\Delta_j^{-1}F_{22} + X_{25}^{(f)T})^T\Lambda_j\Delta_j^{-1} + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}F_{22} - X_{52}^{(f)}) \\ &- (\Delta_j^{-1}F_{22} - X_{52}^{(f)})^T\Lambda_j\Delta_j^{-1}],\end{aligned}$$

$$\Psi = (\psi_{kl}) \in R^{j\times j}, \quad \psi_{kl} = \frac{1}{2\left(1 + \left(\frac{\lambda_{i+k}}{\delta_{i+k}}\right)^2 + \left(\frac{\lambda_{i+l}}{\delta_{i+l}}\right)^2\right)}, \quad 1 \le k, l \le j;$$

$$\hat{Y}_{23} = G * \left[X_{23}^{(f)} - X_{23}^{(f)T} + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}F_{23} + X_{35}^{(f)T}) + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}F_{23} - X_{53}^{(f)})\right],$$

$$G = (g_{kl}) \in R^{i\times t}, \quad g_{kl} = \frac{1}{2}\delta_{i+k}^2, \quad 1 \le k, \le i, 1 \le l \le t;$$

and

$$\hat{Y}_{52}^{(0)} = \Delta_j^{-1}(F_{22} - \Lambda_j\hat{Y}_{22}), \quad \hat{Y}_{53}^{(0)} = \Delta_j^{-1}(F_{23} - \Lambda_j\hat{Y}_{23}).$$

11.4 Conclusions

Using the projection theorem in a Hilbert space, the quotient singular value decomposition (QSVD) and the canonical correlation decomposition (CCD), we obtained explicit analytical expressions of the optimal approximation solutions for the symmetric and skew-symmetric Procrustes problems related to the linear matrix equation $AXB = C$. According to these new results, we can design new algorithms to solve optimal approximation problems for the constrained linear matrix equation and related least-squares problems, which can be applied to some scientific fields, such as inverse problems of vibration theory, control theory, finite elements, and multidimensional approximation and so on. In some aspects, we have generalized the works of Higham [13], Sun [23], Chu [4] and Hua [5].

Acknowledgments The author Deng would like to thank the China Scholarship Council for providing the State Scholarship Fund to pursue his research at the University of Minnesota as a visiting scholar. The author Boley would like to acknowledge partial support for this research from NSF grant IIS-0916750.

References

1. Andersson L, Elfving T (1997) A constrained Procrustes problem. SIAM J Matrix Anal Appl 18(1):124–139
2. Ben-Israel A, Greville T (1974) Generalized inverses: theory and applications. Wiley, New York
3. Chu D, De Moor B (2000) On a variational formulation of the QSVD and the RSVD. Linear Algebra Appl 311:61–78
4. Chu EK (1989) Symmetric solutions of linear matrix equations by matrix decompositions. Linear Algebra Appl 119:35–50
5. Dai H (1990) On the symmetric solutions of linear matrix equations. Linear Algebra Appl 131:1–7
6. Dai H, Lancaster P (1996) Linear matrix equations from an inverse problem of vibration theory. Linear Algebra Appl 246:31–47
7. Deng Y, Hu X, Zhang L (2003) Least squares solution of $BXAT = T$ over symmetric, skew-symmetric, and positive semidefinite X. SIAM J Matrix Anal Appl 25(2):486–494
8. Fausett D, Fulton C (1994) Large least squares problems involving Kronecker products. SIAM J Matrix Anal Appl 15:219–227
9. Golub G, Van Loan C (1989) Matrix computations, 2nd edn. Johns Hopkins University Press, Baltimore
10. Golub G, Zha H (1994) A perturbation analysis of the canonical correlations of matrix pairs. Linear Algebra Appl 210:3–28
11. Hald O (1972) On discrete and numerical Sturm–Liouville problems. Ph.D. Dissertation, Department of Mathematics, New York University, New York
12. Henk Don F (1987) On the symmetric solutions of a linear matrix equation. Linear Algebra Appl 93:1–7
13. Higham N (1988) The symmetric Procrustes problem. BIT 28:133–143
14. Horn R, Johnson C (1985) Matrix analysis. Cambridge University Press, Cambridge
15. Lancaster P (1970) Explicit solutions of linear matrix equations. SIAM Rev 72(4):544–566
16. Lancaster P, Tismenetsky M (1985) The theory of matrices, 2nd edn. Academic Press, Orlando
17. Liao A-P, Bai Z-Z, Lei Y (2005) Best approximate solutions of matrix equation $AXB + CYD = E$. SIAM J Matrix Anal Appl 27(3):675–688
18. Magnus J (1983) L-structured matrices and linear matrix equations. Linear Multilinear Algebra 14:67–88
19. Naylor AW, Sell GR (1982) Linear operator theory in engineering and science. Springer, New York
20. Paige C, Saunders M (1981) Towards a generalized singular value decomposition. SIAM J Numer Anal 18:398–405
21. Penrose R (1955) A generalized inverse for matrices. Proc Cambridge Philos Soc 51:406–413
22. Sun J (1987) Least-squares solutions of a class of inverse eigenvalue problems. Math Numer Sinica 2:206–216 (in Chinese)
23. Sun J (1988) Two kinds of inverse eigenvalue problems. Math Numer Sinica 3:282–290 (in Chinese)
24. Xu G, Wei M, Zheng D (1998) On solutions of matrix equation $AXB + CYD = F$. Linear Algebra Appl 279:93–109
25. Zha H (1995) Comments on large least squares problems involving Kronecker products. SIAM J Matrix Anal Appl 16(4):1172

Chapter 12
Some Inverse Eigenvalue and Pole Placement Problems for Linear and Quadratic Pencils

Sylvan Elhay

Abstract Differential equation models for vibrating systems are associated with matrix eigenvalue problems. Frequently the undamped models lead to problems of the generalized eigenvalue type and damped models lead to problems of the quadratic eigenvalue type. The matrices in these systems are typically real and symmetric and are quite highly structured. The design and stabilisation of systems modelled by these equations (eg., undamped and damped vibrating systems) requires the determination of solutions to the inverse eigenvalue problems which are themselves real, symmetric and possibly have some other structural properties. In this talk we consider some pole assignment problems and inverse spectral problems for generalized and quadratic symmetric pencils, discuss some advances and point to some work that remains to be done.

Dedicated with friendship and respect to Biswa N. Datta for his contributions to mathematics.

S. Elhay 2008

12.1 Introduction: Linear and Quadratic Eigenvalue Problems

The equation

$$\boldsymbol{M}\boldsymbol{v}''(t) + \boldsymbol{C}\boldsymbol{v}'(t) + \boldsymbol{K}\boldsymbol{v}(t) = \boldsymbol{0}, \tag{12.1}$$

$\boldsymbol{M}, \boldsymbol{C}, \boldsymbol{K} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{v}(t) \in \mathbb{R}^n$ is used in many engineering applications to model natural phenomena. In the context of the free vibrations of a linear, time-invariant

S. Elhay (✉)
School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia
e-mail: sylvan.elhay@adelaide.edu.au

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_12,

vibratory system $\boldsymbol{M}, \boldsymbol{C}$ and $\boldsymbol{K}$ are the *mass, damping* and *stiffness* matrices, respectively.

Separation of variables,

$$\boldsymbol{v}(t) = \boldsymbol{x}e^{\lambda t}$$

$\boldsymbol{x} \in \mathbb{R}^n$ a constant, leads to an eigenvalue problem for the quadratic pencil

$$\boldsymbol{Q}(\lambda) = \lambda^2\boldsymbol{M} + \lambda\boldsymbol{C} + \boldsymbol{K}. \tag{12.2}$$

When $\boldsymbol{M}$ is symmetric positive definite (*spd*) and $\boldsymbol{C}, \boldsymbol{K}$ are symmetric we say that $\boldsymbol{Q}$ is *symmetric definite* and throughout this paper we will assume, unless stated otherwise, that (12.2) is real and symmetric definite.

Consider first the linear pencil

$$\boldsymbol{P}(\lambda) = \boldsymbol{K} - \lambda\boldsymbol{M}$$

which is a special case of $\mathbf{Q}(\mu)$ in which $\boldsymbol{C} = \boldsymbol{O}$ and we use the substitution $\mu^2 = -\lambda$. The scalar λ_j and the associated n-vector $\boldsymbol{x}_j \neq \mathbf{0}$ are called an eigenpair of $\boldsymbol{P}$ if they satisfy

$$\boldsymbol{P}(\lambda_j)\boldsymbol{x}_j = \mathbf{0}. \tag{12.3}$$

The pencil $\boldsymbol{P}$ has n real eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and n linearly independent, real eigenvectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ [29] because $\boldsymbol{M}$ is spd. We denote the spectrum, $\{\lambda_j\}_{j=1}^n$, of $\boldsymbol{P}$ variously as

$$\sigma(\boldsymbol{P}(\lambda)) \quad \text{or} \quad \sigma(-\boldsymbol{M}, \boldsymbol{K}).$$

We can assemble all the relations of the form (12.3) into a single matrix equation

$$\boldsymbol{K}\boldsymbol{X} - \boldsymbol{M}\boldsymbol{X} = \mathbf{O}$$

if we define $\boldsymbol{\Lambda} = \operatorname{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ and $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$. It is well known [29] that the eigenvectors in $\boldsymbol{X}$ satisfy the orthogonality relations

$$\left.\begin{array}{l} \boldsymbol{x}_i^T\boldsymbol{K}\boldsymbol{x}_j = 0 \\ \boldsymbol{x}_i^T\boldsymbol{M}\boldsymbol{x}_j = 0 \end{array}\right\} i \neq j$$

and their scaling can be chosen so that

$$\boldsymbol{X}^T\boldsymbol{K}\boldsymbol{X} = \boldsymbol{\Lambda} \quad \text{and} \quad \boldsymbol{X}^T\boldsymbol{M}\boldsymbol{X} = \boldsymbol{I}. \tag{12.4}$$

Thus, $\boldsymbol{X}$ is the matrix which simultaneously diagonalizes the two matrices $\boldsymbol{M}$ and $\boldsymbol{K}$. The eigenvalues of $\boldsymbol{P}$ are given by the *Rayleigh quotients*,

$$\lambda_j = \frac{\boldsymbol{x}_j^T \boldsymbol{K} \boldsymbol{x}_j}{\boldsymbol{x}_j^T \boldsymbol{M} \boldsymbol{x}_j} = \boldsymbol{x}_j^T \boldsymbol{K} \boldsymbol{x}_j$$

if the scaling (12.4) is used. The orthogonality of the eigenvectors has extensive practical application in science and engineering (see [7, pp. 512–531, 23]).

The quadratic pencil (12.2) has $2n$ finite eigenvalues which are the zeros of the degree $2n$ polynomial

$$\det \boldsymbol{Q}(\lambda) = \det \boldsymbol{M} \lambda^{2n} + a_{2n-1} \lambda^{2n-1} + \cdots,$$

and we will assume, unless stated otherwise, that the spectrum of $\boldsymbol{Q}$

$$\sigma(\boldsymbol{Q}(\lambda)) = \sigma(\boldsymbol{M}, \boldsymbol{C}, \boldsymbol{K}) \stackrel{\text{def}}{=} \{\lambda_j\}_{j=1}^{2n}$$

consists of $2n$ *distinct* eigenvalues. The pencil $\boldsymbol{Q}$ then has n linearly independent eigenvectors and we can write the $n \times 2n$ system

$$\boldsymbol{M} \boldsymbol{X} \boldsymbol{\Lambda}^2 + \boldsymbol{C} \boldsymbol{X} \boldsymbol{\Lambda} + \boldsymbol{K} \boldsymbol{X} = \boldsymbol{O} \tag{12.5}$$

in which $\boldsymbol{X} \in \mathcal{C}^{n \times 2n}$, and $\boldsymbol{\Lambda} = \operatorname{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_{2n}\} \in \mathcal{C}^{2n \times 2n}$. Since $\boldsymbol{M}, \boldsymbol{C}, \boldsymbol{K}$ are real and $\boldsymbol{M}$ is invertible, the eigenvectors and eigenvalues of $\boldsymbol{Q}$ are *pairwise self-conjugate* in the sense that they are self-conjugate and $\boldsymbol{x}_i = \bar{\boldsymbol{x}}_j$ whenever $\lambda_i = \bar{\lambda}_j$, for all i and j.

Relation (12.5) is sometimes *linearised* into block companion form as

$$\begin{pmatrix} \boldsymbol{O} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{pmatrix} \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{X}\boldsymbol{\Lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{X}\boldsymbol{\Lambda} \end{pmatrix} \boldsymbol{\Lambda},$$

or its symmetric, generalized eigenvalue problem equivalent

$$\begin{pmatrix} \boldsymbol{O} & \boldsymbol{K} \\ \boldsymbol{K} & \boldsymbol{C} \end{pmatrix} \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{X}\boldsymbol{\Lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{K} & \\ & -\boldsymbol{M} \end{pmatrix} \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{X}\boldsymbol{\Lambda} \end{pmatrix} \boldsymbol{\Lambda}.$$

There is [9] a set of three orthogonality relations[1] for the quadratic pencil which generalize the orthogonality relations (12.4) for the linear pencil:

$$\left.\begin{aligned} \boldsymbol{\Lambda} \boldsymbol{X}^T \boldsymbol{M} \boldsymbol{X} \boldsymbol{\Lambda} - \boldsymbol{X}^T \boldsymbol{K} \boldsymbol{X} &= \boldsymbol{D}_1, \\ \boldsymbol{\Lambda} \boldsymbol{X}^T \boldsymbol{C} \boldsymbol{X} \boldsymbol{\Lambda} + \boldsymbol{\Lambda} \boldsymbol{X}^T \boldsymbol{K} \boldsymbol{X} + \boldsymbol{X}^T \boldsymbol{K} \boldsymbol{X} \boldsymbol{\Lambda} &= \boldsymbol{D}_2, \\ \boldsymbol{\Lambda} \boldsymbol{X}^T \boldsymbol{M} \boldsymbol{X} + \boldsymbol{X}^T \boldsymbol{M} \boldsymbol{X} \boldsymbol{\Lambda} + \boldsymbol{X}^T \boldsymbol{C} \boldsymbol{X} &= \boldsymbol{D}_3, \end{aligned}\right\} \tag{12.6}$$

where the three diagonal matrices, $\boldsymbol{D}_{1,2,3}$, satisfy

$$\boldsymbol{D}_1 = \boldsymbol{D}_3 \boldsymbol{\Lambda}, \quad \boldsymbol{D}_2 = -\boldsymbol{D}_1 \boldsymbol{\Lambda} \quad \text{and} \quad \boldsymbol{D}_2 = -\boldsymbol{D}_3 \boldsymbol{\Lambda}^2.$$

[1] In this paper a conjugate transpose is denoted by a superscript H while transposition, which is denoted by a superscript T, does not mean conjugate transpose even for complex quantities.

These relations, written componentwise, are

$$\left.\begin{array}{c} \boldsymbol{x}_i^T(\lambda_i\lambda_j\boldsymbol{M}-\boldsymbol{K})\boldsymbol{x}_j=0 \\ \boldsymbol{x}_i^T(\lambda_i\lambda_j\boldsymbol{C}+(\lambda_i+\lambda_j)\boldsymbol{K})\boldsymbol{x}_j=0 \\ \boldsymbol{x}_i^T((\lambda_i+\lambda_j)\boldsymbol{M}+\boldsymbol{C})\boldsymbol{x}_j=0 \end{array}\right\}, \quad i\neq j.$$

Provided the denominators do not vanish, we can also write

$$\left.\begin{array}{rcl} \lambda_i\lambda_j & = & \frac{\boldsymbol{x}_i^T\boldsymbol{K}\boldsymbol{x}_j}{\boldsymbol{x}_i^T\boldsymbol{M}\boldsymbol{x}_j} \\ -\frac{\lambda_i+\lambda_j}{\lambda_i\lambda_j} & = & \frac{\boldsymbol{x}_i^T\boldsymbol{C}\boldsymbol{x}_j}{\boldsymbol{x}_i^T\boldsymbol{K}\boldsymbol{x}_j} \\ -(\lambda_i+\lambda_j) & = & \frac{\boldsymbol{x}_i^T\boldsymbol{C}\boldsymbol{x}_j}{\boldsymbol{x}_i^T\boldsymbol{M}\boldsymbol{x}_j} \end{array}\right\}, \quad i\neq j$$

and the Rayleigh-quotient-like expressions

$$\left.\begin{array}{rcl} \lambda_i & = & \frac{\boldsymbol{x}_i^T(\lambda_i^2\boldsymbol{M}-\boldsymbol{K})\boldsymbol{x}_i}{\boldsymbol{x}_i^T(2\lambda_i\boldsymbol{M}+\boldsymbol{C})\boldsymbol{x}_i} \\ -\lambda_i & = & \frac{\boldsymbol{x}_i^T(\lambda_i^2\boldsymbol{C}+2\lambda_i\boldsymbol{K})\boldsymbol{x}_i}{\boldsymbol{x}_i^T(\lambda_i^2\boldsymbol{M}-\boldsymbol{K})\boldsymbol{x}_i} \\ -\lambda_i^2 & = & \frac{\boldsymbol{x}_i^T(\lambda_i^2\boldsymbol{C}+2\lambda_i\boldsymbol{K})\boldsymbol{x}_i}{\boldsymbol{x}_i^T(2\lambda_i\boldsymbol{M}+\boldsymbol{C})\boldsymbol{x}_i} \end{array}\right\}, \quad i=1,2,\ldots,2n.$$

Note that when $\boldsymbol{C}=\boldsymbol{O}$, this last relation simplifies to the Rayleigh quotient $\boldsymbol{x}_j^T\boldsymbol{K}\boldsymbol{x}_j=\lambda_j$ (recall the substitution $-\lambda^2=\mu$). Unlike the linear case, however, there is in general no way to simultaneously diagonalize *three* symmetric matrices.

Table 12.1 shows information about the eigendata of quadratic pencils and is drawn from the excellent survey of the quadratic eigenvalue problem by Tissuer and Meerbergen [35]. As is pointed out in their survey, quadratic eigenvalue problems arise in the dynamic analysis of structural mechanical, and acoustic systems, in electrical circuit simulation, in fluid mechanics, signal processing and in modeling microelectronic mechanical systems.

In this paper we describe some methods for the solution of certain inverse eigenvalue and pole placement problems for the linear and quadratic pencils. Our interest here is in the linear algebra although we will sometimes also make reference to the application of the problems we discuss in the study of vibrations.

The rest of the paper is organised as follows. In Sect. 12.2 we discuss an inverse eigenvalue problem for a matrix pair that arises in the modelling of the axial vibrations of a rod. The problem can be solved [33] by fixed point iteration and has an interesting reformulation as either an inverse eigenvalue problem for a (symmetric, tridiagonal, unreduced) Jacobi matrix or an inverse singular value problem for a bidiagonal, unit lower triangular matrix.

In Sect. 12.3 we briefly mention the solution [31] to an inverse eigenvalue problem for the symmetric definite quadratic pencil which has application to the study of damped oscillatory systems.

Table 12.1 Some properties of the eigenvalues and eigenvectors of quadratic pencils $\gamma(\boldsymbol{M},\boldsymbol{C},\boldsymbol{K}) = \min_{\|\boldsymbol{x}\|_2=1}(\boldsymbol{x}^H\boldsymbol{C}\boldsymbol{x})^2 - 4(\boldsymbol{x}^H\boldsymbol{M}\boldsymbol{x})(\boldsymbol{x}^H\boldsymbol{K}\boldsymbol{x})$

Matrix	Eigenvalues	Eigenvectors
$\boldsymbol{M}$ nonsingular	$2n$ finite eigenvalues	
$\boldsymbol{M}$ singular	Finite and infinite eigenvalues	
$\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{K}$ real	Finite eigenvalues are real or conjugate pairs	If x is a right eigenvector of λ then $\bar{x}$ is a right eigenvector of $\bar{\lambda}$
$\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{K}$ Hermitian	Finite eigenvalues are real or conjugate pairs	If x is a right eigenvector of λ then x is a left eigenvector of $\bar{\lambda}$
$\boldsymbol{M}$ Hermitian positive definite, $\boldsymbol{C}$, $\boldsymbol{K}$ Hermitian positive semidefinite	$Re(\lambda) \leq 0$	
$\boldsymbol{M}$ Hermitian positive definite, $\boldsymbol{C}$ Hermitian, $\boldsymbol{K}$ Hermitian negative definite	Real eigenvalues	Real eigenvectors
$\boldsymbol{M}$, $\boldsymbol{C}$ symmetric positive definite, $\boldsymbol{K}$ symmetric positive semidefinite, $\gamma(\boldsymbol{M},\boldsymbol{C},\boldsymbol{K}) > 0$ (overdamped)	λ's are real and negative, gap between n largest and n smallest eigenvalues	n linearly independent eigenvectors associated with the n largest (n smallest) eigenvalues
$\boldsymbol{M}$, $\boldsymbol{K}$ Hermitian, $\boldsymbol{M}$ positive definite, $\boldsymbol{C} = -\boldsymbol{C}^H$	Eigenvalues are pure imaginary or come in pairs $(\lambda, -\bar{\lambda})$	If $\boldsymbol{x}$ is a right eigenvector of λ then $\boldsymbol{x}$ is a left eigenvector of $-\bar{\lambda}$
$\boldsymbol{M}$, $\boldsymbol{K}$ real symmetric and positive definite, $\boldsymbol{C} = -\boldsymbol{C}^T$	Eigenvalues are pure imaginary	

In Sect. 12.4 we consider problems of partial pole assignment by single-input control for the symmetric definite quadratic pencil. We describe an explicit and a computational solution [9] which both derive from the orthogonality relations (12.6).

In Sect. 12.5 we consider three methods [8, 32] for partial pole assignment by multi-input control for the symmetric definite quadratic pencil. The first parallels the computational solution for the single-input case. The second, not described in detail, again uses the orthogonality relations (12.6) to construct a multi-step solution and the last is based on the Cholesky factoring and the Singular value decomposition.

In Sect. 12.6 we describe a technique [10] for solving the problem of partial eigenstructure assignment by multi-input control for the symmetric definite quadratic pencil. This method is again based on the exploitation of the orthogonality relations (12.6).

In Sect. 12.7 we describe two methods [17, 18] of pole assignment for the symmetric definite quadratic pencil by affine sums and in Sect. 12.8 we describe an explicit formula for symmetry preserving partial pole assignment to a symmetric definite matrix pair.

Finally, we summarise in Sect. 12.9 and indicate some further worthwhile work.

All the methods described in this paper are the result of work done variously with Professors B.N. Datta, G.H. Golub and Y.M. Ram. The author tenders grateful tribute to their contributions.

12.2 An Inverse Eigenvalue Problem for a Linear Pencil Arising in the Vibration of Rods

To begin with we consider an inverse eigenvalue problem for a rather special matrix pair. The problem arises in a discretization of the eigenvalue problem $(EAy')' + \lambda\rho Ay = 0, 0 < x < 1$, with boundary conditions $y'(0) = 0, y(1) = 0$, associated with the axial oscillations of a non-uniform rod. Here E, ρ and A are problem parameters.

Denote the Kroenecker delta by δ_{ij} and denote by $\boldsymbol{I}$, $\boldsymbol{S}$ and $\boldsymbol{E}$, respectively, the $n \times n$ *identity*, *shift*, and *exchange* matrices

$$[\boldsymbol{I}]_{ij} = \delta_{ij},\ [\boldsymbol{S}]_{ij} = \delta_{i,j-1},\ [\boldsymbol{E}]_{ij} = \delta_{i,n-j+1}.$$

Further, denote by $\boldsymbol{F} = \boldsymbol{I} - \boldsymbol{S}$ the *finite difference* matrix.

Problem 2.1 Given a set $S = \{\lambda_j\}_{j=1}^n$ of real numbers such that $\prod_{j=1}^n \lambda_j = 1$ find a diagonal matrix $\boldsymbol{D} = \text{diag}\,\{d_1, d_2, \ldots, d_n\}$, $d_i > 0$ such that the pencil $\boldsymbol{P}(\lambda) = \boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F} - \lambda\boldsymbol{D}$ has spectrum S.

Note that if $\boldsymbol{D}$ is a solution to Problem 2.1 then $\alpha\boldsymbol{D}$, α scalar, is also a solution so we can, without loss of generality, set $d_1 = 1$. Note also that $\sigma(\boldsymbol{P}) = \sigma(\boldsymbol{D}^{-1}\boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F})$ and, in view of the fact that $\det(\boldsymbol{F}) = 1$ we know that the eigenvalues of this system must therefore satisfy $\prod_{k=1}^n \lambda_k = 1$. Thus,

a. $n - 1$ eigenvalues uniquely determine the nth and
b. there exist a finite number, at most $(n-1)!$, different (generally complex) solutions $\boldsymbol{D}$ even though only real positive solutions have a physical meaning in the context of vibrations.

Define $\boldsymbol{B} = \boldsymbol{D}^2\boldsymbol{S}\boldsymbol{D}^{-1}\boldsymbol{S}^T - \boldsymbol{S}^T\boldsymbol{D}\boldsymbol{S}$ and $\hat{\boldsymbol{D}} = d_n\boldsymbol{E}\boldsymbol{D}^{-1}\boldsymbol{E}$. Interestingly, the three systems

$$\boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F} - \lambda\boldsymbol{D}, \quad \boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F} + \boldsymbol{B} - \lambda\boldsymbol{D}, \quad \text{and} \quad \boldsymbol{F}^T\hat{\boldsymbol{D}}\boldsymbol{F} - \lambda\hat{\boldsymbol{D}},$$

are isospectral (see [33] for details).

We can recast Problem 2.1 into an interesting equivalent inverse standard eigenvalue problem form. Denote by $\boldsymbol{H}$ the sign matrix $[\boldsymbol{H}]_{ij} = (-)^{i+1}\delta_{ij}$. Then the matrix $\boldsymbol{H}\boldsymbol{D}^{-1/2}\boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F}\boldsymbol{D}^{-1/2}\boldsymbol{H}$ has the same eigenvalues as $\boldsymbol{P}(\lambda) = \boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F} - \lambda\boldsymbol{D}$ and the (symmetric, tridiagonal, unreduced) Jacobi matrix $\boldsymbol{J} = \boldsymbol{H}\boldsymbol{D}^{-1/2}\boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F}\boldsymbol{D}^{-1/2}\boldsymbol{H} - \boldsymbol{I}$ which has eigenvalues $\omega_i = \lambda_i - 1$ has the quite special structure

$$\boldsymbol{J} = \begin{pmatrix} 0 & \beta_1 & & & \\ \beta_1 & \beta_1^2 & \beta_2 & & \\ & \beta_2 & \beta_2^2 & \beta_3 & \\ & & \cdot & \cdot & \\ & & & \beta_{n-1} & \beta_{n-1}^2 \end{pmatrix},$$

where

$$\beta_i = \sqrt{d_i/d_{i+1}}, \quad i = 1, 2, \ldots, n-1.$$

Our problem is now to reconstruct $\boldsymbol{J}$ from its spectrum only. There are just $n-1$ free parameters to be found from $n-1$ independent pieces of data: ω_i, $i = 1, 2, \ldots, n$ constrained by $\prod_{k=1}^{n}(1+\omega_k) = 1$. The equivalent singular value assignment problem is

Problem 2.2 Find, if it exists, a unit lower bidiagonal matrix

$$\boldsymbol{L} = \begin{pmatrix} 1 & & & & \\ \beta_1 & 1 & & & \\ & \beta_2 & 1 & & \\ & & \cdot & \cdot & \\ & & & \beta_{n-1} & 1 \end{pmatrix}$$

which has prescribed singular values $\sigma_1, \sigma_2, \ldots, \sigma_n$ constrained by $\prod_{k=1}^{n}\sigma_k = 1$.

$\boldsymbol{L}$ here is the Cholesky factor $\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{J} + \boldsymbol{I}$.

It turns out [33] that the $\{\beta_j\}_{j=1}^{n-1}$ satisfy a fixed point equation $\boldsymbol{A}\boldsymbol{\Omega}\boldsymbol{A}^{-1}\boldsymbol{b} = \boldsymbol{c}$ where $\boldsymbol{\Omega} = \text{diag}\{\omega_1, \omega_2, \ldots, \omega\}_n$

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \boldsymbol{a}_3^T \\ \vdots \\ \boldsymbol{a}_n^T \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1 \\ \beta_1^2 \\ \beta_2^2 \\ \vdots \\ \beta_{n-1}^2 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 0 \\ \beta_1^4 \\ \beta_2^4 \\ \vdots \\ \beta_{n-1}^4 \end{pmatrix}$$

and each $\boldsymbol{a}_i$ defines a diagonal scaling matrix that turns the first row of the eigenvector matrix of $\boldsymbol{J}$ into the ith row. A linear convergence fixed point iteration scheme for the solution of this problem is given in Ram and Elhay [33].

12.3 An Inverse Eigenvalue Problem for the Tridiagonal, Symmetric Definite Quadratic Pencil

We now consider the system of monic, time differential equations

$$\boldsymbol{I}\boldsymbol{v}'' + \boldsymbol{C}\boldsymbol{v}' + \boldsymbol{K}\boldsymbol{v} = \boldsymbol{0},$$

in which the matrices $\boldsymbol{C}, \boldsymbol{K} \in \mathbb{R}^{n\times n}$ are tridiagonal, symmetric,

$$\boldsymbol{C} = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \ldots, & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \ldots, & 0 \\ 0 & \beta_2 & \alpha_3 & \ldots, & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_n \end{pmatrix}, \quad \boldsymbol{K} = \begin{pmatrix} \gamma_1 & \delta_1 & 0 & \ldots, & 0 \\ \delta_1 & \gamma_2 & \delta_2 & \ldots, & 0 \\ 0 & \delta_2 & \gamma_3 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \gamma_n \end{pmatrix},$$

$\boldsymbol{I}$ is an identity, and $\boldsymbol{v} = \boldsymbol{v}(t)$. The corresponding quadratic pencil is

$$\boldsymbol{Q}(\lambda) = \lambda^2\boldsymbol{I} + \lambda\boldsymbol{C} + \boldsymbol{K}. \tag{12.7}$$

Our problem here is to find $\boldsymbol{C}$ and $\boldsymbol{K}$ from the spectrum of the full pencil and the spectrum of the reduced pencil in which the last row and column have been deleted. More precisely, we need to solve

Problem 3.1 Given two sets of distinct numbers $\{\lambda_k\}_{k=1}^{2n}$and$\{\mu_k\}_{k=1}^{2n-2}$, Find tridiagonal symmetric $\boldsymbol{C}$ and $\boldsymbol{K}$ such that $\boldsymbol{Q}(\lambda) = \lambda^2\boldsymbol{I} + \lambda\boldsymbol{C} + \boldsymbol{K}$ satisfies

$$det(\boldsymbol{Q}(\lambda)) \text{ has zeros } \{\lambda_k\}_{k=1}^{2n}$$

and the matrix $\hat{\boldsymbol{Q}}(\lambda)$, obtained by deleting the last row and column of $\boldsymbol{Q}(\lambda)$ satisfies,

$$det(\hat{\boldsymbol{Q}}(\lambda)) \text{ has zeros } \{\mu_k\}_{k=1}^{2n-2}.$$

The problem of reconstructing one unreduced, symmetric, tridiagonal matrix from its n eigenvalues and those of its leading principal submatrix of dimension $n-1$ is related to Problem 3.1 (it is a special case of (12.7) with $\boldsymbol{K}$ a Jacobi matrix and $\boldsymbol{C} = \boldsymbol{O}$ and has received much attention in the literature (see eg., [3, 13, 21, 22]). In vibrations, this may be regarded as identifying the spring configurations of an undamped system from its spectrum and the spectrum of the *constrained* system where the last mass is restricted to have no motion. Problem 3.1 corresponds to determining the spring *and damper* configurations of a non-conservative vibratory system which has a prescribed spectrum and which is such that the associated constrained system has a prescribed spectrum.

We have shown by construction [31] that this problem is always soluble over the complex field, it has at most $2^n(2n-3)!/(n-2)!$ solutions and *all* solutions can be found by a method that, aside from finding the roots of certain polynomials, requires only a finite number of steps.

The solution matrices should (a) have positive diagonal elements, (b) have negative off-diagonal elements and (c) be weakly diagonally dominant, for practical realizations in vibrations. Finding solutions that satisfy these constraints for large, real problems remains a challenge.

12.4 Partial Pole Assignment by Single-Input Control for the Symmetric Definite Quadratic Pencil

Partial pole assignment by state feedback control for a system modeled by a set of second order differential equations is used in the stabilization and control of flexible, large, space structures where only a small part of the spectrum is to be reassigned and the rest of the spectrum is required to remain unchanged.

In this problem the homogeneous differential equation (12.1) is replaced by an equation with a forcing function $\boldsymbol{b}u(t)$, $\boldsymbol{b} \in \mathbb{R}^n$ a constant, and $u(t)$ a scalar, by which we want to control this system. Thus the differential equation is now

$$\boldsymbol{M}\boldsymbol{v}'' + \boldsymbol{C}\boldsymbol{v}' + \boldsymbol{K}\boldsymbol{v} = \boldsymbol{b}u(t) \tag{12.8}$$

and we seek constant $\boldsymbol{f}, \boldsymbol{g} \in \mathbb{R}^n$ which define the control

$$u(t) = \boldsymbol{f}^T\boldsymbol{v}'(t) + \boldsymbol{g}^T\boldsymbol{v}(t) \tag{12.9}$$

which will assign all, or part, of the spectrum of the system. Substituting for $u(t)$ in (12.8) with (12.9) leads to the *closed loop system*

$$\boldsymbol{M}\boldsymbol{v}'' + (\boldsymbol{C} - \boldsymbol{b}\boldsymbol{f}^T)\boldsymbol{v}' + (\boldsymbol{K} - \boldsymbol{b}\boldsymbol{g}^T)\boldsymbol{v} = \boldsymbol{0},$$

the dynamics of which are characterised by the eigenvalues of the *closed loop pencil*

$$\boldsymbol{Q}_c(\lambda) = \boldsymbol{M}\lambda^2 + (\boldsymbol{C} - \boldsymbol{b}\boldsymbol{f}^T)\lambda + (\boldsymbol{K} - \boldsymbol{b}\boldsymbol{g}^T).$$

It is well known [27] that the system is *completely controllable* if and only if

$$\operatorname{rank}\{\lambda^2\boldsymbol{M} + \lambda\boldsymbol{C} + \boldsymbol{K}, \boldsymbol{b}\} = n,$$

for every eigenvalue of $\boldsymbol{Q}$. Complete controllability is a necessary and sufficient condition for the existence of $\boldsymbol{f}$ and $\boldsymbol{g}$ such that the closed-loop pencil has a spectrum that can be assigned arbitrarily. However, if the system is only *partially controllable*, i.e., if

$$\operatorname{rank}\{\lambda^2\boldsymbol{M} + \lambda\boldsymbol{C} + \boldsymbol{K}, \boldsymbol{b}\} = n,$$

only for m of the eigenvalues $\lambda = \lambda_{i_k}$, $k = 1, 2, \ldots, m, m < n$, of the pencil, then only those eigenvalues can be arbitrarily assigned by an appropriate choice of $\boldsymbol{f}$ and $\boldsymbol{g}$. The system (12.8) is partially controllable iff the vector $\boldsymbol{b}$ is not orthogonal to $\{\boldsymbol{x}_{i_k}\}_{k=1}^m$, the eigenvectors corresponding to the assignable eigenvalues $\{\lambda_{i_k}\}_{k=1}^m$.

This problem can approached by using the block companion first order realization i.e. by finding $\hat{\boldsymbol{f}}$ such that the $2n \times 2n$ matrix $\boldsymbol{A} - \hat{\boldsymbol{b}}\hat{\boldsymbol{f}}^T$, where

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{O} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{pmatrix}, \quad \hat{\boldsymbol{b}} = \begin{pmatrix} \boldsymbol{O} \\ \boldsymbol{M}^{-1}\boldsymbol{b} \end{pmatrix}, \quad \hat{\boldsymbol{f}} = \begin{pmatrix} -\boldsymbol{g} \\ -\boldsymbol{f} \end{pmatrix},$$

has the desired spectrum. The first order realization solution, however, is not always suitable since it does not respect structural properties such as sparsity, bandedness or positive definiteness that are sometimes assets for this problem.

It is often also an important practical requirement that no *spill-over*, the phenomenon in which eigenvalues not intended to be changed are modified by the process, occurs [1, 2]. In fact, partial pole assignment without spillover is possible [9] knowing only the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ which are to be assigned and their associated eigenvectors. Where n is large and only $m \ll n$ eigenvalues are to be assigned this can be a significant advantage. The eigendata can be found by computation using a Krylov subspace methods [30], or by modal analysis measurements when the physical structure is available [23] More precisely, the problem we now have is:

Problem 4.1 *Let* (12.5), λ_i distinct, be an eigendecomposition of the quadratic open loop pencil (12.2).

Given a self-conjugate set of $m \leq 2n$ complex numbers $\mu_1, \mu_1, \ldots, \mu_m$ and a vector $\boldsymbol{b} \in \mathbb{R}^n$,

Find $\boldsymbol{f}, \boldsymbol{g} \in \mathcal{C}^n$ which are such that the closed loop pencil

$$\boldsymbol{Q}_c(\lambda) = \boldsymbol{M}\lambda^2 + (\boldsymbol{C} - \boldsymbol{b}\boldsymbol{f}^T)\lambda + (\boldsymbol{K} - \boldsymbol{b}\boldsymbol{g}^T), \tag{12.10}$$

has spectrum $\{\mu_1, \mu_2, \ldots, \mu_m, \lambda_{m+1}, \ldots, \lambda_{2n}\}$.

Let us partition the $n \times 2n$ eigenvector matrix and $2n \times 2n$ eigenvalue matrix as

$$\underset{m}{\boldsymbol{X}} = (\underset{m}{\boldsymbol{X}_1} \quad \underset{2n-m}{\boldsymbol{X}_2}), \qquad \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda_1} & \\ & \boldsymbol{\Lambda_2} \\ m & 2n-m. \end{pmatrix} \begin{matrix} m \\ 2n-m \end{matrix} \tag{12.11}$$

The theorem that follows gives conditions on the vectors $\boldsymbol{f}, \boldsymbol{g}$ which ensure that the eigenvalues not being assigned remain unchanged after the assignment, thus avoiding spillover.

Theorem 4.1 *Let*

$$\boldsymbol{f} = \boldsymbol{M}\boldsymbol{X}_1\boldsymbol{\Lambda}_1\boldsymbol{\beta}, \quad \boldsymbol{g} = -\boldsymbol{K}\boldsymbol{X}_1\boldsymbol{\beta}, \quad \beta \in \mathcal{C}^m. \tag{12.12}$$

Then, for any choice of β *we have*

$$\boldsymbol{M}\boldsymbol{X}_2\boldsymbol{\Lambda}_2^2 + (\boldsymbol{C} - \boldsymbol{b}\boldsymbol{f}^T)\boldsymbol{X}_2\boldsymbol{\Lambda}_2 + (\boldsymbol{K} - \boldsymbol{b}\boldsymbol{g}^T)\boldsymbol{X}_2 = \boldsymbol{O}.$$

If there exists such a vector $\boldsymbol{\beta}$, then there exist an eigenvector matrix $\boldsymbol{Y} \in \mathcal{C}^{n \times m}$,

$$\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m), \quad \boldsymbol{y}_j \neq 0, \quad j = 1, 2, \ldots, m,$$

and a diagonal matrix, $\boldsymbol{D} = \text{diag}\{\mu_1, \mu_2, \ldots, \mu_m\}$ of the eigenvalues to be assigned, which are such that

$$\boldsymbol{MYD}^2 + (\boldsymbol{C} - \boldsymbol{bf}^T)\boldsymbol{YD} + (\boldsymbol{K} - \boldsymbol{bg}^T)\boldsymbol{Y} = \boldsymbol{O}.$$

So

$$\begin{aligned}\boldsymbol{MYD}^2 + \boldsymbol{CYD} + \boldsymbol{KY} &= \boldsymbol{b\beta}^T(\boldsymbol{\Lambda}_1\boldsymbol{X}_1^T\boldsymbol{MYD} - \boldsymbol{X}_1^T\boldsymbol{KY}) \\ &= \boldsymbol{b\beta}^T\boldsymbol{Z}_1^T \\ &= \boldsymbol{bc}^T,\end{aligned}$$

where $\boldsymbol{Z}_1 = \boldsymbol{DY}^T\boldsymbol{MX}_1\boldsymbol{\Lambda}_1 - \boldsymbol{Y}^T\boldsymbol{KX}_1$ and $\boldsymbol{c} = \boldsymbol{Z}_1\boldsymbol{\beta}$ is a vector that will depend on the scaling chosen for the eigenvectors in $\boldsymbol{Y}$. To obtain $\boldsymbol{Y}$, we can solve in turn for each of the eigenvectors $\boldsymbol{y}_i$ using the equations

$$(\mu_j^2\boldsymbol{M} + \mu_j\boldsymbol{C} + \boldsymbol{K})\boldsymbol{y}_j = \boldsymbol{b}, \quad j = 1, 2, \ldots, m.$$

This corresponds to choosing the vector $\boldsymbol{c} = (1, 1, \ldots, 1)^T$, so, having computed the eigenvectors we could solve the m-square system $\boldsymbol{Z}_1\boldsymbol{\beta} = (1, 1, \ldots, 1)^T$ for $\boldsymbol{\beta}$, and hence determine the vectors $\boldsymbol{f}, \boldsymbol{g}$ using Theorem 4.1. However, there exists an explicit solution for this problem:

Theorem 4.2 *Suppose the open loop quadratic pencil* (12.2) *has eigendecomposition* (12.5) *and that* $\boldsymbol{f}$ and $\boldsymbol{g}$ *are as in* (12.12) with the components β_j *of* $\boldsymbol{\beta}$ *chosen as*

$$\beta_j = \frac{1}{\boldsymbol{b}^T\boldsymbol{x}_j}\frac{\mu_j - \lambda_j}{\lambda_j}\prod_{\substack{i=1 \\ i\neq j}}^{m}\frac{\mu_i - \lambda_j}{\lambda_i - \lambda_j}, \quad j = 1, 2, \ldots, m. \tag{12.13}$$

Then, the closed loop pencil (12.10) *has spectrum* $\{\mu_1, \mu_2, \ldots, \mu_m, \lambda_{m+1}, \ldots, \lambda_{2n}\}$ *and its first m eigenvectors can be scaled to satisfy* $(\mu_j^2\boldsymbol{M} + \mu_j\boldsymbol{C} + \boldsymbol{K})\boldsymbol{y}_j = \boldsymbol{b}$.

Formula (12.13) reveals three conditions (for the existence of $\boldsymbol{\beta}$) that apply to the m eigenvalues which will be replaced, and their associated eigenvectors:

a. no λ_j, $j = 1, 2, \ldots, m$ may vanish,
b. the $\{\lambda_j\}_{j=1}^m$ must be distinct, and
c. $\boldsymbol{b}$ must be not orthogonal to $\boldsymbol{x}_j$, $j = 1, 2, \ldots, m$.

We note that if all the $\{\lambda_j\}_{j=1}^m$ are real, then $\boldsymbol{X}_1$ is real as well. If, in addition, all the $\{\mu_j\}_{j=1}^m$ are real, then $\boldsymbol{\beta}$ and so $\boldsymbol{f}, \boldsymbol{g}$ are also real.

Furthermore, if the set of eigenvalues which are to be replaced is self-conjugate then $\boldsymbol{f}$ and $\boldsymbol{g}$ are real and they specify a solution which can be physically realized. Indeed, the whole calculation can be done in real arithmetic [9].

Finally, we note that any problem in which condition (a) is violated can be handled with a *shift of origin*:

Lemma 4.3 *Let*

$$\boldsymbol{Q}(\lambda) = \lambda^2 \boldsymbol{U} + \lambda \boldsymbol{V} + \boldsymbol{W}, \quad \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W} \in \mathcal{C}^{n \times n}$$

$\boldsymbol{U}$ *invertible, have spectrum* $\{\lambda_j\}_{j=1}^{2n}$. *Then, the pencil*

$$\hat{\boldsymbol{Q}}(\lambda) = \lambda^2 \boldsymbol{U} + \lambda \hat{\boldsymbol{V}} + \hat{\boldsymbol{W}}$$

with

$$\hat{\boldsymbol{V}} = \boldsymbol{V} + 2p\boldsymbol{U}, \quad \hat{\boldsymbol{W}} = \boldsymbol{W} + p\boldsymbol{V} + p^2\boldsymbol{U},$$

scalar p, *has spectrum* $\{\lambda_j - p\}_{j=1}^{2n}$. *If, in addition,* $\boldsymbol{Q}(\lambda)$ *is symmetric definite, then* $\hat{\boldsymbol{Q}}(\lambda)$ *is also symmetric definite*.

12.5 Partial Pole Assignment by Multi-Input Control for the Symmetric Definite Quadratic Pencil

Although the pole assignment problem by single input control and its solution are satisfactory from a theoretical standpoint, one may still encounter some difficulties when using this solution in a practical vibrations context. It may happen that the control force required to relocate the poles is so large that it cannot be implemented in practice without causing an early structural fatigue. This practical difficulty may be overcome by using a *multi-input* control, one in which the vector $\boldsymbol{\beta}$ of (12.8) replaced by a matrix $\boldsymbol{B} \in \mathbb{R}^{n \times m}$ and the scalar $u(t)$ is replaced be a vector $\boldsymbol{u}(t) \in \mathbb{R}^m$. Our differential equation is now

$$\boldsymbol{M}\boldsymbol{v}'' + \boldsymbol{C}\boldsymbol{v}' + \boldsymbol{K}\boldsymbol{v} = \boldsymbol{B}\boldsymbol{u}(t)$$

and we seek $\boldsymbol{F}, \boldsymbol{G} \in \mathbb{R}^{n \times m}$ to assign part or all of the spectrum of (12.2).

In this section we describe methods that are applicable to two slightly different problems. The first deals with the case where we make no extra assumptions about the given matrices $\boldsymbol{C}, \boldsymbol{K}$ and the second concerns that case where $\boldsymbol{C}$ and $\boldsymbol{K}$ have the addtional property that they are non-negative definite. Recall that this extra property ensures that all the eigenvalues of the pencil $\boldsymbol{Q}$ have non-positive real part, something which is important in the stability of dynamic systems. The second method achieves the assignment while ensuring that all the eigenvalues not assigned, even though they may have been moved, still have non-positive real part.

For alternative treatments of this problem which also address the issue of robustness see [4, 5].

12.5.1 Method 1

Problem 5.1 *Given* $2p \le 2n$ self-conjugate numbers $\mu_1, \mu_2, \ldots, \mu_{2p}$, and $\boldsymbol{B} \in \mathbb{R}^{n \times m}$,

Find $\boldsymbol{F}, \boldsymbol{G} \in \mathbb{R}^{n\times m}$ such that the closed-loop system

$$\boldsymbol{Q}_c(\lambda) = \boldsymbol{M}\lambda^2 + (\boldsymbol{C} - \boldsymbol{B}\boldsymbol{F}^T)\lambda + (\boldsymbol{K} - \boldsymbol{B}\boldsymbol{G}^T), \tag{12.14}$$

has spectrum

$$\{\mu_1, \mu_2, \ldots, \mu_{2p}, \lambda_{2p+1}, \ldots, \lambda_{2n}\}.$$

As before, the problem has a computational solution [32] that avoids spillover but an explicit solution for this problem has yet to be found.

Let the eigenvalue and eigenvector matrices of $\boldsymbol{Q}$ be partitioned as in (12.11). The following theorem is the matrix counterpart of Theorem 4.1 and gives conditions on $\boldsymbol{B}$ which ensure that there is no spillover. The multi-input control form (12.15) directly generalises the single-input form (12.12).

Theorem 5.1 *Let*

$$\boldsymbol{F} = \boldsymbol{M}\boldsymbol{X}_1\boldsymbol{\Lambda}_1\boldsymbol{B}, \quad \boldsymbol{G} = -\boldsymbol{K}\boldsymbol{X}_1\boldsymbol{B}, \quad \boldsymbol{B} \in \mathcal{C}^{2p\times m}. \tag{12.15}$$

Then, for any choice of $\boldsymbol{B}$ *we have*

$$\boldsymbol{M}\boldsymbol{X}_2\boldsymbol{\Lambda}_2^2 + (\boldsymbol{C} - \boldsymbol{B}\boldsymbol{F}^T)\boldsymbol{X}_2\boldsymbol{\Lambda}_2 + (\boldsymbol{K} - \boldsymbol{B}\boldsymbol{G}^T)\boldsymbol{X}_2 = \boldsymbol{O}.$$

Any choice of $\boldsymbol{B}$ with $\boldsymbol{F}, \boldsymbol{G}$ chosen thus guarantees that the last $2n - 2p$ eigenpairs $(\boldsymbol{\Lambda_2}, \boldsymbol{X_2})$ are also eigenpairs of the closed loop pencil.

In a development analogous to the vector case we can find [32] the matrix $\boldsymbol{B}$ and from it $\boldsymbol{F}, \boldsymbol{G}$ which solve the problem in one step. An alternative, multi-step solution, is also available which uses the explicit solution (12.13) to the single input case. One application of the method based on Theorem 4.1 leaves the pencil unsymmetric so we cannot use that technique repeatedly. However, it is possible to use the orthogonality relations (12.6) to compute $\boldsymbol{F}$ and $\boldsymbol{G}$ which will do the job [32].

12.5.2 Method 2

We now consider multi-input control systems where the matrices $\boldsymbol{C}, \boldsymbol{K}$ are non-negative definite, thus ensuring that all eigenvalues of the pencil have non-positive real part. The technique of this section [8] assigns part of the spectrum and avoids spillover, not by preserving the eigenvalues which are not assigned, but by ensuring only that the unassigned eigenvalues, even if they have been moved by the assignment, have non-positive real part.

Problem 5.2 Given the pencil (12.2) with $\boldsymbol{M}$ spd, $\boldsymbol{C}, \boldsymbol{K}$ non-negative definite, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$and a self-conjugate set$\{\mu_k\}_{k=1}^{2p}$ of $2p \leq 2n$ *scalars*, Find *matrices* $\boldsymbol{F}$, $\boldsymbol{G} \in \mathbb{R}^{n\times m}$ such that the spectrum of the closed-loop pencil (12.14) contains the set $\{\mu_k\}_{nk=1}^{2p}$and the complementary part of the spectrum has non-positive real part.

We begin by constructing diagonal $p \times n$ matrices

$$\boldsymbol{D}_\alpha = \underset{p}{\big(}\hat{\boldsymbol{D}}_\alpha \quad \underset{n-p}{\boldsymbol{O}}\big), \quad \hat{\boldsymbol{D}}_\beta = (\underset{p}{\boldsymbol{D}_\beta} \quad \underset{n-p}{\boldsymbol{O}},)$$

which give $\lambda^2\boldsymbol{I} + \lambda\boldsymbol{D}_\alpha + \boldsymbol{D}_\beta$ the required eigenvalues. We then compute the Cholesky and SVD factorings

$$\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{M}, \quad \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \boldsymbol{L}^{-1}\boldsymbol{B},$$

$\boldsymbol{U}$ and $\boldsymbol{V}$ orthogonal and $\boldsymbol{S}$ diagonal. Since the spectrum of $\boldsymbol{Q}$ is invariant to multiplication by an invertible matrix, we can write

$$\sigma(\boldsymbol{M},\ \boldsymbol{C} - \boldsymbol{B}\boldsymbol{F}^T,\ \boldsymbol{K} - \boldsymbol{B}\boldsymbol{G}^T) = \sigma(\boldsymbol{I},\ \hat{\boldsymbol{C}} - \boldsymbol{S}\hat{\boldsymbol{F}}^T,\ \hat{\boldsymbol{K}} - \boldsymbol{S}\hat{\boldsymbol{G}}^T),$$

where we have defined

$$\hat{\boldsymbol{C}} \overset{\text{def}}{=} \boldsymbol{U}^T\boldsymbol{L}^{-1}\boldsymbol{C}\boldsymbol{L}^{-T}\boldsymbol{U}, \quad \hat{\boldsymbol{K}} \overset{\text{def}}{=} \boldsymbol{U}^T\boldsymbol{L}^{-1}\boldsymbol{K}\boldsymbol{L}^{-T}\boldsymbol{U}$$
$$\hat{\boldsymbol{F}}^T \overset{\text{def}}{=} \boldsymbol{V}^T\boldsymbol{F}^T\boldsymbol{L}^{-T}\boldsymbol{U}, \quad \hat{\boldsymbol{G}}^T \overset{\text{def}}{=} \boldsymbol{V}^T\boldsymbol{G}^T\boldsymbol{L}^{-T}\boldsymbol{U}.$$

To find $\hat{\boldsymbol{F}}, \hat{\boldsymbol{G}}$ we note that, with the blocking

$$\hat{\boldsymbol{C}} = \begin{pmatrix} \hat{\boldsymbol{C}}_1 \\ \hat{\boldsymbol{C}}_2 \end{pmatrix} \begin{matrix} p \\ n-p \\ n \end{matrix}, \quad \hat{\boldsymbol{K}} = \begin{pmatrix} \hat{\boldsymbol{K}}_1 \\ \hat{\boldsymbol{K}}_2 \end{pmatrix} \begin{matrix} p \\ n-p \\ n \end{matrix}, \quad \boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_1 \\ \boldsymbol{O} \end{pmatrix} \begin{matrix} p \\ n-p \\ p \end{matrix},$$

the choice $\hat{\boldsymbol{F}} = (\hat{\boldsymbol{C}}_1^T - \boldsymbol{D}_\alpha^T)\boldsymbol{S}_1^{-1}$ makes the first block row of $\hat{\boldsymbol{C}} - \boldsymbol{S}\hat{\boldsymbol{F}}^T = (\hat{\boldsymbol{D}}_\alpha \quad \boldsymbol{O})$ and leaves the other $n-p$ rows unchanged. Similarly, the choice $\hat{\boldsymbol{G}} = (\hat{\boldsymbol{K}}_1^T - \boldsymbol{D}_\beta^T)\boldsymbol{S}_1^{-1}$ makes the first block row of $\hat{\boldsymbol{K}} - \boldsymbol{S}\hat{\boldsymbol{G}}^T = (\hat{\boldsymbol{D}}_\beta \quad \boldsymbol{O})$ and leaves the other $n-p$ rows unchanged. Thus

$$\hat{\boldsymbol{Q}}_c(\lambda) = \lambda^2\boldsymbol{I} + \lambda(\hat{\boldsymbol{C}} - \boldsymbol{S}\hat{\boldsymbol{F}}^T) + (\hat{\boldsymbol{K}} - \boldsymbol{S}\hat{\boldsymbol{G}}^T) = \begin{pmatrix} \boldsymbol{Q}_1 & \boldsymbol{O} \\ \boldsymbol{Q}_2 & \boldsymbol{Q}_3 \end{pmatrix} \begin{matrix} p \\ n-p \end{matrix}$$
$$\begin{matrix} \\ p \end{matrix} \quad \begin{matrix} \\ n-p \end{matrix}$$

and $\sigma(\hat{\boldsymbol{Q}}_c(\lambda)) = \sigma(\boldsymbol{Q}_1) \cup \sigma(\boldsymbol{Q}_3)$. In addition $\boldsymbol{Q}_1(\lambda) = \lambda^2\boldsymbol{I}_p + \lambda\hat{\boldsymbol{D}}_\alpha + \hat{\boldsymbol{D}}_\beta$ is diagonal and has the assigned eigenvalues. It's easy to show [8] that $\boldsymbol{Q}_3$ is a positive semi-definite pencil and all its eigenvalues have real part that is non-positive.

12.6 Partial Eigenstructure Assignment by Multi-Input Control for the Symmetric Definite Quadratic Pencil

We now consider the case where the problem is to find all three control matrices $\boldsymbol{F}, \boldsymbol{G}$ and $\boldsymbol{B}$ to replace some eigenpairs of the quadratic pencil (12.2). Assume we have the eigendecomposition (12.5) and the partitioning (12.11).

Problem 6.1 *Given*

a. $\boldsymbol{Q}$ *as in* (12.2) with eigen decomposition (12.5)
b. $\boldsymbol{X}_1$ *and* $\boldsymbol{\Lambda}_1$ *pairwise self-conjugate and partitioned as in* (12.11)
c. $\boldsymbol{Y}_1 \in \mathcal{C}^{n\times m}$, $\boldsymbol{D}_1 \in \mathcal{C}^{m\times m}$, *pairwise self-conjugate such that with*

$$\underset{m}{\boldsymbol{Y}}\,(\underset{m}{\boldsymbol{Y}_1}\ \ \underset{2n-m}{\boldsymbol{X}_2}), \quad \boldsymbol{D} = \begin{pmatrix} \underset{m}{\boldsymbol{D}_1} & \\ & \underset{2n-m}{\boldsymbol{\Lambda_2}} \end{pmatrix}^{\begin{matrix} m \\ 2n-m \end{matrix}}$$

the matrix $\begin{pmatrix} \boldsymbol{Y} \\ \boldsymbol{YD} \end{pmatrix}$ *is invertible*,
Find $\boldsymbol{B}, \boldsymbol{F}, \boldsymbol{G} \in \mathbb{R}^{n\times m}$ *which satisfy*

$$\boldsymbol{MYD}^2 + (\boldsymbol{C} - \boldsymbol{BF}^T)\boldsymbol{YD} + (\boldsymbol{K} - \boldsymbol{BG}^T)\boldsymbol{Y} = \boldsymbol{O}. \tag{12.16}$$

The process of finding the $\boldsymbol{F}, \boldsymbol{G}$ and $\boldsymbol{B}$ which will assign these m eigenvalues and their associated eigenvectors is done in two stages:

a. First, determine matrices $\hat{\boldsymbol{B}}, \hat{\boldsymbol{F}}$ and $\hat{\boldsymbol{G}}$ which are generally complex and which satisfy

$$\boldsymbol{MYD}^2 + (\boldsymbol{C} - \hat{\boldsymbol{B}}\hat{\boldsymbol{F}}^T)\boldsymbol{YD} + (\boldsymbol{K} - \hat{\boldsymbol{B}}\hat{\boldsymbol{G}}^T)\boldsymbol{Y} = \boldsymbol{O}. \tag{12.17}$$

b. Second, from $\hat{\boldsymbol{B}}, \hat{\boldsymbol{F}}$ and $\hat{\boldsymbol{G}}$ find real $\boldsymbol{B}, \boldsymbol{F}$, and $\boldsymbol{G}$ such that $\boldsymbol{BF}^T = \hat{\boldsymbol{B}}\hat{\boldsymbol{F}}^T$ and $\boldsymbol{BG}^T = \hat{\boldsymbol{B}}\hat{\boldsymbol{G}}^T$.

The first stage proceeds as follows. Let $\boldsymbol{W} \in \mathcal{C}^{m\times p}$, $p \geq m$ have pseudoinverse $\boldsymbol{W}^+ \in \mathcal{C}^{p\times m}$ such that $\boldsymbol{WW}^+ = \boldsymbol{I} \in \mathbb{R}^{m\times m}$. If $\tilde{\boldsymbol{B}}$, $\tilde{\boldsymbol{F}}$ and $\tilde{\boldsymbol{G}}$ is a solution of Problem 6.1 then

$$\hat{\boldsymbol{B}} = \tilde{\boldsymbol{B}}\boldsymbol{W}, \hat{\boldsymbol{F}} = \tilde{\boldsymbol{F}}(\boldsymbol{W}^+)^T \quad \hat{\boldsymbol{G}} = \tilde{\boldsymbol{G}}(\boldsymbol{W}^+)^T$$

is another solution because $\tilde{\boldsymbol{B}}\tilde{\boldsymbol{F}}^T = \hat{\boldsymbol{B}}\hat{\boldsymbol{F}}^T$ and $\tilde{\boldsymbol{B}}\tilde{\boldsymbol{G}}^T = \hat{\boldsymbol{B}}\hat{\boldsymbol{G}}^T$. Now, if $\tilde{\boldsymbol{B}}, \tilde{\boldsymbol{F}}$ and $\tilde{\boldsymbol{G}}$ is a solution and satisfies (12.16) then

$$\boldsymbol{MY}_1\boldsymbol{D}_1^2 + (\boldsymbol{C} - \tilde{\boldsymbol{B}}\tilde{\boldsymbol{F}}^T)\boldsymbol{Y}_1\boldsymbol{D}_1 + (\boldsymbol{K} - \tilde{\boldsymbol{B}}\tilde{\boldsymbol{G}}^T)\boldsymbol{Y}_1 = \boldsymbol{O}$$

and it is evident that we can take $\hat{\boldsymbol{B}}$ and $\boldsymbol{W}$ as

$$\underbrace{\boldsymbol{MY}_1\boldsymbol{D}_1^2 + \boldsymbol{CY}_1\boldsymbol{D}_1 + \boldsymbol{KY}_1}_{\hat{\boldsymbol{B}}} = \tilde{\boldsymbol{B}}\,\underbrace{(\tilde{\boldsymbol{F}}^T\boldsymbol{Y}_1\boldsymbol{D}_1 + \tilde{\boldsymbol{G}}^T\boldsymbol{Y}_1)}_{\boldsymbol{W}}. \tag{12.18}$$

provided that $\boldsymbol{W}$ is invertible. Then for some $\hat{\boldsymbol{F}}$ and $\hat{\boldsymbol{G}}$ this $\hat{\boldsymbol{B}} = \tilde{\boldsymbol{B}}\boldsymbol{W}$ is an admissable solution. Equation (12.17) blocked as

$$\boldsymbol{M}(\boldsymbol{Y}_1,\boldsymbol{X}_2)+\begin{pmatrix}\boldsymbol{D}_1^2 & \\ & \boldsymbol{\Lambda}_2^2\end{pmatrix}+(\boldsymbol{C}-\hat{\boldsymbol{B}}\hat{\boldsymbol{F}}^T)(\boldsymbol{Y}_1,\boldsymbol{X}_2)\begin{pmatrix}\boldsymbol{D}_1 & \\ & \boldsymbol{\Lambda}_2\end{pmatrix}+(\boldsymbol{K}-\hat{\boldsymbol{B}}\hat{\boldsymbol{G}}^T)\boldsymbol{Y}=\boldsymbol{O}$$

has first block row

$$\boldsymbol{M}\boldsymbol{Y}_1\boldsymbol{D}_1^2+\boldsymbol{C}\boldsymbol{Y}_1\boldsymbol{D}_1+\boldsymbol{K}\boldsymbol{Y}_1-\hat{\boldsymbol{B}}\hat{\boldsymbol{F}}^T\boldsymbol{Y}_1\boldsymbol{D}_1-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{G}}^T\boldsymbol{Y}_1=\boldsymbol{O}$$

and, taking the definition of $\hat{\boldsymbol{B}}$ from (12.18), this gives $\hat{\boldsymbol{B}}\left(\boldsymbol{I}-\tilde{\boldsymbol{F}}^T\boldsymbol{Y}_1\boldsymbol{D}_1-\tilde{\boldsymbol{G}}^T\boldsymbol{Y}_1\right)=\boldsymbol{O}$. The full rank of $\tilde{\boldsymbol{B}}$ implies that

$$\hat{\boldsymbol{F}}^T\boldsymbol{Y}_1\boldsymbol{D}_1+\hat{\boldsymbol{G}}^T\boldsymbol{Y}_1=\boldsymbol{I}. \tag{12.19}$$

The following theorem is a variation of Theorem 5.1.

Theorem 6.1 *For any* $\boldsymbol{B}\in\mathcal{C}^{m\times m}$, $\hat{\boldsymbol{F}}=\boldsymbol{M}\boldsymbol{X}_1\Lambda_1\boldsymbol{B}$ *and* $\tilde{\boldsymbol{G}}=-\boldsymbol{K}\boldsymbol{X}_1\boldsymbol{B}$ *satisfy*

$$\boldsymbol{M}\boldsymbol{X}_2\boldsymbol{\Lambda}_2^2+(\boldsymbol{C}-\hat{\boldsymbol{B}}\hat{\boldsymbol{F}}^T)\boldsymbol{X}_2\boldsymbol{\Lambda}_2+(\boldsymbol{K}-\hat{\boldsymbol{B}}\hat{\boldsymbol{G}}^T)\boldsymbol{X}_2=\boldsymbol{O}.$$

Putting these expressions for $\hat{\boldsymbol{F}},\hat{\boldsymbol{G}}$ into (12.19) gives

$$\boldsymbol{B}=(\boldsymbol{\Lambda}_1\boldsymbol{X}_1^T\boldsymbol{M}\boldsymbol{Y}_1\boldsymbol{D}_1-\boldsymbol{X}_1^T\boldsymbol{K}\boldsymbol{Y}_1)^{-1}$$

from which $\hat{\boldsymbol{F}}$ and $\hat{\boldsymbol{G}}$ can easily be found using the expressions in Theorem 6.1.

It is easy to show that although $\hat{\boldsymbol{B}}$, $\hat{\boldsymbol{F}}$, $\hat{\boldsymbol{G}}$ are generally complex, the products $\hat{\boldsymbol{B}}\hat{\boldsymbol{F}}^T$ and $\hat{\boldsymbol{B}}\hat{\boldsymbol{G}}^T$ are always pure real. The second stage of the solution, that of finding real $\boldsymbol{B},\boldsymbol{F},\boldsymbol{G}$ from the generally complex $\hat{\boldsymbol{B}}$, $\hat{\boldsymbol{F}}$, $\hat{\boldsymbol{G}}$ can be carried out using any one of several factorings of the form $\boldsymbol{L}\boldsymbol{R}=\boldsymbol{H},\boldsymbol{L}\in\mathbb{R}^{n\times m},\boldsymbol{R}\in\mathbb{R}^{m\times 2n}$ of the composite $n\times 2n$ product matrix

$$\boldsymbol{H}=\hat{\boldsymbol{B}}\left[\hat{\boldsymbol{F}}^T|\hat{\boldsymbol{G}}^T\right].$$

We take $\boldsymbol{B}$ to be the $\boldsymbol{L}$ matrix, the first n columns of $\boldsymbol{R}$ to be $\boldsymbol{F}^T$ and the last n columns to be $\boldsymbol{G}^T$. The two factorings which immediately come to mind are the truncated QR and compact SVD [15]. Details can be found in Datta et al. [10].

12.7 Pole Assignment for the Symmetric Definite Quadratic Pencil by Affine Sums

In this section we consider the pole assignment problem by means of linear combinations of matrix sets which span the space of matrices in which our target matrix lies. Two methods, in which the solutions are found by zeroing two

different nonlinear functions, are described. We motivate the methods by first considering the standard eigenvalue problem which arises from a set of first order, time differential equations

$$x' = \boldsymbol{A}x, \quad \boldsymbol{A} = \boldsymbol{A}^T \in \mathbb{R}^{n\times n}, \quad x \in \mathbb{R}^n.$$

This system's behaviour is characterized by eigenpairs $(\mu_k, \boldsymbol{x}_k), \lambda_k$ scalar, $\mathbf{0} \neq \boldsymbol{x}_\mathbf{k} \in \mathbb{R}^\mathbf{n}$, which satisfy $(\boldsymbol{A} - \mu \boldsymbol{I})\boldsymbol{x} = \mathbf{0}$. The *Affine Inverse Eigenvalue Problem* requires us to determine the vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$ of coefficients, if they exist, which define the matrix

$$\boldsymbol{A} = \boldsymbol{A}_0 + \sum_{k=1}^{n} \alpha_k \boldsymbol{A}_k,$$

from its spectrum. The elements $\boldsymbol{A}_k \in \mathbb{R}^{n\times n}$, $k = 0, 1, 2, \ldots, n$ in the linear combination comprise the *affine family*. As an example, The eigenvalues of the matrix

$$\begin{pmatrix} \kappa_1 + \kappa_2 & -\kappa_2 & & & \\ -\kappa_2 & \kappa_2 + \kappa_3 & -\kappa_3 & & \\ & -\kappa_3 & \kappa_3 + \kappa_4 & -\kappa_4 & \\ & & \ddots & \ddots & \ddots \\ & & & -\kappa_n & \kappa_n \end{pmatrix}$$

determine the natural frequencies of a certain n-degrees-of-freedom mass-spring system with spring constants $\kappa_i > 0$ and unit masses. An affine family suitable for such a 4×4 problem might consist of the set $\boldsymbol{A}_{0,1,2,3}$ defined by

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}. \tag{12.20}$$

We note that the matrices in the affine set are real and symmetric and tridiagonal. Hence the spectrum of $\boldsymbol{A}$ is real.

Friedland et al. [20] consider several Newton-based algorithms for solving the affine inverse eigenvalue problem. Starting with some initial estimate of the solution $\boldsymbol{\alpha}^{(0)} = \left(\alpha_1^{(0)}, \alpha_2^{(0)}, \ldots, \alpha_n^{(0)}\right)^T$, the methods find the zero $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$ of $\boldsymbol{f}(\boldsymbol{\alpha}) = (\lambda_1(\boldsymbol{\alpha}) - \mu_1, \lambda_2(\boldsymbol{\alpha}) - \mu_2, \ldots, \lambda_n(\boldsymbol{\alpha}) - \mu_n)^T$ where $\{\mu_k\}_{k=1}^n$ is the *ordered* target (real) spectrum, and $\lambda_j\left(\boldsymbol{\alpha}^{(m)}\right)$ is the j-th eigenvalue of the similarly ordered spectrum of

$$\boldsymbol{A} = \boldsymbol{A}_0 + \sum_{k=1}^{n} \alpha_k^{(m)} \boldsymbol{A}_k.$$

A correction vector $\boldsymbol{\xi}^{(m)}$ to $\boldsymbol{\alpha}^{(m)}$ is found from $\boldsymbol{J}^{(m)}\boldsymbol{\xi}^{(m)} = -\boldsymbol{f}(m)$ where the Jacobian is

$$\boldsymbol{J}^{(m)} = \begin{pmatrix} \boldsymbol{x}_1^T\boldsymbol{A}_1\boldsymbol{x}_1 & \boldsymbol{x}_1^T\boldsymbol{A}_2\boldsymbol{x}_1 & \cdots & \boldsymbol{x}_1^T\boldsymbol{A}_n\boldsymbol{x}_1 \\ \boldsymbol{x}_2^T\boldsymbol{A}_1\boldsymbol{x}_2 & \boldsymbol{x}_2^T\boldsymbol{A}_2\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_2^T\boldsymbol{A}_n\boldsymbol{x}_2 \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{x}_n^T\boldsymbol{A}_1\boldsymbol{x}_n & \boldsymbol{x}_n^T\boldsymbol{A}_2\boldsymbol{x}_n & \cdots & \boldsymbol{x}_n^T\boldsymbol{A}_n\boldsymbol{x}_n \end{pmatrix} \tag{12.21}$$

and $\boldsymbol{x}_k$ is the eigenvector associated with $\lambda_k\left(\boldsymbol{\alpha}^{(m)}\right)$. The next iterate is found from $\boldsymbol{\alpha}^{(m+1)} = \boldsymbol{\xi}^{(m)} + \boldsymbol{\alpha}^{(m)}$ and the process continues until convergence or divergence occurs.

For the quadratic problem we are interested in

Problem 7.1 *Given*

a. $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ *spd*
b. $\{\boldsymbol{C}_k\}_{k=0}^n, \{\boldsymbol{K}_k\}_{k=0}^n, \boldsymbol{C}_k, \boldsymbol{K}_k \in \mathbb{R}^{n\times n}$, symmetric, linearly independent,
c. $S = \{\mu_k\}_{k=1}^{2n}$, a self-conjugate set of scalars.

Define $\boldsymbol{C} = \boldsymbol{C}_0 + \sum_{k=1}^n \alpha_k\boldsymbol{C}_k$ and, $\boldsymbol{K} = \boldsymbol{K}_0 + \sum_{k=1}^n \beta_k\boldsymbol{K}_k$.

Find real scalars$\{\alpha_k, \beta_k\}_{k=1}^n$, if they exist, such that the pencil (12.2) has spectrum S.

Affine methods are attractive because, unlike some other methods, they preserve structural properties like symmetry, bandedness and sparseness. We now describe two affine methods of pole assignment for the quadratic symmetric definite pencil. In each a Newton method is used to solve the system of nonlinear equations that give the affine coefficients which achieve the assignment.

12.7.1 Affine Method 1

Denote the vector of target eigenvalues by $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_{2n})^T$ and the vectors of the unknown coefficients by $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)^T$. denote also the vector of open loop eigenvalues by $\lambda(\boldsymbol{\alpha}, \boldsymbol{\beta}) = (\lambda_1(\boldsymbol{\alpha}, \boldsymbol{\beta}), \lambda_2(\boldsymbol{\alpha}, \boldsymbol{\beta}), \ldots, \lambda_{2n}(\boldsymbol{\alpha}, \boldsymbol{\beta}))^T$. We will assume that

- the target spectrum S defines a solution,
- there exists an open neighbourhood of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ where $\boldsymbol{Q}(\lambda)$ has eigenvalues and eigenvectors which are analytic,
- the target eigenvalues and eigenvectors for the current $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ iterates have the same number of reals and complex pairs.

Let us assume that the target real eigenvalues are ordered $\mu_1 \le \mu_2 \le \cdots \le \mu_r$ and that $2n - r = 2c$ complex eigenvalues are paired

$$\begin{array}{ll} \mu_{r+1} = \rho_1 + i\eta_1, & \mu_{r+2} = \rho_1 - i\eta_1, \\ \mu_{r+3} = \rho_2 + i\eta_2, & \mu_{r+4} = \rho_2 - i\eta_2, \\ \vdots & \vdots \\ \mu_{2n-1} = \rho_c + i\eta_c, & \mu_{2n} = \rho_c - i\eta_c. \end{array}.$$

Suppose also that the eigenvalues to be replaced are similarly ordered $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_r$

$$\begin{array}{ll} \lambda_{r+1} = \phi_1 + i\psi_1, & \lambda_{r+2} = \phi_1 - i\psi_1, \\ \lambda_{r+3} = \phi_2 + i\psi_2, & \lambda_{r+4} = \phi_2 - i\psi_2, \\ \vdots & \vdots \\ \lambda_{2n-1} = \phi_c + i\psi_c, & \lambda_{2n} = \phi_c - i\psi_c \end{array}$$

as are their pure real eigenvectors $\boldsymbol{z}_i$, $i = 1, 2, \ldots, r$ and their complex eigenvectors

$$\begin{array}{ll} z_{r+1} = \boldsymbol{x}_1 + i\boldsymbol{y}_1, & z_{r+2} = \boldsymbol{x}_1 - i\boldsymbol{y}_1, \\ z_{r+3} = \boldsymbol{x}_2 + i\boldsymbol{y}_2, & z_{r+4} = \boldsymbol{x}_2 - i\boldsymbol{y}_2, \\ \vdots & \vdots \\ z_{2n-1} = \boldsymbol{x}_c + i\boldsymbol{y}_c, & z_{2n} = \boldsymbol{x}_c - i\boldsymbol{y}_c. \end{array}$$

A Newton method for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ which parallels the method described earlier for the inverse standard eigenvalue problem zeros the function

$$\boldsymbol{f}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \begin{pmatrix} \lambda_1(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \mu_1 \\ \lambda_2(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \mu_2 \\ \vdots \\ \lambda_r(\boldsymbol{\alpha}, \boldsymbol{\beta})' - \mu_r \\ \hline \phi_1(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \rho_1 \\ \phi_2(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \rho_2 \\ \vdots \\ \phi_c(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \rho_c \\ \hline \psi_1(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \eta_1 \\ \psi_2(\alpha, \beta) - \eta_2 \\ \vdots \\ \psi_c(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \eta_c \end{pmatrix}$$

We now derive the Jacobian for this function. Every eigenpair λ_i, z_i satisfies

$$z_i^T(\lambda_i^2 \boldsymbol{M} + \lambda_i C + \boldsymbol{K})z_i = 0. \tag{12.22}$$

Denote a derivative with respect to either α_j or β_j by a dot. Differentiating (12.22) gives

$$2\dot{\boldsymbol{z}}_i^T(\lambda_i^2\boldsymbol{M}+\lambda_i\boldsymbol{C}+\boldsymbol{K})\boldsymbol{z}_i+\boldsymbol{z}_i^T(2\lambda_i\dot{\lambda}_i\boldsymbol{M}+\dot{\lambda}_i\boldsymbol{C}+\lambda_i\dot{\boldsymbol{C}}+\dot{\boldsymbol{K}})\boldsymbol{z}_i=0$$

which simplifies immediately to

$$\boldsymbol{z}_i^T(\dot{\lambda}_i(2\lambda_i\boldsymbol{M}+\boldsymbol{C})+\lambda_i\dot{\boldsymbol{C}}+\dot{\boldsymbol{K}})\boldsymbol{z}_i=0.$$

Isolating $\dot{\lambda}$ we gives

$$\dot{\lambda}_i=-\frac{\boldsymbol{z}_i^T(\lambda_i\dot{\boldsymbol{C}}+\dot{\boldsymbol{K}})\boldsymbol{z}_i}{\boldsymbol{z}_i^T(2\lambda\boldsymbol{M}+\boldsymbol{C})\boldsymbol{z}_i}.$$

Since

$$\frac{\partial\boldsymbol{C}}{\partial\alpha_j}=\boldsymbol{C}_j,\quad \frac{\partial\boldsymbol{C}}{\partial\beta_j}=\frac{\partial\boldsymbol{K}}{\partial\alpha_j}=\boldsymbol{O},\quad \frac{\partial\boldsymbol{K}}{\partial\beta_j}=\boldsymbol{K}_j,$$

we get, provided the denominators do not vanish,

$$\frac{\partial\lambda_i}{\partial\alpha_j}=-\frac{\lambda_i\boldsymbol{z}_i^T\boldsymbol{C}_j\boldsymbol{z}_i}{\boldsymbol{z}_i^T(2\lambda_i\boldsymbol{M}+\boldsymbol{C})\boldsymbol{z}_i},\quad\text{and}\quad \frac{\partial\lambda_i}{\partial\beta_j}=-\frac{\boldsymbol{z}_i^T\boldsymbol{K}_j\boldsymbol{z}_i}{\boldsymbol{z}_i^T(2\lambda_i\boldsymbol{M}+\boldsymbol{C})\boldsymbol{z}_i}.$$

The Newton method is now

$$\boldsymbol{J}\left(\boldsymbol{\alpha}^{(k)},\boldsymbol{\beta}^{(k)}\right)\left(\begin{pmatrix}\boldsymbol{\alpha}^{(k+1)}\\ \boldsymbol{\beta}^{(k+1)}\end{pmatrix}-\begin{pmatrix}\boldsymbol{\alpha}^{(k)}\\ \boldsymbol{\beta}^{(k)}\end{pmatrix}\right)=-\boldsymbol{f}\left(\boldsymbol{\alpha}^{(k)},\boldsymbol{\beta}^{(k)}\right).$$

For details see [18].

Some simple examples illustrate the technique. All calculations here were performed in Matlab with IEEE Standard Double Precision arithmetic ie. using $\epsilon\approx 2\times 10^{-16}$. All numbers are correctly rounded to the number of figures shown.

Example 7.1 Let $n=5,\boldsymbol{M}=\boldsymbol{I}$, and let the affine family be

$$\boldsymbol{C}_0=\begin{pmatrix}10&-10&0&0&0\\-10&18&-8&0&0\\0&-8&12&-4&0\\0&0&-4&12&-8\\0&0&0&-8&11\end{pmatrix},\quad \boldsymbol{K}_0=\begin{pmatrix}10&-10&0&0&0\\-10&18&-8&0&0\\0&-8&12&-4&0\\0&0&-4&12&-8\\0&0&0&-8&11\end{pmatrix}.$$

together with the rank-2, symmetric, tridiagonal elements

a. $\boldsymbol{C}_k=\boldsymbol{K}_k=(\boldsymbol{e}_k-\boldsymbol{e}_{k+1})(\boldsymbol{e}_k-\boldsymbol{e}_{k+1})^T,\quad k=1,2,\ldots,n-1,$
b. $\boldsymbol{C}_n=\boldsymbol{K}_n=\boldsymbol{e}_n\boldsymbol{e}_n^T.$

These elements are of the same type as those shown in (12.20). For

$$\boldsymbol{\alpha}=-\boldsymbol{\beta}=(-1,\quad 1,\quad -1,\quad 1,\quad -1)^T \tag{12.23}$$

Table 12.2 Eigenvalues of $\boldsymbol{Q}$ with $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ defined by (12.23)

$\boldsymbol{\lambda}(\boldsymbol{\alpha}, \boldsymbol{\beta})$
-26.07397
-18.47433
-8.91370
-2.48789
-1.11603
-0.36370
$-1.69234 - 2.43496i$
$-1.69234 + 2.43496i$
$-0.09285 - 0.72845i$
$-0.09285 + 0.72845i$

Table 12.3 A second solution which assigns the eigenvalues in Table 12.2

$\boldsymbol{\alpha}^{(0)}$	$\boldsymbol{\beta}^{(0)}$	$\boldsymbol{\alpha}^{(8)}$	$\boldsymbol{\beta}^{(8)}$
-0.54550	1.08070	-0.87850	1.11642
1.29810	-0.58530	0.89416	-1.08588
-0.83550	1.47810	-0.99899	1.05144
1.23910	-0.70220	0.99300	-1.16071
-0.70140	1.01440	-1.00968	1.01043

the pencil $\boldsymbol{Q}$ has eigenvalues shown in Table 12.2. Using the starting values $\boldsymbol{\alpha} = -\boldsymbol{\beta} = (-1.2, 1.4, -1.6, 1.8, 2)^T$ to assign the eigenvalues in Table 12.2, the Newton method finds the solution (12.23) with the characteristic quadratic convergence and after 9 iterations we get $\| \boldsymbol{\lambda}\left(\boldsymbol{\alpha}^{(9)}, \boldsymbol{\beta}^{(9)}\right) - \boldsymbol{\lambda}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \|_2 < 10^{-14}$. Table 12.3 shows a second starting value and solution (of the many possible). This solution is obtained to the same accuracy as the first solution in 8 iterations.

12.7.2 Affine Method 2

A difficulty with the method in Sect. 12.7 is that it requires an uncomfortable assumption: that the number of real and complex pairs of eigenvalues remain the same throughout the iteration process. However, these can change from one iteration to the next even though after some point close to convergence they will remain the same.

The crucial element is the ordering of the eigenvalues. In the real, symmetric, inverse standard eigenvalue problem there is a natural ordering of the (always real) eigenvalues at every step. We are thus able to pair $\lambda_k(\boldsymbol{\alpha}^m)$ to μ_k correctly in $\boldsymbol{f}^m$ at each iterative step.

In principle the algorithm described at the start of Sect. 12.7 for the real, symmetric inverse standard eigenvalue problem above can be extended to solve

affine inverse eigenvalue problems associated with non-symmetric matrices. The Jacobian matrix elements of (12.21) are replaced by

$$\frac{\partial \lambda_i\left(\boldsymbol{\alpha}^{(m)}\right)}{\partial \alpha_j} = \frac{\boldsymbol{y}_i^T \boldsymbol{A}_j \boldsymbol{x}_i}{\boldsymbol{y}_i^T \boldsymbol{x}_i}$$

where $\boldsymbol{y}_i$ here is the ith left eigenvector of $\boldsymbol{A}$. The algorithm now can frequently fail however since the eigenvalues of this (real) $\boldsymbol{A}$ are generally complex and there is no natural way to order the eigenvalues consistently throughout the iterations. A simple example illustrates the problem. Consider the affine sum

$$\boldsymbol{A}_0 + \delta \boldsymbol{A}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -2 \end{pmatrix} + \delta \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & -2 & 1 \\ -4 & 0 & -5 & 2 \end{pmatrix}.$$

a. For $\delta = 0$ this affine sum has eigenvalues $\pm 1, \pm 2$.
b. For $\delta = 1$ it has eigenvalues $\pm i, \pm 2i$.

The bifurcation point where the four real eigenvalues become two complex conjugate pairs is at $\delta = 1/(1+\sqrt{3})$. Any labelling scheme based on an ordering which is used for the real eigenvalues when $\delta < 1/(1+\sqrt{3})$ has no natural extension once $\delta > 1/(1+\sqrt{3})$.

A technique for overcoming this problem [17] of ordering is based on zeroing the function

$$f_i(\boldsymbol{\alpha}) = \det\left(\boldsymbol{A}_0 - \mu_i \boldsymbol{I}_n + \sum_{k=1}^{n} \alpha_k \boldsymbol{A}_k\right), \quad i = 1, 2, \ldots, n,$$

rather than $f_i(\boldsymbol{\alpha}) = \lambda_i(\boldsymbol{\alpha}) - \mu_i, i = 1, 2, \ldots, n$. Since no pairing of the eigenvalues is needed, the method is applicable to

- non-symmetric affine inverse linear eigenvalue problems with complex eigenvalues,
- non-symmetric inverse generalized eigenvalue problems, and
- inverse quadratic eigenvalue problems and higher order matrix polynomial inverse eigenvalue problems.

The *regular* pencil $\boldsymbol{P}(\lambda) = \lambda \boldsymbol{A} + \boldsymbol{B}$ ($\det(P(\lambda))$ not identically 0) has as its eigenvalues

a. the zeros of $p_k(\lambda) = \det(\boldsymbol{P}(\lambda))$ and
b. ∞ with multiplicity $n - k$ if $k < n$.

If $\boldsymbol{P}(\lambda)$ is regular, there is a generalized Schur decomposition [14] which gives $\boldsymbol{Q}, \boldsymbol{R} \in C^{n \times n}$, unitary, such that

$$\boldsymbol{Q}\boldsymbol{P}(\lambda)\boldsymbol{R} = \lambda\boldsymbol{T}_A + \boldsymbol{T}_B,$$

$\boldsymbol{T}_A, \boldsymbol{T}_B \in \mathcal{C}^{n\times n}$ upper triangular. Quotients of the diagonal elements $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ of $\boldsymbol{T}_A$ and $\boldsymbol{T}_B$, define the eigenvalues of the pencil. Importantly, the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be chosen so that

- (a_i, b_i) are in any order on the diagonals of $\boldsymbol{T}_A$ and $\boldsymbol{T}_B$ and
- $\det(\boldsymbol{Q}) = \det(\boldsymbol{R}) = 1$.

Now, $\det(\boldsymbol{P}(\lambda)) = \prod_{j=1}^{n}(\lambda a_j + b_j)$. If $a_i \neq 0 \neq b_i$ then we can differentiate to get

$$\frac{\partial}{\partial\lambda}\det(\lambda\boldsymbol{A} + \boldsymbol{B}) = \sum_{i=1}^{n} a_i \prod_{\substack{j=1\\ j\neq i}}^{n}(\lambda a_j - b_j).$$

Otherwise, if $a_1, a_2, \ldots, a_r$ are nonzero and $a_{r+1} = a_{r+2} = \cdots = a_n = 0$, then we have (none of the b_i can vanish by the assumption of regularity)

$$\frac{\partial}{\partial\lambda}\det(\lambda\boldsymbol{A} + \boldsymbol{B}) = \left(\prod_{i=r+1}^{n} b_i\right)\sum_{i=1}^{r} a_i \prod_{\substack{j=1\\ j\neq i}}^{n}(\lambda a_j - b_j).$$

We can therefore use Newton's method to solve for the coefficients $\{\alpha_j\}_{j=1}^{n}$ which zero the function

$$\boldsymbol{f}(\boldsymbol{\alpha}) = \begin{pmatrix} f_1(\boldsymbol{\alpha}) \\ f_2(\boldsymbol{\alpha}) \\ \vdots \\ f_n(\boldsymbol{\alpha}) \end{pmatrix} = \begin{pmatrix} \det(\boldsymbol{A}_0 - \mu_1\boldsymbol{I}_n + \sum_{k=1}^{n}\alpha_k\boldsymbol{A}_k) \\ \det(\boldsymbol{A}_0 - \mu_2\boldsymbol{I}_n + \sum_{k=1}^{n}\alpha_k\boldsymbol{A}_k) \\ \vdots \\ \det(\boldsymbol{A}_0 - \mu_n\boldsymbol{I}_n + \sum_{k=1}^{n}\alpha_k\boldsymbol{A}_k) \end{pmatrix}$$

because we can get the Jacobian elements $[\boldsymbol{J}]_{ij} = \partial f_i(\boldsymbol{\alpha})/\partial\alpha_j$ by setting $\lambda = \alpha_j$, $\boldsymbol{A} = \boldsymbol{A}_j$ and

$$\boldsymbol{B} = \boldsymbol{A}_0 - \mu_i\boldsymbol{I}_n + \sum_{\substack{k=1\\ k\neq j}}^{n}\alpha_k\boldsymbol{A}_k.$$

This is consistent with $\lambda\boldsymbol{A} + \boldsymbol{B} = f_i(\boldsymbol{\alpha})$.

These expressions appear computationally costly but in many situations the $\boldsymbol{A}_j$ have rank one or perhaps two and this reduces the computational cost significantly. In addition, there is no need to compute eigenvectors at each stage, as is normally required [29].

A Newton method based on these results takes as input an affine set $\{\boldsymbol{A}_k\}_{k=0}^{n}$, an initial estimate $\alpha^{(0)}$ and a target spectrum $S = \{\mu_k\}_{k=1}^{n}$ and finds, if the process converges, the affine coefficients $\boldsymbol{\alpha}$ such that $\sigma(\boldsymbol{A}_0 + \sum_{k=1}^{n}\alpha_k\boldsymbol{A}_k) = S$. We describe

the algorithm only for the step in which the Jacobian matrix is computed (see [17] for details).

a. For each $i = 1, 2, \ldots, n$
   i. Compute $\det(\boldsymbol{H})$, $\boldsymbol{H} = \boldsymbol{A}_0 - \mu_i \boldsymbol{I} + \sum_{k=1}^{n} \alpha_k^{(m)} \boldsymbol{A}_k$.
   ii. for each $k = 1, 2, \ldots, n$

1. Compute $\boldsymbol{B} = \boldsymbol{A}_0 - \mu_i \boldsymbol{I}_n + \sum_{\substack{j=1 \\ j \neq k}}^{n} \alpha_j \boldsymbol{A}_j$.
2. Use the QZ algorithm to get unitary $\boldsymbol{Q}, \boldsymbol{R}$, $\det(\boldsymbol{Q}) = \det(\boldsymbol{R}) = 1$ which simultaneously triangularize $\boldsymbol{A}_k, \boldsymbol{B}$
3. Reorder (a_i, b_i) so that $a_{r+1} = a_{r+2} = \cdots = a_n = 0$.
4. $[\boldsymbol{J}(\boldsymbol{\alpha}^m)]_{ik} = \left(\prod_{i=r+1}^{n} b_i\right) \sum_{i=1}^{r} a_i \prod_{\substack{j=1 \\ j \neq i}}^{n} \left(\alpha_k^{(m)} a_j - b_j\right)$.

Remarks

- Any factoring from which the determinant can be easily computed will suffice.
- Matrix $\boldsymbol{H}$ changes only slightly between values of i so a factoring based on updates would save considerable effort.
- The Jacobian calculation is well suited to parallel computation.
- Quite good heuristics for starting values exist.

12.7.2.1 Inverse Eigenvalue Problems for Higher Degree Matrix Polynomials

The method of Sect. 12.7 is applicable to the solution of inverse eigenvalue problems with higher degree matrix polynomials. To illustrate the technique we show how it can be used on quadratic matrix polynomial problems.

We define $f_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$, $i = 1, 2, 3, \ldots, 2n$ by

$$f_i(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \det\left(\mu_i^2 \boldsymbol{M} + \mu_i \left(\boldsymbol{C}_0 + \sum_{k=1}^{n} \alpha_k \boldsymbol{C}_k\right) + \boldsymbol{K}_0 + \sum_{k=1}^{n} \beta_k \boldsymbol{K}_k\right).$$

The $2n \times 2n$ Jacobian for f_i has two $2n \times n$ blocks $\boldsymbol{J}\left(\boldsymbol{\alpha}^{(m)}, \boldsymbol{\beta}^{(m)}\right) = \begin{pmatrix} \boldsymbol{J}_1 & \boldsymbol{J}_2 \end{pmatrix}$ where $\boldsymbol{J}_1$ has partial derivatives of $\boldsymbol{f}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ with respect to $(\boldsymbol{\alpha})$ and $\boldsymbol{J}_2$ has the partial derivatives of $\boldsymbol{f}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$. We replace the matrix $\boldsymbol{H}$ of $((a)i)$ by

$$\boldsymbol{H} = \mu_i^2 \boldsymbol{M} + \mu_i \left(\boldsymbol{C}_0 + \sum_{k=1}^{n} \alpha_k C_k\right) + \boldsymbol{K}_0 + \sum_{k=1}^{n} \beta_k \boldsymbol{K}_k$$

and the matrix $\boldsymbol{B}$ of step $(a)ii.1.$ is replaced by

$$\boldsymbol{B} = \mu_i^2 \boldsymbol{M} + \mu_i \left(\boldsymbol{C}_0 + \sum_{\substack{j=1 \\ j \neq k}}^{n} \alpha_j \boldsymbol{C}_j \right) + \boldsymbol{K}_0 + \sum_{j=1}^{n} \beta_j \boldsymbol{K}_j$$

when computing $\boldsymbol{J}_1$ and by

$$\boldsymbol{B} = \mu_i^2 \boldsymbol{M} + \mu_i \left(\boldsymbol{C}_0 + \sum_{j=1}^{n} \alpha_j \boldsymbol{C}_j \right) + \boldsymbol{K}_0 + \sum_{\substack{j=1 \\ j \neq k}}^{n} \beta_j \boldsymbol{K}_j$$

when computing $\boldsymbol{J}_2$.

Example 7.2 Construct a real symmetric, triple with prescribed real eigenvalues. This is an example of Problem 7.1 and has application in the passive vibration control of a mass-spring-damper system.

The system is $\boldsymbol{M} = \text{diag}\{1, 2, 3\}$,

$$\boldsymbol{C}_0 = \begin{pmatrix} 3 & -2 & 0 \\ -2 & 3 & -1 \\ 0 & -1 & 1 \end{pmatrix} \quad \boldsymbol{K}_0 = \begin{pmatrix} 8 & -4 & 0 \\ -4 & 11 & -7 \\ 0 & -7 & 7 \end{pmatrix}$$

and the affine family elements

$$\boldsymbol{C}_1 = \boldsymbol{K}_1 = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{C}_2 = \boldsymbol{K}_2 = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix},$$

$$\boldsymbol{C}_3 = \boldsymbol{K}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The method finds many real and complex solutions. Taking six steps from the starting value

$$\boldsymbol{\alpha}^{(0)} = (\, 0 \quad 0 \quad 10 \,)^T, \quad \boldsymbol{\beta}^{(0)} = (\, 10 \quad 50 \quad 10 \,)^T.$$

the method finds the real affine coefficients

$$\boldsymbol{\alpha} = \begin{pmatrix} -0.700525 \\ 0.128550 \\ 14.046055 \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} 4.608802 \\ 70.160636 \\ 8.151792 \end{pmatrix}$$

which assign the eigenvalues $-3 \pm i, -3 \pm 2i, -3 \pm 3i$. Starting instead from

$$\boldsymbol{\alpha}^{(0)} = (\, 0 \quad 0 \quad 0 \,)^T, \quad \boldsymbol{\beta}^{(0)} = (\, 50 \quad 10 \quad 10 \,)^T.$$

the method takes five steps to find a different solution

$$\boldsymbol{\alpha} = \begin{pmatrix} -0.324894 \\ 0.367479 \\ 13.164036 \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} 53.234508 \\ 13.732452 \\ 6.945330 \end{pmatrix}$$

which assigns the same eigenvalues.

Example 7.3 Finding the damping matrix. This example is a special case of Problem 7.1 in which we are given the matrices $\boldsymbol{M}$ and $\boldsymbol{K}$ and we seek the damping matrix $\boldsymbol{C}$. More precisely we want to solve

Problem 7.2 Given a symmetric tridiagonal $n \times n$ *matrix* $\boldsymbol{K}$ *and a set* $S = \{\mu_k\}_{k=1}^{2n}$ of self conjugate scalars such that $\det(\boldsymbol{K}) = \prod_{k=1}^{2n} \mu_k$, Find a symmetric, tridiagonal $\boldsymbol{C}, n \times n$, such that $\sigma(\lambda^2\boldsymbol{I} + \lambda\boldsymbol{C} + \boldsymbol{K})$ *is* S.

The necessity of $\det(\boldsymbol{K}) = \prod_{k=1}^{2n} \mu_k$ *follows from substituting* $\lambda = 0$ in $\det(\lambda^2\boldsymbol{I} + \lambda\boldsymbol{C} + \boldsymbol{K}) = \prod_{k=1}^{2n}(\lambda - \mu_k)$. *The problem generalizes easily to diagonal* $\boldsymbol{M}$ *replacing* $\boldsymbol{I}$.

We use $\boldsymbol{C} = \sum_{k=1}^{2n-1} \alpha_k \boldsymbol{C}_k$, *with the affine family*

$$\begin{cases} \boldsymbol{C}_i = \boldsymbol{e}_i\boldsymbol{e}_i^T, \quad i = 1, 2, \ldots, n \\ \boldsymbol{C}_{n+i} = (\boldsymbol{e}_i - \boldsymbol{e}_{i+1})(\boldsymbol{e}_i - \boldsymbol{e}_{i+1})^T, \quad i = 1, 2, \ldots, n-1, \end{cases}$$

As before $\boldsymbol{f}(\boldsymbol{\alpha})$ *is defined by* $f_i(\boldsymbol{\alpha}) = \det\left(\mu_i^2\boldsymbol{I} + \mu_i \sum_{k=1}^{2n-1} \alpha_k \boldsymbol{C}_k + \boldsymbol{K}\right)$, $i = 1, 2, \ldots, 2n-1$ *and the Jacobian is now* $2n-1$ *square*

$$[\boldsymbol{J}]_{ij} = \frac{\partial \lambda_i(\boldsymbol{\alpha})}{\partial \alpha_j}, \quad i,\ j = 1, 2, \ldots, 2n-1,$$

with

$$\boldsymbol{H} = \mu_i^2\boldsymbol{I} + \mu_i \sum_{k=1}^{2n-1} \alpha_k \boldsymbol{C}_k + \boldsymbol{K} \quad \text{and} \quad \boldsymbol{B} = \mu_i^2\boldsymbol{I} + \mu_i \sum_{\substack{j=1 \\ j \neq k}}^{2n-1} \alpha_j \boldsymbol{C}_j + \boldsymbol{K}.$$

Example

$$\boldsymbol{K} = \begin{pmatrix} 2 & -2 & \\ -2 & 9 & -4 \\ & -4 & 8 \end{pmatrix}, \quad \boldsymbol{C}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{C}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{C}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{C}_4 \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{C}_5 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix}.$$

The starting value $\boldsymbol{\alpha}^{(0)} = (1,1,1,1,1)^T$, *finds*

$$\boldsymbol{\alpha} = (0.722105, 2.002012, 0.209320, 1.936412, 2.132404)^T$$

to an accuracy of 10^{-15}in eight steps. This solution $\boldsymbol{\alpha}$ corresponds to the damping matrix

$$\boldsymbol{C} = \begin{pmatrix} 2.658517 & -1.936412 & \\ -1.936412 & 6.070828 & -2.132404 \\ & -2.132404 & 2.341724 \end{pmatrix}$$

which is such that $\sigma(\lambda^2\boldsymbol{I} + \lambda\boldsymbol{C} + \boldsymbol{K}) = \{-4\sqrt{2},\ -\sqrt{2},\ -1 \pm i,\ -1 \pm 2i\}$.

12.8 Symmetry Preserving Partial Pole Assignment for the Standard and the Generalized Eigenvalue Problems

The problem of symmetry preserving pole assignment for the quadratic pencil has received much attention [6, 11, 12, 24, 25, 26, 34] but remains a considerable challenge. In this section we describe a pole assignment method for the generalized inverse eigenvalue problem which has an explicit solution and which preserves symmetry.

Consider a system modelled by the differential equation

$$\boldsymbol{B}\boldsymbol{x}'(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t), \quad \boldsymbol{x}(t) = \boldsymbol{x}_0,$$

with $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$, symmetric and $\boldsymbol{b} \in \mathbb{R}^n$. We seek a single input control $u(t) = \boldsymbol{f}^T\boldsymbol{x}(t) - \boldsymbol{g}^T\boldsymbol{x}'(t)$ which is such that the closed loop system

$$\big(\boldsymbol{B} + \boldsymbol{b}\boldsymbol{g}^T\big)\boldsymbol{x}'(t) = \big(\boldsymbol{A} + \boldsymbol{b}\boldsymbol{f}^T\big)\boldsymbol{x}(t)$$

has prescribed frequencies. This leads to the linear algebra problem of finding vectors $\boldsymbol{f}, \boldsymbol{g}$ which assign part or all of the spectrum of the modified pencil

$$\big(\boldsymbol{A} + \boldsymbol{b}\boldsymbol{f}^T\big) - \lambda\big(\boldsymbol{B} + \boldsymbol{b}\boldsymbol{g}^T\big).$$

Now, many finite element models lead to real symmetric $\boldsymbol{A}, \boldsymbol{B}$ but rank-one updates of the form $\boldsymbol{b}\boldsymbol{f}^T, \boldsymbol{b}\boldsymbol{g}^T$ destroy symmetry. Furthermore, sometimes (the control of vibratory systems by passive elements is an example) we need the closed-loop system to satisfy a reciprocity law: *the force at* x_1 *due to a unit displacement at* x_2 *should equal the force at* x_2 *due to a unit displacement at* x_1. In such a case we need to find symmetric controls such as

$$P_c(\lambda) = \big(\boldsymbol{A} + \alpha\boldsymbol{u}\boldsymbol{u}^T\big) - \lambda\big(\boldsymbol{B} + \beta\boldsymbol{u}\boldsymbol{u}^T\big). \tag{12.24}$$

It is well known [29] that the eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$, and eigenvectors of

- the symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ are real and those of
- the real, symmetric definite ($\boldsymbol{B}$ spd) pair $\boldsymbol{A}, \boldsymbol{B}$ are real.

In addition, the eigenvalues of $\boldsymbol{A}$ and those $\mu_1 \le \mu_2 \le \cdots \le \mu_n$, of $\boldsymbol{A} + \sigma \boldsymbol{v}\boldsymbol{v}^T$, $\sigma = \pm 1$, interlace:

$$\lambda_j \le \mu_j \le \lambda_{j+1}, \quad \text{if } \sigma = 1 \text{ and } \mu_j \le \lambda_j \le \mu_{j+1}, \text{if } \sigma = -1 \quad j = 1, 2, \ldots, n$$

($\lambda_{n+1} = \mu_{n+1} = \infty$). Similarly, the eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ of $\boldsymbol{A} - \lambda \boldsymbol{B}$ and those $\mu_1 \le \mu_2 \le \cdots \le \mu_n$, of $\boldsymbol{A} + \alpha \boldsymbol{u}\boldsymbol{u}^T - \lambda(\boldsymbol{B} + \beta \boldsymbol{u}\boldsymbol{u}^T)$ interlace [19]

$$\left.\begin{array}{ll} \alpha/\beta \le \mu_1 \le \lambda_1 & \text{if } \quad \alpha/\beta \le \lambda_1, \\ \lambda_j \le \mu_j \le \lambda_{j+1} & \text{if } \lambda_j \le \alpha/\beta, \\ \lambda_j \le \mu_j \le \alpha/\beta \le \mu_{j+1} \le \lambda_{j+1} & \text{if } \lambda_j \le \alpha/\beta \le \lambda_{j+1}, \\ \lambda_j \le \mu_{j+1} \le \lambda_{j+1} & \text{if } \alpha/\beta \le \lambda_j, \\ \lambda_n \le \mu_n \le \alpha/\beta & \text{if } \lambda_n \le \alpha/\beta \end{array}\right\} j = 1, 2, \ldots, n. \tag{12.25}$$

Thus, only poles which interlace appropriately can be assigned.

Lowner [28] solved the symmetry preserving partial pole assignment for the standard eigenvalue problem by showing that choosing

$$\hat{\boldsymbol{u}}_i^2 = -\frac{\prod_{k=1}^{n}(\lambda_i - \mu_k)}{\prod_{\substack{k=1 \\ k\neq i}}^{n}(\lambda_i - \lambda_k)}, \quad i = 1, 2, \ldots, n \tag{12.26}$$

where, $\hat{\boldsymbol{u}} = \boldsymbol{Q}^T\boldsymbol{u}$, $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$, $\boldsymbol{Q}$ orthogonal, $\boldsymbol{\Lambda} = \operatorname{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$, assigns an appropriately interlacing set of scalars $\{\mu_j\}_{j=1}^n$ to the spectrum of $\boldsymbol{A} + \sigma \boldsymbol{v}\boldsymbol{v}^T$.

Similarly, there is an explicit formula [16] for the vector $\boldsymbol{u}$ which assigns part or all of the spectrum of $\boldsymbol{A} + \alpha \boldsymbol{u}\boldsymbol{u}^T - \lambda(\boldsymbol{B} + \beta \boldsymbol{u}\boldsymbol{u}^T)$.

The method leaves unchanged the eigenvalues not to be replaced, again ensuring no spillover and only those λ_j which are to be replaced need to be known. Our problem is now stated as

Problem 8.1 *Given*

a. $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$, symmetric and $\boldsymbol{B}$ spd with the spectrum of $\boldsymbol{A} - \lambda \boldsymbol{B}$ labelled $\lambda_1 \le \lambda_2 \le \lambda_3 \le \cdots \le \lambda_n$,
b. scalars $\alpha \ge 0, \beta > 0$, $\alpha/\beta \ne \lambda_j$, $\forall j$, and
c. $\{\mu_j\}_{j=1}^r$, $r \le n$ which satisfy the interlacing property (12.25)

Find $\boldsymbol{u} \in \mathbb{R}^n$ such that the spectrum of $\boldsymbol{A} + \alpha \boldsymbol{u}\boldsymbol{u}^T, \boldsymbol{B} + \beta \boldsymbol{u}\boldsymbol{u}^T$ is $\{\mu_1, \mu_2, \ldots, \mu_r, \lambda_{r+1}, \lambda_{r+2}, \ldots, \lambda_n\}$.

Our method uses the secular equation for a symmetric pair of matrices [19] which we briefly review here.

Suppose that $\boldsymbol{Y}$ simultaneously diagonalizes $\boldsymbol{A}$ and $\boldsymbol{B}$ with the scaling $\boldsymbol{Y}^T\boldsymbol{A}\boldsymbol{Y} = \boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}, \boldsymbol{Y}^T\boldsymbol{B}\boldsymbol{Y} = \boldsymbol{I}$. If $\boldsymbol{x}, \mu$ is an eigenpair of (12.24) then $(\boldsymbol{A} + \alpha\boldsymbol{u}\boldsymbol{u}^T)\boldsymbol{x} = \mu(\boldsymbol{B} + \beta\boldsymbol{u}\boldsymbol{u}^T)\boldsymbol{x}$ and using the substitution $\hat{\boldsymbol{u}} = \boldsymbol{Y}^T\boldsymbol{u}$ and $\boldsymbol{Y}\hat{\boldsymbol{x}} = \boldsymbol{x}$ we quickly get

$$(\Lambda + \alpha\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}^T)\hat{\boldsymbol{x}} = \mu(\boldsymbol{I} + \beta\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}^T)\hat{\boldsymbol{x}}. \tag{12.27}$$

Lemma 8.1 *Suppose λ_j are distinct and assume also $\boldsymbol{e}_j^T\hat{\boldsymbol{u}} \neq 0$ and $\lambda_j \neq \alpha/\beta$ for all j. Then, (a) $(\mu\boldsymbol{I} - \boldsymbol{\Lambda})$ is invertible, (b) $\mu \neq \alpha/\beta$, and (c) $\hat{\boldsymbol{x}}^T\hat{\boldsymbol{u}} \neq 0$.*

Multiplying (12.27) on the left by $\boldsymbol{x}^T$ and rearranging yields a secular equation for the symmetric pair $(\alpha - \beta\mu)\hat{\boldsymbol{u}}^T(\mu\boldsymbol{I} - \Lambda)^{-1}\hat{\boldsymbol{u}} = 1$ which is sometimes written componentwise as $g(\mu) = 1 - (\alpha - \beta\mu)\sum_{j=1}^{n}\frac{\hat{\boldsymbol{u}}_j^2}{\mu - \lambda_j} = 0$. The zeros of g are the eigenvalues of the pencil (12.24) and its poles are the eigenvalues of the pencil $\boldsymbol{A} - \lambda\boldsymbol{B}$. Figure 12.1 shows the example secular equation function

$$\boldsymbol{g}(\mu) = 1 - (3\,\mu - 7)\left(\frac{1/4}{1-\mu} + \frac{1/9}{2-\mu} + \frac{1/16}{3-\mu}\right). \tag{12.28}$$

The poles, at $1, 2, 3$, are the eigenvalues of the original system and the zeros, approximately $2.9233, 2.0913, 1.4196$ are the eigenvalues of the system after rank-one corrections are added to each of the matrices in the pencil.

Using the secular equation we can devise a symmetry preserving full and partial pole assignment method for the symmetric definite matrix pair.

Suppose first that we wish to perform a full pole assignment with $\{\mu_j\}_{j=1}^n$ appropriately defined. The equations $g(\mu_j) = 0$, $j = 1, 2, \ldots, n$ can be assembled into a matrix equation

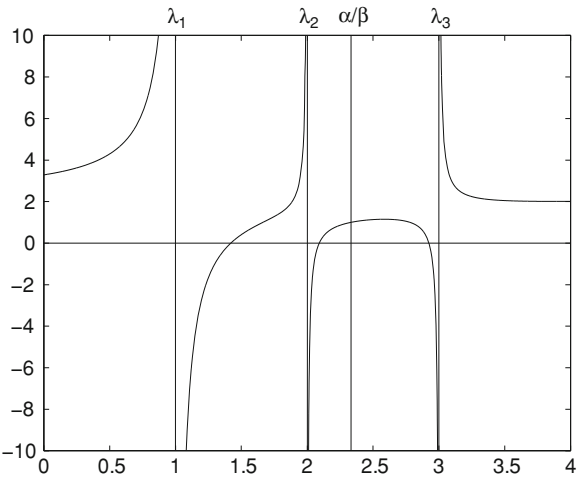


Fig. 12.1 The Secular equation function $g(\mu)$ of (12.28)

$$\begin{pmatrix} \frac{1}{\mu_1-\lambda_1} & \frac{1}{\mu_1-\lambda_2} & \cdots & \frac{1}{\mu_1-\lambda_n} \\ \frac{1}{\mu_2-\lambda_1} & \frac{1}{\mu_2-\lambda_2} & \cdots & \frac{1}{\mu_2-\lambda_n} \\ \cdots & & & \\ \frac{1}{\mu_n-\lambda_1} & \frac{1}{\mu_n-\lambda_2} & \cdots & \frac{1}{\mu_n-\lambda_n} \end{pmatrix} \begin{pmatrix} \hat{u}_1^2 \\ \hat{u}_2^2 \\ \vdots \\ \hat{u}_n^2 \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha-\beta\mu_1} \\ \frac{1}{\alpha-\beta\mu_2} \\ \vdots \\ \frac{1}{\alpha-\beta\mu_n}, \end{pmatrix}$$

in which the *Cauchy matrix*, $\boldsymbol{C}$, on the left, has explicit inverse

$$[\boldsymbol{C}^{-1}]_{ij} = \frac{\prod_{k=1}^{n}(\lambda_i-\mu_k)\prod_{k=1}^{n}(\mu_j-\lambda_k)}{(\lambda_i-\mu_j)\prod_{\substack{k=1\\k\neq i}}^{n}(\lambda_i-\lambda_k)\prod_{\substack{k=1\\k\neq j}}^{n}(\mu_j-\mu_k)}$$

from which we can quickly get

$$\begin{aligned} \hat{u}_i^2 &= \sum_{j=1}^{n} \frac{\prod_{k=1}^{n}(\lambda_i-\mu_k)\prod_{k=1}^{n}(\mu_j-\lambda_k)}{(\lambda_i-\mu_j)(\alpha-\beta\mu_j)\prod_{\substack{k=1\\k\neq i}}^{n}(\lambda_i-\lambda_k)\prod_{\substack{k=1\\k\neq j}}^{n}(\mu_j-\mu_k)} \\ &= -\frac{\prod_{k=1}^{n}(\lambda_i-\mu_k)}{\prod_{\substack{k=1\\k\neq i}}^{n}(\lambda_i-\lambda_k)} \sum_{j=1}^{n} \frac{\prod_{\substack{k=1\\k\neq i}}^{n}(\mu_j-\lambda_k)}{(\alpha-\beta\mu_j)\prod_{\substack{k=1\\k\neq j}}^{n}(\mu_j-\mu_k)} \\ &= \prod_{k=1}^{n}\frac{(\lambda_i-\mu_k)}{(\beta\mu_k-\alpha)} \prod_{\substack{k=1\\k\neq i}}^{n}\frac{(\beta\lambda_k-\alpha)}{(\lambda_i-\lambda_k)}, \quad i=1,2,\ldots,n. \end{aligned}$$

Thus, taking signs into account shows there are 2^n different solutions. We note that computing the $\hat{u}_i^2$ by this explicit formula requires $6n(n-1)\times,\div$ operations and is more economical than the $(n^3+6n^2+2n)/3$ operations required to solve the system numerically. There are, however, numerical considerations which may make the explicit formula undesirable for certain distributions of the λ_j and μ_j.

Partial pole assignment is now straightforward. Suppose that $S=\{i_1,i_2,\ldots,i_r\}$ is a subset of $1,2,\ldots,n$, $r\leq n$, and that T is the complement of S. Let α,β, $\{\lambda_j\}_1^n$, and $\{\mu_j\}_1^n$ satisfy the interlacing property (12.25). We set $\hat{u}_i=0$ for all $i\in T$ and compute

$$\hat{u}_i^2 = \prod_{k\in S}\frac{(\lambda_i-\mu_k)}{(\beta\mu_k-\alpha)} \prod_{\substack{k\in S\\k\neq i}}^{n}\frac{(\beta\lambda_k-\alpha)}{(\lambda_i-\lambda_k)}, \quad i\in S. \tag{12.29}$$

For each component of $\hat{u}$ which is zero, the secular equation has one fewer term and the Cauchy matrix has one fewer row and column.

The numerical properties of this method will be reported elsewhere. While the tests conducted so far have not been exhaustive, it is clear that the method, even applied to problems with $n=1,024$ and with assigned eigenvalues placed very close to the open loop system eigenvalues, can return assigned eigenvalues which have lost no more than about four decimals of accuracy. However, the distribution

of the existing and assigned eigenvalues and their respective condition numbers in the problem will certainly play a important role in determining the accuracy of the method in a particular case. Based on the test results obtained so far we believe that the method will produce very accurate assignments for even quite large systems provided that the point distributions are not pathological.

12.9 Conclusions

Our interest in most of the problems discussed in this paper was motivated by the study of damped vibrating systems and the control of damped oscillatory systems. The pole assignment and inverse problems for the symmetric definite linear pencils have the very useful property that all the eigenpairs in the system are real. Thus, we have described the solution to an inverse problem for a particular symmetric definite pair and we have shown how to assign part or all of the spectrum of a symmetric definite pair while preserving the system's symmetry.

The eigenpairs of real, symmetric definite quadratic pencils may contain complex elements. This fact considerably complicates the solution of inverse and pole assignment problems for these pencils. We have described a solution for the inverse problem of constructing a symmetric, tridiagonal, monic quadratic pencil with two spectral sets prescribed. We have described methods for pole assignment and partial eigenstructure assignment in symmetric definite quadratic pencils controlled by either single or multi-input controls. We have also described symmetry preserving pole assignment to symmetric definite quadratic pencils by affine methods.

Further work needs to be done on the characterization of those eigendata which, when assigned, lead to symmetric definite quadratic pencils with the kind of properties that can be realized in a physical system.

References

1. Balas MJ (1982) Trends in large space structure control theory: Fondest dreams, wildest hopes. IEEE Trans Automat Control AC-22:522–535
2. Bhaya A, Desoer C (1985) On the design of large flexible space structures (lfss). IEEE Trans Automat Control AC-30(11):1118–1120
3. Boley D, Golub GH (1987) A survey of matrix inverse eigenvalue problems. Inverse Probl. 3:595–622
4. Brahma S, Datta BN (2007) A norm-minimizing parametric algorithm for quadratic partial eigenvalue assignment via Sylvester equation. In: Proceedings of European control conference 2007, pp 490–496 (to appear)
5. Brahma S, Datta BN (2007) A Sylvester-equation based approach for minimum norm and robust partial quadratic eigenvalue assignment problems. Mediterranean conference on control and automation. MED '07, pp 1–6
6. Carvalho J, Datta BN, Lin WW, Wang CS (2006) Symmetry preserving eigenvalue embedding in finite element model updating of vibrating structures. J Sound Vib 290(3–5):839–864

7. Datta BN (1995) Numerical linear algebra and applications. Brooks/Cole Publishing Company, Pacific Grove
8. Datta BN, Elhay S, Ram YM (1996) An algorithm for the partial multi-input pole assignment problem of a second-order control system. In: Proceedings of the 35th IEEE conference on decision and control, vol 2. pp 2025–2029 (ISBN: 0780335910 0780335902 0780335929 0780335937)
9. Datta BN, Elhay S, Ram YM (1997) Orthogonality and partial pole assignment for the symmetric definite quadratic pencil. Linear Alg Appl 257:29–48
10. Datta BN, Elhay S, Ram YM, Sarkissian D (2000) Partial eigstructure assignemnt for the quadratic pencil. J Sound Vib 230:101–110
11. Datta BN, Sarkissian D (2001) Theory and computations of some inverse eigenvalue problems for the quadratic pencil. Contemp Math 280:221–240
12. Datta BN, Sokolov VO, Sarkissian DR (2008) An optimization procedure for model updating via physical parameters. Mechanical Systems and Signal Processing. (To appear in a special issue on Inverse Problems)
13. de Boor C, Golub GH (1978) The numerically stable reconstruction of a Jacobi matrix from spectral data. Lin Alg Appl 21:245–260
14. Demmel JW (1997) Applied numerical linear algebra. SIAM, Philadelphia
15. Dongarra JJ, Bunch JR, Moler CB, Stewart GW (1979) LINPACK User guide. SIAM, Philadelphia
16. Elhay S (2007) Symmetry preserving partial pole assignment for the standard and the generalized eigenvalue problems. In: Read Wayne, Roberts AJ (eds) Proceedings of the 13th biennial computational techniques and applications conference, CTAC-2006, vol 48. ANZIAM J, pp C264–C279. http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/106. Accessed 20 July 2007
17. Elhay S, Ram YM (2002) An affine inverse eigenvalue problem. Inverse Probl 18(2):455–466
18. Elhay S, Ram YM (2004) Quadratic pencil pole assignment by affine sums. In: Crawfordb J, Roberts AJ, (eds) Proceedings of 11th computational techniques and applications conference CTAC-2003, vol 45. pp. C592–C603. http://anziamj.austms.org.au/V45/CTAC2003/Elha. Accessed 4 July 2004
19. Elhay S, Golub GH, Ram YM (2003) On the spectrum of a modified linear pencil. Comput Maths Appl 46:1413–1426
20. Friedland S, Nocedal J, Overton ML (1987) The formulation and analysis of numerical methods for inverse eigenvalue problems. SIAM J Numer Anal 24(3):634–667
21. Hald O (1976) Inverse eigenvalue problems for Jacobi matrices. Lin Alg Appl 14:63–85
22. Hochstadt H (1974) On construction of a Jacobi matrix from spectral data. Lin Alg Appl 8:435–446
23. Inman D (1989) Vibration with control, measurement and stability. Prentice-Hall, Englewood Cilffs
24. Kuo YC, Lin WW, Xu SF (2006) Solutions of the partially described inverse quadratic eigenvalue problem. SIAM Matrix Anal Appl 29(1):33–53
25. Lancaster P, Prells U (2005) Inverse problems for damped vibrating systems. J Sound Vib 283:891–914
26. Lancaster P, Ye Qiang (1988) Inverse spectral problems for linear and quadratic matrix pencils. Linear Alg Appl 107:293–309
27. Laub AJ, Arnold WF (1984) Controllability and observability criteria for multivariate linear second order models. IEEE Trans Automat Control AC-29:163–165
28. Lowner K (1934) Uber monotone matrixfunktionen. Math Z 38:177–216
29. Parlett BN (1980) The symmetric eigenvalue problem. Prentice Hall, Englewood Cliffs
30. Parlett BN, Chen HC (1990) Use of indefinite pencils for computing damped natural modes. Linear Alg Appl 140:53–88
31. Ram YM, Elhay S (1996) An inverse eigenvalue problem for the symmetric tridiagonal quadratic pencil with application to damped oscillatory systems. SAIM J Appl Math 56(1):232–244

32. Ram YM, Elhay S (2000) Pole assignment in vibratory systems by multi input control. J Sound Vib 230:309–321
33. Ram YM, Elhay S (1998) Constructing the shape of a rod from eigenvalues. Commun Numer Methods Eng 14(7):597–608. ISSN: 1069-8299
34. Starek L, Inman D (1995) A symmetric inverse vibration problem with overdamped modes. J Sound Vib 181(5):893–903
35. Tissuer F, Meerbergen K (2001) The quadratic eigenvalue problem. SIAM Rev 43(3):235–286

Chapter 13
Descent Methods for Nonnegative Matrix Factorization

Ngoc-Diep Ho, Paul Van Dooren and Vincent D. Blondel

Abstract In this paper, we present several descent methods that can be applied to nonnegative matrix factorization and we analyze a recently developed fast block coordinate method called Rank-one Residue Iteration (RRI). We also give a comparison of these different methods and show that the new block coordinate method has better properties in terms of approximation error and complexity. By interpreting this method as a rank-one approximation of the residue matrix, we prove that it *converges* and also extend it to the nonnegative tensor factorization and introduce some variants of the method by imposing some additional controllable constraints such as: sparsity, discreteness and smoothness.

13.1 Introduction

Linear algebra has become a key tool in almost all modern techniques for data analysis. Most of these techniques make use of linear subspaces represented by eigenvectors of a particular matrix. In this paper, we consider a set of n data points $a_1, a_2, \ldots, a_n$, where each point is a real vector of size $m, a_i \in \mathbb{R}^m$. We then approximate these data points by linear combinations of r basis vectors $u_i \in \mathbb{R}^m$:

N.-D. Ho (✉) · P. Van Dooren · V. D. Blondel
CESAME, Université catholique de Louvain, Av. Georges Lemaître 4, 1348, Louvain-la-Neuve, Belgium
e-mail: hndiep@gmail.com

P. Van Dooren
e-mail: paul.vandooren@uclouvain.be

V. D. Blondel
e-mail: vincent.blondel@uclouvain.be

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_13,

$$a_i \approx \sum_{j=1}^{r} v_{ij} u_j, \quad v_{ij} \in \mathbb{R},\ u_j \in \mathbb{R}^m.$$

This can be rewritten in matrix form as $A \approx UV^T$, where a_i and u_i are respectively the columns of A and U and the v_{ij}'s are the elements of V. Optimal solutions of this approximation in terms of the Euclidean (or Frobenius) norm can be obtained by the Singular Value Decomposition (SVD) [11].

In many cases, data points are constrained to a subset of $\mathbb{R}^m$. For example, light intensities, concentrations of substances, absolute temperatures are, by their nature, nonnegative (or even positive) and lie in the nonnegative orthant $\mathbb{R}^m_+$. The input matrix A then becomes elementwise nonnegative and it is then natural to constrain the basis vectors v_i and the coefficients v_{ij} to be nonnegative as well. In order to satisfy this constraint, we need to approximate the columns of A by the following additive model:

$$a_i \approx \sum_{j=1}^{r} v_{ij} u_j, \quad v_{ij} \in \mathbb{R}_+,\ u_j \in \mathbb{R}^m_+.$$

where the v_{ij} coefficients and u_j vectors are nonnegative, $v_{ij} \in \mathbb{R}_+,\ u_j \in \mathbb{R}^m_+$.

Many algorithms have been proposed to find such a representation, which is referred to as a Nonnegative Matrix Factorization (NMF). The earliest algorithms were introduced by Paatero [23, 24]. But the topic became quite popular with the publication of the algorithm of Lee and Seung in 1999 [18] where multiplicative rules were introduced to solve the problem. This algorithm is very simple and elegant but it lacks a complete convergence analysis. Other methods and variants can be found in [15, 19, 20].

The quality of the approximation is often measured by a distance. Two popular choices are the Euclidean (Frobenius) norm and the generalized Kullback–Leibler divergence. In this paper, we focus on the Euclidean distance and we investigate descent methods for this measure. One characteristic of descent methods is their monotonic decrease until they reach a stationary point. This point maybe located in the interior of the nonnegative orthant or on its boundary. In the second case, the constraints become active and may prohibit any further decrease of the distance measure. This is a key issue to be analyzed for any descent method.

In this paper, $\mathbb{R}^m_+$ denotes the set of nonnegative real vectors (elementwise) and $[v]_+$ the projection of the vector v on $\mathbb{R}^m_+$. We use $v \geq 0$ and $A \geq 0$ to denote nonnegative vectors and matrices and $v > 0$ and $A > 0$ to denote positive vectors and matrices. $A \circ B$ and $\frac{[A]}{[B]}$ are respectively the Hadamard (elementwise) product and quotient. $A_{:i}$ and $A_{i:}$ are the ith column and ith row of A.

This paper is an extension of the internal report [14], where we proposed to decouple the problem based on rank one approximations to create a new algorithm called Rank-one Residue Iteration (RRI). During the revision of this report, we were informed that essentially the same algorithm was independently proposed

and published in [7] under the name Hierarchical Alternative Least Squares (HALS). But the present paper gives several additional results wherein the major contributions are *the convergence proof* of the method and its *extensions* to many practical situations and constraints. The paper also compares a selection of some recent descent methods from the literature and aims at providing a survey of such methods for nonnegative matrix factorizations. For that reason, we try to be self-contained and hence recall some well-known results. We also provide short proofs when useful for a better understanding of the rest of the paper.

We first give a short introduction of low rank approximations, both unconstrained and constrained. In Sect. 13.3 we discuss error bounds of various approximations and in Sect. 13.4 we give a number of descent methods for Nonnegative Matrix Factorizations. In Sect. 13.5 we describe the method based on successive rank one approximations. This method is then also extended to approximate higher order tensor and to take into account other constraints than nonnegativity. In Sect. 13.5 we discuss various regularization methods and in Sect. 13.6, we present numerical experiments comparing the different methods. We end with some concluding remarks.

13.2 Low-Rank Matrix Approximation

Low-rank approximation is a special case of matrix nearness problem [12]. When only a rank constraint is imposed, the optimal approximation with respect to the Frobenius norm can be obtained from the Singular Value Decomposition.

We first investigate the problem without the nonnegativity constraint on the low-rank approximation. This is useful for understanding properties of the approximation when the nonnegativity constraints are imposed but inactive. We begin with the well-known Eckart–Young Theorem.

Theorem 2.1 (Eckart–Young) *Let* $A \in \mathbb{R}^{m\times n}(m \geq n)$ *have the singular value decomposition*

$$A = P\Sigma Q^T, \quad \Sigma = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ *are the singular values of* A *and where* $P \in \mathbb{R}^{m\times m}$ *and* $Q \in \mathbb{R}^{n\times n}$ *are orthogonal matrices. Then for* $1 \leq r \leq n$, *the matrix*

$$A_r = P\Sigma_r Q^T, \quad \Sigma_r = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \dots & \sigma_r & \dots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix}$$

is a global minimizer of the problem

$$\min_{B \in \mathbb{R}^{m\times n}\ rank(B) \le r} \frac{1}{2}\|A - B\|_F^2 \tag{13.1}$$

and its error is

$$\frac{1}{2}\|A - B\|_F^2 = \frac{1}{2}\sum_{i=r+1}^{n} \sigma_i^2.$$

Moreover, if $\sigma_r > \sigma_{r+1}$ then A_r is the unique global minimizer.

The proof and other implications can be found for instance in [11]. The columns of P and Q are called singular vectors of A, in which vectors corresponding to the largest singular values are referred to as the dominant singular vectors.

Let us now look at the following modified problem

$$\min_{X \in \mathbb{R}^{m\times r}\ Y \in \mathbb{R}^{n\times r}} \frac{1}{2}\|A - XY^T\|_F^2, \tag{13.2}$$

where the rank constraint is implicit in the product XY^T since the dimensions of X and Y guarantee that $\text{rank}(XY^T) \le r$. Conversely, every matrix of rank less than r can be trivially rewritten as a product XY^T, where $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$. Therefore Problems (13.1) and (13.2) are equivalent. But even when the product $A_r = XY^T$ is unique, the pairs (XR^T, YR^{-1}) with R invertible, yield the same product XY^T. In order to avoid this, we can always choose X and Y such that

$$X = PD^{\frac{1}{2}} \quad \text{and} \quad Y = QD^{\frac{1}{2}}, \tag{13.3}$$

where $P^TP = I_{r\times r}, Q^TQ = I_{r\times r}$ and D is $r \times r$ nonnegative diagonal matrix. Doing this is equivalent to computing a compact SVD decomposition of the product $A_r = XY^T = PDQ^T$.

As usual for optimization problems, we calculate the gradient with respect to X and Y and set them equal to 0.

$$\nabla_X = XY^TY - AY = 0 \quad \nabla_Y = YX^TX - A^TX = 0. \tag{13.4}$$

If we then premultiply A^T with ∇_X and A with ∇_Y, we obtain

$$(A^TA)Y = (A^TX)Y^TY \quad (AA^T)X = (AY)X^TX. \tag{13.5}$$

Replacing $A^TX = YX^TX$ and $AY = XY^TY$ into (13.5) yields

$$(A^TA)Y = YX^TXY^TY \quad (AA^T)X = XY^TYX^TX. \tag{13.6}$$

Replacing (13.3) into (13.6) yields

$$(A^TA)QD^{\frac{1}{2}} = QDP^TPDQ^TQD^{\frac{1}{2}} \quad \text{and} \quad (AA^T)PD^{\frac{1}{2}} = PDQ^TQDP^TPD^{\frac{1}{2}}.$$

When D is invertible, this finally yields

$$(A^TA)Q = QD^2 \quad \text{and} \quad (AA^T)P = PD^2.$$

This shows that the columns of P and Q are singular vectors and D_{ii}'s are nonzero singular values of A. Notice that if D is singular, one can throw away the corresponding columns of P and Q and reduce it to a smaller-rank approximation with the same properties. Without loss of generality, we therefore can focus on approximations of Problem (13.2) which are of exact rank r. We can summarize the above reasoning in the following theorem.

Theorem 2.2 *Let $A \in \mathbb{R}^{m\times n}(m > n$ and $\text{rank}(A) = t)$. If $A_r(1 \le r \le t)$ is a rank r stationary point of Problem (13.2), then there exists two orthogonal matrices $P \in \mathbb{R}^{m\times m}$ and $Q \in \mathbb{R}^{n\times n}$ such that:*

$$A = P\hat{\Sigma}Q^T \quad \text{and} \quad A_r = P\hat{\Sigma}_rQ^T$$

where

$$\hat{\Sigma} = \begin{pmatrix} \hat{\sigma}_1 & 0 & \dots & 0 \\ 0 & \hat{\sigma}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{\sigma}_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}, \quad \hat{\Sigma}_r = \begin{pmatrix} \hat{\sigma}_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \hat{\sigma}_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \dots & \hat{\sigma}_r & \dots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix}$$

and the $\hat{\sigma}_i$'s are unsorted singular values of A. Moreover, the approximation error is:

$$\frac{1}{2}\|A - A_r\|_F^2 = \frac{1}{2}\sum_{i=r+1}^{t}\hat{\sigma}_i^2.$$

This result shows that, if the singular values are all different, there are $\frac{n!}{r!(n-r)!}$ possible stationary points A_r. When there are multiple singular values, there will be infinitely many stationary points A_r since there are infinitely many singular subspaces. The next result will identify the minima among all stationary points. Other stationary points are saddle points whose every neighborhood contains both smaller and higher points.

Theorem 2.3 *The only minima of Problem* (13.2) *are given by Theorem 2.1 and are global minima. All other stationary points are saddle points.*

Proof Let us assume that A_r is a stationary point given by Theorem 2.2 but not by Theorem 2.1. Then there always exists a permutation of the columns of P and Q, and of the diagonal elements of $\hat{\Sigma}$ and $\hat{\Sigma}_r$ such that $\hat{\sigma}_{r+1} > \hat{\sigma}_r$. We then construct two points in the ϵ-neighborhood of A_r that yield an increase and a decrease, respectively, of the distance measure. They are obtained by taking:

$$\overline{\Sigma}_r(\epsilon) = \begin{pmatrix} \hat{\sigma}_1 + \epsilon & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots \\ 0 & \dots & \hat{\sigma}_r & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{pmatrix}, \quad \overline{A}_r(\epsilon) = P\overline{\Sigma}_r(\epsilon)Q^T$$

and

$$\underline{\Sigma}_r(\epsilon) = \begin{pmatrix} \hat{\sigma}_1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & \hat{\sigma}_r & \epsilon\sqrt{\hat{\sigma}_r} & \vdots & 0 \\ 0 & \dots & \epsilon\sqrt{\hat{\sigma}_r} & \epsilon^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix}, \quad \underline{A}_r(\epsilon) = P\underline{\Sigma}_r(\epsilon)Q^T.$$

Clearly $\overline{A}_r(\epsilon)$ and $\underline{A}_r(\epsilon)$ are of rank r. Evaluating the distance measure yields

$$\begin{aligned} \|A - \underline{A}_r(\epsilon)\|_F^2 &= 2\hat{\sigma}_r\epsilon^2 + (\hat{\sigma}_{r+1} - \epsilon^2)^2 + \sum_{i=r+2}^{t} \hat{\sigma}_i^2 \\ &= \epsilon^2[\epsilon^2 - 2(\hat{\sigma}_{r+1} - \hat{\sigma}_r)] + \sum_{i=r+1}^{t} \hat{\sigma}_i^2 \\ &< \sum_{i=r+1}^{t} \hat{\sigma}_i^2 = \|A - A_r\|_F^2 \end{aligned}$$

for all $\epsilon \in (0, \sqrt{2(\hat{\sigma}_{r+1} - \hat{\sigma}_r)})$ and

$$\|A - \overline{A}_r(\epsilon)\|_F^2 = \epsilon^2 + \sum_{i=r+1}^{t} \hat{\sigma}_i^2 > \sum_{i=r+1}^{t} \hat{\sigma}_i^2 = \|A - A_r\|_F^2$$

for all $\epsilon > 0$. Hence, for an arbitrarily small positive ϵ, we obtain

$$\|A - \underline{A}_r(\epsilon)\|_F^2 < \|A - A_r\|_F^2 < \|A - \overline{A}_r(\epsilon)\|_F^2$$

which shows that A_r is a saddle point of the distance measure. □

When we add a nonnegativity constraint in the next section, the results of this section will help to identify stationary points at which all the nonnegativity constraints are inactive.

13.3 Nonnegativity Constraint

In this section, we investigate the problem of Nonnegative Matrix Factorization. This problem differs Problem (13.2) in the previous section because of the additional nonnegativity constraints on the factors. We first discuss the effects of adding such a constraint. By doing so, the problem is no longer easy because of the existence of local minima at the boundary of the nonnegative orthant. Determining the lowest minimum among these minima is far from trivial. On the other hand, a minimum that coincides with a minimum of the unconstrained problem (i.e. Problem (13.2)) may be easily reached by standard descent methods, as we will see.

Problem 1 *(Nonnegative matrix factorization—NMF)* Given a $m \times n$ nonnegative matrix A and an integer $r < \min(m, n)$, solve

$$\min_{U \in \mathbb{R}_+^{m \times r}} \min_{V \in \mathbb{R}_+^{n \times r}} \frac{1}{2} \|A - UV^T\|_F^2.$$

where r is called the reduced rank. From now on, m and n will be used to denote the size of the target matrix A and r is the reduced rank of a factorization.

We rewrite the nonnegative matrix factorization as a standard nonlinear optimization problem:

$$\min_{-U \leq 0} \min_{-V \leq 0} \frac{1}{2} \|A - UV^T\|_F^2.$$

The associated Lagrangian function is

$$L(U, V, \mu, \nu) = \frac{1}{2} \|A - UV^T\|_F^2 - \mu \circ U - \nu \circ V,$$

where μ and ν are two *matrices* of the same size of U and V, respectively, containing the Lagrange multipliers associated with the nonnegativity constraints $U_{ij} \geq 0$ and $V_{ij} \geq 0$. Then the Karush–Kuhn–Tucker conditions for the nonnegative matrix factorization problem say that if (U, V) is a local minimum, then there exist $\mu_{ij} \geq 0$ and $\nu_{ij} \geq 0$ such that:

$$U \geq 0, \quad V \geq 0, \tag{13.7}$$

$$\nabla L_U = 0, \quad \nabla L_V = 0, \tag{13.8}$$

$$\mu \circ U = 0, \quad v \circ V = 0. \tag{13.9}$$

Developing (13.8) we have:

$$AV - UV^T V - \mu = 0, \quad A^T U - VU^T U - v = 0$$

or

$$\mu = -(UV^T V - AV), \quad v = -(VU^T U - A^T U).$$

Combining this with $\mu_{ij} \geq 0, v_{ij} \geq 0$ and (13.9) gives the following conditions:

$$U \geq 0, \quad V \geq 0, \tag{13.10}$$

$$\nabla F_U = UV^T V - AV \geq 0, \quad \nabla F_V = VU^T U - A^T U \geq 0, \tag{13.11}$$

$$U \circ (UV^T V - AV) = 0, \quad V \circ (VU^T U - A^T U) = 0, \tag{13.12}$$

where the corresponding Lagrange multipliers for U and V are also the gradient of F with respect to U and V. Since the Euclidean distance is not convex with respect to both variables U and V at the same time, these conditions are only necessary. This is implied because of the existence of saddle points and maxima. We then call all the points that satisfy the above conditions, the stationary points.

Definition 1 (*NMF stationary point*) We call (U, V) a stationary point of the NMF Problem if and only if U and V satisfy the KKT conditions (13.10), (13.11) and (13.12).

Alternatively, a stationary point (U, V) of the NMF problem can also be defined by using the following necessary condition (see for example [4]) on the convex sets $\mathbb{R}_+^{m \times r}$ and $\mathbb{R}_+^{n \times r}$, that is

$$\left\langle \begin{pmatrix} \nabla F_U \\ \nabla F_V \end{pmatrix}, \begin{pmatrix} X - U \\ Y - V \end{pmatrix} \right\rangle \geq 0, \quad \forall\, X \in \mathbb{R}_+^{m \times r},\, Y \in \mathbb{R}_+^{n \times r}, \tag{13.13}$$

which can be shown to be equivalent to the KKT conditions (13.10), (13.11) and (13.12). Indeed, it is trivial that the KKT conditions imply (13.13). And by carefully choosing different values of X and Y from (13.13), one can easily prove that the KKT conditions hold.

There are two values of reduced rank r for which we can trivially identify the global solution which are $r = 1$ and $r = \min(m, n)$. For $r = 1$, a pair of dominant singular vectors are a global minimizer. And for $r = \min(m, n)$, $(U = A, V = I)$ is a global minimizer. Since most of existing methods for the nonnegative matrix factorization are descent algorithms, we should pay attention to all local minimizers. For the rank-one case, they can easily be characterized.

13.3.1 Rank One Case

The rank-one NMF problem of a nonnegative matrix A can be rewritten as

$$\min_{u\in\mathbb{R}^m_+,\, v\in\mathbb{R}^n_+} \frac{1}{2}\|A - uv^T\|_F^2 \tag{13.14}$$

and a complete analysis can be carried out. It is well known that any pair of nonnegative Perron vectors of AA^T and A^TA yields a global minimizer of this problem, but we can also show that the *only* stationary points of (13.14) are given by such vectors. The following theorem excludes the case where $u = 0$ and/or $v = 0$.

Theorem 3.1 *The pair (u, v) is a local minimizer of* (13.14) *if and only if u* and *v are nonnegative eigenvectors of AA^T and A^TA respectively of the eigenvalue $\sigma = \|u\|_2^2\|v\|_2^2$.*

Proof The *if part* easily follows from Theorem 2.2. For the *only if part* we proceed as follows. Without loss of generality, we can permute the rows and columns of A such that the corresponding vectors u and v are partitioned as $(u_+\ 0)^T$ and $(v_+\ 0)^T$ respectively, where $u_+, v_+ > 0$. Partition the corresponding matrix A conformably as follows

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

then from (13.11) we have

$$\begin{pmatrix} u_+v_+^T & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} v_+ \\ 0 \end{pmatrix} - \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} v_+ \\ 0 \end{pmatrix} \geq 0$$

and

$$\begin{pmatrix} v_+u_+^T & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} u_+ \\ 0 \end{pmatrix} - \begin{pmatrix} A_{11}^T & A_{21}^T \\ A_{12}^T & A_{22}^T \end{pmatrix}\begin{pmatrix} u_+ \\ 0 \end{pmatrix} \geq 0$$

implying that $A_{21}v_+ \leq 0$ and $A_{12}^Tu_+ \leq 0$. Since $A_{21}, A_{12} \geq 0$ and $u_+, v_+ > 0$, we can conclude that $A_{12} = 0$ and $A_{21} = 0$. Then from (13.12) we have:

$$u_+ \circ (\|v_+\|_2^2u_+ - A_{11}v_+) = 0 \quad \text{and} \quad v_+ \circ (\|u_+\|_2^2v_+ - A_{11}^+u_+) = 0.$$

Since $u_+, v_+ > 0$, we have:

$$\|v_+\|_2^2u_+ = A_{11}v_+ \quad \text{and} \quad \|u_+\|_2^2v_+ = A_{11}^Tu_+$$

or

$$\|u_+\|_2^2\|v_+\|_2^2u_+ = A_{11}A_{11}^Tu_+ \quad \text{and} \quad \|u_+\|_2^2\|v_+\|_2^2v_+ = A_{11}^TA_{11}v_+.$$

Setting $\sigma = \|u_+\|_2^2\|v_+\|_2^2$ and using the block *diagonal* structure of A yields the desired result. □

Theorem 3.1 guarantees that all stationary points of the rank-one case are nonnegative singular vectors of a submatrix of A. These results imply that a global minimizer of the rank-one NMF can be calculated correctly based on the largest singular value and corresponding singular vectors of the matrix A.

For ranks other than 1 and $\min(m,n)$, there are no longer trivial stationary points. In the next section, we try to derive some simple characteristics of the local minima of the nonnegative matrix factorization.

The KKT conditions (13.12) help to characterize the stationary points of the NMF problem. Summing up all the elements of one of the conditions (13.12), we get:

$$\begin{aligned} 0 &= \sum_{ij} \big(U \circ (UV^TV - AV)\big)_{ij} \\ &= \big\langle U, UV^TV - AV \big\rangle \\ &= \big\langle UV^T, UV^T - A \big\rangle. \end{aligned} \tag{13.15}$$

From that, we have some simple characteristics of the NMF solutions:

Theorem 3.2 *Let (U,V) be a stationary point of the NMF problem, then $UV^T \in \mathcal{B}\big(\frac{A}{2}, \frac{1}{2}\|A\|_F\big)$, the ball centered at $\frac{A}{2}$ and with radius $= \frac{1}{2}\|A\|_F$.*

Proof From (13.15) it immediately follows that

$$\left\langle \frac{A}{2} - UV^T, \frac{A}{2} - UV^T \right\rangle = \left\langle \frac{A}{2}, \frac{A}{2} \right\rangle$$

which implies

$$UV^T \in \mathcal{B}\left(\frac{A}{2}, \frac{1}{2}\|A\|_F\right).$$

□

Theorem 3.3 *Let (U,V) be a stationary of the NMF problem, then*

$$\frac{1}{2}\|A - UV^T\|_F^2 = \frac{1}{2}(\|A\|_F^2 - \|UV^T\|_F^2).$$

Proof From (13.15), we have $\langle UV^T, A\rangle = \langle UV^T, UV^T\rangle$. Therefore,

$$\begin{aligned} \frac{1}{2}\big\langle A - UV^T, A - UV^T \big\rangle &= \frac{1}{2}(\|A\|_F^2 - 2\big\langle UV^T, A\big\rangle + \|UV^T\|_F^2) \\ &= \frac{1}{2}(\|A\|_F^2 - \|UV^T\|_F^2). \end{aligned}$$

□

Theorem 3.3 also suggests that at a stationary point (U, V) of the NMF problem, we should have $\|A\|_F^2 \geq \|UV^T\|_F^2$. This norm inequality can be also found in [6] for less general cases where we have $\nabla F_U = 0$ and $\nabla F_V = 0$ at a stationary point. For this particular class of NMF stationary point, all the nonnegativity constraints on U and $V$are inactive. And all such stationary points are also stationary points of the unconstrained problem, characterized by Theorem 2.2.

We have seen in Theorem 2.2 that, for the unconstrained least-square problem the only stable stationary points are in fact global minima. Therefore, if the stationary points of the constrained problem are inside the nonnegative orthant (i.e. all constraints are inactive), we can then probably reach the global minimum of the NMF problem. This can be expected because the constraints may no longer prohibit the descent of the update.

Let A_r be the optimal rank-r approximation of a nonnegative matrix A, which we obtain from the singular value decomposition, as indicated in Theorem 2.2. Then we can easily construct its nonnegative part $[A_r]_+$, which is obtained from A_r by just setting all its negative elements equal to zero. This is in fact the closest matrix in the cone of nonnegative matrices to the matrix A_r, in the Frobenius norm (in that sense, it is its projection on that cone). We now derive some bounds for the error $\|A - [A_r]_+\|_F$.

Theorem 3.4 *Let A_r be the best rank r approximation of a nonnegative matrix A, and let $[A_r]_+$ be its nonnegative part, then*

$$\|A - [A_r]_+\|_F \leq \|A - A_r\|_F.$$

Proof This follows easily from the convexity of the cone of nonnegative matrices. Since both A and $[A_r]_+$ are nonnegative and since $[A_r]_+$ is the closest matrix in that cone to A_r we immediately obtain the inequality

$$\|A - A_r\|_F^2 \geq \|A - [A_r]_+\|_F^2 + \|A_r - [A_r]_+\|_F^2 \geq \|A - [A_r]_+\|_F^2$$

from which the result readily follows. □

The approximation $[A_r]_+$ has the merit of requiring as much storage as a rank r approximation, even though its rank is larger than r whenever $A_r \neq [A_r]_+$. We will look at the quality of this approximation in Sect. 13.6. If we now compare this bound with the nonnegative approximations then we obtain the following inequalities. Let $U_* V_*^T$ be an optimal nonnegative rank r approximation of A and let UV^T be any stationary point of the KKT conditions for a nonnegative rank r approximation, then we have:

$$\|A - [A_r]_+\|_F^2 \leq \|A - A_r\|_F^2 = \sum_{i=r+1}^{n} \sigma_i^2 \leq \|A - U_* V_*^T\|_F^2 \leq \|A - UV^T\|_F^2.$$

For more implications of the NMF problem, see [13].

13.4 Existing Descent Algorithms

We focus on descent algorithms that guarantee a non increasing update at each iteration. Based on the search space, we have two categories: *Full-space search* and *(Block) Coordinate search*.

Algorithms in the former category try to find updates for both U and V at the same time. This requires a search for a descent direction in the $(m+n)r$-dimensional space. Note also that the NMF problem in this full space is not convex but the optimality conditions may be easier to achieve.

Algorithms in the latter category, on the other hand, find updates for each (block) coordinate in order to guarantee the descent of the objective function. Usually, search subspaces are chosen to make the objective function convex so that efficient methods can be applied. Such a simplification might lead to the loss of some convergence properties. Most of the algorithms use the following column partitioning:

$$\frac{1}{2}\|A - UV^T\|_F^2 = \frac{1}{2}\sum_{i=1}^{n} \|A_{:,i} - U(V_{i,:})^T\|_2^2, \tag{13.16}$$

which shows that one can minimize with respect to each of the rows of V independently. The problem thus decouples into smaller convex problems. This leads to the solution of quadratic problems of the form

$$\min_{v \geq 0} \frac{1}{2}\|a - Uv\|_2^2. \tag{13.17}$$

Updates for the rows of V are then alternated with updates for the rows of U in a similar manner by transposing A and UV^T.

Independent on the search space, most of algorithms use the Projected Gradient scheme for which three basic steps are carried out in each iteration:

- Calculating the gradient $\nabla F(x^k)$,
- Choosing the step size α^k,
- Projecting the update on the nonnegative orthant

$$x^{k+1} = [x^k - \alpha^k \nabla F(x^k)]_+,$$

where x^k is the variable in the selected search space. The last two steps can be merged in one iterative process and must guarantee a sufficient decrease of the objective function as well as the nonnegativity of the new point.

13.4.1 Multiplicative Rules (Mult)

Multiplicative rules were introduced in [18]. The algorithm applies a block coordinate type search and uses the above column partition to formulate the updates. A special feature of this method is that the step size is calculated for each element of the vector. For the elementary problem (13.17) it is given by

$$v^{k+1} = v^k - \alpha^k \circ \nabla F(v^{k+1}) = v^k \circ \frac{[U^T a]}{[U^T U v^k]}$$

where $[\alpha^k]_i = \frac{v_i}{[U^T U v]_i}$. Applying this to all rows of V and U gives the updating rule of Algorithm 1 to compute

$$(U^*, V^*) = \underset{U \geq 0\ V \geq 0}{\operatorname{argmin}} \|A - UV^T\|_F^2.$$

Algorithm 1 (Mult)

1: Initialize U^0, V^0 and $k = 0$
2: **repeat**
3: $\quad U^{k+1} = U^k \circ \frac{[AV^k]}{[U^k (V^k)^T (V^k)]}$
4: $\quad V^{k+1} = V^k \circ \frac{[A^T U^{k+1}]}{[V^k (U^{k+1})^T (U^{k+1})]}$
5: $\quad k = k + 1$
6: **until** Stopping condition

These updates guarantee automatically the nonnegativity of the factors but may fail to give a sufficient decrease of the objective function. It may also get stuck in a non-stationary point and hence suffer from a poor convergence. Variants can be found in [20, 22].

13.4.2 Line Search Using Armijo Criterion (Line)

In order to ensure a sufficient descent, the following projected gradient scheme with Armijo criterion [19, 21] can be applied to minimize

$$x^* = \underset{x}{\operatorname{argmin}} F(x).$$

Algorithm 2 needs two parameters σ and β that may affect its convergence. It requires only the gradient information, and is applied in [19] for two different strategies: for the whole space (U, V) (Algorithm FLine) and for U and V

separately in an alternating fashion (Algorithm CLine). With a good choice of parameters ($\sigma = 0.01$ and $\beta = 0.1$) and a good strategy of alternating between variables, it was reported in [19] to be the faster than the multiplicative rules.

Algorithm 2 (Line)

Initialize x^0, σ, β, $\alpha_0 = 1$ and $k = 1$
repeat
  $\alpha_k = \alpha_{k-1}$
  $y = [x^k - \alpha_k \nabla F(x^k)]_+$
  if $F(y) - F(x^k) > \sigma \left\langle \nabla F(x^k), y - x^k \right\rangle$ **then**
    repeat
      $\alpha_k = \alpha_k \cdot \beta$
      $y = [x^k - \alpha_k \nabla F(x^k)]_+$
    until $F(y) - F(x^k) \leq \sigma \left\langle \nabla F(x^k), y - x^k \right\rangle$
  else
    repeat
      $lasty = y$
      $\alpha_k = \alpha_k / \beta$
      $y = [x^k - \alpha_k \nabla F(x^k)]_+$
    until $F(y) - F(x^k) > \sigma \left\langle \nabla F(x^k), y - x^k \right\rangle$
    $y = lasty$
  end if
  $x^{k+1} = y$
  $k = k + 1$
until Stopping condition

13.4.3 Projected Gradient with First-Order Approximation (FO)

In order to find the solution to

$$x^* = \underset{x}{\operatorname{argmin}} F(x)$$

we can also approximate at each iteration the function $F(X)$ using:

$$\tilde{F}(x) = F(x^k) + \left\langle \nabla_x F(x^k), x - x^k \right\rangle + \frac{L}{2} \|x^k - x\|_2^2,$$

where L is a Lipshitz constant satisfying $F(x) \leq \tilde{F}(x)$, $\forall x$. Because of this inequality, the solution of the following problem

$$x_{k+1} = \underset{x \geq 0}{\operatorname{argmin}} \tilde{F}(x)$$

also is a point of descent for the function $F(x)$ since

$$F(x_{k+1}) \leq \tilde{F}(x_{k+1}) \leq \tilde{F}(x_k) = F(x_k).$$

Since the constant L is not known a priori, an inner loop is needed. Algorithm 3 presents an iterative way to carry out this scheme. As in the previous algorithm this also requires only the gradient information and can therefore can be applied to two different strategies: to the whole space (U, V) (Algorithm FFO) and to U and V separately in an alternating fashion (Algorithm CFO).

A main difference with the previous algorithm is its stopping criterion for the inner loop. This algorithm requires also a parameter β for which the practical choice is 2.

13.4.4 Alternative Least Squares Methods

The first algorithm proposed for solving the nonnegative matrix factorization was the alternative least squares method [24]. It is known that, fixing either U or V, the problem becomes a least squares problem with nonnegativity constraint.

Since the least squares problems in Algorithm 4 can be perfectly decoupled into smaller problems corresponding to the columns or rows of A, we can directly apply methods for the Nonnegative Least Square problem to each of the small problem. Methods that can be applied are [5, 17], etc.

Algorithm 3 (FO)

Initialize x^0, L_0 and $k = 0$
repeat
  $y = [x^k - \frac{1}{L_k}\nabla F(x^k)]_+$
  while $F(y) - F(x^k) > \langle \nabla F(x^k), y - x^k \rangle + \frac{L_k}{2}\|y - x^k\|_2^2$ **do**
    $L_k = L_k/\beta$
    $Y = [x^k - \frac{1}{L_k}\nabla F(x^k)]_+$
  end while
  $x^{k+1} = y$
  $L_{k+1} = L_k \cdot \beta$
  $k = k + 1$
until Stopping condition

Algorithm 4 Alternative Least Square (ALS)

Initialize U and V
repeat
  Solve: $\min_{V \geq 0} \frac{1}{2}\|A - UV^T\|_F^2$
  Solve: $\min_{U \geq 0} \frac{1}{2}\|A^T - VU^T\|_F^2$
until Stopping condition

13.4.5 Implementation

The most time-consuming job is the test for the sufficient decrease, which is also the stopping condition for the inner loop. As mentioned at the beginning of the section, the above methods can be carried out using two different strategies: full space search or coordinate search. In some cases, it is required to evaluate repeatedly the function $F(U, V)$. We mention here how to do this efficiently with the coordinate search.

Full space search: The exact evaluation of $F(x) = F(U, V) = \|A - UV^T\|_F^2$ need $O(mnr)$ operations. When there is a correction $y = (U + \Delta U, V + \Delta V)$, we have to calculate $F(y)$ which also requires $O(mnr)$ operations. Hence, it requires $O(tmnr)$ operations to determine a stepsize in t iterations of the inner loop.

Coordinate search: when V is fixed, the Euclidean distance is a quadratic function on U:

$$\begin{aligned} F(U) = \|A - UV^T\|_F^2 &= \langle A, A\rangle - 2\langle UV^T, A\rangle + \langle UV^T, UV^T\rangle \\ &= \|A\|_F^2 - 2\langle U, AV\rangle + \langle U, U(V^TV)\rangle. \end{aligned}$$

The most expensive step is the computation of AV, which requires $O(mnr)$ operations. But when V is fixed, AV can be calculated once at the beginning of the inner loop. The remaining computations are $\langle U, AV\rangle$ and $\langle U, U(V^TV)\rangle$, which requires $O(nr)$ and $O(nr^2 + nr)$ operations. Therefore, it requires $O(tnr^2)$ operations to determine a stepsize in t iterations of the inner loop which is much less than $O(tmnr)$ operations. This is due to the assumption $r \ll n$. Similarly, when U fixed, $O(tmr^2)$ operations are needed to determine a stepsize.

If we consider an iteration is a sweep, i.e. once all the variables are updated, the following table summarizes the complexity of each sweep of the described algorithms:

Algorithm	Complexity per iteration
Mult	$O(mnr)$
FLine	$O(tmnr)$
CLine	$O(t_1nr^2 + t_2mr^2)$
FFO	$O(tmnr)$
CFO	$O(t_1nr^2 + t_2mr^2)$
ALS	$O(2^r mnr)^*$
IALS	$O(mnr)$

where t, t_1 and t_2 are the number of iterations of inner loops, which can not be bounded in general. For algorithm *ALS*, the complexity is reported for the case where the active set method [17] is used. Although $O(2^r mnr)$ is a very high *theorical* upper bound that count all the possible subsets of r variables of each

subproblem, in practice, the active set method needs much less iterations to converge. One might as well use more efficient convex optimization tools to solve the subproblems instead of the active set method.

13.4.6 Scaling and Stopping Criterion

For descent methods, several stopping conditions are used in the literature. We now discuss some problems when implementing these conditions for NMF.

The very first condition is the decrease of the objective function. The algorithm should stop when it fails to make the objective function decrease with a certain amount:

$$F(U^{k+1}, V^{k+1}) - F(U^k, V^k) < \epsilon \quad \text{or} \quad \frac{F(U^{k+1}, V^{k+1}) - F(U^k, V^k)}{F(U^k, V^k)} < \epsilon.$$

This is not a good choice for all cases since the algorithm may stop at a point very far from a stationary point. Time and iteration bounds can also be imposed for very slowly converging algorithms. But here again this may not be good for the optimality conditions. A better choice is probably the norm of the projected gradient as suggested in [19]. For the NMF problem it is defined as follows:

$$[\nabla_X^P]_{ij} = \begin{cases} [\nabla_X]_{ij} & \text{if } X_{ij} > 0 \\ \min(0, [\nabla_X]_{ij}) & \text{if } X_{ij} = 0 \end{cases}$$

where X stands for U or V. The proposed condition then becomes

$$\left\| \begin{pmatrix} \nabla_{U^k}^P \\ \nabla_{V^k}^P \end{pmatrix} \right\|_F \leq \epsilon \left\| \begin{pmatrix} \nabla_{U^1} \\ \nabla_{V^1} \end{pmatrix} \right\|_F. \tag{13.18}$$

We should also take into account the scaling invariance between U and V. Putting $\bar{U} = \gamma U$ and $\bar{V} = \frac{1}{\gamma} V$ does not change the approximation UV^T but the above projected gradient norm is affected:

$$\begin{aligned} \left\| \begin{pmatrix} \nabla_{\bar{U}}^P \\ \nabla_{\bar{V}}^P \end{pmatrix} \right\|_F^2 &= \|\nabla_{\bar{U}}^P\|_F^2 + \|\nabla_{\bar{V}}^P\|_F^2 = \frac{1}{\gamma^2}\|\nabla_U^P\|_F^2 + \gamma^2 \|\nabla_V^P\|_F^2 \\ &\neq \left\| \begin{pmatrix} \nabla_U^P \\ \nabla_V^P \end{pmatrix} \right\|_F^2. \end{aligned} \tag{3.19}$$

Two approximate factorizations $UV^T = \bar{U}\bar{V}^T$ resulting in the same approximation should be considered equivalent in terms of precision. One could choose $\gamma^2 := \|\nabla_U^P\|_F / \|\nabla_V^P\|_F$, which minimizes (13.19) and forces $\|\nabla_{\bar{U}}^P\|_F = \|\nabla_{\bar{V}}^P\|_F$, but this may not be a good choice when only one of the gradients $\|\nabla_{\bar{U}}^P\|_F$ and $\|\nabla_{\bar{V}}^P\|_F$ is nearly zero.

In fact, the gradient $\begin{pmatrix} \nabla_U \\ \nabla_V \end{pmatrix}$ is scale dependent in the NMF problem and any stopping criterion that uses gradient information is affected by this scaling. To limit that effect, we suggest the following scaling after each iteration:

$$\tilde{U}_k \leftarrow U_k D_k \quad \tilde{V}_k \leftarrow V_k D_k^{-1}$$

where D_k is a positive diagonal matrix:

$$[D_k]_{ii} = \sqrt{\frac{\|V_{:i}\|_2}{\|U_{:i}\|_2}}.$$

This ensures that $\|\tilde{U}_{:i}\|_F^2 = \|\tilde{V}_{:i}\|_F^2$ and hopefully reduces also the difference between $\|\nabla_{\tilde{U}}^P\|_F^2$ and $\|\nabla_{\tilde{V}}^P\|_F^2$. Moreover, it may help to avoid some numerically unstable situations.

The same scaling should be applied to the initial point as well (U_1, V_1) when using (13.18) as the stopping condition.

13.5 Rank-One Residue Iteration

In the previous section, we have seen that it is very appealing to decouple the problem into convex subproblems. But this may "converge" to solutions that are far from the global minimizers of the problem.

In this section, we analyze a different decoupling of the problem based on rank one approximations. This also allows us to formulate a very simple basic subproblem. This scheme has a major advantage over other methods: the subproblems can be optimally solved in closed form. Therefore it can be proved to have a strong convergence results through its *damped* version and it can be extended to more general types of factorizations such as for nonnegative tensors and to some practical constraints such as sparsity and smoothness. Moreover, the experiments in Sect. 13.6 suggest that this method outperforms the other ones in most cases. During the completion of the revised version of this report, we were informed that an independent report [9] had also proposed this decoupling without any convergence investigation and extentions.

13.5.1 New Partition of Variables

Let the u_i's and v_i's be respectively the columns of U and V. Then the NMF problem can be rewritten as follows:

Problem 2 (*Nonnegative Matrix Factorization*) Given a $m \times n$ nonnegative matrix A, solve

$$\min_{u_i \geq 0\ v_i \geq 0} \frac{1}{2} \left\| A - \sum_{i=1}^{r} u_i v_i^T \right\|_F^2 .$$

Let us fix all the variables, except for a single vector v_t and consider the following least squares problem:

$$\min_{v \geq 0} \frac{1}{2} \|R_t - u_t v^T\|_F^2, \tag{13.20}$$

where $R_t = A - \sum_{i \neq t} u_i v_i^T$. We have:

$$\|R_t - u_t v^T\|_F^2 = \operatorname{trace}\left[(R_t - u_t v^T)^T (R_t - u_t v^T) \right] \tag{13.21}$$

$$\|R_t - u_t v^T\|_F^2 = \|R_t\|_F^2 - 2 v^T R_t^T u_t + \|u_t\|_2^2 \|v\|_2^2. \tag{13.22}$$

From this formulation, one now derives the following lemma.

Lemma 5.1 *If* $[R_t^T u_t]_+ \neq 0$, *then* $v_* := \frac{[R_t^T u_t]_+}{\|u_t\|_2^2}$ *is the unique global minimizer of* (13.20) *and the function value equals* $\|R_t\|_F^2 - \frac{\|[R_t^T u_t]_+\|_2^2}{\|u_t\|_2^2}$.

Proof Let us permute the elements of the vectors $x := R_t^T u_t$ and v such that

$$Px = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad Pv = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \quad \text{with} \quad x_1 \geq 0,\ x_2 < 0$$

and P is the permutation matrix. Then

$$\|R_t - u_t v^T\|_F^2 = \|R_t\|_F^2 - 2 v_1^T x_1 - 2 v_2^T x_2 + \|u_t\|_2^2 (v_1^T v_1 + v_2^T v_2).$$

Since $x_2 < 0$ and $v_2 \geq 0$, it is obvious that $\|R_t - u_t v^T\|_F^2$ can only be minimal if $v_2 = 0$. Our assumption implies that x_1 is nonempty and $x_1 > 0$. Moreover $[R_t^T u_t]_+ \neq 0$ and $u_t \geq 0$ imply $\|u_t\|_2^2 > 0$, one can then find the optimal v_1 by minimizing the remaining quadratic function

$$\|R_t\|_F^2 - 2 v_1^T x_1 + \|u_t\|_2^2 v_1^T v_1$$

which yields the solution $v_1 = \frac{x_1}{\|u_t\|_2^2}$. Putting the two components together yields the result

$$v_* = \frac{[R_t^T u_t]_+}{\|u_t\|_2^2} \quad \text{and} \quad \|R_t - u_t v_*^T\|_F^2 = \|R_t\|_F^2 - \frac{\|[R_t^T u_t]_+\|_2^2}{\|u_t\|_2^2}.$$

□

Algorithm 5 (RRI)

Initialize u_i's, v_i's, for $i = 1$ to r
repeat
  for $t = 1$ to r **do**
    $R_t = A - \sum_{i \neq t} u_i v_i^T$

    if $[R_t^T u_t]_+ \neq 0$ **then**
      $v_t \leftarrow \frac{[R_t^T u_t]_+}{\|u_t\|_2^2}$
    else
      $v_t = 0$
    end if

    if $[R_t v_t]_+ \neq 0$ **then**
      $u_t \leftarrow \frac{[R_t v_t]_+}{\|v_t\|_2^2}$
    else
      $u_t = 0$
    end if
  end for
until Stopping condition

Remark 1 The above lemma has of course a dual form, where one fixes v_t but solves for the optimal u to minimize $\|R_t - uv_t^T\|_F^2$. This would yield the updating rules

$$v_t \leftarrow \frac{[R_t^T u_t]_+}{\|u_t\|_2^2} \quad \text{and} \quad u_t \leftarrow \frac{[R_t v_t]_+}{\|v_t\|_2^2} \tag{13.23}$$

which can be used to recursively update approximations $\sum_{i=1}^r u_i v_i^T$ by modifying each rank-one matrix $u_t v_t^T$ in a cyclic manner. This problem is different from the NMF, since the error matrices $R_t = A - \sum_{i \neq t} u_i v_i^T$ are no longer nonnegative. We will therefore call this method the *Rank-one Residue Iteration* (RRI), i.e. Algorithm 5. The same algorithm was independently reported as Hierarchical Alternating Least Squares (HALS) [7].

Remark 2 In case where $[R_t^T u_t]_+ = 0$, we have a trivial solution for $v = 0$ that is not covered by Lemma 5.1. In addition, if $u_t = 0$, this solution is no longer unique. In fact, v can be arbitrarily taken to construct a rank-deficient approximation. The effect of this on the convergence of the algorithm will be discussed further in the next section.

Remark 3 Notice that the optimality of Lemma 5.1 implies that $\|A - UV^T\|$ can not increase. And since $A \geq 0$ fixed, $UV^T \geq 0$ must be bounded. Therefore, its component $u_i v_i^t (i = 1 \ldots r)$ must be bounded as well. One can moreover scale the vector pairs (u_i, v_i) at each stage as explained in Sect. 13.4 without affecting the

local optimality of Lemma 5.1. It then follows that the rank one products $u_i v_i^T$ and their scaled vectors remain bounded.

13.5.2 Convergence

In the previous section, we have established the partial updates for each of the variable u_i or v_i. And for a NMF problem where the reduced rank is r, we have in total $2r$ vector variables (the u_i's and v_i's). The described algorithm can be also considered as a projected gradient method since the update (13.23) can be rewritten as:

$$u_t \leftarrow \frac{[R_t v_t]_+}{\|v_t\|_2^2} = \frac{[(A - \sum_{i \neq t} u_i v_i^T) v_t]_+}{\|v_t\|_2^2} = \frac{[(A - \sum_i u_i v_i^T + u_t v_t^T) v_t]_+}{\|v_t\|_2^2}$$
$$= \frac{[(A - \sum_i u_i v_i^T) v_t + u_t v_t^T v_t]_+}{\|v_t\|_2^2} = \left[u_t - \frac{1}{\|v_t\|_2^2} \nabla_{u_t} \right]_+ .$$

Similarly, the update for v_i can be rewritten as

$$v_t \leftarrow \left[v_t - \frac{1}{\|u_t\|_2^2} \nabla_{v_t} \right]_+ .$$

Therefore, the new method follows the projected gradient scheme described in the previous section. But it produces the optimal solution in closed form. For each update of a column v_t (or u_t), the proposed algorithm requires just a matrix–vector multiplication $R_t^T u_t$ (or $R_t v_t$), wherein the residue matrix $R_t = A - \sum_{i \neq t} u_i v_i^T$ does not have to be calculated explicitly. Indeed, by calculating $R_t^T u_t$ (or $R_t v_t$) from $A^T u_t$ (or $A v_t$) and $\sum_{i \neq t} v_i (u_i^T u_t)$ (or $\sum_{i \neq t} u_i (v_i^T v_t)$), the complexity is reduced from $O(mnr + mn)$ to only $O(mn + (m + n)(r - 1))$ which is majored by $O(mn)$. This implies that the complexity of each sweep through the $2r$ variables u_t's and v_t's requires only $O(mnr)$ operations, which is equivalent to a sweep of the multiplicative rules and to an inner loop of any gradient methods. This is very low since the evaluation of the whole gradient requires already the same complexity.

Because at each step of the $2r$ basic steps of Algorithm 5, we compute an optimal rank-one nonnegative correction to the corresponding error matrix R_t the Frobenius norm of the error can not increase. This is a reassuring property but it does not imply convergence of the algorithm.

Each vector u_t or v_t lies in a convex set $\mathbb{U}_t \subset \mathbb{R}_+^m$ or $\mathbb{V}_t \subset \mathbb{R}_+^n$. Moreover, because of the possibility to include scaling we can set an upper bound for $\|U\|$ and $\|V\|$, in such a way that all the $\mathbb{U}_t$ and $\mathbb{V}_t$ sets can be considered as closed convex. Then, we can use the following Theorem 5.1, to prove a stronger convergence result for Algorithm 5.

Theorem 5.1 *Every limit point generated by Algorithm 1 is a stationary point.*

Proof We notice that, if $u_t = 0$ and $v_t = 0$ at some stages of Algorithm 5, they will remain zero and no longer take part in all subsequent iterations. We can divide the execution of Algorithm 1 into two phases.

During the first phase, some of the pairs (u_t, v_t) become zero. Because there are only a finite number $(2r)$ of such vectors, the number of iterations in this phase is also finite. At the end of this phase, we can rearrange and partition the matrices U and V such that

$$U = (U_+ \; 0) \quad \text{and} \quad V = (V_+ \; 0),$$

where U_+ and V_+ do not have any zero column. We temporarily remove zero columns out of the approximation.

During the second phase, no column of U_+ and V_+ becomes zero, which guarantees the updates for the columns of U_+ and V_+ are unique and optimal. Moreover, $\frac{1}{2}\|A - \sum_{i=1}^{r} u_i v_i^T\|_F^2$ is continuously differentiable over the set $\mathbb{U}_1 \times \cdots \times \mathbb{U}_r \times \mathbb{V}_1 \times \cdots \times \mathbb{V}_r$, and the $\mathbb{U}_i$'s and $\mathbb{V}_i$'s are closed convex. A direct application of Proposition 2.7.1 in [4] proves that every stationary point (U_+^*, V_+^*) is a stationary point. It is then easy to prove that if there are zero columns removed at the end of the first phase, adding them back yields another stationary point: $U^* = (U_+^* \; 0)$ and $V^* = (V_+^* \; 0)$ of the required dimension. However, in this case, the rank of the approximation will then be lower than the requested dimension r. □

In Algorithm 5, variables are updated in this order: $u_1, v_1, u_2, v_2, \ldots$. We can alternate the variables in a different order as well, for example $u_1, u_2, \ldots, u_r v_1$, $v_2, \ldots, v_r, \ldots$. Whenever this is carried out in a cyclic fashion, the Theorem 5.1 still holds and this does not increase the complexity of each iteration of the algorithm.

As pointed above, stationary points given by Algorithm 5 may contain useless zero components. To improve this, one could replace $u_t v_t^T (\equiv 0)$ by any nonnegative rank-one approximation that reduces the norm of the error matrix. For example, the substitution

$$u_t = e_{i^*} \quad v_t = [R_t^T u_t]_+, \tag{13.24}$$

where $i^* = \text{argmax}_i \|[R_t^T e_i]_+\|_2^2$, reduces the error norm by $\|[R_t^T e_i]_+\|_2^2 > 0$ unless $R_t \leq 0$. These substitutions can be done as soon as u_t and v_t start to be zero. If we do these substitutions in only a *finite* number of times before the algorithm starts to converge, Theorem 5.1 still holds. In practice, only a few such substitutions in total are usually needed by the algorithm to converge to a stationary point without any zero component. Note that the matrix rank of the approximation might not be r, even when all u_t's and v_t's $(t = 1 \ldots r)$ are nonzero.

A possibly better way to fix the problem due to zero components is to use the following *damped RRI algorithm* in which we introduce new $2r$ dummy variables $w_i \in \mathbb{U}_i$ and $z_i \in \mathbb{V}_i$, where $i = 1 \ldots r$. The new problem to solve is:

Problem 3 *(Damped Nonnegative Matrix Factorization)*

$$\min_{\substack{u_i \geq 0\ v_i \geq 0 \\ w_i \geq 0\ z_i \geq 0}} \frac{1}{2}\|A - \sum_{i=1}^{r} u_i v_i^T\|_F^2 + \frac{\psi}{2}\sum_i \|u_i - w_i\|_2^2 + \frac{\psi}{2}\sum_i \|v_i - z_i\|_2^2,$$

where the damping factor ψ is a positive constant.

Again, the coordinate descent scheme is applied with the cyclic update order: $u_1, w_1, v_1, z_1, u_2, w_2, v_2, z_2, \ldots$ to result in the following optimal updates for u_t, v_t, w_t and z_t:

$$u_t = \frac{[R_t v_t]_+ + \psi w_t}{\|v_t\|_2^2 + \psi}, \quad w_t = u_t, \quad v_t = \frac{[R_t^T u_t]_+ + \psi z_t}{\|u_t\|_2^2 + \psi} \quad \text{and} \quad z_t = v_t \tag{13.25}$$

where $t = 1 \ldots r$. The updates $w_t = u_t$ and $z_t = v_t$ can be integrated in the updates of u_t and v_t to yield Algorithm 6. We have the following results:

Theorem 5.2 *Every limit point generated by Algorithm 6 is a stationary point of NMF Problem 2.*

Algorithm 6 (Damped RRI)

1: Initialize u_i's, v_i's, for $i = 1$ to r
2: **repeat**
3: **for** $t = 1$ to r **do**
4: $R_t = A - \sum_{i \neq t} u_i v_i^T$
5: $v_t \leftarrow \frac{[R_t^T u_t + \psi v_t]_+}{\|u_t\|_2^2 + \psi}$
6: $u_t \leftarrow \frac{[R_t v_t + \psi u_t]_+}{\|v_t\|_2^2 + \psi}$
7: **end for**
8: **until** Stopping condition

Proof Clearly the cost function in Problem 3 is continuously differentiable over the set $\mathbb{U}_1 \times \cdots \times \mathbb{U}_r \times \mathbb{U}_1 \times \cdots \times \mathbb{U}_r \times \mathbb{V}_1 \times \cdots \times \mathbb{V}_r \times \mathbb{V}_1 \times \cdots \times \mathbb{V}_r$, and the $\mathbb{U}_i$'s and $\mathbb{V}_i$'s are closed convex. The uniqueness of the global minimum of the elementary problems and a direct application of Proposition 2.7.1 in [4] prove that every limit point of Algorithm 6 is a stationary point of Problem 3.

Moreover, at a stationary point of Problem 3, we have $u_t = w_t$ and $v_t = z_t, t = 1 \ldots r$. The cost function in Problem 3 becomes the cost function of the NMF Problem 2. This implies that every stationary point of Problem 3 yields a stationary point of the standard NMF Problem 2. □

This *damped* version not only helps to eliminate the problem of zero components in the convergence analysis but may also help to avoid zero columns in the approximation when ψ is carefully chosen. But it is not an easy task. Small values of ψ provide an automatic treatment of zeros while not changing much the updates

of RRI. Larger values of ψ might help to prevent the vectors u_t and v_t $(t = 1 \ldots r)$ from becoming zero too soon. But too large values of ψ limit the updates to only small changes, which will slow down the convergence.

In general, the rank of the approximation can still be lower than the requested dimension. Patches may still be needed when a zero component appears. Therefore, in our experiments, using the *undamped* RRI Algorithm 5 with the substitution (13.24) is still the best choice.

13.5.3 Variants of the RRI Method

We now extend the Rank-one Residue Iteration by using a factorization of the type XDY^T where D is diagonal and nonnegative and the columns of the nonnegative matrices X and Y are normalized. The NMF formulation then becomes

$$\min_{x_i \in \mathbb{X}_i \; y_i \in \mathbb{Y}_i d_i \in \mathbb{R}_+} \frac{1}{2} \left\| A - \sum_{i=1}^{r} d_i x_i y_i^T \right\|_F^2,$$

where $\mathbb{X}_i$'s and $\mathbb{Y}_i$'s are sets of normed vectors.

The variants that we present here depend on the choice of $\mathbb{X}_i$'s and $\mathbb{Y}_i$'s. A generalized Rank-one Residue Iteration method for low-rank approximation is given in Algorithm 7. This algorithm needs to solve a sequence of elementary problems of the type:

$$\max_{s \in \mathbb{S}} y^T s \tag{13.26}$$

where $y \in \mathbb{R}^n$ and $\mathbb{S} \subset \mathbb{R}^n$ is a set of normed vectors. We first introduce a permutation vector $I_y = (i_1 \; i_2 \; \ldots \; i_n)$ which reorders the elements of y in nonincreasing order: $y_{i_k} \geq y_{i_{k+1}}, k = 1 \ldots (n-1)$. The function $p(y)$ returns the number of positive entries of y.

Algorithm 7 GRRI

1: Initialize x_i's, y_i's and d_i's, for $i = 1$ to r
2: **repeat**
3: **for** $i = 1$ to r **do**
4: $R_i = A - \sum_{j \neq i} d_j x_j y_j^T$
5: $y_i \leftarrow \text{argmax}_{s \in \mathbb{Y}_i} \left(x_i^T R_i s \right)$
6: $x_i \leftarrow \text{argmax}_{s \in \mathbb{X}_i} \left(y_i^T R_i^T c \right)$
7: $d_i = x_i^T R_i y_i$
8: **end for**
9: **until** Stopping condition

Let us first point out that for the set of normed nonnegative vectors the solution of problem (13.26) is given by $s^* = \frac{y_+}{\|y_+\|_2}$. It then follows that Algorithm 7 is essentially the same as Algorithm 5 since the solutions v_i and u_i of each step of Algorithm 7, given by (13.23), correspond exactly to those of problem (13.26) via the relations $y_i = u_i/\|u_i\|_2, y_i = v_i/\|v_i\|_2$ and $d_i = \|u_i\|_2\|v_i\|_2$.

Below we list the sets for which the solution s^* of (13.26) can be easily computed.

- *Set of normed vectors*: $s = \frac{y}{\|y\|_2}$. This is useful when one wants to create factorizations where only one of the factor U or V is nonnegative and the other is real matrix.
- *Set of normed nonnegative vectors*: $s = \frac{y_+}{\|y_+\|_2}$.
- *Set of normed bounded nonnegative vectors* $\{s\}$: where $0 \le l_i \le s_i \le p_i$. The optimal solution of (13.26) is given by:

$$s = \max\left(l,\ \min\left(p,\ \frac{y_+}{\|y_+\|_2}\right)\right).$$

- *Set of normed binary vectors* $\{s\}$: where $s = \frac{b}{\|b\|}$ and $b \in \{0,1\}^n$. The optimal solution of (13.26) is given by:

$$[s^*]_{i_t} = \begin{cases} \frac{1}{\sqrt{k^*}} & \text{if } t \le k^* \\ 0 & \text{otherwise} \end{cases} \quad \text{where } k^* = \underset{k}{\operatorname{argmax}} \frac{\sum_{t=1}^{k} y_{i_t}}{\sqrt{k}}.$$

- *Set of normed sparse nonnegative vectors*: all normed nonnegative vectors having at most K nonzero entries. The optimal solution for (13.26) is given by norming the following vector p^*

$$[p^*]_{i_t} = \begin{cases} y_{i_t} & \text{if } t \le \min(p(y), K) \\ 0 & \text{otherwise} \end{cases}$$

- *Set of normed fixed-sparsity nonnegative vectors*: all nonnegative vectors s a fixed sparsity, where

$$\text{sparsity}(s) = \frac{\sqrt{n} - \|s\|_1/\|s\|_2}{\sqrt{n} - 1}.$$

The optimal solution for (13.26) is given by using the projection scheme in [15].

One can also imagine other variants, for instance by combining the above ones. Depending on how data need to be approximated, one can create new algorithms provided it is relatively simple to solve problem (13.26). There have been some particular ideas in the literatures such as NMF with sparseness constraint [15], Semidiscrete Matrix Decomposition [16] and Semi-Nonnegative Matrix

Factorization [9] for which variants of the above scheme can offer an alternative choice of algorithm.

Remark Only the first three sets are the normed version of a closed convex set, as required for the convergence by Theorem 5.1. Therefore the algorithms might not converge to a stationary point with the other sets. However, the algorithm always guarantees a non-increasing update even in those cases and can therefore be expected to return a *good* approximation.

13.5.4 Nonnegative Tensor Factorization

If we refer to the problem of finding the nearest nonnegative vector to a given vector a as the nonnegative approximation in one dimension, the NMF is its generalization in two dimensions and naturally, it can be extended to even higher-order tensor approximation problems. Algorithms described in the previous section use the closed form solution of the one dimensional problem to solve the two-dimensional problem. We now generalize this to higher orders. Since in one dimension such an approximation is easy to construct, we continue to use this approach to build the solutions for higher order problems.

For a low-rank tensor, there are two popular kinds of factored tensors, namely those of Tucker and Kruskal [2]. We only give an algorithm for finding approximations of Kruskal type. It is easy to extend this to tensors of Tucker type, but this is omitted here.

Given a d dimensional tensor T, we will derive an algorithm for approximating a nonnegative tensor by a rank-r nonnegative Kruskal tensor $S \in \mathbb{R}_+^{n_1 \times n_2 \times \cdots \times n_d}$ represented as a sum of r rank-one tensors:

$$S = \sum_{i=1}^{r} \sigma_i u_{1i} \star u_{2i} \star \cdots \star u_{di}$$

where $\sigma_i \in \mathbb{R}_+$ is a scaling factor, $u_{ti} \in \mathbb{R}_+^{n_t}$ is a normed vector (i.e. $\|u_{ti}\|_2 = 1$) and $a \star b$ stands for the outer product between two vectors or tensors a and b.

The following update rules are the generalization of the matrix case to the higher order tensor:

$$y = (\ldots((\ldots(R_k u_{1k}) \ldots u_{(t-1)k}) u_{(t+1)k}) \ldots) u_{dk} \tag{13.27}$$

$$\sigma_k = \|[y]_+\|_2, \quad u_{tk} = \frac{[y]_+}{\sigma_k}, \tag{13.28}$$

where $R_k = T - \sum_{i \neq k} \sigma_i u_{1i} \star u_{2i} \star \cdots \star u_{di}$ is the residue tensor calculated without the kth component of S and $R_k u_{ij}$ is the ordinary tensor/vector product in the corresponding dimension.

We can then produce an algorithm which updates in a cyclic fashion all vectors u_{ji}. This is in fact a direct extension to Algorithm 5, one can carry out the same discussion about the convergence here to guarantee that each limit point of this algorithm is a stationary point for the nonnegative tensor factorization problem and to improve the approximation quality.

Again, as we have seen in the previous section, we can extend the procedure to take into account different constraints on the vectors u_{ij} such as discreteness, sparseness, etc.

The approach proposed here is again different from that in [8] where a similar cascade procedure for multilayer nonnegative matrix factorization is used to compute a 3D tensor approximation. Clearly, the approximation error will be higher than our proposed method, since the cost function is minimized by taking into account all the dimensions.

13.5.5 Regularizations

The regularizations are common methods to cope with the ill-posedness of inverse problems. Having known some additional information about the solution, one may want to imposed a priori some constraints to algorithms, such as: smoothness, sparsity, discreteness, etc. To add such regularizations in to the RRI algorithms, it is possible to modify the NMF cost function by adding some regularizing terms. We will list here the update for u_i's and v_i's when some simple regularizations are added to the original cost function. The proofs of these updates are straightforward and hence omitted.

- One-Norm $\|.\|_1$ regularization: the one-norm of the vector variable can be added as a heuristic for finding a sparse solution. This is an alternative to the fixed-sparsity variant presented above. The regularized cost function with respect to the variable v_t will be

$$\frac{1}{2}\|R_t - u_t v^T\|_F^2 + \beta\|v\|_1, \quad \beta > 0$$

  where the optimal update is given by

$$v_t^* = [\frac{R_t^T u_t - \beta \mathbf{1}_n \times 1]_+}{\|u_t\|_2^2}$$

  The constant $\beta > 0$ can be varied to control the trade-off between the approximation error $\frac{1}{2}\|R_t - u_t v^T\|_F^2$ and $\|v\|_1$. From this update, one can see that this works by zeroing out elements of $R_t^T u_t$ which are smaller than β, hence reducing the number of nonzero elements of v_t^*.

- Smoothness regularization $\|v - B\hat{v}_t\|_F^2$: where $\hat{v}_t$ is the current value of v_t and the matrix B helps to calculate the average of the neighboring elements at each element of v. When v is a 1D smooth function, B can be the following $n \times n$ matrix:

$$B = \begin{pmatrix} 0 & 1 & \dots & \dots & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix}. \tag{13.29}$$

  This matrix can be defined in a different way to take the true topology of v into account, for instance $v = \text{vec}(F)$ where F is a matrix. The regularized cost function with respect to the variable v_t will be

$$\frac{1}{2}\|R_t - u_t v^T\|_F^2 + \frac{\delta}{2}\|v - B\hat{v}_t\|_F^2, \quad \delta > 0$$

  where the optimal update is given by

$$v_t^* = \frac{[R_t^T u_t + \delta B\hat{v}_t]_+}{\|u_t\|_2^2 + \delta}.$$

  The constant $\delta \geq 0$ can be varied to control the trade-off between the approximation error $\frac{1}{2}\|R_t - u_t v^T\|_F^2$ and the smoothness of v_t at the fixed point. From the update, one can see that this works by searching for the optimal update v_t^* with some preference for the neighborhood of $B\hat{v}_i$, i.e., a smoothed vector of the current value $\hat{v}_t$.

The two above regularizations can be added independently to each of the columns of U and/or V. The trade-off factor β (or δ) can be different for each column. A combination of different regularizations on a column (for instance v_t) can also be used to solve the multi-criterion problem

$$\frac{1}{2}\|R_t - u_t v^T\|_F^2 + \frac{\gamma}{2}\|v\|_2^2 + \frac{\delta}{2}\|v - B\hat{v}_t\|_F^2, \quad \beta, \gamma, \delta > 0$$

where the optimal update is given by

$$v_t^* = \frac{[R_t^T u_t - \beta \mathbf{1}_{n\times 1} + \delta B\hat{v}_t]_+}{\|u_t\|_2^2 + \delta}.$$

The one-norm regularizations as well as the two-norm regularization can be found in [1, 3]. A major difference with that method is that the norm constraints is added to the rows rather than on the columns of V or U as done here. However, for

the two versions of the one-norm regularization, the effects are somehow similar. While the two-norm regularization on the columns of U and V are simply scaling effects, which yield nothing in the RRI algorithm. We therefore only test the smoothness regularization at the end of the chapter with some numerical generated data.

For more extensions and variants, see [13].

13.6 Experiments

Here we present several experiments to compare the different descent algorithms presented in this paper. For all the algorithms, the scaling scheme proposed in Sect. 13.4 was applied.

13.6.1 Random Matrices

We generated 100 random nonnegative matrices of different sizes. We used seven different algorithms to approximate each matrix:

- the multiplicative rule (*Mult*),
- alternative least squares using Matlab function *lsqnonneg* (*ALS*),
- a full space search using line search and Armijo criterion (*FLine*),
- a coordinate search alternating on U and V, and using line search and Armijo criterion (*CLine*),
- a full space search using first-order approximation (*FFO*),
- a coordinate search alternating on U and V, and using first-order approximation (*CFO*)
- an iterative rank-one residue approximation (*RRI*).

For each matrix, the same starting point is used for every algorithm. We create a starting point by randomly generating two matrices U and V and then rescaling them to yield a first approximation of the original matrix A as proposed in Sect. 13.4:

$$U = UD\sqrt{\alpha}, \quad V = VD^{-1}\sqrt{\alpha},$$

where

$$\alpha := \frac{\langle A, UV^T \rangle}{\langle UV^T, UV^T \rangle} \quad \text{and} \quad D_{ij} = \begin{cases} \sqrt{\frac{\|V_{:i}\|_2}{\|U_{:i}\|_2}} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}.$$

From (13.15), we see that when approaching a KKT stationary point of the problem, the above scaling factor $\alpha \to 1$. This implies that every KKT stationary point of this problem is scale-invariant.

The algorithms are all stopped when the projected gradient norm is lower than ϵ times the gradient norm at the starting point or when it takes more than 45 s. The relative precisions ϵ are chosen equal to $10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$. No limit was imposed on the number of iterations.

For alternative gradient algorithms *CLine* and *CFO*, we use different precisions ϵ_U and ϵ_V for each of the inner iteration for U and for V as suggested in [19] where ϵ_U and ϵ_V are initialized by 10^{-3}. And when the inner loop for U or V needs no iteration to reach the precision ϵ_U or ϵ_V, one more digit of precision will be added into ϵ_U or ϵ_V (i.e. $\epsilon_U = \epsilon_U/10$ or $\epsilon_V = \epsilon_V/10$).

Table 13.1 shows that for all sizes and ranks, Algorithm *RRI* is the fastest to reach the required precision. Even though it is widely used in practice, algorithm *Mult* fails to provide solutions to the NMF problem within the allocated time. A further investigation shows that the algorithm gets easily trapped in boundary points where some U_{ij} and/or V_{ij} is zero while $\nabla_{U_{ij}}$ and/or $\nabla_{V_{ij}}$ is negative, hence violating one of the KKT conditions (13.11). The multiplicative rules then fail to move and do not return to a local minimizer. A slightly modified version of this algorithm was given in [20], but it needs to wait to get sufficiently close to such points before attempting an escape, and is therefore also not efficient. The *ALS* algorithm can return a stationary point, but it takes too long.

We select five methods: *FLine, CLine, FFO, CFO* and *RRI* for a more detailed comparison. For each matrix A, we run these algorithms with 100 different starting points. Figures 13.1, 13.2, 13.3 and 13.4 show the results with some different settings. One can see that, when the approximated errors are almost the same between the algorithms, *RRI* is the best overall in terms of running times. It is probably because the RRI algorithm chooses only one vector u_t or v_t to optimize at once. This allows the algorithm to move *optimally* down on partial direction rather than just a *small step* on a more global direction. Furthermore, the computational load for an update is very small, only one matrix–vector multiplication is needed. All these factors make the running time of the RRI algorithm very attractive.

13.6.2 Image Data

The following experiments use the Cambridge ORL face database as the input data. The database contains 400 images of 40 persons (10 images per person). The size of each image is 112×92 with 256 gray levels per pixel representing a front view of the face of a person. The images are then transformed into 400 "face vectors" in $\mathbb{R}^{10,304}(112 \times 92 = 10,304)$ to form the data matrix A of size $10,304 \times 400$. We used three weight matrices of the same size of A (ie. $10,304 \times 400$). Since it was used in [18], this data has become the standard benchmark for NMF algorithms.

In the first experiment, we run six NMF algorithms described above on this data for the reduced rank of 49. The original matrix A is constituted by transforming each image into one of its column. Figure 13.5 shows for the six algorithms the evolution of the error versus the number of iterations. Because the minimization

Table 13.1 Comparison of average successful running time of algorithms over 100 random matrices

ϵ	Mult	ALS	FLine	CLine	FFO	CFO	RRI
$(m = 30,\ n = 20,\ r = 2)$							
10^{-2}	0.02(96)	0.40	0.04	0.02	0.02	0.01	0.01
10^{-3}	0.08(74)	1.36	0.12	0.09	0.05	0.04	0.03
10^{-4}	0.17(71)	2.81	0.24	0.17	0.11	0.08	0.05
10^{-5}	0.36(64)	4.10	0.31	0.25	0.15	0.11	0.07
10^{-6}	0.31(76)	4.74	0.40	0.29	0.19	0.15	0.09
$(m = 100,\ n = 50,\ r = 5)$							
10^{-2}	$45*(0)$	3.48	0.10	0.09	0.09	0.04	0.02
10^{-3}	$45*(0)$	24.30(96)	0.59	0.63	0.78	0.25	0.15
10^{-4}	$45*(0)$	$45*(0)$	2.74	2.18	3.34	0.86	0.45
10^{-5}	$45*(0)$	$45*(0)$	5.93	4.06	6.71	1.58	0.89
10^{-6}	$45*(0)$	$45*(0)$	7.23	4.75	8.98	1.93	1.30
$(m = 100,\ n = 50,\ r = 10)$							
10^{-2}	$45*(0)$	11.61	0.28	0.27	0.18	0.11	0.05
10^{-3}	$45*(0)$	41.89(5)	1.90	2.11	1.50	0.74	0.35
10^{-4}	$45*(0)$	$45*(0)$	7.20	5.57	5.08	2.29	1.13
10^{-5}	$45*(0)$	$45*(0)$	12.90	9.69	10.30	4.01	1.71
10^{-6}	$45*(0)$	$45*(0)$	14.62(99)	11.68(99)	13.19	5.26	2.11
$(m = 100,\ n = 50,\ r = 15)$							
10^{-2}	$45*(0)$	25.98	0.66	0.59	0.40	0.20	0.09
10^{-3}	$45*(0)$	$45*(0)$	3.90	4.58	3.18	1.57	0.61
10^{-4}	$45*(0)$	$45*(0)$	16.55(98)	13.61(99)	9.74	6.12	1.87
10^{-5}	$45*(0)$	$45*(0)$	21.72(97)	17.31(92)	16.59(98)	7.08	2.39
10^{-6}	$45*(0)$	$45*(0)$	25.88(89)	19.76(98)	19.20(98)	10.34	3.66
$(m = 100,\ n = 100,\ r = 20)$							
10^{-2}	$45*(0)$	42.51(4)	1.16	0.80	0.89	0.55	0.17
10^{-3}	$45*(0)$	$45*(0)$	9.19	8.58	10.51	5.45	1.41
10^{-4}	$45*(0)$	$45*(0)$	28.59(86)	20.63(94)	29.89(69)	12.59	4.02
10^{-5}	$45*(0)$	$45*(0)$	32.89(42)	27.94(68)	34.59(34)	18.83(90)	6.59
10^{-6}	$45*(0)$	$45*(0)$	37.14(20)	30.75(60)	36.48(8)	22.80(87)	8.71
$(m = 200,\ n = 100,\ r = 30)$							
10^{-2}	$45*(0)$	$45*(0)$	2.56	2.20	2.68	1.31	0.44
10^{-3}	$45*(0)$	$45*(0)$	22.60(99)	25.03(98)	29.67(90)	12.94	4.12
10^{-4}	$45*(0)$	$45*(0)$	36.49(2)	39.13(13)	$45*(0)$	33.33(45)	14.03
10^{-5}	$45*(0)$	$45*(0)$	$45*(0)$	39.84(2)	$45*(0)$	37.60(6)	21.96(92)
10^{-6}	$45*(0)$	$45*(0)$	$45*(0)$	$45*(0)$	$45*(0)$	$45*(0)$	25.61(87)

Time limit is 45 s. 0.02(96) means that a result is returned with the required precision ϵ within 45 s for 96 (of 100) matrices of which the average running time is 0.02 s. $45*(0)$: failed in all 100 matrices

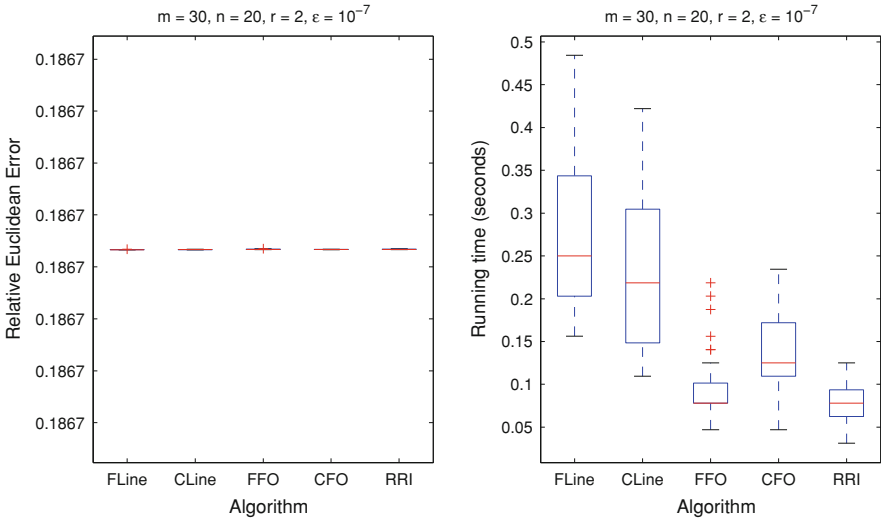


Fig. 13.1 Comparison of selected algorithms for $\epsilon = 10^{-7}$

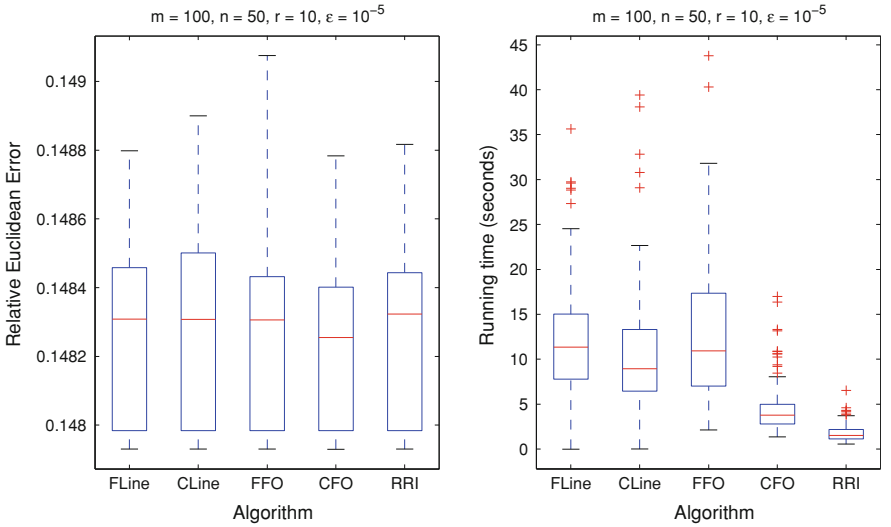


Fig. 13.2 Comparison of selected algorithms for $\epsilon = 10^{-5}$

process is different in each algorithm, we will say that one iteration corresponds to all elements of both U and V being updated. Figure 13.6 shows the evolution of the error versus time. Since the work of one iteration varies from one algorithm to another, it is crucial to plot the error versus time to get a fair comparison between the different algorithms. In the two figures, we can see that the RRI algorithm

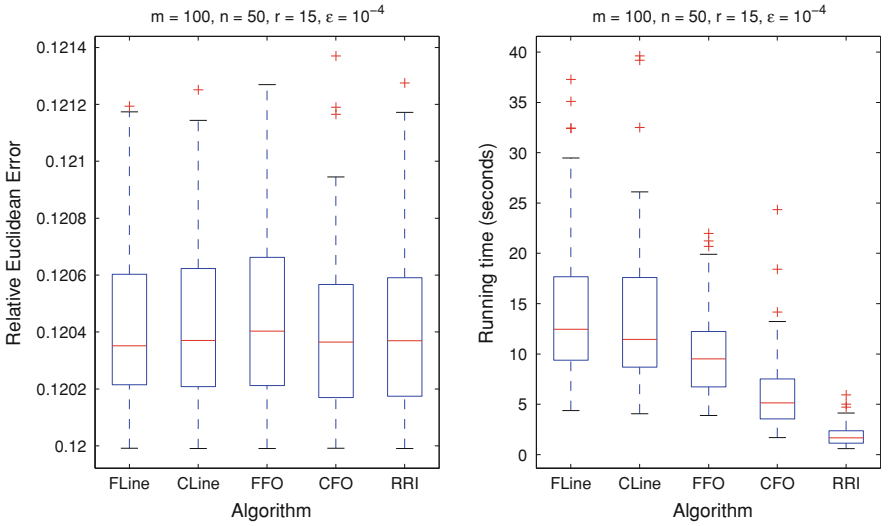


Fig. 13.3 Comparison of selected algorithms for $\epsilon = 10^{-4}$

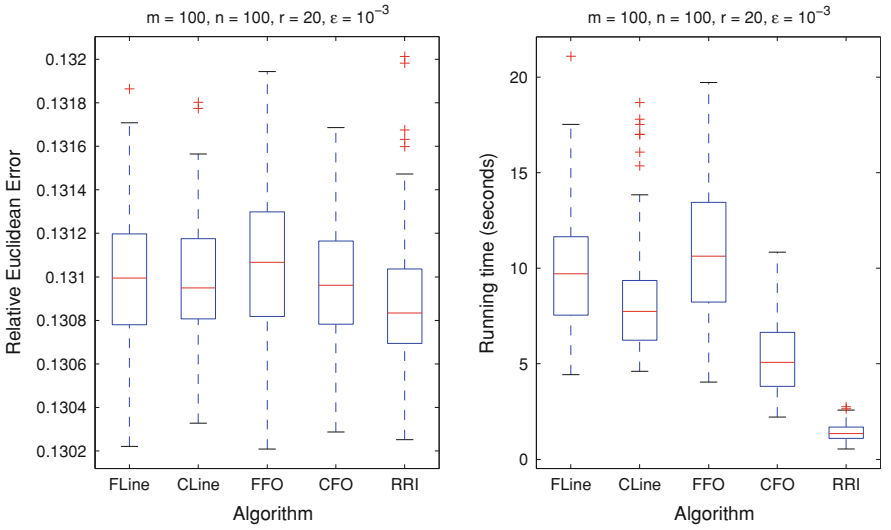


Fig. 13.4 Comparison of selected algorithms for $\epsilon = 10^{-3}$

behaves very well on this dataset. And since its computation load of each iteration is small and constant (without inner loop), this algorithm converges faster than the others.

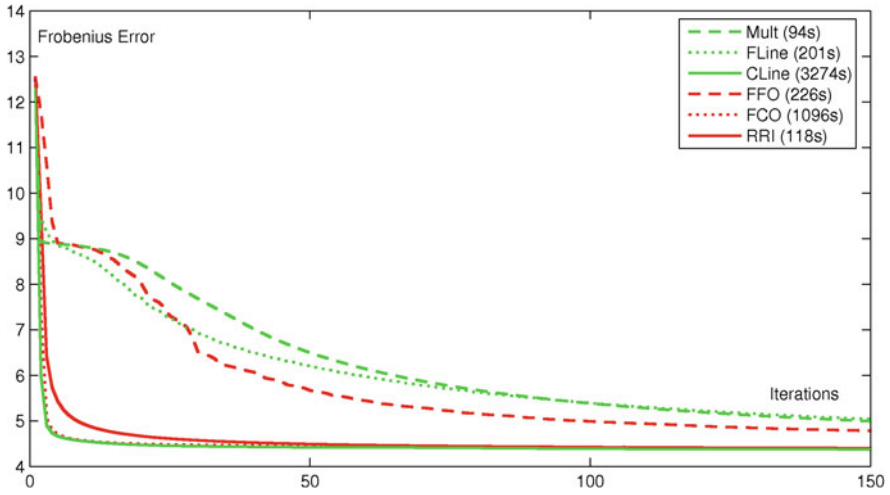


Fig. 13.5 NMF: error vs. iterations

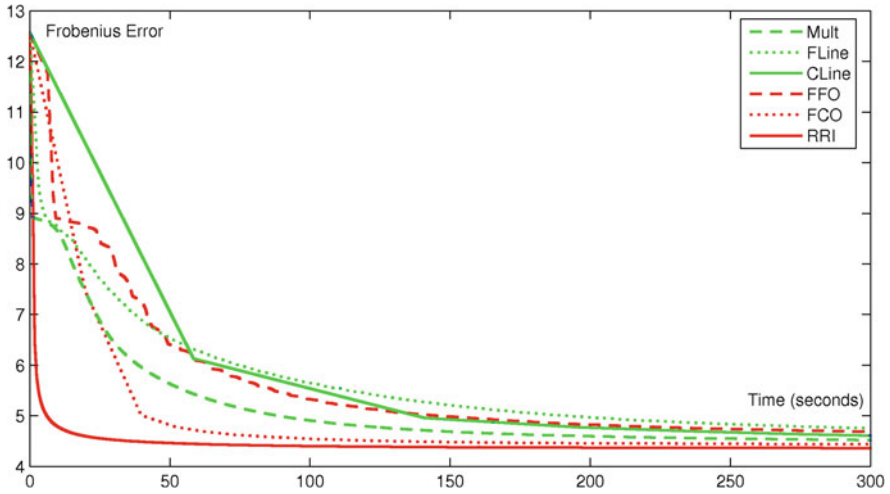


Fig. 13.6 NMF: error vs. time

In the second experiment, we construct a third-order nonnegative tensor approximation. We first build a tensor by stacking all 400 images to have a $112 \times 92 \times 400$ nonnegative tensor. Using the proposed algorithm, a rank-142 nonnegative tensor is calculated to approximate this tensor. Figure 13.7 shows the result for six images chosen randomly from the 400 images. Their approximations given by the rank-142 nonnegative tensor are much better than that given by the rank-8 nonnegative matrix, even though they require similar storage space:

Fig. 13.7 Tensor factorization vs. matrix factorization on facial data. Six randomly chosen images from 400 of ORL dataset. From top to bottom: original images, their rank-8 truncated SVD approximation, their rank-142 nonnegative tensor approximation (150 RRI iterations) and their rank-8 nonnegative matrix approximation (150 RRI iterations)

$8 * (112 * 92 + 400) = 85632$ and $142 * (112 + 92 + 400) = 85768$. The rank-8 truncated SVD approximation (i.e. $[A_8]_+$) is also included for reference.

In the third experiment, we apply the variants of RRI algorithm mentioned in Sect. 13.5 to the face databases. The following settings are compared:

Original: original faces from the databases.
49NMF: standard factorization (nonnegative vectors), $r = 49$.
100Binary: columns of U are limited to the scaled binary vectors, $r = 100$.
49Sparse10: columns of U are sparse. Not more than 10% of the elements of each column of A are positive. $r = 49$.
49Sparse20: columns of U are sparse. Not more than 20% of the elements of each column of A are positive. $r = 49$.
49HSparse60: columns of U are sparse. The Hoyer sparsity of each column of U are 0.6. $r = 49$.
49HSparse70: columns of U are sparse. The Hoyer sparsity of each column of U are 0.7. $r = 49$.

49HBSparse60: columns of U are sparse. The Hoyer sparsity of each column of U are 0.6. Columns of V are scaled binary. $r = 49$.
49HBSparse70: columns of U are sparse. The Hoyer sparsity of each column of U are 0.7. Columns of V are scaled binary. $r = 49$.

For each setting, we use RRI algorithm to compute the corresponding factorization. Some randomly selected faces are reconstructed by these settings as shown

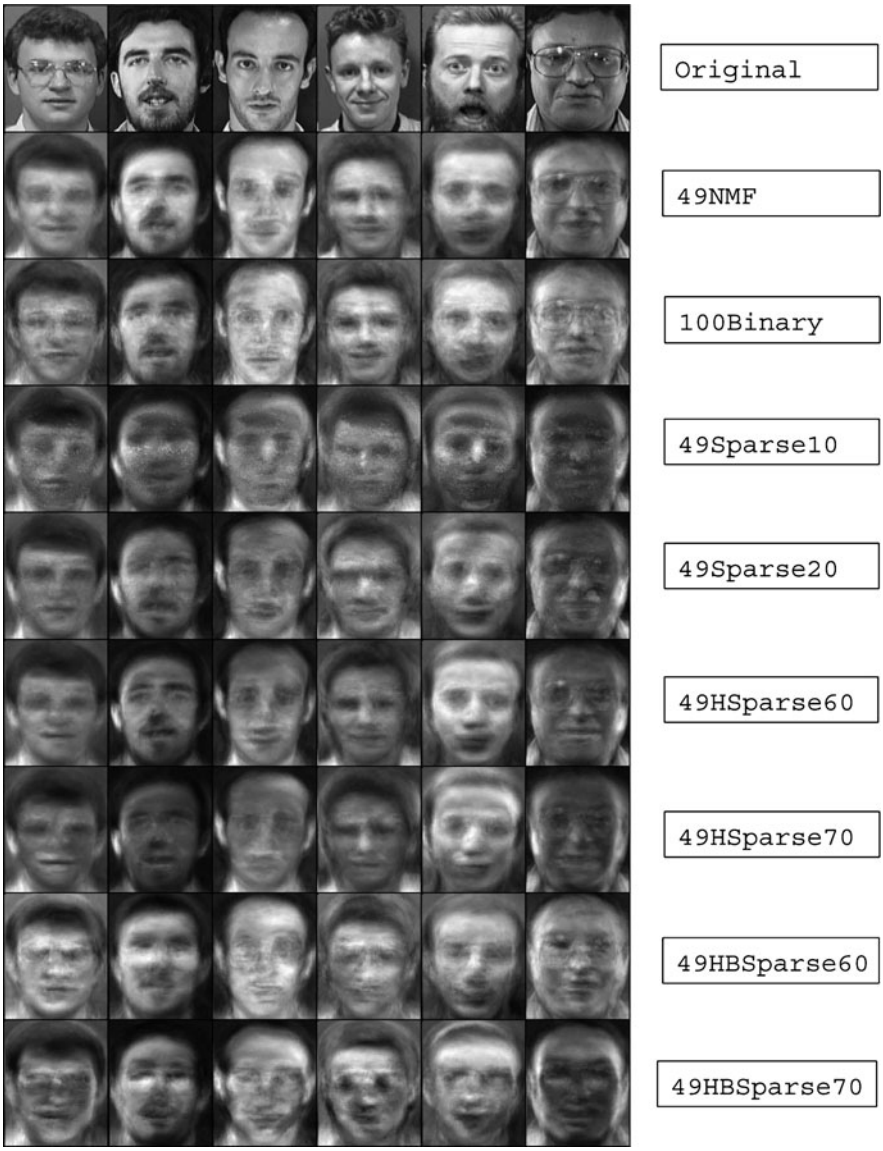


Fig. 13.8 Nonnegative matrix factorization with several sparse settings

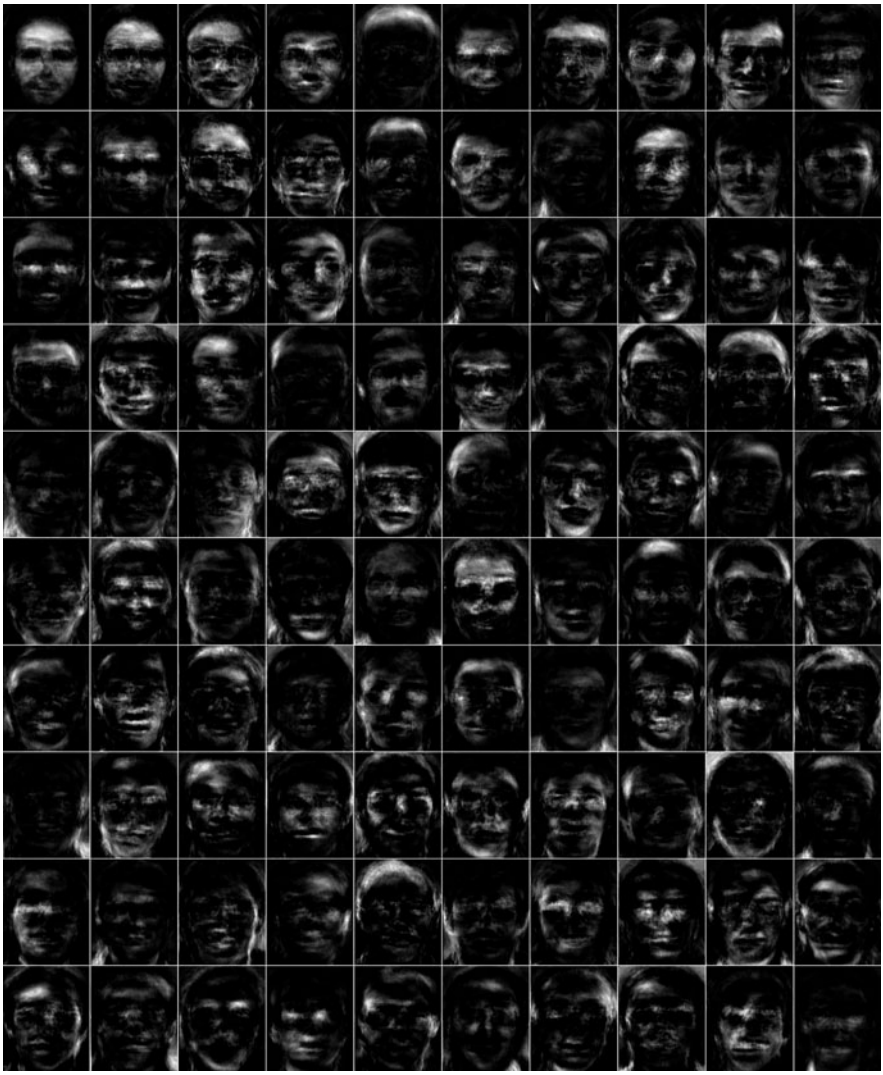


Fig. 13.9 Bases from *100Binary* setting

in Fig. 13.8. For each setting, RRI algorithm produces a different set of bases to approximate the original faces. When the columns of V are constrained to scaled binary vectors (*100Binary*), the factorization can be rewritten as $UV^T = \hat{U}B^T$, where B is a binary matrix. This implies that each image is reconstructed by just the presence or absence of 100 bases shown in Fig. 13.9.

Figures 13.10 and 13.11 show nonnegative bases obtained by imposing some sparsity on the columns of V. The sparsity can be easily controlled by the percentages of positive elements or by the Hoyer sparsity measure.

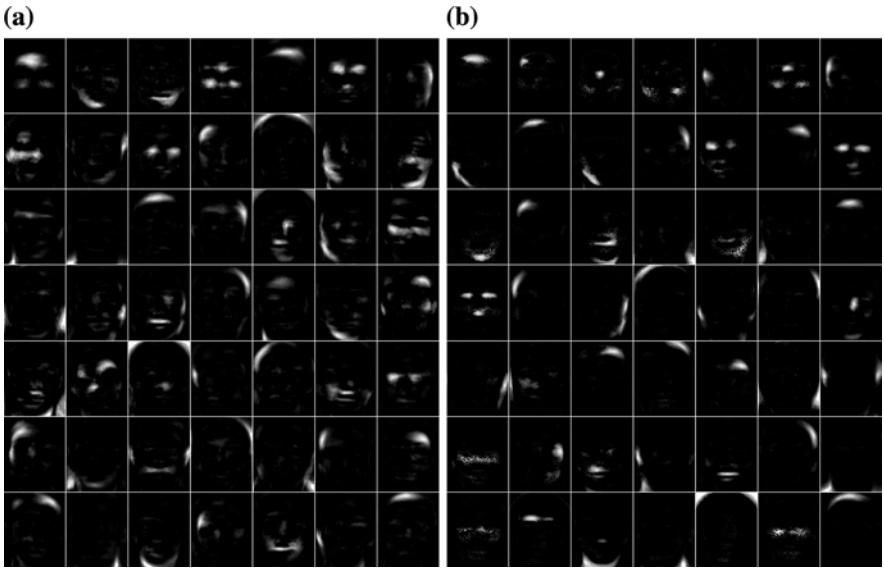


Fig. 13.10 Sparse bases *49Sparse20* and *49Sparse10*. Maximal percentage of positive elements is 20% (**a**) and 10% (**b**)

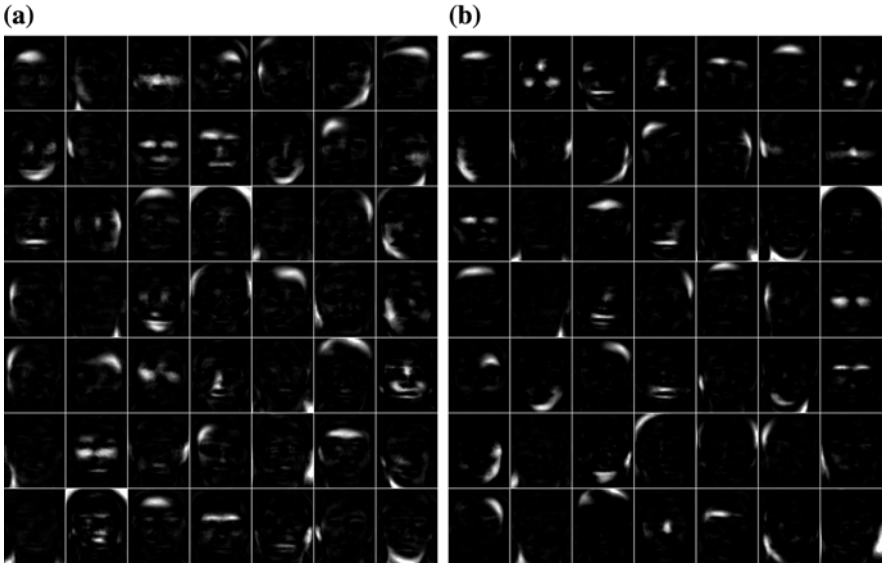


Fig. 13.11 Hoyer sparse bases *49HSparse60* and *49HSparse70*. Sparsity of bases is 0.6 (**a**) and 0.7 (**b**)

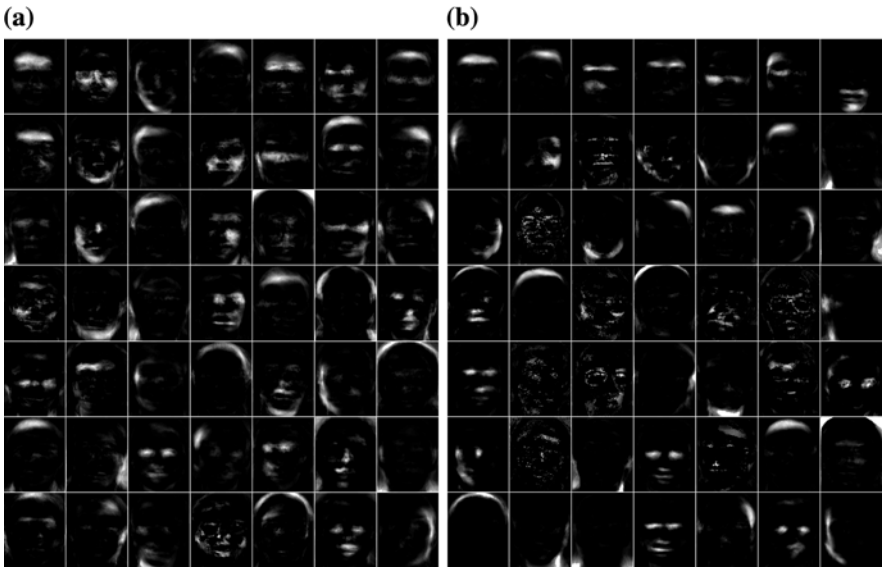


Fig. 13.12 Hoyer sparse bases *49HBSparse60* and *49HBSparse70*. Sparsity of bases is (**a**) and 0.7 (**b**). V is binary matrix

Figure 13.12 combines the sparsity of the bases (columns of U) and the binary representation of V. The sparsity is measured by the Hoyer measure as in Fig. 13.11. Only with the absence or presence of these 49 *features*, faces are approximated as showed in the last two rows of Fig. 13.8.

The above examples show how to use the variants of the RRI algorithm to control the sparsity of the bases. One can see that the sparser the bases are, the less storage is needed to store the approximation. Moreover, this provides a part-based decomposition using *local* features of the faces.

13.6.3 Smooth Approximation

We carry out this experiment to test the new smoothness constraint introduced in the previous section:

$$\frac{1}{2}\|R_i - u_i v^T\|_F^2 + \frac{\delta}{2}\|v - B\hat{v}_i\|_F^2, \quad \delta > 0$$

where B is defined in (13.29).

We generate the data using four smooth nonnegative functions f_1, f_2, f_3 and f_4, described in Fig. 13.13, where each function is represented as a nonnegative vector of size 200.

We then generate a matrix A containing 100 mixtures of these functions as follows

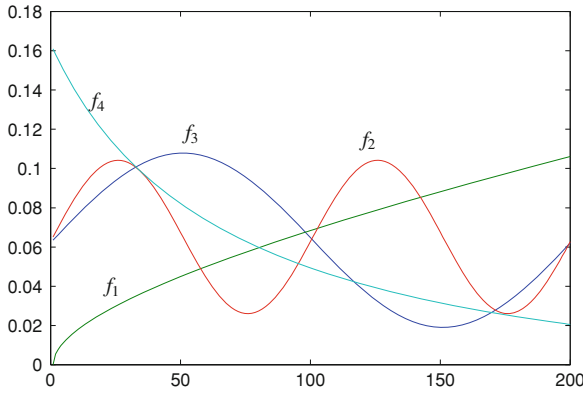


Fig. 13.13 Smooth functions

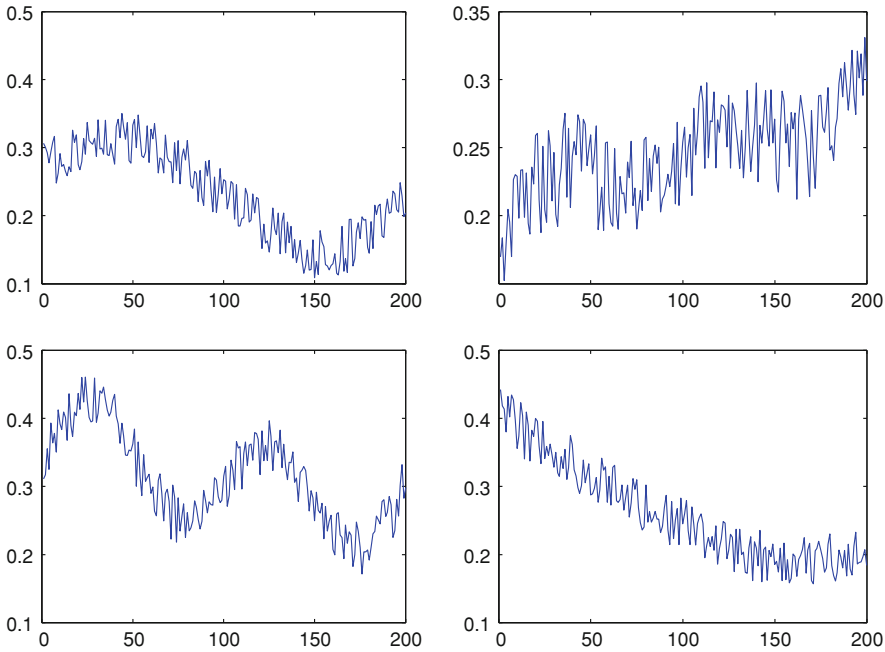


Fig. 13.14 Randomly selected generated data

$$A = \max(FE^T + N, 0)$$

where $F = [f_1\ f_2\ f_3\ f_4], E$ is a random nonnegative matrix and N is normally distributed random noise with $\|N\|_F = 0.2\|FE^T\|_F$. Four randomly selected columns of A are plotted in Fig. 13.14.

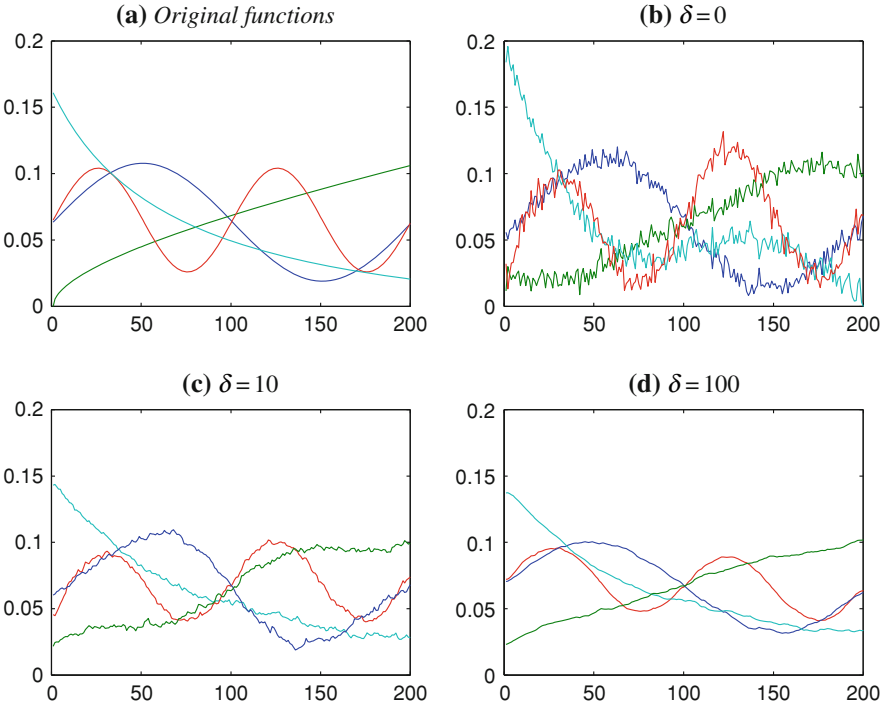


Fig. 13.15 Original functions vs. reconstructed functions

We run the regularized RRI algorithm to force the smoothness of columns of U. We apply, for each run, the same value of δ for all the columns of $U : \delta = 0, 10, 100$. The results obtained through these runs are presented in Fig. 13.15. We see that, without regularization, i.e. $\delta = 0$, the noise is present in the approximation, which produces nonsmooth solutions. When increasing the regularizing terms, i.e. $\delta = 10, 100$, the reconstructed functions become smoother and the shape of the original functions are well preserved.

This smoothing technique can be used for applications like that in [27], where smooth spectral reflectance data from space objects is unmixed. The multiplicative rules are modified by adding the two-norm regularizations on the factor U and V to enforce the smoothness. This is a different approach, therefore, a comparison should be carried out.

We have described a new method for nonnegative matrix factorization that has a good and fast convergence. Moreover, it is also very flexible to create variants and to add some constraints as well. The numerical experiments show that this method and its derived variants behave very well with different types of data. This gives enough motivations to extend to other types of data and applications in the future.

13.7 Conclusion

This paper focuses on the descent methods for Nonnegative Matrix Factorization, which are characterized by nonincreasing updates at each iteration.

We present also the Rank-one Residue Iteration algorithm for computing an approximate Nonnegative Matrix Factorization. It uses recursively nonnegative rank one approximations of a residual matrix that is not necessarily nonnegative. This algorithm requires no parameter tuning, has nice properties and typically converges quite fast. It also has many potential extensions. During the revision of this report, we were informed that essentially the same algorithm was published in an independent contribution [7] and also mentioned later in an independent personal communication [10].

Acknowledgments This paper presents research results of the Concerted Research Action(ARC) "Large Graphs and Networks" of the French Community of Belgium and the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors. Ngoc-Diep Ho is a FRIA fellow.

References

1. Albright R, Cox J, Duling D, Langville AN, Meyer CD (2006) Algorithms, initializations, and convergence for the nonnegative matrix factorization. NCSU technical report math 81706, North Caroline State University, USA
2. Bader BW, Kolda TG (2006) Efficient MATLAB computations with sparse and factored tensors. Technical report SAND2006-7592, Sandia National Laboratories, Albuquerque, NM and Livermore, CA
3. Berry MW, Browne M, Langville AN, Pauca VP, Plemmons RJ (2007) Algorithms and applications for approximate nonnegative matrix factorization. Comput Stat Data Anal 52(1):155–173
4. Bertsekas DP (1999) Nonlinear programming. Athena Scientific, Belmont
5. Bro R, De Jong S (1997) A fast non-negativity constrained least squares algorithm. J Chemom 11(5):393–401
6. Catral M, Han L, Neumann M, Plemmons RJ (2004) On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices. Linear Algebra Appl 393:107–126
7. Cichocki A, Zdunek R, Amari S (2007) Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization. In: Proceedings of independent component analysis, ICA 2007, London, UK, September 9–12, 2007, Lecture Notes in Computer Science, vol 4666, pp 169–176. Springer
8. Cichocki A, Zdunek R, Amari S (2008) Nonnegative matrix and tensor factorization. IEEE Signal Process. Mag. 25:142–145
9. Ding C, Li T, Jordan MI (2006) Convex and semi-nonnegative matrix factorizations. Technical report, LBNL Tech Report 60428
10. Gillis N, Glineur F (2008) Nonnegative factorization and the maximum edge biclique problem. CORE Discussion paper, no. 64, Université catholique de Louvain, Belgium
11. Golub G, Van Loan CF (1996) Matrix computations. vol xxvii, 3rd edn. The Johns Hopkins Univ. Press, Baltimore, p 694

12. Higham NJ (1989) Matrix nearness problems and applications. In: Applications of matrix theory. Oxford University Press, pp 1–27
13. Ho N-D (2008) Nonnegative matrix factorization—algorithms and applications. PhD Thesis, Université catholique de Louvain, Belgium
14. Ho N-D, Van Dooren P, Blondel VD (2007) Descent algorithms for nonnegative matrix factorization. Technical report 2007-57, Cesame. Université catholique de Louvain, Belgium
15. Hoyer PO (2004) Non-negative matrix factorization with sparseness constraints. J. Mach. Learn. Res. 5:1457–1469
16. Kolda TG, O'Leary DP (1998) A semidiscrete matrix decomposition for latent semantic indexing information retrieval. ACM Trans. Inform Syst (TOIS) 16(4):322–346
17. Lawson CL, Hanson RJ (1974) Solving least squares problems. Prentice-Hall, Englewood Cliffs
18. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788–791
19. Lin C-J (2005) Projected gradient methods for non-negative matrix factorization. Technical report information and support services technical report ISSTECH-95-013, Department of Computer Science, National Taiwan University
20. Lin C-J (2007) On the convergence of multiplicative update algorithms for non-negative matrix factorization. IEEE Trans Neural Netw 18(6):1589–1596
21. Lin C-J (2007) Projected gradient methods for non-negative matrix factorization. Neural Comput 19(10):2756–2779
22. Merritt M, Zhang Y (2005) Interior-point gradient method for large-scale totally nonnegative least squares problems. J Optim Theory Appl 126(1):191–202
23. Paatero P (1997) A weighted non-negative least squares algorithm for three-way 'parafac' factor analysis. Chemom Intell Lab Syst 38(2):223–242
24. Paatero P, Tapper U (1994) Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5(1):111–126
25. Pauca VP, Piper J, Plemmons RJ (2006) Nonnegative matrix factorization for spectral data analysis. Linear Algebra Appl 416(1):29–47

Chapter 14
A Computational Method for Symmetric Stein Matrix Equations

K. Jbilou and A. Messaoudi

Abstract In the present paper, we propose a numerical method for solving the sparse symmetric Stein equation $AXA^T - X + BB^T = 0$. Such problems appear in control problems, filtering and image restoration. The proposed method is a Krylov subspace method based on the global Arnoldi algorithm. We apply the global Arnoldi algorithm to extract low-rank approximate solutions to Stein matrix equations. We give some theoretical results and report some numerical experiments to show the effectiveness of the proposed method.

14.1 Introduction

In this paper, we present a numerical Krylov subspace method for solving the Stein (discrete-time Lyapunov) matrix equation

$$AXA^T - X + BB^T = 0 \tag{14.1}$$

where A is an $n \times n$ real and sparse matrix, $X \in \mathrm{IR}^{n\times n}$ and $B \in \mathrm{IR}^{n\times s}$ with $\mathrm{rank}(B) = s$ and $s \ll n$.

K. Jbilou (✉)
Université du Littoral, zone universitaire de la Mi-voix, batiment H. Poincarré, 50 rue F. Buisson, B.P. 699, 62280 Calais Cedex, France
e-mail: jbilou@univ-littoral.fr

A. Messaoudi
Département d'Informatique, Ecole Normale Supérieure Takaddoum, Av. Oued Akreuch, Takaddoum, B.P. 5118 Rabat, Marocco
e-mail: abderrahim.messaoudi@gmail.com

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_14,

We assume throughout this paper that $\lambda_i(A)\lambda_j(A) \neq 1$ for all $i, j = 1, \ldots, n$ ($\lambda_i(A)$ denotes the ith eigenvalue of the matrix A) and this condition ensures that the solution X of the problem (1) exists and is unique.

Lyapunov and discrete-time Lyapunov equation play a crucial role in linear control and filtering theory for continuous or discrete-time large-scale dynamical systems and other problems; see [3–5, 14, 15, 20, 23] and the references therein. They also appear in each step of Newton's method for discrete-time algebraic Riccati equations [8, 13]. Equation (14.1) is also referred to as discrete-time Lyapunov equation.

Direct methods for solving the matrix equation (14.1) such as those proposed in [1] are attractive if the matrices are of small size. These methods are based on the Bartels-Stewart algorithm [2] or on the Hessenberg-Schur method [7].

Iterative methods such as the squared Smith method [6, 16, 21] have been proposed for dense Stein equations. All of these methods compute the solution in dense form and hence require $O(n^2)$ storage and $O(n^3)$ operations. Notice that the matrix equation (14.1) can be formulated as an $n^2 \times n^2$ large linear system using the Kronecker formulation $(A \otimes A - I_{n^2})\text{vec}(X) = -\text{vec}(BB^T)$ where $\otimes$ denotes the Kronecker product; $(F \otimes D = [f_{i,j}D])$, $\text{vec}(X)$ is the vector of IR^{n^2} formed by stacking the columns of the matrix X and I_n is the identity matrix of order $n \times n$. Krylov subspace methods could be used to solve the above linear system. However, for large problems this approach is very expensive and doesn't take into account the low rank of the right hand side.

The observability G_o and controllability G_g Gramians of the discrete-time linear time-invariant (LTI) system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Dx(k), \quad k = 0, 1, \ldots \end{aligned}$$

are the solutions of the Stein equations

$$A^T G_o A - G_o + D^T D = 0$$

and

$$AG_g A^T - G_g + BB^T = 0$$

where the input $u(\cdot) \in \text{IR}^s$ and the output $y(\cdot) \in \text{IR}^q$.

In many applications, n is quite large while the number of inputs s and outputs q usually satisfy $s, q \ll n$.

The Gramians of LTI systems play an important role in many analysis and design problems for LTI systems such as model reduction, the computation of the Hankel norm or H_2-norm of the system; see [4, 20, 23]. The problem of model reduction is to produce a low dimensional LTI system that will have approximately the same response as the original system to any given input u.

Recently, several schemes using Krylov subspace methods have been developed to produce low-rank approximate solution to Lyapunov and discrete-time Lyapunov equations with low-rank right hand sides [9, 10, 12, 18, 22].

If the symmetric Stein equation (14.1) is Schur stable, that is if $\rho(A) < 1$, where $\rho(A)$ denotes the spectral radius of A, then (14.1) has a unique solution X given by [13]:

$$X = \sum_{i=0}^{\infty} A^i B B^T A^{T^i}.$$

In this paper, we present a new Krylov subspace approach to solve the problem (14.1). The proposed method takes advantage of the low-rank structure of (14.1) to obtain low-rank approximate solutions in factored form. Our algorithm uses a Galerkin projection method and is based on the global Arnoldi algorithm [11].

The remainder of the paper is organized as follows: In Sect. 14.2, we review the global Arnoldi algorithm. In Sect. 14.3, new expressions of the exact solution of the matrix equation (14.1) are given. Section 14.4 is devoted to the Stein global Arnoldi method. The new method is developed and some theoretical results are given. In Sect. 14.5 we give some numerical examples to show the effectiveness of the proposed method for large sparse problems.

Throughout this paper we use the following notations. For two matrices X and Y in $\mathrm{IR}^{n\times s}$, we define the inner product $\langle X, Y\rangle_F = \mathrm{trace}(X^T Y)$. The associated norm is the Frobenius norm or F-norm denoted by $\|.\|_F$. A system of vectors (matrices) of $\mathrm{IR}^{n\times s}$ is said to be F-orthonormal if it is orthonormal with respect to $\langle ., .\rangle_F$. The matrices I_n and O_n will denote the $n \times n$ identity and the null matrices respectively. Finally, for a matrix Z, $\|Z\|_2$ will denote the 2-norm of Z.

14.2 The Global Arnoldi Algorithm

The global Arnoldi algorithm [11] constructs an F-orthonormal basis $V_1, V_2, \ldots, V_m$ of the matrix Krylov subspace $\mathcal{K}_m(A, B)$, i.e.,

$$\begin{aligned} &\langle V_i, V_j\rangle_F = 0 \quad \text{for } i \neq j;\ i, j = 1, \ldots, m \quad \text{and} \\ &\langle V_i, V_i\rangle_F = 1. \end{aligned}$$

We recall that the minimal polynomial P (scalar polynomial) of A with respect to $B \in \mathrm{IR}^{n\times s}$ is the nonzero monic polynomial of lowest degree such that $P(A)B = 0$. The degree p of this polynomial is called the grade of B and we have $p \leq n$.

The modified global Arnoldi algorithm is described as follows:

Algorithm **1**

The Modified Global Arnoldi algorithm

Set $V_1 = B/\|B\|_F$
For $j = 1, \ldots, m$
$\tilde{V} = AV_j$.
For $i = 1, 2, \ldots, j$,
$h_{i,j} = \text{trace}(V_i^T \tilde{V})$,
$\tilde{V} = \tilde{V} - h_{i,j} V_i$.
End.
$h_{j+1,j} = \| \tilde{V} \|_F$,
$V_{j+1} = \tilde{V}/h_{j+1,j}$.
End.

Basically, the global Arnoldi algorithm is the standard Arnoldi algorithm applied to the matrix pair $(\mathcal{A}, b)$ where $\mathcal{A} = I_s \otimes A$ and $b = \text{vec}(B)$. When $s = 1$, Algorithm 1 reduces to the classical Arnoldi algorithm [17]. The global Arnoldi algorithm breaks down at step j if and only if $h_{j+1,j} = 0$ and in this case an invariant subspace is obtained. However, a near breakdown may occur when a subspace is A-invariant to machine precision (when, for some $j, h_{j+1,j}$ is close to zero). We note that the global Arnoldi algorithm generates an F-orthonormal basis of the matrix Krylov subspace $\mathcal{K}_m(A, V_1) \subseteq \mathcal{M}_{n,s}$ where $\mathcal{M}_{n,s}$ is the space of real matrices having dimension $n \times s$.

Let us now introduce some notations: $\mathcal{V}_m$ denotes the $n \times ms$ matrix $\mathcal{V}_m = [V_1, \ldots, V_m]$.$\tilde{H}_m$ is the $(m+1) \times m$ upper Hessenberg matrix whose entries $h_{i,j}$ are defined by Algorithm 1 and H_m is the $m \times m$ matrix obtained from $\tilde{H}_m$ by deleting its last row.

With $\mathcal{V}_m$ and H_m defined above, and using the Kronecker product $\otimes$, the following relation is satisfied [11]

$$A\mathcal{V}_m = \mathcal{V}_m(H_m \otimes I_s) + h_{m+1,m} V_{m+1} E_m^T \tag{14.2}$$

and

$$A\mathcal{V}_m = \mathcal{V}_{m+1}(\tilde{H}_m \otimes I_s) \tag{14.3}$$

where $E_m^T = [0_s, \ldots, 0_s, I_s]$ and $\mathcal{V}_{m+1} = [\mathcal{V}_m, V_{m+1}]$. Note that $\| V_i \|_F = 1, i = 1, \ldots, m$ and $\| \mathcal{V}_m \|_F = \sqrt{m}$.

We have the following properties [11].

Proposition 1 *Let $p (p \leq n)$ be the degree of the minimal polynomial of A with respect to V_1, then the following statements are true.*

1. *The matrix Krylov subspace $\mathcal{K}_p(A, V_1)$ is invariant under A and $\mathcal{K}_m(A, V_1)$ is of dimension m if and only if p is greater than $m - 1$.*
2. *The global Arnoldi algorithm will stop at step m if and only if $m = p$.*
3. *For $m \leq p$, $\{V_1, V_2, \ldots, V_m\}$ is an F-orthonormal basis of the matrix Krylov subspace $\mathcal{K}_m(A, V_1)$.*
4. $A\mathcal{V}_p = \mathcal{V}_p(H_p \otimes I_s)$.

14.3 Expressions for the Exact Solution of the Stein Equation

In this section we will give new expressions for the solution X of the Stein equation (14.1). If we assume that $\lambda_i(A)\lambda_j(A) \neq 1$, for $i = 1, \ldots, n$ and $j = 1, \ldots, n$, where $\lambda_i(A)$ denotes the ith eigenvalue of A, then the solution X of the Eq. (14.1) exists and is unique. Let P be the minimal polynomial of A with respect to B of degree p:

$$P(A)B = \sum_{i=0}^{p} \alpha_i A^i B = 0, \quad \alpha_p = 1.$$

Associated with the polynomial P, we define the $p \times p$ companion matrix K

$$K = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \ldots & \ldots & 0 & 1 \\ -\alpha_0 & \ldots & \ldots & -\alpha_{p-2} & -\alpha_{p-1} \end{bmatrix},$$

M denotes the following matrix

$$M = [B, AB, \ldots, A^{p-1}B].$$

Remark that, if I_s denotes the identity matrix of $\mathrm{IR}^{s\times s}$, then we have

$$AM = M(K^T \otimes I_s). \tag{14.4}$$

Next, involving the minimal polynomial of A for B, we give a closed-form finite series representation of the solution X of the Eq. (14.1).

Theorem 1.1 *The unique solution X of (14.1) has the following representation*

$$X = \sum_{j=1}^{p} \sum_{i=1}^{p} \gamma_{i,j} A^{i-1} B B^T (A^T)^{j-1}, \tag{14.5}$$

where the matrix $\Gamma = (\gamma_{i,j})_{1 \leq i,j \leq p}$ is the solution of

$$K^T \Gamma K - \Gamma + e_1 e_1^T = 0, \tag{14.6}$$

and e_1 denotes the first vector of the canonical basis of IR^p.

Proof Remark first that since the eigenvalues of K are eigenvalues of A, we have $\lambda_i(K)\lambda_j(K) \neq 1$ for all $i, j = 1, \ldots, p$ and this implies that equation (6) has a unique solution. Let Y be the following matrix:

$$Y = \sum_{j=1}^{p}\sum_{i=1}^{p} \gamma_{i,j} A^{i-1} B B^T (A^T)^{j-1},$$

where Γ solves the low-order Stein equation (14.6). Then Y can also be expressed as

$$\begin{aligned} Y &= [B, AB, \ldots, A^{p-1}B](\Gamma \otimes I_s) \begin{bmatrix} B^T \\ B^T A^T \\ \vdots \\ B^T (A^T)^{p-1} \end{bmatrix} \\ &= M(\Gamma \otimes I_s) M^T. \end{aligned}$$

Let us show that the matrix Y is a solution of the Stein equation (14.1). Using the relation (14.4) we obtain

$$\begin{aligned} AYA^T &= AM(\Gamma \otimes I_s) M^T A^T \\ &= M(K^T \otimes I_s)(\Gamma \otimes I_s)(K \otimes I_s) M^T \\ &= M(K^T \Gamma K \otimes I_s) M^T. \end{aligned}$$

On the other hand, since

$$B = [B, AB, A^2B, \ldots, A^{p-1}B][I_s, O_s, \ldots, O_s]^T$$

we have

$$\begin{aligned} BB^T &= [B, AB, \ldots, A^{p-1}B](e_1 \otimes I_s)(e_1^T \otimes I_s)[B, AB, \ldots, A^{p-1}B]^T \\ &= M(e_1 e_1^T \otimes I_s) M^T. \end{aligned}$$

Then

$$\begin{aligned} AYA^T - Y + BB^T &= M(K^T \Gamma K \otimes I_s) M^T - M(\Gamma \otimes I_s) M^T + M(e_1 e_1^T \otimes I_s) M^T \\ &= M(K^T \Gamma K \otimes I_s) - (\Gamma \otimes I_s) + (e_1 e_1^T \otimes I_s) M^T \\ &= M[(K^T \Gamma K - \Gamma + e_1 e_1^T) \otimes I_s] M^T. \end{aligned}$$

Therefore, since Γ solves the low-order Stein equation (14.6), we obtain $AYA^T - Y + BB^T = 0$. As the Eq. (14.1) has a unique solution it follows that $X = Y$ which completes the proof.

The following result states that the solution X of (14.1) can be expressed in terms of the blocks $V_1, \ldots, V_p$.

Theorem 1.2 *Let p be the grade of B and let $\mathcal{V}_p$ be the matrix defined by $\mathcal{V}_p = [V_1, \ldots, V_p]$, where the matrices $V_1, \ldots, V_p$ are constructed by the global Arnoldi algorithm with $V_1 = B/ \parallel B \parallel_F$. Then the unique solution X of (14.1) can be expressed as:*

$$X = \mathcal{V}_p(\hat{\Gamma} \otimes I_s)\mathcal{V}_p^T, \tag{14.7}$$

where $\hat{\Gamma}$ *is the solution of the low-order Stein equation:*

$$H_p\hat{\Gamma}H_p^T - \hat{\Gamma} + \|B\|_F^2 e_1 e_1^T = 0. \tag{14.8}$$

Proof Notice that the eigenvalues of H_p are eigenvalues of A, then $\lambda_i(H_p)\lambda_j(H_p) \neq 1$ for all $i,\ j = 1, \ldots, p$ and this ensures that Eq. (14.8) has a unique solution $\hat{\Gamma}$. Let Y be the matrix defined by $Y = \mathcal{V}_p(\hat{\Gamma} \otimes I_s)\mathcal{V}_p^T$. Then by substituting Y in (14.1) and using the relations:

$$B = \|B\|_F \mathcal{V}_p(e_1 \otimes I_s), \quad BB^T = \|B\|_F^2 \mathcal{V}_p(e_1 e_1^T \otimes I_s)\mathcal{V}_p^T$$

and

$$A\mathcal{V}_p = \mathcal{V}_p(H_p \otimes I_s),$$

we obtain

$$AYA^T - Y + BB^T = \mathcal{V}_p[(H_p\hat{\Gamma}H_p^T - \hat{\Gamma} + \|B\|_F^2 e_1 e_1^T) \otimes I_s]\mathcal{V}_p^T.$$

As $\hat{\Gamma}$ solves the low-order Stein equation (14.8), we get

$$AYA^T - Y + BB^T = 0.$$

Therefore, using the fact that the solution of (14.1) is unique, it follows that $X = Y$.

14.4 The Stein Global Arnoldi Method

Following the results of Theorem 1.2, we will see how to extract low-rank approximations to the solution X of (14.1). Since the exact solution is given by the expressions (14.7) and (14.8), the approximate solution X_m that we will consider is defined by

$$X_m = \mathcal{V}_m(Y_m \otimes I_s)\mathcal{V}_m^T \tag{14.9}$$

where Y_m is the symmetric $m \times m$ matrix satisfying the low-dimensional Stein equation

$$H_m Y_m H_m^T - Y_m + \|B\|_F^2 e_1 e_1^T = 0 \tag{14.10}$$

with $e_1 = (1, 0, \ldots, 0)^T \in \mathrm{IR}^m$.

From now on, we assume that for increasing m, $\rho(H_m) < 1$ which ensures that (14.10) has a unique solution Y_m. For the solution of the projected problem (14.10), we have the following result

Theorem 1.3 *Assume that* $\| A \|_2 < 1$, *then* $\forall m \geq 1$, *the projected Stein matrix equation* (14.10) *has a unique solution.*

Proof Notice first that if $\| A \|_2 < 1$ then $\rho(A) < 1$ which implies that the Stein matrix equation (14.1) has a unique solution. Let $\lambda_1^{(m)}$ be the largest (in modulus) eigenvalue of the matrix H_m and let $\alpha_1^{(m)}$ denote it's largest singular value. Then using the Weyl's inequalities [24], it is known that $| \lambda_1^{(m)} | < \alpha_1^{(m)}$. On the other hand, it was shown in [19] that $\alpha_1^{(m)} \leq \| A \|_2$. Therefore as $\| A \|_2 < 1$, it follows that $\rho(H_m) < 1$ which implies that the projected Stein matrix equation has a unique solution.

The low-dimensional Stein equation (14.10) will be solved by a standard direct method such as the Hessenberg-Schur method [1, 7]. Note that, at step m, the method proposed in [10], which is based on the block Arnoldi algorithm, yields a reduced order Stein equation of dimension $ms \times ms$ while the projected Eq. (14.10) is of dimension $m \times m$. In [10], it was addressed that rank degradation is often observed in the block Arnoldi process and this leads to equations of dimension strictly smaller than ms.

Next, we give an upper bound for the residual norm that can be used to stop the iterations in the Stein global Arnoldi algorithm without having to compute extra products involving the matrix A. We first give the following lemma to be used later.

Lemma 1 *Let* $\mathcal{V}_m = [V_1, \ldots, V_m]$, *where* $V_1, \ldots, V_m$ *are the matrices generated by Algorithm 1. Let* $Z = [z_{i,j}]$ *be a matrix in* $\mathrm{IR}^{m \times r}$ *and let* $G = [g_{i,j}]$ *be a matrix of* $\mathrm{IR}^{ms \times q}$ *where* r *and* q *are any integers. Then we have*

$$\|\mathcal{V}_m(Z \otimes I_s)\|_F = \|Z\|_F, \tag{14.11}$$

and

$$\|\mathcal{V}_m G\|_F \leq \|G\|_F. \tag{14.12}$$

Proof If $z_{.,j}, j = 1, \ldots, r$, denotes the jth column of the matrix Z, we have

$$\begin{aligned} \mathcal{V}_m(Z \otimes I_s) &= \mathcal{V}_m[z_{.,1} \otimes I_s, \ldots, z_{.,r} \otimes I_s] \\ &= [\mathcal{V}_m(z_{.,1} \otimes I_s), \ldots, \mathcal{V}_m(z_{.,r} \otimes I_s)]. \end{aligned}$$

As $\{V_1, \ldots, V_m\}$ is F-orthonormal, it results that

$$\begin{aligned} \| \mathcal{V}_m(z_{.,j} \otimes I_s) \|_F^2 &= \| \sum_{i=1}^{m} z_{i,j} V_i \|_F^2 \\ &= \sum_{i=1}^{m} | z_{i,j} |^2 \\ &= \| z_{.,j} \|_2^2, \; j = 1, \ldots, r, \end{aligned}$$

and then

$$\begin{aligned} \| \mathcal{V}_m(Z \otimes I_s) \|_F^2 &= \sum_{j=1}^{r} \sum_{i=1}^{m} | z_{i,j} |^2 \\ &= \| Z \|_F^2 . \end{aligned}$$

We express the matrix $\mathcal{V}_m G$ as $\mathcal{V}_m G = [\mathcal{V}_m g_{.1}, \ldots, \mathcal{V}_m g_{.q}]$ where $g_{.j}$ is the jth column of G. For $j = 1, \ldots, q$, the vector $\mathcal{V}_m g_{.j}$ can be written as follows

$$\mathcal{V}_m g_{.j} = \sum_{i=1}^{m} V_i \begin{pmatrix} g_{(i-1)s+1,j} \\ \vdots \\ g_{is,j} \end{pmatrix}.$$

Now, since $\|V_i\|_F = 1$ for $i = 1, \ldots, m$, we obtain

$$\begin{aligned} \|\mathcal{V}_m g_{.j}\|_F^2 &\leq \sum_{i=1}^{m} \left\| \begin{pmatrix} g_{(i-1)s+1,j} \\ \vdots \\ g_{is,j} \end{pmatrix} \right\|_2^2 \\ &= \|g_{.j}\|_2^2; j = 1, \ldots, q. \end{aligned}$$

Therefore

$$\begin{aligned} \|\mathcal{V}_m G\|_F^2 &= \sum_{j=1}^{q} \|\mathcal{V}_m g_{.j}\|_2^2 \\ &\leq \|G\|_F^2. \end{aligned}$$

In the following theorem, we give an upper bound for the residual norm.

Theorem 1.4 *Let X_m be the approximate solution obtained, at step m, by the Stein global Arnoldi algorithm and let $R(X_m) = AX_mA^T - X_m + BB^T$ be the corresponding residual. Then*

$$\|R(X_m)\|_F \leq h_{m+1,m} \sqrt{2\|H_m \tilde{Y}_m\|_2^2 + h_{m+1,m}^2 (\tilde{Y}_m^{(m)})^2}, \tag{14.13}$$

where $\tilde{Y}_m$ is the last column of the matrix Y_m and $\tilde{Y}_m^{(m)}$ denotes the last component of $\tilde{Y}_m$.

Proof At step m, the residual $R(X_m)$ is written as

$$R(X_m) = A\mathcal{V}_m(Y_m \otimes I_s)\mathcal{V}_m^T A^T - \mathcal{V}_m(Y_m \otimes I_s)\mathcal{V}_m^T + BB^T.$$

Using the relations (14.2)–(14.3) and the fact that $E_m = e_m \otimes I_s$, we obtain

$$R(X_m) = \mathcal{V}_{m+1}\left[\begin{pmatrix} H_m Y_m H_m^T - Y_m + \|B\|_F^2 e_1 e_1^T & h_{m+1,m} H_m Y_m e_m \\ h_{m+1,m} e_m^T Y_m H_m^T & h_{m+1,m}^2 e_m^T Y_m e_m \end{pmatrix} \otimes I_s\right]\mathcal{V}_{m+1}^T.$$

Invoking (14.10) it follows that

$$\|R(X_m)\|_F^2 = \left\|\mathcal{V}_{m+1}\left[\begin{pmatrix} 0 & h_{m+1,m} H_m Y_m e_m \\ h_{m+1,m} e_m^T Y_m H_m^T & h_{m+1,m}^2 e_m^T Y_m e_m \end{pmatrix} \otimes I_s\right]\mathcal{V}_{m+1}^T\right\|_F^2.$$

Now, using the relation (14.12) of Lemma 1, we get

$$\|R(X_m)\|_F^2 \leq \left\|\left[\begin{pmatrix} 0 & h_{m+1,m} H_m Y_m e_m \\ h_{m+1,m} e_m^T Y_m H_m^T & h_{m+1,m}^2 e_m^T Y_m e_m \end{pmatrix} \otimes I_s\right]\mathcal{V}_{m+1}^T\right\|_F^2.$$

On the other hand, we set

$$Z = \begin{pmatrix} 0 & h_{m+1,m} H_m Y_m e_m \\ h_{m+1,m} e_m^T Y_m H_m^T & h_{m+1,m}^2 e_m^T Y_m e_m \end{pmatrix},$$

and

$$\alpha_m = \left\|(Z \otimes I_s)\mathcal{V}_{m+1}^T\right\|_F^2.$$

Note that α_m is also expressed as

$$\alpha_m = \|\mathcal{V}_{m+1}(Z \otimes I_s)\|_F^2.$$

Then, setting $G = Z \otimes I_s$ and applying (14.11) we obtain

$$\begin{aligned} \alpha_m &= \| Z \|_F^2 \\ &= 2\left\|h_{m+1,m} H_m Y_m e_m\right\|_2^2 + (h_{m+1,m}^2 e_m^T Y_m e_m)^2 \\ &= 2h_{m+1,m}^2 \|H_m Y_m e_m\|_2^2 + (h_{m+1,m}^2 e_m^T Y_m e_m)^2 \\ &= h_{m+1,m}^2\left[2\left\|H_m \tilde{Y}_m\right\|_2^2 + h_{m+1,m}^2 (\tilde{Y}_m^{(m)})^2\right]. \end{aligned}$$

Therefore

$$\|R(X_m)\|_F^2 \leq h_{m+1,m}^2\{2\|H_m \tilde{Y}_m\|_2^2 + h_{m+1,m}^2 (\tilde{Y}_m^{(m)})^2\}.$$

The result of Theorem 1.4 allows us to compute an upper bound for the residual norm without computing the residual which requires the construction of the approximation X_m and two matrix-matrix products. This provides a useful stopping criterion in a practical implementation of the algorithm.

The Stein global Arnoldi algorithm is summarized as follows:

Algorithm **2**:

The Stein global Arnoldi algorithm
Choose a tolerance $\epsilon > 0$, **an integer parameter** k_1
and set $k = 0$, $m = k_1$.
For $j = k+1, k+2, \ldots, k+k_1$
construct the F-orthonormal basis $V_{k+1}, \ldots, V_{k+k_1}$
and H_m **by Algorithm 1.**
End.
Solve the low-dimensional problem:
$H_m Y_m H_m^T - Y_m + \|B\|_F^2 e_1 e_1^T = 0.$
Compute the upper bound for the F-norm of the residual:
$r_m = h_{m+1,m}\sqrt{2\|H_m \tilde{Y}_m\|_2^2 + h_{m+1,m}^2 (\tilde{Y}_m^{(m)})^2}.$
If $r_m > \epsilon$, set $k := k + k_1$, $m = k + k_1$ and return.
The approximate solution is given as $X_m = \mathcal{V}_m (Y_m \otimes I_s)\mathcal{V}_m^T$.

Remarks As m increases, the computation of Y_m becomes expensive. To avoid this, the procedure above introduces a parameter k_1, to be chosen, such that the low-order Stein equation is solved every k_1 iterations. Note also that, when convergence is achieved, the computed approximate solution X_m is stored as the product of smaller matrices. The next perturbation result shows that the approximation X_m is an exact solution of a perturbed Stein equation.

Theorem 1.5 *Assume that, at step m, the matrix* $\mathcal{V}_m$ *is of full rank. Then the approximate solution of* (14.1) *solves the following Stein equation:*

$$(A - \Delta_m)X_m(A - \Delta_m)^T - X_m + BB^T = 0, \tag{14.14}$$

where $\Delta_m = h_{m+1,m} V_{m+1} E_m^T \mathcal{V}_m^+$ *and* $\mathcal{V}_m^+ = (\mathcal{V}_m^T \mathcal{V}_m)^{-1} \mathcal{V}_m^T$ *is the pseudo-inverse of the matrix* $\mathcal{V}_m$.

Proof Applying the Kronecker product to (14.10), we obtain

$$(H_m \otimes I_s)(Y_m \otimes I_s)(H_m^T \otimes I_s) - (Y_m \otimes I_s) + \|B\|_F^2 (e_1 \otimes I_s)(e_1^T \otimes I_s) = 0. \tag{14.15}$$

Multiplying (14.15) on the right by $\mathcal{V}_m^T$, on the left by $\mathcal{V}_m$ and using (14.2) we get

$$\begin{aligned}&[A\mathcal{V}_m - h_{m+1,m} V_{m+1} E_m^T](Y_m \otimes I_s)[A\mathcal{V}_m - h_{m+1,m} V_{m+1} E_m^T]^T - \mathcal{V}_m (Y_m \otimes I_s)\mathcal{V}_m^T \\ &\quad + \|B\|_F^2 \mathcal{V}_m (e_1 \otimes I_s)(e_1^T \otimes I_s)\mathcal{V}_m^T \\ &= 0.\end{aligned}$$

Now, using the fact that $\mathcal{V}_m(e_1 \otimes I_s) = V_1$ and $X_m = \mathcal{V}_m (Y_m \otimes I_s)\mathcal{V}_m^T$, we obtain

$$(A - \Delta_m)X_m(A - \Delta_m)^T - X_m + BB^T = 0.$$

The next result gives an upper bound for the Frobenius norm of the error $X - X_m$ where X is the exact solution of (14.1).

Theorem 1.6 *Assume that m steps of Algorithm 2 have been run. Let X_m be the obtained approximate solution of (14.1). If $\|A\|_2 < 1$, then we have*

$$\|X - X_m\|_2 \leq h_{m+1,m} \frac{2\|H_m \tilde{Y}_m\|_2 + |h_{m+1,m} \tilde{Y}_m^{(m)}|}{1 - \|A\|_2^2},$$

where $\tilde{Y}_m$ is the last column of Y_m (the solution of the low-order problem (14.10)) and $\tilde{Y}_m^{(m)}$ denotes the last component of $\tilde{Y}_m$.

Proof Remark first that $\|A\|_2 < 1$ implies that $\rho(A) < 1$ which insures that the Stein matrix equation (14.1) has a unique solution. The matrix equation (14.14) can be expressed as

$$AX_mA^T - X_m + BB^T = L_m, \tag{14.16}$$

where

$$L_m = AX_m\Delta_m^T + \Delta_m X_m A^T - \Delta_m X_m \Delta_m^T.$$

Subtracting (14.16) from the initial Stein equation (14.1), we get

$$A(X - X_m)A^T - (X - X_m) + L_m = 0.$$

Now, as $\rho(A) < 1$, the error $X - X_m$ can be written as

$$X - X_m = \sum_{i=0}^{\infty} A^i L_m (A^i)^T.$$

Hence

$$\begin{aligned} \| X - X_m \|_2 &\leq \| L_m \|_2 \sum_{i=0}^{\infty} \|A\|_2^{2i} \\ &\leq \| L_m \|_2 \frac{1}{1 - \|A\|_2^2}. \end{aligned}$$

Invoking (14.9) and (14.2), L_m is given by

$$\begin{aligned} L_m = & h_{m+1,m} \mathcal{V}_m (H_m Y_m e_m \otimes I_s) V_{m+1}^T \\ & + h_{m+1,m} V_{m+1} (e_m^T Y_m H_m^T \otimes I_s) \mathcal{V}_m^T \\ & + h_{m+1,m}^2 V_{m+1} (e_m^T Y_m e_m \otimes I_s) V_{m+1}^T, \end{aligned}$$

therefore

$$\begin{aligned} \| L_m \|_2 &\leq \| L_m \|_F \\ &\leq 2h_{m+1,m} \| \mathcal{V}_m (H_m Y_m e_m \otimes I_s) V_{m+1}^T \|_F \\ &\quad + h_{m+1,m}^2 \| V_{m+1} (e_m^T Y_m e_m \otimes I_s) V_{m+1}^T \|_F . \end{aligned}$$

Table 14.1 Costs for Stein global Arnoldi and Stein block Arnoldi algorithms

Cost	Stein block Arnoldi	Stein global Arnoldi
Matrix-vector	$s(m+1)$	$s(m+1)$
Mult. of blocks $n \times s$ and $s \times s$	$m(m+1)/2$	
n-vector DOT	$m(m+1)s^2/2$	$m(m+1)s/2$
MGS on $n \times s$ blocks	m	
Solving low-order Stein equation	$O(m^4 s^3)$	$O(m^4)$

On the other hand, as V_{m+1} is F-orthonormal, we have $\| V_{m+1}^T \|_F = 1$. Using the relation (14.11), we get $\| \mathcal{V}_m(H_m Y_m e_m \otimes I_s) \|_F = \| H_m Y_m e_m \|_F$ and finally, as $e_m^T Y_m e_m$ is a scalar it follows that $\| V_{m+1}(e_m^T Y_m e_m \otimes I_s) \|_F = | e_m^T Y_m e_m |$ and then the result follows.

In Table 14.1, we listed the major work, at each iteration m, used for the Stein block and global Arnoldi algorithms. In addition to matrix-vector products the Stein block Arnoldi algorithm requires the application of the modified Gram-Schmidt process on matrices of dimension $n \times s$ and the solution of Stein equations of order at most ms.

14.5 Numerical Examples

All the experiments were performed on a computer of Intel Pentium processor at 1.6 GHz and 3 GBytes of RAM using Matlab 7.4. The matrix test A was divided by $\| A \|_1$. The entries of the matrix B were random values uniformly distributed on [0 1]. The starting block V_1 is $V_1 = B/\| B \|_F$.

Example 1 For the first experiment, we compared the effectiveness of the Stein global Arnoldi algorithm with the fixed point iteration method defined as follows:

$$X_0 = BB^T,\ A_0 = AX_{m+1} = A_m X_m A_m^T + X_m \text{ and } A_{m+1} = A_m^2,\ m = 0, 1, \ldots.$$

This method is known as the Squared Smith Method (SQSM) (see [6, 21]). For this experiment, the matrix A was the matrix test PDE900 ($n = 900$ and $nnz(A) = 4380$) from the Harwell Boeing collection. We used $s = 4$ and $k_1 = 1$.

We compared the CPU-time (in seconds) used for convergence with the two methods. For the SQSM method, the iterations were stopped when the F-norm of the residual $R_m = AX_m A^T - X_m + BB^T$ is less than some tolerance tol $= 10^{-12}$ and for the Stein global Arnoldi algorithm, the iterations were stopped when the upper bound r_m for the residual norm is less than 10^{-12}. The results are reported in Table 14.2.

Table 14.2 $n = 900, s = 4$ and $k_1 = 1$

	Stein global Arnoldi	SQSM
Residual norms	3.5×10^{-13}	2.7×10^{-13}
CPU-time in sec.	4.4	59

Example 2 The matrix A is generated from the 5-point discretization of the operator

$$L(u) = \Delta u - f_1(x,y)\frac{\partial u}{\partial x} - f_2(x,y)\frac{\partial u}{\partial y} - g(x,y)u$$

on the unit square $[0,1] \times [0,1]$ with homogeneous Dirichlet boundary conditions. We set $f_1(x,y) = x^2$, $f_2(x,y) = e^y$ and $g(x,y) = xy$. The dimension of the matrix A is $n = n_0^2$ where n_0 is the number of inner grid points in each direction.

Experiment 2.1 For this experiment, we set $n = 900$ and $s = 4$. In Fig. 14.1 (left), we plotted the true residual norm (solid-line) and the upper bound (dotted-line) given by Theorem 1.4, versus the number of iterations. We also plotted the norm of the error (solid-line) and the corresponding upper bound (dashed-line) given by Theorem 1.6. For this small example, the exact solution was computed by the Matlab function 'dlyap'.

Experiment 2.2 In the last experiment we compared the performance of the Stein global and Stein block algorithms for different values of n and s. The iterations were stopped when the relative residual norm was less than $\epsilon = 10^{-6}$ for the Stein block Arnoldi algorithm and when the upper bound given in Theorem 1.4 was less than $\epsilon \times \|R_0\|_F$. In Table 14.3, we listed the obtained CPU-times (in seconds) and also the total number of iterations in parentheses. For the two algorithms, the projected Stein equations were solved every $k_1 = 5$ iterations.

As shown in Table 14.3, the Stein global Arnoldi algorithm is more effective than the Stein block Arnoldi Solver. As A is a sparse matrix, the larger CPU times needed for the Stein block Arnoldi solver [10] are attributed to the computational

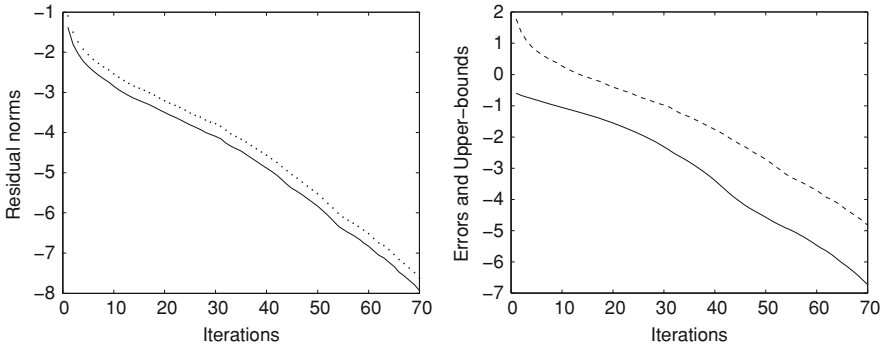


Fig. 14.1 *Left* The norm of the residual (*solid*) and the upper bound (*dotted*). *Right* The norm of the error (*solid*) and the upper bound (*dashed*)

Table 14.3 CPU-time (in seconds) and the total number of iterations (in parentheses)

	Stein global Arnoldi	Stein block Arnoldi
$n = 22500, s = 4$	209(265)	399(215)
$n = 44100, s = 3$	572(350)	788(280)
$n = 52900, s = 2$	554(380)	853(355)
$n = 62500, s = 2$	639(395)	912(375)

Table 14.4 CPU-time (in seconds) for Stein global Arnoldi and GMRES(10)

Method	Stein global Arnoldi	GMRES(10)
$n = 400, N = 1.6 \times 10^5$ unknowns	10	52
$n = 900, N = 8.1 \times 10^5$ unknowns	18	85
$n = 1600, N = 2.56 \times 10^6$ unknowns	31	270

expenses of the block Arnoldi algorithm and to the computation of the solution of the projected Stein equation of order at most ms for increasing m. In the Stein global Arnoldi solver, the projected problem is of order m.

Example 3. In this experiment, we compared the performance of the Stein global Arnoldi method for solving (14.1) with the restarted GMRES algorithm applied to the $n^2 \times n^2$ equivalent linear system $(A \otimes A - I_{n^2})\text{vec}(X) = -\text{vec}(BB^T)$. We notice that the construction of the matrix $\mathcal{A} = A \otimes A - I_{n^2}$ is very expensive for large problems. Therefore, using preconditioners such as the incomplete LU factorization is not possible in this case. So in the GMRES algorithm, we computed the matrix-vector products by using the relation $w = \mathcal{A}v \Leftrightarrow W = AVA^T - V$ where $w = vec(V)$ and $w = vec(W)$. This formulation reduces the cost and the memory requirements. The $n \times n$ matrix A was the same as the one given in Example 2 and the $n \times s$ matrix B was chosen to be random with $s = 4$. The iterations were stopped when the relative residual norms were less than 10^{-8}. We used different medium values of the dimension $n : n = 400, n = 900$ and $n = 1600$ corresponding to $N = 1.6 \times 10^5$, $N = 8.1 \times 10^5$ and $N = 2.56 \times 10^6$ unknowns, respectively. We notice that for large values of n and due to memory limitation, it was impossible to run the GMRES algorithm on our computer. Here also, the projected Stein matrix equation (14.10) was solved every $k_1 = 5$ iterations. In Table 14.4, we reported the CPU time (in seconds) obtained by the two approaches.

14.6 Summary

We presented in this paper a new Krylov subspace method for solving symmetric Stein matrix equations. Some new theoretical results such as expressions of the exact solution and upper bounds for the error and residual norms are given. The numerical tests and comparisons with other known methods show that the

proposed method is effective. In conclusion, global methods are competitive for sparse matrices and should not be used for relatively dense problems. The block Arnoldi algorithm is advantageous if a moderately low number of iterations is accompanied by a high cost of matrix vector operations with the coefficient matrix A.

Acknowledgements We would like to thank the referees for helpful remarks and useful suggestions.

References

1. Barraud AY (1977) A numerical algorithm to solve $A^TXA - X = Q$. IEEE Trans Autom Contr AC-22:883–885
2. Bartels RH, Stewart GW (1994) Algorithm 432: Solution of the matrix equation $AX + XB = C$. Circ Syst and Signal Proc 13:820–826
3. Calvetti D, Levenberg N, Reichel L (1997) Iterative methods for $X - AXB = C$. J Comput Appl Math 86:73–101
4. Datta BN, Datta K (1986) Theoretical and computational aspects of some linear algebra problems in control theory. In: Byrnes CI, Lindquist A (eds) Computational and Combinatorial Methods in Systems Theory. Elsevier, Amsterdam, pp 201–212
5. Datta BN (2003) Numerical methods for linear control systems design and applications. Academic Press, New York
6. Davison EJ, Tan FT (1968) The numerical solution of $A'Q + QA = -C$. IEEE Trans Automat Contr 13:448–449
7. Golub GH, Nash S, Van Loan C (1979) A Hessenberg-Schur method for the problem $AX + XB = C$. IEEC Trans Autom Contr AC-24:909–913
8. Hewer GA (1971) An iterative technique for the computation of steady state gains for the discrete optimal regulator. IEEE Trans Autom Contr AC-16:382–384
9. Hu DY, Reichel L (1992) Krylov subspace methods for the Sylvester equation. Lin Alg and Appl 174:283–314
10. Jaimoukha IM, Kasenally EM (1994) Krylov subspace methods for solving large Lyapunov equations. SIAM J Numer Anal 31:227–251
11. Jbilou K, Messaoudi A, Sadok H (1999) Global FOM and GMRES algorithms for linear systems with multiple right-hand sides. App Num Math 31:49–63
12. Jbilou K, Riquet AJ (2006) Projection methods for large Lyapunov matrix equations. Lin Alg and Appl 415(2):344–358
13. Lancaster P, Rodman L (1995) The algebraic Riccati equations. Clarendon Press, Oxford
14. Li J-R, Wang F, White J (1999) An efficient Lyapunov equation-based approach for generating reduced-order models of interconnect. In: 36th IEEE/ACM Design Automation Conference, New Orleans, LA, pp 1–6
15. Moore BC (1981) Principal component analysis in linear systems: controllability, observability and model reduction. IEEE Trans Auto Contr AC-26:17–31
16. Penzl T (2000) A cyclic low-rank Smith method for large sparse Lyapunov equations. SIAM J Sci Comput 21(4):1401–1418
17. Saad Y (1995) Iterative methods for sparse linear systems. PWS Press, New York
18. Saad Y (1990) Numerical solution of large Lyapunov equations. In: Kaashoek MA, van Schuppen JH, Ran AC (eds) Signal Processing, Scattering, Operator Theory and Numerical Methods. Proceedings of the international symposium MTNS-89, vol 3. Birkhauser, Boston, pp 503–511

19. Sadok H (2005) Analysis of the convergence of the minimal and the orthogonal residual methods. Numer Algo 40:201–216
20. Sima V (1996) Algorithms for Linear-Quadratic Optimization, Pure and applied Mathematics, vol. 200. Marcel Dekker, Inc., New York
21. Smith RA (1968) Matrix equation $XA + BX = C$. SIAM J Appl Math 16(1):198–201
22. Stykel T (2002) Analysis and numerical solution of generalized Lyapunov equations. Dissertation, TU Berlin
23. Van Dooren P (2000) Gramian based model reduction of large-scale dynamical systems. In: Numerical Analysis. Chapman and Hall/CRC Press, London, pp 231–247
24. Weyl H (1949) Inequalities between the two kinds of eigenvalues of a linear transformation. Proc Nat Sci USA 30:408–411

Chapter 15
Optimal Control for Linear Descriptor Systems with Variable Coefficients

Peter Kunkel and Volker Mehrmann

Abstract We study optimal control problems for general linear descriptor systems with variable coefficients. We derive necessary and sufficient optimality conditions for optimal solution. We also show how to solve these optimality systems via the solution of generalized Riccati-differential equations. and discussed how a modification of the cost functional leads to better solvability properties for the optimality system.

15.1 Introduction

We study the *linear-quadratic optimal control problem* to minimize the cost functional

$$\mathcal{J}(x,u) = \frac{1}{2}x(t_f)^T Mx(t_f) + \frac{1}{2}\int_{t_0}^{t_f} (x^T Wx + 2x^T Su + u^T Ru)\, dt, \tag{15.1}$$

Supported by *Deutsche Forschungsgemeinschaft*, through MATHEON, the DFG Research Center "Mathematics for Key Technologies" in Berlin.

P. Kunkel (✉)
Mathematisches Institut, Universität Leipzig, Augustusplatz 10–11, 04109 Leipzig, Germany
e-mail: kunkel@math.uni-leipzig.de

V. Mehrmann
Institut für Mathematik, MA 4-5, Technische Universität Berlin, 10623 Berlin, Germany
e-mail: mehrmann@math.tu-berlin.de

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_15,

subject to a general linear differential-algebraic equation (DAE), sometimes also called *descriptor system*, with variable coefficients of the form

$$E\dot{x} = Ax + Bu + f, \tag{15.2}$$

$$x(t_0) = x_0. \tag{15.3}$$

In this system, x is the state vector and u is the input (control) vector. Denoting by $\mathbb{R}^{n,k}$ the set of real $n \times k$ matrices and by $C^l(\mathbb{I}, \mathbb{R}^{n,k})$ the l-times continuously differentiable functions from an interval $\mathbb{I} = [t_0, t_f]$ to $\mathbb{R}^{n,k}$, we assume for the coefficients in the cost functional that $W \in C^0(\mathbb{I}, \mathbb{R}^{n,n})$, $S \in C^0(\mathbb{I}, \mathbb{R}^{n,m})$, $R \in C^0(\mathbb{I}, \mathbb{R}^{m,m})$. Furthermore, we require that W and R are pointwise symmetric and also that $M \in \mathbb{R}^{n,n}$ is symmetric. In the constraint descriptor system (15.2) we have coefficients $E \in C^0(\mathbb{I}, \mathbb{R}^{n,n})$, $A \in C^0(\mathbb{I}, \mathbb{R}^{n,n})$, $B \in C^0(\mathbb{I}, \mathbb{R}^{n,m})$, $f \in C^0(\mathbb{I}, \mathbb{R}^n)$, and we take $x_0 \in \mathbb{R}^n$.

Linear quadratic optimal control problems for and DAEs arise in the control of mechanical multibody systems, Eich-Soellner [12] and Gerdts [14]; electrical circuits, Günther and Feldmann [15]; chemical engineering, Diehl et al. [11] or heterogeneous systems, where different models are coupled together [26]. They usually represent local linearizations of general nonlinear control problems.

For ordinary differential equations, the theory of optimal control problems is well established, see, e.g., Gabasov and Kirillova [13] and Vinter [29] and the references therein. For systems where the constraint is a DAE, the situation is much more difficult and the existing literature is more recent. Results for special cases such as linear constant coefficient systems or special semi-explicit systems were e.g. obtained in Bender and Laub [5], Cobb [9], Lin and Yang [24], Mehrmann [25] and Pinho and Vinter [27].

A major difficulty in deriving adjoint equations and optimality systems is that for the potential candidates of adjoint equations and optimality systems, existence and uniqueness of solutions and correctness of initial contions cannot be guaranteed. See Refs. [1, 3, 10, 18, 23, 27, 28] for examples and a discussion of the difficulties.

Due to these difficulties, the standard approach to deal with optimal control problems for DAEs is to first perform some transformations, including *regularization and index reduction*. Such transformations exist. See Refs. [1, 22, 23].

There also exist some papers that derive optimality conditions for specially structured DAE systems directly [1, 14, 23, 27, 28]. Here, we discuss general unstructured linear systems with variable coefficients. We follow the so-called *strangeness index concept*, see Kunkel and Mehrmann [20], and consider the system in a behavior setting as a general over- or underdetermined differential-algebraic system.

15.2 Preliminaries

In this section we will introduce some notation and recall some results on differential-algebraic equations and on optimization theory.

Throughout the paper we assume that all functions are sufficiently smooth, i.e., sufficiently often continuously differentiable. We will make frequent use of the *Moore–Penrose pseudoinverse* of a matrix valued function $A : \mathbb{I} \to \mathbb{R}^{l,n}$, which is the unique matrix function $A^+ : \mathbb{I} \to \mathbb{R}^{n,l}$ that satisfies the four Penrose axioms

$$AA^+A = A, \quad A^+AA^+ = A^+, \quad (AA^+)^T = AA^+, \quad (A^+A)^T = A^+A \tag{15.4}$$

pointwise, see, e.g. Campbell and Meyer [8]. Note that if $A \in C^k(\mathbb{I}, \mathbb{R}^{l,n})$ and has constant rank on $\mathbb{I}$ then $A^+ \in C^k(\mathbb{I}, \mathbb{R}^{n,l})$.

We will generate our optimality conditions via some results from general optimization theory, see, e.g., Zeidler [30]. For this consider the optimization problem

$$\mathcal{J}(z) = \min ! \tag{15.5}$$

subject to the constraint

$$\mathcal{F}(z) = 0, \tag{15.6}$$

where

$$\mathcal{J} : \mathbb{D} \to \mathbb{R}, \quad \mathcal{F} : \mathbb{D} \to \mathbb{Y}, \quad \mathbb{D} \subseteq \mathbb{Z} \,\text{open},$$

with real Banach spaces $\mathbb{Z}, \mathbb{Y}$. Let, furthermore,

$$z^* \in \mathbb{M} = \{z \in \mathbb{D} \mid \mathcal{F}(z) = 0\}.$$

Then we will make use of the following theorem.

Theorem 1.1 *Let $\mathcal{J}$ be Fréchet differentiable in z^* and let $\mathcal{F}$ be a submersion in z^*, i.e., let $\mathcal{F}$ be Fréchet differentiable in a neighborhood of z^* with Fréchet derivative $D\mathcal{F}(z^*) : \mathbb{Z} \to \mathbb{Y}$ surjective and* $\operatorname{kernel} D\mathcal{F}(z^*)$ *continuously projectable.*

If z^* is a local minimum of (15.5)*, then there exists a unique Λ in the dual space $\mathbb{Y}^*$ of $\mathbb{Y}$ with*

$$D\mathcal{J}(z^*)\Delta z + \Lambda(D\mathcal{F}(z^*)\Delta z) = 0 \quad \textit{for all } \Delta z \in \mathbb{Z}. \tag{15.7}$$

The functional Λ in Theorem 1.1 is called the *Lagrange multiplier* associated with the constraint (15.6).

In general we are interested in function representations of the Lagrange multiplier functional Λ. Such representations are obtained by the following theorem.

Theorem 1.2 *Let* $\mathbb{Y} = C^0(\mathbb{I}, \mathbb{R}^l) \times \mathbb{V}$ *with a vector space* $\mathbb{V} \subseteq \mathbb{R}^l$ *and let* $(\lambda, \gamma) \in \mathbb{Y}$. *Then*

$$\Lambda(g, r) = \int_{t_0}^{t_f} \lambda(t)^T g(t) dt + \gamma^T r$$

defines a linear form $\Lambda \in \mathbb{Y}^*$, *which conversely uniquely determines* $(\lambda, \gamma) \in \mathbb{Y}$.

A sufficient condition that guarantees that also the minimum is unique is given by the following theorem, which, e.g., covers linear-quadratic control problems with positive definite reduced Hessian.

Theorem 1.3 *Suppose that* $\mathcal{F} : \mathbb{Z} \to \mathbb{Y}$ *is affine linear and that* $\mathcal{J} : \mathbb{Z} \to \mathbb{R}$ *is strictly convex on* $\mathbb{M}$, *i.e.*,

$$\mathcal{J}(\alpha z_1 + (1-\alpha) z_2) < \alpha \mathcal{J}(z_1) + (1-\alpha)\mathcal{J}(z_2) \quad \text{for all } z_1, z_2 \in \mathbb{M} \text{ with } z_1 \neq z_2 \text{ for all } \alpha \in (0,1),$$

then the optimization problem (15.5) *subject to* (15.6) *has a unique minimum.*

For our analysis, we will make use of some basic results on DAE theory. We follow [20] in notation and style of presentation.

When studying DAE control problems, one can distinguish two viewpoints. Either one takes the behavior approach and merges the variables x, u into one vector z, i.e. one studies

$$\mathcal{E}\dot{z} = \mathcal{A}z + f, \tag{15.8}$$

with

$$\mathcal{E} = [E\ 0], \quad \mathcal{A} = [A\ B] \in C^0(\mathbb{I}, \mathbb{R}^{n,n+m}).$$

For the underdetermined system (15.8) one then studies existence and uniqueness of solutions. The alternative is to keep the variables u and x separate. In this case one has to distinguish whether solutions exist for all controls in a given input set $\mathbb{U}$ or whether there exist controls at all for which the system is solvable, using the following solution concept.

Definition 1.4 Consider system (15.2) with a given fixed input function u that is sufficiently smooth. A function $x : \mathbb{I} \to \mathbb{R}^n$ is called a solution of (15.2) if $x \in C^1(\mathbb{I}, \mathbb{R}^n)$ and x satisfies (15.3) pointwise. It is called a solution of the initial value problem (15.2)–(15.3) if x is a solution of (15.2) and satisfies (15.3). An initial condition (15.3) is called consistent if the corresponding initial value problem has at least one solution.

In the sequel it will be necessary to slightly weaken this solution concept. Note, however, that under the assumption of sufficient smoothness we will always be in the case of Definition 1.4.

Definition 1.5 A control problem of the form (15.2) with a given set of controls $\mathbb{U}$ is called regular (locally with respect to a given solution $(\hat{x}, \hat{u})$ of (15.2)) if it has a unique solution for every sufficiently smooth input function u in a neighborhood of $\hat{u}$ and every initial value in a neighborhood of $\hat{x}(t_0)$ that is consistent for the system with input function u.

In order to analyze the properties of the system, in Kunkel and Mehrmann [19], Kunkel [22], hypotheses have been formulated which lead to an index concept, the so-called *strangeness index*, see Kunkel and Mehrmann [20] for a detailed derivation and analysis of this concept. Consider the constraint system in the behavior form (15.8). As in Campbell [7], we introduce a *derivative array*, which stacks the original equation and all its derivatives up to level ℓ in one large system,

$$M_\ell(t)\dot{z}_\ell = N_\ell(t)z_\ell + g_\ell(t), \tag{15.9}$$

where

$$\begin{aligned}
(M_\ell)_{i,j} &= \binom{i}{j}\mathcal{E}^{(i-j)} - \binom{i}{j+1}\mathcal{A}^{(i-j-1)}, \quad j = 0, \ldots, \ell,\\
(N_\ell)_{i,j} &= \begin{cases} \mathcal{A}^{(i)} & \text{for } i = 0, \ldots, \ell,\ j = 0,\\ 0 & \text{otherwise,} \end{cases}\\
(z_\ell)_j &= z^{(j)}, \quad j = 0, \ldots, \ell,\\
(g_\ell)_i &= f^{(i)}, \quad i = 0, \ldots, \ell.
\end{aligned}$$

To characterize the solution set we require the following hypothesis which can be proved for any linear system under some constant rank assumptions, see Kunkel and Mehrmann [20].

Hypothesis 1.6 There exist integers μ, d, a, and v such that the pair (M_μ, N_μ) associated with (15.9) has the following properties:

1. For all $t \in \mathbb{I}$ we have rank $M_\mu(t) = (\mu+1)n - a - v$. This implies the existence of a smooth matrix function Z of size $(\mu+1)n \times (a+v)$ and pointwise maximal rank satisfying $Z^T M_\mu = 0$.
2. For all $t \in \mathbb{I}$ we have rank $Z^T N_\mu [I_{n+m}\ 0 \ldots 0]^T = a$. This implies that without loss of generality Z can be partitioned as $Z = [\,Z_2 \quad Z_3\,]$, *with* Z_2 of size $(\mu+1)$ $n \times a$ and Z_3 of size $(\mu+1)n \times v$, such that $\hat{A}_2 = Z_2^T N_\mu [I_{n+m} 0 \ldots 0]^T$ has full row rank a and $Z_3^T N_\mu [I_{n+m} 0 \ldots 0]^T = 0$. Furthermore, there exists a smooth matrix function T_2 of size $(n+m) \times d, d = n - a$,and pointwise maximal rank satisfying $\hat{A}_2 T_2 = 0$.
3. For all $t \in \mathbb{I}$ we have rank $\mathcal{E}(t)T_2(t) = d$. This implies the existence of a smooth matrix function Z_1 of size $n \times d$ and pointwise maximal rank satisfying rank $\hat{E}_1 = \hat{d}$ with $\hat{E}_1 = Z_1^T E$.

The smallest μ for which this hypothesis holds is called the *strangeness-index* of the system and system (15.8) has the same solution set as the *reduced system*

$$\begin{bmatrix} \hat{E}_1(t) \\ 0 \\ 0 \end{bmatrix} \dot{z} = \begin{bmatrix} \hat{A}_1(t) \\ \hat{A}_2(t) \\ 0 \end{bmatrix} z + \begin{bmatrix} \hat{f}_1(t) \\ \hat{f}_2(t) \\ \hat{f}_3(t) \end{bmatrix}, \tag{15.10}$$

where $\hat{A}_1 = Z_1^T \mathcal{A}$, $\hat{f}_1 = Z_1^T f$, $\hat{f}_i = Z_i^T g_\mu$ for $i = 2, 3$.

If in this reduced system $\hat{f}_3$ does not vanish identically, then the system has no solution regardless how the input is chosen. If $\hat{f}_3 \equiv 0$, however, then we can just leave off the last v equations. For this reason, in the following analysis we assume w.l.o.g. that $v = 0$.

We can then rewrite the reduced system again in terms of the original variables u, x and obtain the system

$$\hat{E}\dot{x} = \hat{A}x + \hat{B}u + \hat{f}, \quad x(t_0) = x_0, \tag{15.11}$$

where

$$\hat{E} = \begin{bmatrix} \hat{E}_1 \\ 0 \end{bmatrix}, \quad \hat{A} = \begin{bmatrix} \hat{A}_1 \\ \hat{A}_2 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix}, \quad \hat{f} = \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \end{bmatrix} \tag{15.12}$$

with

$$\begin{aligned} \hat{E}_1 = Z_1^T E, \quad \begin{bmatrix} \hat{A}_1 & \hat{B}_1 \end{bmatrix} &= Z_1^T [A \quad B], \qquad \hat{f}_1 = Z_1^T f, \\ \begin{bmatrix} \hat{A}_2 & \hat{B}_2 \end{bmatrix} &= Z_2^T N_\mu [I_{n+m} 0 \cdots 0]^T, \quad \hat{f}_2 = Z_2^T g_\mu. \end{aligned}$$

By construction, in the reduced system (15.11), the matrix function $\hat{E}_1$ has full row rank d and $\begin{bmatrix} \hat{A}_2 T_2' & \hat{B}_2 \end{bmatrix}$ has full row rank a with a matrix function T_2' satisfying $\hat{E}_1 T_2' = 0$ and $T_2'^T T_2' = I_a$. Due to the fact that the solution set has not changed, one can consider the minimization of (15.1) subject to (15.11) instead of (15.2). Unfortunately, (15.11) still may not be solvable for all $u \in \mathbb{U} = C^0(\mathbb{I}, \mathbb{R}^m)$. But, since $\begin{bmatrix} \hat{A}_2 T_2' & \hat{B}_2 \end{bmatrix}$ has full row rank, it has been shown in Kunkel et al. [22], that there exists a linear feedback

$$u = Kx + w, \tag{15.13}$$

with $K \in C^0(\mathbb{I}, \mathbb{R}^{m,n})$ such that in the closed loop system

$$\hat{E}\dot{x} = (\hat{A} + \hat{B}K)x + \hat{B}w + \hat{f}, \quad x(t_0) = x_0, \tag{15.14}$$

the matrix function $(\hat{A}_2 + \hat{B}_2 K) T_2'$ is pointwise nonsingular, implying that the DAE in (15.14) is regular and strangeness-free for every given $w \in \mathbb{U} = C^0(\mathbb{I}, \mathbb{R}^m)$.

If we insert the feedback (15.13) in (15.11), then we obtain an optimization problem for the variables x, w instead of x, u, and (see Ref. [21]) these problems

and the solutions are directly transferable to each other. For this reason we may in the following assume w.l.o.g. that the differential-algebraic system (15.2) is regular and strangeness-free as a free system without control, i.e., when $u = 0$.

Under these assumptions it is then known, see, e.g., Kunkel and Mehrmann [20], that there exist $P \in C^0(\mathbb{I}, \mathbb{R}^{n,n})$ and $Q \in C^1(\mathbb{I}, \mathbb{R}^{n,n})$ pointwise orthogonal such that

$$\begin{aligned} \tilde{E} &= PEQ = \begin{bmatrix} E_{1,1} & 0 \\ 0 & 0 \end{bmatrix}, \quad \tilde{A} = PAQ - PE\dot{Q} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}, \\ \tilde{B} &= PB = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad \tilde{f} = Pf = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \\ x &= Q\tilde{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad x_0 = Q\tilde{x}_0 = \begin{bmatrix} x_{0,1} \\ x_{0,2} \end{bmatrix}, \end{aligned} \tag{15.15}$$

with $E_{1,1} \in C(\mathbb{I}, \mathbb{R}^{d,d})$ and $A_{2,2} \in C(\mathbb{I}, \mathbb{R}^{a,a})$ pointwise nonsingular. To get solvability of (15.2) for arbitrary $u \in C^0(\mathbb{I}, \mathbb{R}^m)$ and $f \in C^0(\mathbb{I}, \mathbb{R}^n)$, in view of

$$E\dot{x} = EE^+E\dot{x} = E\frac{d}{dt}(E^+Ex) - E\frac{d}{dt}(E^+E)x,$$

we have to interpret (15.2) as

$$E\frac{d}{dt}(E^+Ex) = (A + E\frac{d}{dt}(E^+E))x + Bu + f, \quad (E^+Ex)(t_0) = x_0, \tag{15.16}$$

which allows the larger solution space, see Kunkel and Mehrmann [17],

$$\mathbb{X} = C^1_{E^+E}(\mathbb{I}, \mathbb{R}^n) = \left\{ x \in C^0(\mathbb{I}, \mathbb{R}^n) \mid E^+Ex \in C^1(\mathbb{I}, \mathbb{R}^n) \right\} \tag{15.17}$$

equipped with the norm

$$\|x\|_{\mathbb{X}} = \|x\|_{C^0} + \|\frac{d}{dt}(E^+Ex)\|_{C^0}. \tag{15.18}$$

One should note that the choice of the initial value x_0 is restricted by the requirement in (15.16).

15.3 Necessary Optimality Conditions

In this section we derive necessary conditions for the linear quadratic optimal control problem to minimize (15.1) subject to (15.2) and (15.3). Following Kunkel and Mehrmann [17], we can use in (15.6) the constraint function

$$\mathcal{F} : \mathbb{X} \to \mathbb{Y} = C^0(\mathbb{I}, \mathbb{R}^n) \times \text{range } E^+(t_0)E(t_0)$$

given by

$$\mathcal{F}(x) = \left(E\frac{d}{dt}(E^+Ex) - (A + E\frac{d}{dt}(E^+E))x - Bu - f, (E^+Ex)(t_0) - x_0 \right).$$

Then from (15.16) we obtain

$$\begin{aligned} &PEQQ^T\frac{d}{dt}(QQ^TE^+P^TPEQQ^Tx) \\ &\quad = \left(PAQ + PEQQ^T\frac{d}{dt}(QQ^TE^+P^TPEQQ^T)Q \right) Q^Tx + PBu + Pf, \end{aligned}$$

or equivalently

$$\begin{aligned} &\tilde{E}Q^T\frac{d}{dt}(Q\tilde{E}^+\tilde{E}\tilde{x}) \\ &\quad = \left(\tilde{A} + PP^T\tilde{E}Q^T\dot{Q} + \tilde{E}Q^T\frac{d}{dt}(Q\tilde{E}^+\tilde{E}Q^T)Q \right)\tilde{x} + \tilde{B}u + \tilde{f}. \end{aligned}$$

Using the product rule and cancelling equal terms on both sides we obtain

$$\tilde{E}Q^TQ\frac{d}{dt}(\tilde{E}^+\tilde{E}\tilde{x}) = \left(\tilde{A} + \tilde{E}Q^T\dot{Q} + \tilde{E}\frac{d}{dt}(\tilde{E}^+\tilde{E}) + \tilde{E}\tilde{E}^+\tilde{E}\dot{Q}^TQ \right)\tilde{x} + \tilde{B}u + \tilde{f}.$$

Since by definition $\tilde{E}\tilde{E}^+\tilde{E} = \tilde{E}$ and $\dot{Q}^TQ + Q^T\dot{Q} = 0$, we then obtain

$$\tilde{E}\frac{d}{dt}(\tilde{E}^+\tilde{E}\tilde{x}) = \left(\tilde{A} + \tilde{E}\frac{d}{dt}(\tilde{E}^+\tilde{E}) \right)\tilde{x} + \tilde{B}u + \tilde{f}, \ (\tilde{E}^+\tilde{E}\tilde{x})(t_0) = \tilde{x}_0, \tag{15.19}$$

i.e., (15.16) transforms covariantly with pointwise orthogonal P and Q. If we partition P and Q conformably to (15.15) as

$$P = \begin{bmatrix} Z'^T \\ Z^T \end{bmatrix}, \quad Q = [\,T' \quad T\,],$$

then $Z^TE = 0, \ ET = 0$, and we can write (15.19) as

$$\begin{bmatrix} E_{1,1} & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}u + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \begin{bmatrix} x_1(t_0) \\ 0 \end{bmatrix} = \begin{bmatrix} x_{0,1} \\ 0 \end{bmatrix}.$$

Since $A_{2,2}$ is pointwise nonsingular, this system is uniquely solvable for arbitrary continuous functions u, f_1, and f_2, and for any $x_{0,1}$, with solution components satisfying

$$x_1 \in C^1(\mathbb{I}, \mathbb{R}^d), \quad x_2 \in C^0(\mathbb{I}, \mathbb{R}^a)$$

such that

$$x = Q\tilde{x} = [T'T]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{X}.$$

In particular, this construction defines a solution operator of the form

$$\mathcal{S} : \mathbb{U} \times \mathbb{Y} \to \mathbb{X}, \ (u, f, x_0) \mapsto x, \quad \mathbb{U} = C^0(\mathbb{I}, \mathbb{R}^m). \tag{15.20}$$

The Fréchet derivative $D\mathcal{F}(z)$ of $\mathcal{F}$ at $z \in \mathbb{Z} = \mathbb{X} \times \mathbb{U}$ is given by

$$D\mathcal{F}(z)\Delta z = \left(E\frac{d}{dt}(E^+E\Delta x) - (A + E\frac{d}{dt}(E^+E))\Delta x - B\Delta u, (E^+E\Delta x)(t_0) \right).$$

For $(g, r) \in \mathbb{Y}$, the equation $D\mathcal{F}(z) = (g, r)$ then takes the form

$$E\frac{d}{dt}(E^+E\Delta x) - (A + E\frac{d}{dt}(E^+E))\Delta x - B\Delta u = g, \ (E^+E\Delta x)(t_0) = r.$$

A possible solution is given by $u = 0$ and $\Delta x = \mathcal{S}(0, g, r)$, hence $D\mathcal{F}(z)$ is surjective. Moreover, the kernel is given by

$$\begin{aligned}
&\text{kernel}(D\mathcal{F}(z)) \\
&\quad = \{(\Delta x, \Delta u) \mid E\frac{d}{dt}(E^+E\Delta x) - (A + E\frac{d}{dt}(E^+E))\Delta x - B\Delta u = 0, \\
&\qquad (E^+E\Delta x)(t_0) = 0\} \\
&\quad = \{(\Delta x, \Delta u) \mid \Delta x = \mathcal{S}(\Delta u, 0, 0), \ \Delta u \in \mathbb{U}\} \subseteq \mathbb{X} \times \mathbb{U}.
\end{aligned}$$

Observe that kernel$(D\mathcal{F}(z))$ is parameterized with respect to Δu and that

$$\mathcal{P}(z) = \mathcal{P}(x, u) = (\mathcal{S}(u, 0, 0), u)$$

defines a projection $\mathcal{P} : \mathbb{Z} \to \mathbb{Z}$ onto kernel$(D\mathcal{F}(z))$. Here,

$$\|(\mathcal{S}(u, 0, 0), u)\|_{\mathbb{Z}} = \|\mathcal{S}(u, 0, 0)\|_{\mathbb{X}} + \|u\|_{\mathbb{U}}, \quad \text{and} \quad \|\mathcal{S}(u, 0, 0)\|_{\mathbb{X}} = \|x\|_{\mathbb{X}},$$

where x is the solution of the homogeneous problem

$$E\frac{d}{dt}(E^+Ex) - (A + E\frac{d}{dt}(E^+E))x - Bu = 0, \ (E^+Ex)(t_0) = 0. \tag{15.21}$$

Replacing again $x = Q\tilde{x}$ as in (15.15), we can write (15.21) as

$$\begin{bmatrix} E_{1,1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u, \ x_1(t_0) = 0,$$

or equivalently

$$E_{1,1}\dot{x}_1 = (A_{1,1} - A_{1,2}A_{2,2}^{-1}A_{2,1})x_1 + (B_1 - A_{1,2}A_{2,2}^{-1}B_2)u, \ x_1(t_0) = 0, \tag{15.22}$$

$$x_2 = -A_{2,2}^{-1}(A_{2,1}x_1 + B_2u). \tag{15.23}$$

The variation of the constant formula for the ODE in (15.22) yields the estimate $\|x_1\|_{C^0} + \|\dot{x}_1\|_{C^0} \le c_1\|u\|_{\mathbb{U}}$, with a constant c_1, and thus $\|x_2\|_{C^0} \le c_2\|u\|_{\mathbb{U}}$ with a constant c_2. Altogether, using (15.18) we then get the estimate

$$\|x\|_{\mathbb{X}} = \|x\|_{C^0} + \|\frac{d}{dt}(E^+Ex)\|_{C^0} = \|Q\tilde{x}\|_{C^0} + \|\frac{d}{dt}(E^+ET'x_1)\|_{C^0}$$
$$= \|Q\tilde{x}\|_{C^0} + \|\frac{d}{dt}(E^+ET')x_1 + (E^+ET')\dot{x}_1)\|_{C^0} \leq c_3\|u\|_{\mathbb{U}},$$

with a constant c_3. With this we have shown that $\mathcal{P}$ is continuous and thus kernel$(D\mathcal{F}(z))$ is continuously projectable. Hence, we can apply Theorem 1.2 and obtain the existence of a unique Lagrange multiplier $\Lambda \in \mathbb{Y}^*$. To determine Λ, we make the ansatz

$$\Lambda(g, r) = \int_{t_0}^{t_f} \lambda^T g \; dt + \gamma^T r. \tag{15.24}$$

Using the cost function (15.1) we have

$$D\mathcal{J}(z)\Delta z = x(t_f)^T M\Delta x(t_f) + \int_{t_0}^{t_f} (x^T W\Delta x + x^T S\Delta u + u^T S^T\Delta x + u^T R\Delta u) \; dt,$$

and in a local minimum $z = (x, u)$ we obtain that for all $(\Delta x, \Delta u) \in \mathbb{X} \times \mathbb{U}$ the relationship

$$\begin{aligned} 0 &= x(t_f)^T M\Delta x(t_f) + \gamma^T(E^+E\Delta x)(t_0)) \\ &+ \int_{t_0}^{t_f} (x^T W\Delta x + x^T S\Delta u + u^T S^T\Delta x + u^T R\Delta u) \; dt \\ &+ \int_{t_0}^{t_f} \lambda^T \left(E\frac{d}{dt}(E^+E\Delta x) - (A + E\frac{d}{dt}(E^+E))\Delta x - B\Delta u \right) \; dt \end{aligned} \tag{15.25}$$

has to hold. If $\lambda \in C^1_{E^+E}(\mathbb{I}, \mathbb{R}^n)$, then, using the fact that $E = EE^+E = (EE^+)^TE$, we have by partial integration

$$\begin{aligned} \int_{t_0}^{t_f} \lambda^T E\frac{d}{dt}(E^+E\Delta x)\, dt &= \int_{t_0}^{t_f} \lambda^T (EE^+)^T E\frac{d}{dt}(E^+E\Delta x) dt \\ &= \int_{t_0}^{t_f} (EE^+\lambda)^T E\frac{d}{dt}(E^+E\Delta x)\, dt \\ &= \lambda^T EE^+E\Delta x\Big|_{t_0}^{t_f} - \int_{t_0}^{t_f} \frac{d}{dt}\big[(EE^+\lambda)^T E\big](E^+E\Delta x) dt \end{aligned}$$

$$= \lambda^T E\Delta x\Big|_{t_0}^{t_f} - \int_{t_0}^{t_f} \left[\frac{d}{dt}(EE^+\lambda)^T E + (EE^+\lambda)^T \dot{E}\right](E^+E\Delta x)\,dt$$

$$= \lambda^T E\Delta x\Big|_{t_0}^{t_f} - \int_{t_0}^{t_f} \left[\frac{d}{dt}(EE^+\lambda)^T E\Delta x + (EE^+\lambda)^T \dot{E}E^+E\Delta x\right] dt.$$

Therefore, we can rewrite (15.25) as

$$0 = \int_{t_0}^{t_f} \Bigg(x^T W + u^T S^T - \frac{d}{dt}(EE^+\lambda)^T E - (EE^+\lambda)^T \dot{E}E^+E - \lambda^T A$$
$$- \lambda^T E\frac{d}{dt}(E^+E)\Bigg)\Delta x\,dt + \int_{t_0}^{t_f} (x^T S + u^T R - \lambda^T B)\Delta u\,dt + x(t_f)^T M\Delta x(t_f)$$
$$+ \lambda^T(t_f)E(t_f)\Delta x(t_f) - \lambda^T(t_0)E(t_0)\Delta x(t_0) + \gamma^T(E^+E\Delta x)(t_0).$$

If we first choose $\Delta x = 0$ and vary over all $\Delta u \in \mathbb{U}$, then we obtain the necessary *optimality condition*

$$S^T x + Ru - B^T\lambda = 0. \tag{15.26}$$

Varying then over all $\Delta x \in \mathbb{X}$ with $\Delta x(t_0) = \Delta x(t_f) = 0$, we obtain the *adjoint equation*

$$Wx + Su - E^T\frac{d}{dt}(EE^+\lambda) - E^+E\dot{E}^TEE^+\lambda - A^T\lambda - \frac{d}{dt}(E^+E)E^T\lambda = 0. \tag{15.27}$$

Varying finally over $\Delta x(t_0) \in \mathbb{R}^n$ and $\Delta x(t_f) \in \mathbb{R}^n$, respectively, yields the initial condition

$$(E^+(t_0)E(t_0))^T\gamma = E^T(t_0)\lambda(t_0), \text{ i.e., } \gamma = E(t_0)^T\lambda(t_0) \tag{15.28}$$

and the end condition

$$Mx(t_f) + E(t_f)^T\lambda(t_f) = 0, \tag{15.29}$$

respectively.

Observe that the condition (15.29) can only hold if $Mx(t_f) \in$ cokernel $E(t_f)$. This extra requirement for the cost term involving the final state was observed already for constant coefficient systems in Mehrmann [25] and Kurina and März [23]. If this condition on M holds, then from (15.29) we obtain that

$$\lambda(t_f) = -E^+(t_f)^T Mx(t_f).$$

Using the identity

$$\begin{aligned} EE^+\dot{E}E^+E + E\frac{d}{dt}(E^+E) &= EE^+(\dot{E}E^+E + E\frac{d}{dt}(E^+E)) \\ &= EE^+\frac{d}{dt}(EE^+E) = EE^+\dot{E}, \end{aligned}$$

we obtain the initial value problem for the adjoint equation in the form

$$\begin{aligned} &E^T\frac{d}{dt}(EE^+\lambda) = Wx + Su - (A + EE^+\dot{E})^T\lambda, \\ &(EE^+\lambda)(t_f) = -E^+(t_f)^T Mx(t_f). \end{aligned} \tag{15.30}$$

As we had to interpret (15.2) in the form (15.16) for the correct choice of the spaces, (15.30) is the correct interpretation of the problem

$$\frac{d}{dt}(E^T\lambda) = Wx + Su - A^T\lambda, \quad \lambda(t_f) = -E^+(t_f)^T Mx(t_f). \tag{15.31}$$

Note again that these re-interpretations are not crucial when the coefficient functions are sufficiently smooth.

For the adjoint equation and the optimality condition, we will now study the action of the special equivalence transformations of (15.15). Using that $(EE^+)^T = EE^+$, we obtain for (15.30) the transformed system

$$\begin{aligned} &QE^TP^TP\frac{d}{dt}(P^TPEQQ^TE^+P^TP\lambda) \\ &\quad = Q^TWQQ^Tx + Q^TSu - (Q^TA^TP^T + Q^T\dot{E}P^TPEQQ^TE^+P^T)P\lambda. \end{aligned}$$

Setting

$$\tilde{W} = Q^TWQ, \quad \tilde{S} = Q^TS, \quad \tilde{\lambda} = P\lambda, \quad \tilde{M} = Q(t_f)^TMQ(t_f),$$

we obtain

$$\begin{aligned} &\tilde{E}P\frac{d}{dt}(P^T\tilde{E}\tilde{E}^+\tilde{\lambda}) = \tilde{W}\tilde{x} + \tilde{S}u \\ &\quad - \left(\tilde{A}^T + \dot{Q}^TQ\tilde{E}^T + Q^T(Q\tilde{E}^T\dot{P} + Q\dot{\tilde{E}}^TP + \dot{Q}\tilde{E}^TP)P^T\tilde{E}\tilde{E}^+\right)\tilde{\lambda} \end{aligned}$$

or equivalently

$$\begin{aligned} &\tilde{E}P\dot{P}^T\tilde{E}\tilde{E}^+\tilde{\lambda} + \tilde{E}\frac{d}{dt}(\tilde{E}\tilde{E}^+\tilde{\lambda}) = \tilde{W}\tilde{x} + \tilde{S}u \\ &\quad - \left(\tilde{A}^T + \dot{Q}^TQ\tilde{E}^T + \tilde{E}^T\dot{P}P^T\tilde{E}\tilde{E}^+ + \dot{\tilde{E}}^T\tilde{E}\tilde{E}^+ + Q^T\dot{Q}\tilde{E}^T\tilde{E}\tilde{E}^+\right)\tilde{\lambda}. \end{aligned}$$

Using the orthogonality of P, Q, which implies that $\dot{Q}^TQ + Q\dot{Q} = 0$ and $\dot{P}^TP + P\dot{P} = 0$, we obtain

$$\tilde{E}\frac{d}{dt}(\tilde{E}\tilde{E}^+\tilde{\lambda}) = \tilde{W}\tilde{x} + \tilde{S}u - (\tilde{A} + \tilde{E}\dot{\tilde{E}}^+\dot{\tilde{E}})^T\tilde{\lambda}.$$

For the initial condition we obtain accordingly

$$\begin{aligned}(\tilde{E}\tilde{E}^{+}\tilde{\lambda})(t_f) &= (PEQQ^TE^+P^TP\lambda)(t_f) = (PEE^+\lambda)(t_f)\\ &= -P(t_f)E^+(t_f)^TQ(t_f)Q(t_f)^TMQ(t_f)Q(t_f)^Tx(t_f)\\ &= -\tilde{E}^+(t_f)^T\tilde{M}\tilde{x}(t_f).\end{aligned}$$

Thus, we have shown that (15.30) transforms covariantly and that we may consider (15.30) in the condensed form associated with (15.15). Setting (with comfortable partitioning)

$$\tilde{\lambda} = \begin{bmatrix}\lambda_1\\ \lambda_2\end{bmatrix},\quad \tilde{W} = \begin{bmatrix}W_{1,1} & W_{1,2}\\ W_{2,1} & W_{2,2}\end{bmatrix},\quad \tilde{S} = \begin{bmatrix}S_1\\ S_2\end{bmatrix},\quad \tilde{M} = \begin{bmatrix}M_{1,1} & M_{1,2}\\ M_{2,1} & M_{2,2}\end{bmatrix}, \tag{15.32}$$

we obtain the system

$$\begin{aligned}E_{1,1}^T\lambda_1 &= W_{1,1}x_1 + W_{1,2}x_2 + S_1u - (A_{1,1} + \dot{E}_{1,1})^T\lambda_1 - A_{2,1}^T\lambda_2,\\ \lambda_1(t_f) &= -E_{1,1}^{-T}(t_f)(M_{1,1}x_1(t_f) + M_{1,2}x_2(t_f)),\\ 0 &= W_{2,1}x_1 + W_{2,2}x_2 + S_2u - A_{1,2}^T\lambda_1 - A_{2,2}^T\lambda_2.\end{aligned}$$

We immediately see that as a differential-algebraic equation in λ this system is strangeness-free, and, since $A_{2,2}$ is pointwise nonsingular, this system yields a unique solution $\lambda \in C^1_{EE^+}(\mathbb{I},\mathbb{R}^n)$ for every $(x,u) \in \mathbb{Z}$.

If $(x,u) \in \mathbb{Z}$ is a local minimum, then from (15.29) and (15.30) we can determine Lagrange multipliers $\lambda \in C^1_{EE^+}(\mathbb{I},\mathbb{R}^n)$ and $\gamma \in \operatorname{cokernel} E(t_f)$. It has been shown in Kunkel and Mehrmann [21] that this λ also satisfies the optimality condition (15.26).

It thus follows that the functional that is defined via (15.24), (15.30) and $\gamma = E(t_0)^T\lambda(t_0)$ as in (15.28) has the property (15.7) and is, therefore, the desired Lagrange multiplier. Furthermore, it is then clear that $(z,\lambda) = (x,u,\lambda)$ is a local minimum of the unconstrained optimization problem

$$\begin{aligned}\hat{\mathcal{J}}(z,\lambda) &= \mathcal{J}(z) + \Lambda(\mathcal{F}(z))\\ &= \frac{1}{2}x(t_f)^TMx(t_f) + \frac{1}{2}\int_{t_0}^{t_f}(x^TWx + 2x^TSu + u^TRu)\,dt\\ &\quad + \int_{t_0}^{t_f}\lambda^T\left(E(\frac{d}{dt}(E^+Ex) - (A + E\frac{d}{dt}(E^+E))x - Bu - f\right)\,dt\\ &\quad + \gamma^T((E^+Ex)(t_0) - x_0) = \min!\end{aligned} \tag{15.33}$$

We can summarize our analysis in the following theorem.

Theorem 1.7 *Consider the optimal control problem* (15.1) *subject to* (15.2) *with a consistent initial condition. Suppose that* (15.2) *is strangeness-free as a behavior system and that* $Mx(t_f) \in \text{cokernel } E(t_f)$.

If $(x,u) \in \mathbb{X} \times \mathbb{U}$ *is a solution to this optimal control problem, then there exists a Lagrange multiplier function* $\lambda \in C^1_{E^+E}(\mathbb{I},\mathbb{R}^n)$, *such that* (x,λ,u) *satisfy the* optimality boundary value problem

$$\begin{aligned}
&\text{(a)} \quad E\tfrac{d}{dt}(E^+Ex) = (A + E\tfrac{d}{dt}(E^+E))x + Bu + f,\ (E^+Ex)(t_0) = x_0,\\
&\text{(b)} \quad E^T\tfrac{d}{dt}(EE^+\lambda) = Wx + Su - (A + EE^+\dot{E})^T\lambda,\\
&\qquad (EE^+\lambda)(t_f) = -E^+(t_f)^T Mx(t_f),\\
&\text{(c)} \quad 0 = S^Tx + Ru - B^T\lambda.
\end{aligned} \tag{15.34}$$

15.4 The Strangeness Index of the Optimality System

An important question for the numerical computation of optimal controls is when the optimality system (15.34) is regular and strangeness-free and whether the strangeness index of (15.34) is related to the strangeness index of the original system. For other index concepts like the tractability index this question has been discussed in Balla et al. [2], Balla and März [4] and Kurina and März [23].

Theorem 1.8 *The DAE in* (15.34) *is regular and strangeness-free if and only if*

$$\hat{R} = \begin{bmatrix} 0 & A_{2,2} & B_2 \\ A_{2,2}^T & W_{2,2} & S_2 \\ B_2^T & S_2^T & R \end{bmatrix} \tag{15.35}$$

is pointwise nonsingular, where we used the notation of (15.15).

Proof Consider the reduced system (15.11) associated with the DAE (15.2) and derive the boundary value problem (15.34) from this reduced system. If we carry out the change of basis with orthogonal transformations leading to the normal form (15.15), then we obtain the transformed boundary value problem

$$\begin{aligned}
&\text{(a)} \quad E_{1,1}\dot{x}_1 = A_{1,1}x_1 + A_{1,2}x_2 + B_1u + f_1,\ x_1(t_0) = x_{0,1}\\
&\text{(b)} \quad 0 = A_{2,1}x_1 + A_{2,2}x_2 + B_2u + f_2,\\
&\text{(c)} \quad E_{1,1}^T\dot{\lambda}_1 = W_{1,1}x_1 + W_{1,2}x_2 + S_1u - (A_{1,1} + \dot{E}_{1,1})^T\lambda_1 - A_{2,1}^T\lambda_2,\\
&\qquad \lambda_1(t_f) = -E_{1,1}(t_f)^{-T}M_{1,1}x_1(t_f),\\
&\text{(d)} \quad 0 = W_{2,1}x_1 + W_{2,2}x_2 + S_2u - A_{1,2}^T\lambda_1 - A_{2,2}^T\lambda_2,\\
&\text{(e)} \quad 0 = S_1^Tx_1 + S_2^Tx_2 + Ru - B_1^T\lambda_1 - B_2^T\lambda_2.
\end{aligned} \tag{15.36}$$

We can rewrite (15.36) in a symmetrized way as

$$\left[\begin{array}{cc|ccc} 0 & E_{1,1} & 0 & 0 & 0 \\ -E_{1,1}^T & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} -\dot{\lambda}_1 \\ \dot{x}_1 \\ \hline -\dot{\lambda}_2 \\ \dot{x}_2 \\ \dot{u} \end{array}\right]$$

$$= \left[\begin{array}{cc|ccc} 0 & A_{1,1} & 0 & A_{1,2} & B_1 \\ (A_{1,1}+\dot{E}_{1,1})^T & W_{1,1} & A_{2,1}^T & W_{2,1}^T & S_1 \\ \hline 0 & A_{2,1} & 0 & A_{2,2} & B_2 \\ A_{1,2}^T & W_{2,1} & A_{2,2}^T & W_{2,2} & S_2 \\ B_1^T & S_1^T & B_2^T & S_2^T & R \end{array}\right] \left[\begin{array}{c} -\lambda_1 \\ x_1 \\ \hline -\lambda_2 \\ x_2 \\ u \end{array}\right] + \left[\begin{array}{c} f_1 \\ 0 \\ \hline f_2 \\ 0 \\ 0 \end{array}\right]. \tag{15.37}$$

Obviously this DAE is regular and strangeness-free if and only if the symmetric matrix function $\hat{R}$ is pointwise nonsingular. □

If (15.2) with $u = 0$ is regular and strangeness-free, then $A_{2,2}$ is pointwise nonsingular. In our analysis we have shown that this property can always be achieved, but note that we do not need that $A_{2,2}$ is pointwise nonsingular to obtain a regular and strangeness-free optimality system (15.34).

On the other hand for $\hat{R}$ to be pointwise nonsingular, it is clearly necessary that $[A_{2,2}\ B_2]$ has pointwise full row rank. This condition is equivalent to the condition that the behavior system (15.8) belonging to the reduced problem satisfies Hypothesis 1.6 with $\mu = 0$ and $\nu = 0$, see [22] for a detailed discussion of this issue and also for an extension of these results to the case of control systems with output equations.

Example 1.9 An example of a control problem of the form (15.2) that is not directly strangeness-free in the behavior setting is discussed in Backes [1 p. 50]. This linear-quadratic control problem has the coefficients

$$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad f = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad W = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad R = 1,$$

and the initial condition $x_1(0) = \alpha,\ x_2(0) = 0$. A possible reduced system (15.11) is given by

$$\hat{E} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{A} = A, \quad \hat{B} = B.$$

Observe that the corresponding free system of this reduced problem (i.e. with $u = 0$) itself is regular and strangeness-free. It follows that the adjoint equation and the optimality condition are given by

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\lambda}_1 \\ \dot{\lambda}_2 \\ \dot{\lambda}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix},$$

$$0 = -\begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} + u,$$

respectively, with the end condition $\lambda_1(t_f) = -x_1(t_f)$.

We obtain that the matrix function $\hat{R}$ in (15.35) given by

$$\hat{R} = \left[\begin{array}{ccc|cc} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 1 \end{array}\right]$$

is pointwise nonsingular, and hence the boundary value problem (15.34) is regular and strangeness-free. Moreover, it has a unique solution which is given by

$$x_1 = \alpha(1 - \frac{t}{2 + t_f}), \quad x_2 = \lambda_3 = 0, \quad x_3 = u = -\lambda_2 = -\frac{\alpha}{2 + t_f}, \quad \lambda_1 = -\frac{2\alpha}{2 + t_f}.$$

Example 1.10 In Kurina and März [23] the optimal control problem to minimize

$$\mathcal{J}(x, u) = \int_0^{t_f} (x_1(t)^2 + u(t)^2)\, dt$$

subject to

$$\frac{d}{dt}\left(\begin{bmatrix} 0 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \quad x_2(0) = x_{2,0}$$

is discussed. Obviously, x_1 does not enter the DAE and therefore rather plays the role of a control than of a state. Consequently, the corresponding free system is not regular. Rewriting the system as

$$\begin{bmatrix} 0 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \quad x_2(0) = x_{2,0},$$

and analyzing this system in our described framework, we first of all observe that this system possesses a strangeness index and that it is even regular and strangeness-free as a behavior system. A possible reduced system (15.11) is given by

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad x_2(0) = x_{2,0}.$$

The corresponding free system is not regular although it is strangeness-free. Moreover, we can read off

$$\hat{R} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix},$$

which is obviously pointwise nonsingular. Hence, the boundary value problem (15.34) is regular and strangeness-free.

15.5 Sufficient Conditions and the Formal Optimality System

One may be tempted to drop the assumptions of Theorem 1.7 and to consider directly the *formal optimality boundary value problem* given by

$$\begin{aligned} &\text{(a)} \quad E\dot{x} = Ax + Bu + f, \ x(t_0) = x_0 \\ &\text{(b)} \quad \tfrac{d}{dt}(E^T\lambda) = Wx + Su - A^T\lambda, \ (E^T\lambda)(t_f) = -Mx(t_f), \\ &\text{(c)} \quad 0 = S^T x + Ru - B^T\lambda. \end{aligned} \tag{15.38}$$

But it was already observed in Backes [1], Kurina and März [23] and Mehrmann [25] that it is in general not correct to just consider this system. First of all, as we have shown, the cost matrix M for the final state has to be in the correct cokernel, since otherwise the initial value problem may not be solvable due to a wrong number of conditions. An example for this is given in Backes [1], Kurina and März [23]. A further difficulty arises from the fact that the formal adjoint equation (15.38b) may not be strangeness-free in the variable λ and thus extra differentiability conditions may arise which may not be satisfied, see the following example.

Example 1.11 Consider the problem

$$\mathcal{J}(x,u) = \frac{1}{2}\int_0^1 (x_1(t)^2 + u(t)^2)dt = \min!$$

subject to the differential-algebraic system

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}.$$

The reduced system (15.11) in this case is the purely algebraic equation

$$0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} f_1 + \dot{f}_2 \\ f_2 \end{bmatrix}.$$

The associated adjoint equation (15.30) is then

$$0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix},$$

and no initial conditions are necessary. The optimality condition (15.26) is given by

$$0 = u - \lambda_1.$$

A simple calculation yields the optimal solution

$$x_1 = u = \lambda_1 = -\frac{1}{2}(f_1 + \dot{f}_2), \quad x_2 = -f_2, \quad \lambda_2 = 0.$$

If, however, we consider the formal adjoint equation (15.38b) given by

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\lambda}_1 \\ \dot{\lambda}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}, \quad \lambda_1(1) = 0$$

together with the optimality condition (15.38c), then we obtain that

$$x_1 = u = \lambda_1 = -\frac{1}{2}(f_1 + \dot{f}_2), \quad x_2 = -f_2, \quad \lambda_2 = -\frac{1}{2}(\dot{f}_1 + \ddot{f}_2)$$

without using the initial condition $\lambda_1(1) = 0$. Depending on the data, this initial condition may be consistent or not. In view of the correct solution it is obvious that this initial condition should not be present. But this cannot be seen from (15.38). Moreover, the determination of λ_2 requires more smoothness of the inhomogeneity than in (15.34).

As we have demonstrated by Example 1.11, difficulties may arise by working with the formal adjoint equations. In particular, they may not be solvable due to additional initial conditions or due to lack of smoothness. If, however, the cost functional is positive semidefinite, then one can show that any solution of the formal optimality system yields a minimum and thus constitutes a sufficient condition. This was, e.g., shown for ODE optimal control in Campbell [6], for linear constant coefficient DAEs in Mehrmann [25], and in a specific setting for linear DAEs with variable coefficients in Backes [1]. The general result is given by the following theorem.

Theorem 1.12 *Consider the optimal control problem* (15.1) *subject to* (15.2) *with a consistent initial condition and suppose that in the cost functional* (15.1) *we have that*

$$\begin{bmatrix} W & S \\ S^T & R \end{bmatrix}, \ M$$

are (pointwise) positive semidefinite. If (x^*, u^*, λ) *satisfies the formal optimality system* (15.38) *then for any* (x, u) *satisfying* (15.2) *we have*

$$\mathcal{J}(x, u) \geq \mathcal{J}(x^*, u^*).$$

Proof We consider the function

$$\Phi(s) = \mathcal{J}((1-s)x^* + sx, (1-s)u^* + su)$$

and show that $\Phi(s)$ has a minimum at $s = 0$. We have

$$\begin{aligned}\Phi(s) = \frac{1}{2}\int_{t_0}^{t_f} \Bigg(& (1-s)^2 \begin{bmatrix} x^* \\ u^* \end{bmatrix}^T \begin{bmatrix} W & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x^* \\ u^* \end{bmatrix} \\ & + 2s(1-s) \begin{bmatrix} x^* \\ u^* \end{bmatrix}^T \begin{bmatrix} W & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \\ & + s^2 \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} W & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \Bigg) \, dt \\ & + \frac{1}{2}\Big((1-s)^2 x^{*T} M x^* + 2s(1-s) x^{*T} M x + s^2 x^T M x \Big)\Big|_{t=t_f},\end{aligned}$$

and

$$\begin{aligned}\frac{d}{ds}\Phi(0) = \int_{t_0}^{t_f} \Bigg(& \begin{bmatrix} x^* \\ u^* \end{bmatrix}^T \begin{bmatrix} W & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \\ & - \begin{bmatrix} x^* \\ u^* \end{bmatrix}^T \begin{bmatrix} W & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x^* \\ u^* \end{bmatrix} \Bigg) \, dt \\ & + \Big(x^{*T} M x - x^{*T} M x^* \Big)\Big|_{t=t_f}.\end{aligned}$$

If we consider (15.38b) for (x^*, u^*) and multiply from the left by x^{*T}, then we obtain

$$-x^{*T} E^T \dot{\lambda} - x^{*T} \dot{E}^T \lambda + x^{*T} W x^* + x^{*T} S u^* - x^{*T} A^T \lambda = 0.$$

Inserting the transpose of (15.38a) yields

$$-x^{*T} E^T \dot{\lambda} - x^{*T} \dot{E}^T \lambda + x^{*T} W x^* + x^{*T} S u^* - \dot{x}^{*T} E^T \lambda + u^{*T} B^T \lambda + f^T \lambda = 0.$$

Finally, inserting (15.38c) gives

$$\begin{aligned}
&-\frac{d}{dt}\left(x^{*T}E^T\lambda\right) + x^{*T}Wx^* + 2x^{*T}Su^* + u^{*T}Ru^* + f^T\lambda \\
&\quad = -\frac{d}{dt}\left(x^{*T}E^T\lambda\right) + \begin{bmatrix} x^* \\ u^* \end{bmatrix}^T \begin{bmatrix} W & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x^* \\ u^* \end{bmatrix} + f^T\lambda = 0.
\end{aligned}$$

Analogously, for (x, u) we obtain the equation

$$-\frac{d}{dt}\left(x^TE^T\lambda\right) + \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} W & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x^* \\ u^* \end{bmatrix} + f^T\lambda = 0.$$

Thus, we obtain that

$$\begin{aligned}
\frac{d}{ds}\Phi(0) &= \int_{t_0}^{t_f} \left(\frac{d}{dt}\left(x^TE^T\lambda\right) - \frac{d}{dt}\left(x^{*T}E^T\lambda\right)\right)\, dt + \left(x^{*T}M(x - x^*)\right)\Big|_{t=t_f} \\
&= \left((x - x^*)^TE^T\lambda\right)\Big|_{t_0}^{t_f} + \left((x - x^*)^TMx^*\right)\Big|_{t=t_f} = 0,
\end{aligned}$$

since $x(t_0) = x^*(t_0)$ and $(E^T\lambda)(t_f) = -Mx^*(t_f)$. Due to the positive semidefiniteness of the cost functional we have

$$\begin{aligned}
\frac{d^2}{ds^2}\Phi(0) = &\int_{t_0}^{t_f} \begin{bmatrix} x - x^* \\ u - u^* \end{bmatrix}^T \begin{bmatrix} W & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x - x^* \\ u - u^* \end{bmatrix} dt \\
&+ \left((x - x^*)^TM(x - x^*)\right)\Big|_{t=t_f} \geq 0
\end{aligned}$$

and this implies that Φ has a minimum at $s = 0$, which may, however, not be unique. □

We can summarize the results of this section as follows. The necessary optimality condition for the optimal control problem (15.1) subject to (15.2) is given by (15.34) and not by the formal optimality system (15.38). If, however, (15.38) has a solution, then it corresponds to a minimum of the optimal control problem. If no index reduction is performed, then a necessary condition for the DAE in (15.1) to be regular and strangeness-free is that the DAE (15.2) itself is regular and strangeness-free as a behavior system.

15.6 Differential-Algebraic Riccati Equations

One of the classical approaches to solve boundary value problems arising in the linear-quadratic optimal control problem of ordinary differential equations is the use of Riccati differential equations. This approach has also been studied in the

case of differential-algebraic equations, See Refs. [5, 18, 25], and it has been observed in Kunkel and Mehrmann [18] that the Riccati approach is not always possible. If, however, some further conditions hold, then the Riccati approach can be carried out.

Let us first consider the optimality boundary value problem (15.34) in its symmetrized normal form (15.37). If $\hat{R}$ is pointwise nonsingular, then

$$\begin{bmatrix} -\lambda_2 \\ x_2 \\ u \end{bmatrix} = -\hat{R}^{-1}\left(\begin{bmatrix} 0 & A_{2,1} \\ A_{1,2}^T & W_{2,1} \\ B_1^T & S_1^T \end{bmatrix}\begin{bmatrix} -\lambda_1 \\ x_1 \end{bmatrix} + \begin{bmatrix} f_2 \\ 0 \\ 0 \end{bmatrix}\right). \tag{15.39}$$

The remaining equations can be written as

$$\begin{aligned}\begin{bmatrix} E_{1,1}\dot{x}_1 \\ \frac{d}{dt}((-E_{1,1}^T)(-\lambda_1)) \end{bmatrix} &= \begin{bmatrix} 0 & A_{1,1} \\ A_{1,1}^T & W_{1,1} \end{bmatrix}\begin{bmatrix} -\lambda_1 \\ x_1 \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 & A_{1,2} & B_1 \\ A_{2,1}^T & W_{2,1}^T & S_1 \end{bmatrix}\begin{bmatrix} -\lambda_2 \\ x_2 \\ u \end{bmatrix} + \begin{bmatrix} f_1 \\ 0 \end{bmatrix}. \end{aligned} \tag{15.40}$$

Inserting (15.39) and defining

a. $F_1 = E_{1,1}^{-1}\left(A_{1,1} - [0\, A_{1,2}\, B_1]\hat{R}^{-1}[A_{2,1}^T\, W_{2,1}^T\, S_1]^T\right)$
b. $G_1 = E_{1,1}^{-1}[0 \quad A_{1,2} \quad B_1]\,\hat{R}^{-1}\,[0 \quad A_{1,2} \quad B_1]^T\,E_{1,1}^{-T}$
c. $H_1 = W_{1,1} - [A_{2,1}^T \quad W_{2,1}^T \quad S_1]\,\hat{R}^{-1}\,[A_{2,1}^T \quad W_{2,1}^T \quad S_1]^T$,
d. $g_1 = E_{1,1}^{-1}\left(f_1 - [0 \quad A_{1,2} \quad B_1]\,\hat{R}^{-1}\,[f_2^T \quad 0 \quad 0]^T\right)$,
e. $h_1 = -[A_{2,1}^T \quad W_{2,1}^T \quad S_1]\,\hat{R}^{-1}\,[f_2^T \quad 0 \quad 0]^T$,

we obtain the boundary value problem with *Hamiltonian structure* given by

$$\begin{aligned} &\text{(a)} \quad \dot{x}_1 = F_1 x_1 + G_1(E_{1,1}^T\lambda_1) + g_1,\ x_1(t_0) = x_{0,1}, \\ &\text{(b)} \quad \tfrac{d}{dt}(E_{1,1}^T\lambda_1) = H_1 x_1 - F_1^T(E_{1,1}^T\lambda_1) + h_1,\ (E_{1,1}^T\lambda_1)(t_f) = -M_{1,1}x_1(t_f). \end{aligned} \tag{15.41}$$

Making the ansatz

$$E_{1,1}^T\lambda_1 = X_{1,1}x_1 + v_1, \tag{15.42}$$

and using its derivative

$$\frac{d}{dt}(E_{1,1}^T\lambda_1) = \dot{X}_{1,1}x_1 + X_{1,1}\dot{x}_1 + \dot{v}_1,$$

the Hamiltonian boundary value problem (15.41) yields

$$\dot{X}_{1,1}x_1 + X_{1,1}(F_1x_1 + G_1(X_{1,1}x_1 + v_1) + g_1) + \dot{v}_1 = H_1x_1 - F_1^T(X_{1,1}x_1 + v_1) + h_1,$$

or

$$\begin{aligned}&\big(\dot X_{1,1} + X_{1,1}F_1 + F_1^T X_{1,1} + X_{1,1}G_1X_{1,1} - H_1\big)x_1\\ &\quad + (\dot v_1 + X_{1,1}G_1v_1 + F_1^T v_1 + X_{1,1}g_1 - h_1) = 0.\end{aligned}$$

Thus, we can solve the two initial value problems

$$\dot X_{1,1} + X_{1,1}F_1 + F_1^T X_{1,1} + X_{1,1}G_1X_{1,1} - H_1 = 0,\ X_{1,1}(t_f) = -M_{1,1}, \tag{15.43}$$

and

$$\dot v_1 + X_{1,1}G_1v_1 + F_1^T v_1 + X_{1,1}g_1 - h_1 = 0,\ v_1(t_f) = 0, \tag{15.44}$$

to obtain $X_{1,1}$ and v_1 and to decouple the solution to (15.41).

In this way we have obtained a Riccati approach for the dynamic part of the system. Ideally, however, we would like to have a Riccati approach directly for the boundary value problem associated with (15.19), (15.30), and (15.26) in the original data, without carrying out the change of bases and going to normal form. If we make a similar ansatz for the general situation, i.e., $\lambda = Xx + v$, then we face the problem that neither the whole x nor the whole λ may be differentiable. To accomodate for the appropriate solution spaces, we therefore make the modified ansatz

$$\begin{aligned}&\text{(a)}\quad \lambda = XEx + v = XEE^+Ex + v,\\ &\text{(b)}\quad \tfrac{d}{dt}(EE^+\lambda) = \tfrac{d}{dt}(EE^+X)Ex + (EE^+X)\dot EE^+Ex\\ &\qquad\qquad +(EE^+X)E\tfrac{d}{dt}(E^+Ex) + \tfrac{d}{dt}(E^+Ev),\end{aligned} \tag{15.45}$$

where

$$X \in C^1_{EE^+}(\mathbb{I}, \mathbb{R}^{n,n}), \quad v \in C^1_{EE^+}(\mathbb{I}, \mathbb{R}^n). \tag{15.46}$$

In this way we have obtained an ansatz that fits to the solution spaces for x and λ. The disadvantage of this approach, however, is that $X(I - EE^+)$ now can be chosen arbitrarily. Using again the transformation to normal form (15.15) and that

$$P\lambda = PXP^TPEQQ^Tx + Pv,$$

we obtain

$$\tilde\lambda = \begin{bmatrix}\lambda_1\\ \lambda_2\end{bmatrix} = \tilde X\tilde E\tilde x + \tilde v,$$

with

$$\tilde X = PXP^T = \begin{bmatrix}\tilde X_{1,1} & \tilde X_{1,2}\\ \tilde X_{2,1} & \tilde X_{2,2}\end{bmatrix}, \quad \tilde v = \begin{bmatrix}\tilde v_1\\ \tilde v_2\end{bmatrix}.$$

Comparing with (15.42) we obtain

$$X_{1,1} = E_{1,1}^T\tilde X_{1,1}E_{1,1}, \quad v_1 = E_{1,1}^T\tilde v_1.$$

In particular, we obtain that $\tilde{X}_{1,1}$ and $\tilde{v}_1$ are continuously differentiable under the assumption that (15.43) is solvable on the interval $\mathbb{I}$. Furthermore, $\tilde{X}_{1,1}$ is pointwise symmetric. From (15.39) we then obtain

$$\begin{aligned}\lambda_2 &= [I\ 0\ 0]\hat{R}^{-1}\left(\begin{bmatrix} 0 & A_{2,1} \\ A_{1,2}^T & W_{2,1} \\ B_1^T & S_1^T \end{bmatrix}\begin{bmatrix} -(\tilde{X}_{1,1}E_{1,1}x_1 + \tilde{v}_1) \\ x_1 \end{bmatrix} + \begin{bmatrix} f_2 \\ 0 \\ 0 \end{bmatrix}\right) \\ &= \tilde{X}_{2,1}E_{1,1}x_1 + \tilde{v}_2,\end{aligned}$$

with

$$\tilde{X}_{2,1}E_{1,1} = [I\ 0\ 0]\hat{R}^{-1}\begin{bmatrix} 0 & A_{2,1} \\ A_{1,2}^T & W_{2,1} \\ B_1^T & S_1^T \end{bmatrix}\begin{bmatrix} -\tilde{X}_{1,1}E_{1,1} \\ I \end{bmatrix},$$

and

$$\tilde{v}_2 = [I\ 0\ 0]\hat{R}^{-1}\begin{bmatrix} f_2 \\ -A_{1,2}^T\tilde{v}_1 \\ -B_1^T\tilde{v}_1 \end{bmatrix}.$$

If we assume that R itself is pointwise nonsingular (which corresponds to the assumption that all controls are weighted in the cost functional), then from (15.26) we obtain that

$$u = R^{-1}(B^T\lambda - S^Tx)$$

and thus from (15.30) and (15.21) we obtain

$$\begin{aligned}&E^T\frac{d}{dt}(EE^+X)Ex + E^T(EE^+X)\dot{E}E^+Ex + E^T\frac{d}{dt}(EE^+v) + E^T(EE^+X) \\ &\quad\cdot\left(Ax + E\frac{d}{dt}(E^+E)x + BR^{-1}B^T(XEx + v) - BR^{-1}S^Tx + f\right) \\ &= Wx + SR^{-1}B^T(XEx + v) - SR^{-1}S^Tx - (A + EE^+\dot{E})^T(XEx + v),\end{aligned}$$

or

$$\begin{aligned}&\Big(E^T\frac{d}{dt}(EE^+X)E + E^T(EE^+X)\dot{E}E^+E + E^T(EE^+X)E\frac{d}{dt}(E^+E) \\ &\quad + \dot{E}^T(EE^+X)EE^+E + E^TXA + E^TXBR^{-1}B^TXE + A^TXE \\ &\quad - E^TXBR^{-1}S^T - SR^{-1}B^TXE + SR^{-1}S^T - W\Big)x \\ &\quad + \Big(\frac{d}{dt}(EE^+v) + E^TXBR^{-1}B^Tv + E^TXf - SR^{-1}B^Tv + A^Tv + \dot{E}^T(EE^+v)\Big) = 0.\end{aligned}$$

Introducing the notation

$$\begin{aligned} &\text{(a)} \quad F = A - BR^{-1}S^T, \\ &\text{(b)} \quad G = BR^{-1}B^T, \\ &\text{(c)} \quad H = W - SR^{-1}S^T, \end{aligned} \tag{15.47}$$

we obtain

$$\begin{aligned} &\left(\frac{d}{dt}(E^T(EE^+X)E(E^+E)) + E^TXF + F^TXE + E^TXGXE - H\right)x \\ &\quad + \left(E^T\frac{d}{dt}(EE^+v) + \dot{E}^T(EE^+v) + E^TXGv + F^Tv + E^TXf\right) = 0, \end{aligned}$$

which yields the two initial value problems

$$\frac{d}{dt}(E^TXE) + E^TXF + F^TXE + E^TXGXE - H = 0, \ (E^TXE)(t_f) = -M, \tag{15.48}$$

and

$$\frac{d}{dt}(E^Tv) + E^TXGv + F^Tv + E^TXf = 0, (E^Tv)(t_f) = 0. \tag{15.49}$$

Note that we must have $M = E(t_f)^T\tilde{M}E(t_f)$ with suitable $\tilde{M}$ and $H = E^T\tilde{H}E$ with suitable $\tilde{H}$ as necessary condition for the solvability of (15.48). Note also that (as already in the case of ODEs) the optimality boundary value problem (15.34) may be solvable, whereas (15.48) does not allow for a solution on the whole interval $\mathbb{I}$.

The analysis in this section shows that we can obtain a Riccati approach if the system (15.2) is strangeness-free in the behavior setting and if R is invertible.

15.7 A Modified Cost Functional

In the previous sections we have derived necessary conditions for linear-quadratic control problem and studied how these can be solved. In particular, we have seen that extra conditions on the cost functional have to hold for the optimality system or the associated Riccati equation to have a solution.

Since the cost functional is often a matter of choice one could modify it to reduce the requirements. A simple modification is the following cost functional, See Ref. [16] in the case of constant coefficients,

$$\mathcal{J}(x,u) = \frac{1}{2}x(t_f)^T\tilde{M}x(t_f) + \frac{1}{2}\int_{t_0}^{t_f}(x^T\tilde{W}x + 2x^T\tilde{S}u + u^TRu)\,dt, \tag{15.50}$$

with $\tilde{M} = E(t_f)^TME(t_f)$, $\tilde{W} = E^TWE$, and $\tilde{S} = E^TS$.

Assuming again that the original system (15.2) is strangeness-free as a behavior system, the same analysis as before leads to the modified optimality boundary value problem

$$\begin{aligned}
&\text{(a)} \quad E\tfrac{d}{dt}(E^+Ex) = (A + E\tfrac{d}{dt}(E^+E))x + Bu + f, (E^+Ex)(t_0) = x_0,\\
&\text{(b)} \quad E^T\tfrac{d}{dt}(EE^+\lambda) = E^TWEx + E^TSu - (A + EE^+\dot{E})\lambda,\\
&\qquad (EE^+\lambda)(t_f) = -E^+(t_f)^TE(t_f)^TME(t_f)x(t_f),\\
&\text{(c)} \quad 0 = S^TEx + Ru - B^T\lambda.
\end{aligned} \tag{15.51}$$

Considering the conditions that guarantee that the optimality system is again strangeness-free, we obtain the following corollary.

Corollary 1.13 *Consider the optimal control problem to minimize* (15.50) *subject to* (15.2) *and assume that* (15.2) *is strangeness-free as a free system* (*with* $u = 0$). *Then the optimality system* (15.51) *is strangeness-free if and only if* R *is pointwise nonsingular.*

Proof Consider the system (15.2) in the normal form (15.15). By assumption, we have that $A_{2,2}$ is invertible and in the transformed cost functional (15.32) we obtain that $\tilde{S}_2 = 0$ and $\tilde{W}_{2,2} = 0$. The modified matrix $\hat{R}$ then takes the form

$$\hat{R} = \begin{bmatrix} 0 & A_{2,2} & B_2 \\ A_{2,2}^T & 0 & 0 \\ B_2^T & 0 & R \end{bmatrix},$$

which is clearly pointwise nonsingular if and only if R is pointwise nonsingular. □

The Riccati approach also changes when we use the modified cost functional. In particular, we obtain

$$\begin{aligned}
&\text{(a)} \quad \tilde{F} = A - BR^{-1}\tilde{S}^T,\\
&\text{(b)} \quad \tilde{G} = G = BR^{-1}B^T,\\
&\text{(c)} \quad \tilde{H} = \tilde{W} - \tilde{S}R^{-1}\tilde{S}^T,
\end{aligned} \tag{15.52}$$

In this case one obtains the two initial value problems

$$\frac{d}{dt}(E^TXE) + E^TX\tilde{F} + \tilde{F}^TXE + E^TX\tilde{G}XE - \tilde{H} = 0, \quad (E^TXE)(t_f) = -\tilde{M}, \tag{15.53}$$

and

$$\frac{d}{dt}(E^Tv) + E^TX\tilde{G}v + \tilde{F}^Tv + E^TXf = 0, (E^Tv)(t_f) = 0. \tag{15.54}$$

Observe that the necessary conditions for solvability as stated in the end of Sect. 15.6 for (15.48) are now trivially fulfilled.

15.8 Conclusions

We have presented the optimal control theory for general unstructured linear systems of differential-algebraic equations with variable coefficients. We have derived necessary and sufficient conditions as well as a Riccati approach. We have also shown how the cost function may be modified to guarantee that the optimality system is regular and strangeness-free.

The presented results can be generalized to nonlinear control problems and suitable numerical methods for the solution of the optimality system are presented in Kunkel and Mehrmann [21].

References

1. Backes A (2006) Optimale Steuerung der linearen DAE im Fall Index 2. Dissertation, Mathematisch-Naturwissenschaftliche Fakultät, Humboldt-Universität zu Berlin, Berlin, Germany
2. Balla K, Kurina G, März R (2003) Index criteria for differential algebraic equations arising from linear-quadratic optimal control problems. Preprint 2003–14, Institut für Mathematik, Humboldt-Universität zu Berlin, Berlin, Germany
3. Balla K, Linh VH (2005) Adjoint pairs of differential-algebraic equations and Hamiltonian systems. Appl Numer Math 53:131–148
4. Balla K, März R (2004) Linear boundary value problems for differential algebraic equations. Math Notes 5:3–17
5. Bender D, Laub A (1987) The linear quadratic optimal regulator problem for descriptor systems. IEEE Trans Automat Control 32:672–688
6. Campbell SL (1980) Singular systems of differential equations I. Pitman, San Francisco
7. Campbell SL (1987) Comment on controlling generalized state-space (descriptor) systems. Int J Control 46:2229–2230
8. Campbell SL, Meyer CD (1979) Generalized inverses of linear transformations. Pitman, San Francisco
9. Cobb JD (1983) A further interpretation of inconsistent initial conditions in descriptor-variable systems. IEEE Trans Automat Control AC 28:920–922
10. Devdariani EN, Ledyaev YS (1999) Maximum principle for implicit control systems. Appl Math Optim 40:79–103
11. Diehl M, Leineweber DB, Schäfer A, Bock HG, Schlöder JP (2002) Optimization of multiple-fraction batch distillation with recycled waste cuts. AIChE J 48(12):2869–2874
12. Eich-Soellner E, Führer C (1998) Numerical methods in multibody systems. Teubner Verlag, Stuttgart
13. Gabasov R, Kirillova F (1976) The qualitative theory of optimal processes. Marcel Dekker, New York
14. Gerdts M (2005) Local minimum principle for optimal control problems subject to index two differential algebraic equations systems. Technical report, Fakultät für Mathematik, Universität Hamburg, Hamburg, Germany
15. Günther M, Feldmann U (1999) CAD-based electric-circuit modeling in industry I. Mathematical structure and index of network equations. Surv Math Ind 8:97–129
16. Ionescu V, Oara C, Weiss M (1999) Generalized Riccati theory and robust control: a Popov function approach. Wiley, Chichester
17. Kunkel P, Mehrmann V (1996) Generalized inverses of differential-algebraic operators. SIAM J Matr Anal Appl 17:426–442

18. Kunkel P, Mehrmann V (1997) The linear quadratic control problem for linear descriptor systems with variable coefficients. Math Control Signals Sys 10:247–264
19. Kunkel P, Mehrmann V (1998) Regular solutions of nonlinear differential-algebraic equations and their numerical determination. Numer Math 79:581–600
20. Kunkel P, Mehrmann V (2006) Differential-algebraic equations. Analysis and numerical solution. EMS Publishing House, Zürich
21. Kunkel P, Mehrmann V (2006). Necessary and sufficient conditions in the optimal control for general nonlinear differential-algebraic equations. Preprint 355, DFG Research Center MATHEON, TU Berlin, Berlin
22. Kunkel P, Mehrmann V, Rath W (2001) Analysis and numerical solution of control problems in descriptor form. Math Control Signals Sys 14:29–61
23. Kurina GA, März R (2004) On linear-quadratic optimal control problems for time-varying descriptor systems. SIAM J Cont Optim 42:2062–2077
24. Lin J-Y, Yang Z-H (1988) Optimal control for singular systems. Int J Control 47:1915–1924
25. Mehrmann V (1991) The autonomous linear quadratic control problem. Springer, Berlin
26. Otter M, Elmqvist H, Mattson SE (2006) Multi-domain modeling with modelica. In: Fishwick P (eds), CRC handbook of dynamic system modeling. CRC Press (To appear)
27. do Pinho M, de Vinter RB (1997) Necessary conditions for optimal control problems involving nonlinear differential algebraic equations. J Math Anal Appl 212:493–516
28. Roubicek T, Valasek M (2002) Optimal control of causal differential-algebraic systems. J Math Anal Appl 269:616–641
29. Vinter R (2000) Optimal Control. Birkhäuser, Boston
30. Zeidler E (1985) Nonlinear functional analysis and its applications III. Variational methods and optimization. Springer, New-York

Chapter 16
Robust Pole Assignment for Ordinary and Descriptor Systems via the Schur Form

Tiexiang Li, Eric King-wah Chu and Wen-Wei Lin

Abstract In Chu (Syst Control Lett 56:303–314, 2007), the pole assignment problem was considered for the control system $\dot{x} = Ax + Bu$ with linear state-feedback $u = Fx$. An algorithm using the Schur form has been proposed, producing good suboptimal solutions which can be refined further using optimization. In this paper, the algorithm is improved, incorporating the minimization of the feedback gain $\|F\|$. It is also extended for the pole assignment of the descriptor system $E\dot{x} = Ax + Bu$ with linear state- and derivative-feedback $u = Fx - G\dot{x}$. Newton refinement for the solutions is discussed and several illustrative numerical examples are presented.

16.1 Introduction

Let (A, B) denote the *ordinary* system

$$\dot{x} = Ax + Bu \tag{16.1}$$

T. Li (✉)
Department of Mathematics, Southeast University, Nanjing 211189, People's Republic of China
e-mail: feco@sohu.com

E. K. Chu
School of Mathematical Sciences, Building 28, Monash University 3800, Clayton, Australia
e-mail: eric.chu@sci.monash.edu.au

W.-W. Lin
Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan
e-mail: wwlin@math.nctu.edu.tw

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_16,

with the open-loop system matrix $A \in \mathbb{R}^{n\times n}$ and input matrix $B \in \mathbb{R}^{n\times m}(n > m)$. The state-feedback pole assignment problem (SFPAP) seeks a control matrix $F \in \mathbb{R}^{m\times n}$ such that the closed-loop system matrix $A_c \equiv A + BF$ has prescribed eigenvalues or poles. Equivalently, we are seeking a control matrix F such that

$$(A + BF)X = X\Lambda \tag{16.2}$$

for some given Λ with desirable poles and nonsingular matrix X. Notice that Λ does not have to be in Jordan form, and X can be well-conditioned even with defective multiple eigenvalues in some well-chosen Λ.

The SFPAP is solvable for arbitrary closed-loop spectrum when (A, B) is controllable, i.e., when $[sI - A, B]$ $(\forall s \in \mathbb{C})$ or $[B, AB, \ldots, A^{n-1}B]$ are full ranked. The problem has been thoroughly investigated; see [6, 7, 17], the references therein, or any standard textbook in control theory. It is well known that the single-input case $(m = 1)$ has a unique solution, while the multi-input case has some degrees of freedom left in the problem. A notable effort in utilizing these degrees of freedom sensibly was made by Kautsky et al. [13], with the conditioning of the closed-loop spectrum (including $\|X^{-1}\|_F$ where X contains the normalized closed-loop eigenvectors) being optimized.

When solving a pole assignment problem with a particular robustness measure optimized, we call it a robust pole assignment problem (RPAP). It is important to realize that there are many RPAPs, with different robustness measures. In this paper, a weighted sum of the departure from normality and the feedback gain is used as the robustness measure. For other possibilities and a comparison of different robustness measures, see [6, 7].

In [7], (Λ, X) in Schur form is chosen together with the upper triangular part of Λ (the departure from normality) minimized. The resulting non-iterative algorithm `SCHUR` produces a suboptimal solution F for any *given* x_1 (the first Schur vector in X). In this paper, this original `SCHUR` algorithm is improved, minimizing a weighted sum of the departure from normality and the feedback gain. The `SCHUR` algorithm is also extended for the pole assignment of the descriptor system

$$E\dot{x} = Ax + Bu \tag{16.3}$$

with linear state- and derivative-feedback $u = Fx - G\dot{x}$; for the solvability and other algorithms for the problem, see [5, 8, 12, 18]. Furthermore, a Newton refinement procedure is implemented, using the suboptimal but feasible solution produced by `SCHUR` as starting point. Note that a common problem with Newton's method is the lack of a feasible starting point which is close enough to the (locally) optimal solution. This `SCHUR-NEWTON` algorithm will produce a better solution, utilizing the freedom in x_1.

The main contributions of this paper are as follows. For the RPAP, it is not appropriate to compare different algorithms built on different robustness measures. An algorithm is only "better" for a particular measure, which is unlikely to be the sole consideration the control design. However, our algorithm is well-conditioned in the sense that the Schur form, which is well-conditioned or differentiable even

for multiple eigenvalues, is utilized. In addition, the robustness measure is a flexible weighed sum of departure from normality and feedback gains.

After the introduction here, Sect. 16.2 quotes the theorems on the departure from normality measures for ordinary and descriptor systems. Sections 16.3 and 16.4 present the SCHUR-NEWTON algorithms for ordinary and descriptor systems. Section 16.5 contains some numerical examples and Sect. 16.6 some concluding remarks. The (generalized) spectrum is denoted by $\sigma(\cdot)$ and $(A)_{ij}$ is the (i,j) entry in A.

16.2 Departure from Normality

Consider the Schur decomposition $A_c = A + BF = X\Lambda X^\top$ with $\Lambda = D + N$ where N is the off-diagonal part of Λ. The departure of normality measure $\Delta_v(A) \equiv \|N\|_v$ was first considered by Henrici [11] and the following perturbation result was produced:

Theorem 2.1 (Henrici Theorem [11]) *Let* $A,\ \delta A \in \mathbb{C}^{n\times n}, \delta A \neq 0, \mu \in \sigma(A+\delta A)$ *and let* $\|\cdot\|_v$ *be any norm stronger than the spectral norm (with* $\|M\|_2 \leq \|M\|_v$ *for all* M*). Then*

$$\min_{\lambda\in\sigma(A)} |\lambda - \mu| \leq \frac{\eta}{g(\eta)} ||\delta A||_v, \quad \eta = \frac{\Delta_v(A)}{||\delta A||_v}$$

where $g(\eta)$*is the only positive root of* $g + g^2 + \cdots + g^n = \eta (\eta \geq 0)$.

Other related perturbation results involving the departure from normality measure $\Delta_v(A)$ can be found in [1–4, 9, 15].

For generalized eigenvalue problems [16], we have this generalization [1, 9, 11, 15]:

Definition 2.1 Let $\{A, E\}$ define a regular matrix pencil and $\mathscr{U}_{\{A,E\}}$ be the set of all pairs of transformations $\{Z, U\}$ which satisfy the following conditions:

1. $Z, U \in \mathbb{C}^{n\times n}, Z$ is nonsingular and U is unitary;
2. $Z^{-1}AU$ and $Z^{-1}EU$ are both upper triangular; and
3. $|(Z^{-1}AU)_{ii}|^2 + |(Z^{-1}EU)_{ii}|^2 = 1\ (i = 1, \ldots, n)$.

Let $(Z, U) \in \mathscr{U}_{\{A,E\}}$ and $diag(A) \in \mathbb{C}^n$ denote the diagonal matrix sharing the diagonal of A. Denote

$$\mu(Z, U) \equiv \left\| \left[Z^{-1}AU - \mathrm{diag}(Z^{-1}AU), Z^{-1}EU - \mathrm{diag}(Z^{-1}EU) \right] \right\|_2$$

and

$$\Delta_2(A, E) \equiv \inf_{\{Z,U\}\in\mathscr{U}_{\{A,E\}}} \mu(Z, U).$$

Then $\Delta_2(A, E)$ is called the departure from normality measure of $\{A, E\}$.

Definition 2.2 Let $\sigma(A,E) = \{(\alpha_i, \beta_i)\}$ denote the spectrum of the pencil $\alpha A - \beta E$ and let $(\alpha, \beta) \in \sigma(\tilde{A}, \tilde{E})$. The spectral variation of $(\tilde{A}, \tilde{E})$ from (A,E) equals

$$s_{(A,E)}(\tilde{A}, \tilde{E}) \equiv \max_{(\alpha,\beta)} \{s_{(\alpha,\beta)}\}, \quad s_{(\alpha,\beta)} \equiv \min_i \{|\alpha\beta_i - \beta\alpha_i|\}$$

Theorem 2.2 (Henrici Theorem [16]) *Let* $\{A,E\}$ *and* $\{\tilde{A}, \tilde{E}\}$ *define regular pencils of the same dimension,* $\Delta_2(A,E)$ *is the departure from normality measure of* $\{A,B\}$*, and suppose* $\Delta_2(A,E) \neq 0$*. Let* $W = (A,E)$ *and* $\widetilde{W} = (\tilde{A}, \tilde{E})$*, then*

$$s_{(A,E)}(\tilde{A}, \tilde{E}) \leq \frac{\eta}{g(\eta)} [1 + \Delta_2(A,E)]\, d_2(W, \widetilde{W})$$

where $d_2(W, \widetilde{W}) = \| \sin \Theta(W, \widetilde{W}) \|_2$ *denotes the distance between* W *and* $\widetilde{W}$*,*

$$\eta = \frac{\Delta_2(A,E)}{[1 + \Delta_2(A,E)]\, d_2(W, \widetilde{W})},$$

and $g(\eta)$ *is the unique nonnegative root of* $g + g^2 + \cdots + g^n = \eta (\eta > 0)$*.*

Based on Theorems 2.1 and 2.2, we shall minimize the departure from normality (as part of the robustness measure) of the closed-loop matrix pencil in the effort to control the robustness of the closed-loop spectrum or system.

16.3 Ordinary System

Similar to the development in [7] we have

$$(A + BF)X = X\Lambda \tag{16.4}$$

assuming without loss of generality that the feedback matrix B has full rank and possesses the QR decomposition

$$B = [Q_1, Q_2] \left[R_B^\top, 0\right]^\top = Q \left[R_B^\top, 0\right]^\top = Q_1 R_B. \tag{16.5}$$

Pre-multiplying the eigenvalue equation (16.4) by $B^\dagger = R_B^{-1} Q_1^\top$ and $Q_2^\top$, we obtain

$$\begin{aligned} &Q_2^\top (AX - X\Lambda) = 0, \\ &F = R_B^{-1} Q_1^\top (X\Lambda X^{-1} - A). \end{aligned} \tag{16.6}$$

For a given Λ, we can select X from

$$\left[I_n \otimes (Q_2^\top A) - \Lambda^\top \otimes Q_2^\top\right] v(X) = 0$$

where $\otimes$ denote the Kronecker product [9] and $v(X)$ stacks the columns of X [which is different from Stk$(\cdot)$ defined later]. For the selected X, the solution to the SFPAP can then be obtained using (16.6).

16.3.1 Real Eigenvalues

First consider the real case when all the eigenvalues in Λ are real; i.e., let the real $X = [x_1, \ldots, x_n]$ and $\Lambda = D + N$ where $D = \text{diag}\{\lambda_1, \ldots, \lambda_n\}, N$ is the strict upper triangular part of Λ with columns $[\eta_j^\top, 0^\top]^\top$ and $\eta_j \in \mathbb{R}^{j-1} (j = 1, \ldots, n)$. Equation (16.4) and the properties of Schur decompositions imply

$$\begin{bmatrix} Q_2^T(A - \lambda_j I_n) - Q_2^T X_{-j} \\ X_{-j}^T \qquad\qquad 0 \end{bmatrix} \begin{bmatrix} x_j \\ \eta_j \end{bmatrix} = 0, \quad j = 1, \ldots, n. \tag{16.7}$$

Let $S_j \equiv [S_{j1}^\top, S_{j2}^\top]^\top$ be a unitary basis of the above subspace, so that

$$x_j = S_{j1} u_j, \quad \eta_j = S_{j2} u_j. \tag{16.8}$$

The Schur vectors x_j can then be select from the subspaces defined in (16.8) with $\|\eta_j\|$ (and thus $\|N\|_F$) minimized using the GSVD [3, 4, 7]. When $j = 1$, η_1 is degenerate and there will be no minimization to control x_1, making it a free parameter. To overcome this freedom, we can optimize some additional robustness measure. The feedback gain $\|F\|$, which is of some importance in many engineering applications, is then a natural choice. Making use of the orthogonality of X in (16.6), we have

$$Y = [y_1, \ldots, y_n] \equiv FX = B^\dagger (X\Lambda - AX), \quad \|Y\| = \|F\|$$

and

$$y_j = B^\dagger [\lambda_j I_n - A, X_{-j}] \left[x_j^\top, \eta_j^\top \right]^\top = S_{j3} u_j$$

with

$$S_{j3} \equiv B^\dagger \left[\lambda_j I_n - A, X_{-j} \right] S_j, \quad X_{-j} \equiv [x_1, \ldots, x_{j-1}].$$

Incorporating the departure from normality and feedback gain into one weighted robustness measure $r(X, N) \equiv \omega_1^2 \|F\|_F^2 + \omega_2^2 \|N\|_F^2$, we have

$$r(X, N) = \sum_{j=1}^{n} u_j^\top \left(\omega_1^2 S_{j3}^\top S_{j3} + \omega_2^2 S_{j2}^\top S_{j2} \right) u_j.$$

Other more sophisticated scaling schemes can be achieved in a similar fashion. For example, different η_j and y_j can have different weights, or (ω_1, ω_2) can be set to $(1, 0)$ (thus ignoring $\|N\|$).

For $j = 1, \ldots, n$, the vectors x_j and η_j can then be chosen by minimizing $r(X, N)$ while scaling x_j to be a unit vector, by considering the GSVD [3, 4] of (S_{j1}, S_{j4}),

with $S_{j4} \equiv [\omega_1 S_{j2}^\top, \omega_2 S_{j3}^\top]^\top$. Finally, a more direct and efficient way to construct $S_{jk}(k = 1, 2, 3)$ will be to consider, directly from (16.4) and similar to (16.7),

$$\begin{bmatrix} A - \lambda_j I_n & B & -X_{-j} \\ X_{-j}^\top & 0 & 0 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \\ \eta_j \end{bmatrix} = 0$$

so that the columns of $[S_{j1}^\top, S_{j2}^\top, S_{j3}^\top]^\top$ form a (unitary) basis for the above subspace, with the feedback matrix retrievable from $F = YX^\top$. We are then required to choose u_j such that

$$\min_{u_j} = \frac{u_j^\top \hat{S}_{2j}^\top \hat{S}_{2j} u_j}{u_j^\top S_{1j}^\top S_{1j} u_j}, \quad \hat{S}_{2j} = \begin{bmatrix} \omega_1 S_{j2} \\ \omega_2 S_{j3} \end{bmatrix}.$$

From the initial point obtained via the above SCHUR algorithm, we use Newton's algorithm to optimize the problem. Note that the freedom in the order of the poles remains to be utilized.

Optimization Problem 1

$$\min \omega_1^2 \|Y\|_F^2 + \omega_2^2 \|N\|_F^2 \quad \text{s.t.} \quad \begin{cases} AX + BY - X(D + N) = 0 \\ X^\top X - I = 0 \end{cases}$$

where N is $n \times n$ strictly upper triangular and X is $n \times n$ orthogonal.

Optimization problem 1 is equivalent to:

$$\begin{aligned} &\min \omega_1^2 v(Y)^\top v(Y) + \omega_2^2 \mathrm{Stk}(N)^\top \mathrm{Stk}(N) \\ &\text{s.t.} \begin{cases} (I \otimes A)v(X) + (I \otimes B)v(Y) - (D^\top \otimes I)v(X) - (N^\top \otimes I)v(X) = 0 \\ d_0(X)^\top v(X) - \mathrm{Stk}(I) = 0 \end{cases} \end{aligned}$$

where $\mathrm{Stk}(N) = \left[\eta_2^\top, \ldots, \eta_n^\top\right]^\top = \left[\eta_{12}^\top | \eta_{13}^\top, \eta_{23}^\top | \ldots | \eta_{1n}^\top, \ldots, \eta_{n-1,n}^\top\right]^\top \in \mathbb{R}^{n(n-1)/2 \equiv p}$, which stacks the nontrivial elements of η_j $(j = 2, \ldots, n)$. Here, we write $C = [c_1, \ldots, c_n] \in \mathbb{R}^{k \times n}$, so

$$d_0(C) \equiv [c_1 \oplus [c_1, c_2] \oplus \cdots \oplus [c_1, \ldots, c_n]] \in \mathbb{R}^{kn \times q}.$$

We then consider the Lagrange function of the Optimization Problem 1:

$$\begin{aligned} L(\gamma, \delta, v(X), v(Y), \mathrm{Stk}(N)) &= \omega_1^2 v(Y)^\top v(Y) + \omega_2^2 \mathrm{Stk}(N)^\top \mathrm{Stk}(N) \\ &+ \gamma^\top \left[(I \otimes A)v(X) + (I \otimes B)v(Y) - (D^\top \otimes I)v(X) - (N^\top \otimes I)v(X)\right] \\ &+ \delta^\top \left[d_0(X)^\top v(X) - \mathrm{Stk}(I)\right] \end{aligned}$$

where $\gamma = \left[\underbrace{\gamma_1^\top}_{n} \Big| \underbrace{\gamma_2^\top}_{n} \Big| \cdots \Big| \underbrace{\gamma_n^\top}_{n}\right]^\top \in \mathbb{R}^{n^2}, R = [\gamma_1, \gamma_2, \ldots, \gamma_n]$, and

$$\delta = \left[\underbrace{\delta_1^\top}_{1} \Big| \underbrace{\delta_2^\top}_{2} \Big| \cdots \Big| \underbrace{\delta_n^\top}_{n}\right]^\top = \left[\delta_{11} \big| \delta_{21}, \delta_{22} \big| \cdots \big| \delta_{n1}, \delta_{n2}, \ldots, \delta_{nn}\right]^\top$$
$$\in \mathbb{R}^{n(n+1)/2 \equiv q}.$$

The derivatives of L satisfy

$$\begin{aligned} f_1 &\equiv \frac{\partial L}{\partial \gamma} = (I \otimes A - D^\top \otimes I - N^\top \otimes I)v(X) + (I \otimes B)v(Y) = 0, \\ f_2 &\equiv \frac{\partial L}{\partial \delta} = d_0(X)^\top v(X) - \mathrm{Stk}(I) = 0, \\ f_3 &\equiv \frac{\partial L}{\partial v(X)} = (I \otimes A^\top - D \otimes I - N \otimes I)\gamma + v(X\Delta) = 0, \end{aligned} \tag{16.9}$$

$$f_4 \equiv \frac{\partial L}{\partial v(Y)} = (I \otimes B^\top)\gamma + 2\omega_1^2 v(Y) = 0, \tag{16.10}$$

$$f_5 \equiv \frac{\partial L}{\partial \mathrm{Stk}(N)} = 2\omega_2^2\, \mathrm{Stk}(N) - d_1(X)^\top \gamma = 0 \tag{16.11}$$

where $\Delta_{ij} = \delta_{ij}\,(i \neq j), \Delta_{ii} = 2\delta_{ii}$ and

$$d_1(C) \equiv \begin{bmatrix} 0 \cdots & \cdots & \cdots & \cdots 0 \\ c_1 \oplus [c_1, c_2] \oplus \cdots \oplus [c_1, , c_{n-1}] \end{bmatrix} \in \mathbb{R}^{kn \times p}.$$

We apply Newton's method to $f \equiv (f_1^\top, \ldots, f_5^\top)^\top = 0$:

$$\left[\gamma^\top, \delta^\top, v(X)^\top, v(Y)^\top, \mathrm{Stk}(N)^\top\right]^\top_{\mathtt{new}} = \left[\gamma^\top, \delta^\top, v(X)^\top, v(Y)^\top, \mathrm{Stk}(N)^\top\right]^\top - J_f^{-1} f$$
with the symmetric (16.12)

$$J_f = \begin{bmatrix} 0 & 0 & I \otimes A - D^\top \otimes I - N^\top \otimes I & I \otimes B & -d_1(X) \\ & 0 & d_0(X)^\top + d_2(X^\top) & 0 & 0 \\ & & \Delta \otimes I & 0 & -d_3([\gamma_2, \ldots, \gamma_n]) \\ & & & 2\omega_1^2 I_{mn} & 0 \\ & & * & & 2\omega_2^2 I_p \end{bmatrix},$$

$$d_2(C^\top) \equiv \begin{bmatrix} c_1^\top \\ c_2^\top \oplus c_2^\top \\ c_3^\top \oplus c_3^\top \oplus c_3^\top \\ \vdots \\ c_n^\top \oplus \cdots \oplus c_n^\top \end{bmatrix} \in \mathbb{R}^{q \times kn}, \quad \text{and}$$

$$d_3(C) \equiv {}_{k\{}\begin{bmatrix} c_2 c_3 \oplus c_3 c_4 \oplus c_4 \oplus c_4 \cdots c_n \oplus \cdots \oplus c_n \\ 0 \quad \cdots \qquad \cdots \qquad \cdots \qquad 0 \end{bmatrix} \in \mathbb{R}^{kn \times p}.$$

We can now present the algorithm for the RPAP with real eigenvalues:

Algorithm 1

1. Use the Schur Form to find the initial X_0, Y_0 and N_0.
2. Substitute X_0, Y_0, N_0 into (16.9), (16.10) and (16.11), we construct:

$$\begin{bmatrix} I \otimes A^\top - D \otimes I - N_0 \otimes I & d_0(X) + d_2(X^\top)^\top \\ I \otimes B^\top & 0 \\ d_1(X)^\top & 0 \end{bmatrix} \begin{bmatrix} \gamma_0 \\ \delta_0 \end{bmatrix} = \begin{bmatrix} 0 \\ -2\omega_1^2 v(Y_0) \\ 2\omega_2^2 \mathrm{Stk}(N_0) \end{bmatrix} \tag{16.13}$$

   Solve the over-determined system (16.13) in the least squares sense for $(\gamma_0^\top, \delta_0^\top)$.
3. Use $[\gamma_0^\top, \delta_0^\top, v(X_0)^\top, v(Y_0)^\top, \mathrm{Stk}(N_0)^\top]^\top$ as the initial point and run Newton's iteration as in (16.12) until convergence to X, Y, N.
4. Substitute the X, N into (16.6) to obtain the feedback matrix F.

16.3.2 Complex Eigenvalues

When some of the closed-loop eigenvalues are complex, we assume that the given eigenvalues being $\mathcal{L} = \{\lambda_1, \ldots, \lambda_{n-2s}; \alpha_1 \pm \beta_1 i, \ldots, \alpha_s \pm \beta_s i\}$, where s is the number of complex eigenvalues, and λ_i, α_j and β_j are real.

As in the real eigenvalue case, using a modified real Schur form, the real vectors $x_j, x_{j+1} \in \mathbb{R}^n, y_j, y_{j+1} \in \mathbb{R}^m$ and $\eta_j, \eta_{j+1} \in \mathbb{R}^{j-1}$ are chosen via

$$A[x_j, x_{j+1}] + B[y_j, y_{j+1}] - [x_j, x_{j+1}]D_j - X_{-j}[\eta_j, \eta_{j+1}] = 0,$$

with

$$D_j = \Phi(\alpha_j, \beta_j) \equiv \begin{bmatrix} \alpha_j & \beta_j \\ -\beta_j & \alpha_j \end{bmatrix}. \tag{16.14}$$

Consequently, we have $M_j[x_j^\top, x_{j+1}^\top, y_j^\top, y_{j+1}^\top, \eta_j^\top, \eta_{j+1}^\top]^\top = 0$ where

$$M_j \equiv \left[\begin{array}{c|c|c} I_2 \otimes A - D_j^\top \otimes I & I_2 \otimes B & -I_2 \otimes X_{-j} \\ I_2 \otimes X_{-j}^\top & 0 & 0 \end{array}\right]. \tag{16.15}$$

With QR to extract the unitary basis of the null space, we have

$$\left[x_j^\top, x_{j+1}^\top, y_j^\top, y_{j+1}^\top, \eta_j^\top, \eta_{j+1}^\top\right]^\top = \left[S_{1j}^\top, S_{2j}^\top, S_{3j}^\top\right]^\top u_j.$$

We are then required to choose u_j such that

$$\min_{u_j} = \frac{u_j^\top \hat{S}_{2j}^\top \hat{S}_{2j} u_j}{u_j^\top S_{1j}^\top S_{1j} u_j}, \quad \hat{S}_{2j} = \begin{bmatrix} \omega_1 I & 0 \\ 0 & \omega_2 I \end{bmatrix} \begin{bmatrix} S_{2j} \\ S_{3j} \end{bmatrix}$$

where ω_k are some weights in the objective function, to vary the emphasis on the condition of the the feedback gain (or the size of F) and closed-loop eigenvalues (or the departure from normality). The minimization can then be achieved by the GSVD of $\mathcal{S} \equiv (S_{1j}, \hat{S}_{2j})^\top$, by choosing u_j to be the generalized singular vector of $\mathcal{S}$ corresponding to the smallest generalized singular value.

In the Real Schur Decomposition in (16.22) with $a_j, b_j \in \mathbb{R}, a_{2j-1} + a_{2j} = 2\alpha_j, a_{2j-1}a_{2j} - b_{2j-1}b_{2j} = \alpha_j^2 + \beta_j^2$ for $j = 1, \ldots, s$; we have

$$D = \left[\lambda_1 \oplus \cdots \oplus \lambda_{n-2s} \oplus \begin{bmatrix} a_1 & b_2 \\ b_1 & a_2 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} a_{2s-1} & b_{2s} \\ b_{2s-1} & a_{2s} \end{bmatrix} \right]$$

and

$$N = \begin{array}{c} \begin{array}{ccccccccccc} \eta_2 & \eta_3 & \cdots & \eta_{n-2s} & \eta_{n-2s+1} & \eta_{n-2s+2} & \cdots & \eta_{n-1} & \eta_n \\ \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \end{array} \\ \begin{bmatrix} 0 & \eta_{1,2} & \eta_{1,3} & \cdots & \eta_{1,n-2s} & \eta_{1,n-2s+1} & \eta_{1,n-2s+2} & \cdots & \eta_{1,n-1} & \eta_{1,n} \\ & 0 & \eta_{2,3} & & \eta_{2,n-2s} & \eta_{2,n-2s+1} & \eta_{2,n-2s+2} & \cdots & \eta_{2,n-1} & \eta_{2,n} \\ & & 0 & \cdots & \vdots & \vdots & \vdots & & & \vdots \\ & & & & \eta_{n-2s-1,n-2s} & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & \eta_{n-2s,n-2s+1} & \eta_{n-2s,n-2s+2} & \vdots & \vdots & \vdots \\ & & & & & 0 & 0 & \vdots & \vdots & \vdots \\ & & & & & 0 & 0 & \vdots & \vdots & \vdots \\ & & & & & & & \eta_{n-2,n-1} & \eta_{n-2,n} \\ & & & & & & & 0 & 0 \\ & & & & & & & 0 & 0 \end{bmatrix} \end{array} \tag{16.16}$$

From the initial point, which is obtained from a modified real Schur form and is not usually feasible, we use Newton's algorithm to optimize the problem. We arrive at the optimization problem for complex eigenvalues:

Optimization Problem 2

$$\min \omega_1^2 \|Y\|_F^2 + \omega_2^2 \|N\|_F^2 \quad \text{s.t.} \quad \begin{cases} AX + BY - X(D+N) = 0 \\ X^\top X - I = 0 \end{cases}$$

where D and N are as previously defined, and X is $n \times n$ orthogonal. Optimization Problem 2 is equivalent to:

$$\min \omega_1^2 v(Y)^\top v(Y) + \omega_2^2 \mathrm{Stk}(N)^\top \mathrm{Stk}(N)$$

$$\mathbf{s.t.} \begin{cases} (I \otimes A - D^\top \otimes I - N^\top \otimes I)v(X) + (I \otimes B)v(Y) = 0 \\ d_0(X)^\top v(X) - \mathrm{Stk}(I) = 0 \\ \begin{bmatrix} a_1 + a_2 - 2\alpha_1 = 0 \\ \vdots \\ a_{2s-1} + a_{2s} - 2\alpha_s = 0 \end{bmatrix} \\ \begin{bmatrix} a_1 a_2 - b_1 b_2 - (\alpha_1^2 + \beta_1^2) = 0 \\ \vdots \\ a_{2s-1} a_{2s} - b_{2s-1} b_{2s} - (\alpha_s^2 + \beta_s^2) = 0 \end{bmatrix} \end{cases}$$

for which the Lagrangian function equals

$$\begin{aligned} L(\gamma, \delta, \omega, \xi, v(X), v(Y), a, b, \mathrm{Stk}(N)) &= \omega_1^2 v(Y)^\top v(Y) + \omega_2^2 \mathrm{Stk}(N)^\top \mathrm{Stk}(N) \\ &+ \gamma^\top \left[(I \otimes A - D^\top \otimes I - N^\top \otimes I)v(X) + (I \otimes B)v(Y) \right] + \delta^\top \left[d_0(X)^\top v(X) - \mathrm{Stk}(I) \right] \\ &+ \sum_{j=1}^{s} \omega_j (a_{2j-1} + a_{2j} - 2\alpha_j) + \sum_{j=1}^{s} \xi_j \left[a_{2j-1} a_{2j} - b_{2j-1} b_{2j} - \left(\alpha_j^2 + \beta_j^2 \right) \right] \end{aligned}$$

where $\omega = [\omega_1, \omega_2, \ldots, \omega_s]^\top, \xi = [\xi_1, \xi_2, \ldots, \xi_s]^\top, a = [a_1, a_2, \ldots, a_{2s}]^\top, b = [b_1, b_2, \ldots, b_{2s}]^\top$ and $\mathrm{Stk}(N) = \left[\eta_2^\top, \ldots, \eta_n^\top\right]^\top$.

The derivatives of L satisfy

$$f_1 \equiv \frac{\partial L}{\partial \gamma} = (I \otimes A - D^\top \otimes I - N^\top \otimes I)v(X) + (I \otimes B)v(Y) = 0,$$

$$f_2 \equiv \frac{\partial L}{\partial \delta} = d_0(X)^\top v(X) - \mathrm{Stk}(I) = 0, \quad f_3 \equiv \frac{\partial L}{\partial \omega} = \begin{bmatrix} a_1 + a_2 - 2\alpha_1 \\ \vdots \\ a_{2s-1} + a_{2s} - 2\alpha_s \end{bmatrix},$$

$$f_4 \equiv \frac{\partial L}{\partial \xi} = \begin{bmatrix} a_1 a_2 - b_1 b_2 - (\alpha_1^2 + \beta_1^2) = 0 \\ \vdots \\ a_{2s-1} a_{2s} - b_{2s-1} b_{2s} - (\alpha_s^2 + \beta_s^2) = 0 \end{bmatrix},$$

$$f_5 \equiv \frac{\partial L}{\partial v(X)} = (I \otimes A^\top - D \otimes I - N \otimes I)\gamma + v(X\Delta) = 0, \tag{16.17}$$

$$f_6 \equiv \frac{\partial L}{\partial v(Y)} = (I \otimes B^\top)\gamma + 2\omega_1^2 v(Y) = 0, \tag{16.18}$$

$$f_7 \equiv \frac{\partial L}{\partial a} = \begin{bmatrix} \omega_1 \\ \omega_1 \\ \vdots \\ \omega_s \\ \omega_s \end{bmatrix} + \begin{bmatrix} \xi_1 a_2 \\ \xi_1 a_1 \\ \vdots \\ \xi_s a_{2s} \\ \xi_s a_{2s-1} \end{bmatrix} - \begin{bmatrix} \gamma_{n-2s+1}^\top & & \\ & \ddots & \\ & & \gamma_n^\top \end{bmatrix} v([x_{n-2s+1}, \ldots, x_n]) = 0, \tag{16.19}$$

$$f_8 \equiv \frac{\partial L}{\partial b} = -\begin{bmatrix} \xi_1 b_2 \\ \xi_1 b_1 \\ \vdots \\ \xi_s b_{2s} \\ \xi_s b_{2s-1} \end{bmatrix} - \begin{bmatrix} \gamma_{n-2s+1}^\top & & \\ & \ddots & \\ & & \gamma_n^\top \end{bmatrix} \Pi_s \, v([x_{n-2s+1}, \ldots, x_n]) = 0, \tag{16.20}$$

$$f_9 \equiv \frac{\partial L}{\partial \mathrm{Stk}(N)} = 2\omega_2^2 \, \mathrm{Stk}(N) - \widehat{d}_1(X,s)^\top \gamma = 0, \tag{16.21}$$

where

$$\Pi_s = \underbrace{\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}}_{s}, \quad \widehat{d}_1(C,s) \equiv \begin{bmatrix} 0 \\ \mathcal{C}_r \oplus \mathcal{C}_i \end{bmatrix}^{\}k}$$

with $\mathcal{C}_r \equiv c_1 \oplus \cdots \oplus [c_1, \ldots, c_{n-2s-1}]$ and

$$\mathcal{C}_i \equiv \begin{bmatrix} c_1, \ldots, c_{n-2s} & 0 \\ 0 & c_1, \ldots, c_{n-2s} \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} c_1, \ldots, c_{n-2} & 0 \\ 0 & c_1, \ldots, c_{n-2} \end{bmatrix}.$$

We then obtain the symmetric gradient matrix of $f \equiv [f_1^\top, f_2^\top, \ldots, f_9^\top]^\top$:

$$J_f = \begin{bmatrix} 0000 & \Omega & \Psi & -\widehat{d}_4(X,s) & -d_4(X,s) & -\widehat{d}_1(X,s) \\ 000 & \Xi & 0 & 0 & 0 & 0 \\ 00 & 0 & 0 & d_5(e) & 0 & 0 \\ 0 & 0 & 0 & d_5(a) & -d_5(b) & 0 \\ & \Delta \otimes I & 0 & -\widehat{d}_4(R,s) & -d_6(R,s) & -d_8(R,s)^\top \\ & & 2\omega_1^2 I & 0 & 0 & 0 \\ & & & d_7(\xi) & 0 & 0 \\ & * & & & -d_7(\xi) & 0 \\ & & & & & 2\,\omega_2^2 I_{\widehat{p}} \end{bmatrix}$$

where $\Omega = I \otimes A - D^\top \otimes I - N^\top \otimes I, \Psi = I \otimes B, \widehat{p} = (n-2s)(n-2s-1)/2 + 2s(n-s-1), e = [\underbrace{1, 1, \ldots, 1}_{2s}]^\top, \Xi = d_0(X)^\top + d_2(X^\top)$,

$$d_4(C,s) \equiv \begin{bmatrix} 0 & & \cdots & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & & \cdots & \cdots & 0 \\ \begin{bmatrix} c_{n-2s+2} & \\ & c_{n-2s+1} \end{bmatrix} & \oplus \cdots \oplus & & & \begin{bmatrix} c_n & \\ & c_{n-1} \end{bmatrix} \end{bmatrix} \begin{matrix} \} k(n-2s) \\ \\ \} k(2s) \end{matrix},$$

$$\widehat{d}_4(C,s) \equiv \begin{bmatrix} 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ c_{n-2s+1} & \oplus & c_{n-2s+2} & \oplus \cdots \oplus & c_{n-1} & \oplus & c_n \end{bmatrix} \begin{matrix} \} k(n-2s) \\ \\ \} k(2s) \end{matrix},$$

$$d_5(\psi) \equiv d_5([\psi_1, \ldots, \psi_{2s}]^\top)) = [\psi_2, \psi_1] \oplus [\psi_4, \psi_3] \oplus \cdots \oplus [\psi_{2s}, \psi_{2s-1}],$$

$$d_6(C,s) \equiv \begin{bmatrix} 0 & & \cdots & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & & \cdots & \cdots & 0 \\ \begin{bmatrix} 0 & c_{n-2s+2} \\ c_{n-2s+1} & 0 \end{bmatrix} & \oplus \cdots \oplus & & & \begin{bmatrix} 0 & c_n \\ c_{n-1} & 0 \end{bmatrix} \end{bmatrix} \begin{matrix} \} k(k-2s) \\ \\ \} k(2s) \end{matrix},$$

$$d_7(\psi) \equiv d_7([\psi_1, \ldots, \psi_{2s}]^\top)) = \begin{bmatrix} 0 & \psi_1 \\ \psi_1 & 0 \end{bmatrix} \oplus \begin{bmatrix} 0 & \psi_2 \\ \psi_2 & 0 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} 0 & \psi_s \\ \psi_s & 0 \end{bmatrix},$$

and

$$d_8(C,s) \equiv \begin{bmatrix} (c_2)^\top & 0 \\ (c_3)^\top \oplus (c_3)^\top & 0 \\ (c_4)^\top \oplus (c_4)^\top \oplus (c_4)^\top & 0 \\ \vdots & \\ (c_{n-2s})^\top \oplus \ldots \oplus (c_{n-2s})^\top & 0 \\ (c_{n-2s+1})^\top \oplus \ldots \oplus (c_{n-2s+1})^\top & 0 \\ (c_{n-2s+2})^\top \oplus \ldots \oplus (c_{n-2s+2})^\top & 0 \\ \vdots & \\ (c_{n-1})^\top \oplus \ldots \oplus (c_{n-1})^\top & 0 \\ (c_n)^\top \oplus \ldots \oplus (c_n)^\top & 0 \end{bmatrix} \begin{matrix} \} n-2s-1 \\ \} n-2s \\ \} n-2s \\ \} n-2 \\ \} n-2 \end{matrix}$$

Modifying Algorithm 1, we solve Optimization Problem 2 by Newton's iteration to obtain a real feedback matrix F:

Algorithm 2

1. Use SCHUR to find the initial X_0 and N_0.
2. Substitute X_0, N_0 into (16.17–16.20), we obtain $[\gamma_0^\top, \delta_0^\top, \omega_0^\top, \xi_0^\top]^\top$.

3. Use $[\gamma_0^\top, \delta_0^\top, \omega_0^\top, \xi_0^\top, v(X_0)^\top, v(Y_0)^\top, a_0^\top, b_0^\top, \mathrm{Stk}(N_0)^\top]^\top$ as the initial guess, apply Newton's iteration (16.12) until convergence to X, Y, N.
4. Substitute the X, N and Λ into (16.6) to obtain the feedback matrix F.

Remarks At step 3, the initial point $[\gamma_0^\top, \delta_0^\top, \omega_0^\top, \xi_0^\top, v(X_0)^\top, v(Y_0)^\top, a_0^\top, b_0^\top, \mathrm{Stk}(N_0)^\top]^\top$ is sometimes far away from the optimal point. In such an event, we apply the GBB Gradient method [10] to decrease the objective function sufficiently, before Newton's iteration is applied. At step 4, since the matrix X is orthogonal, we can use $X^\top$ in place of X^{-1}.

16.4 Descriptor System

Multiplying the nonsingular matrix Z^{-1} and orthogonal matrix X on the both sides of $A + BF$ and $E + BG$, we get

$$(A + BF)X = Z(D_\alpha + N_\alpha), \quad (E + BG)X = Z(D_\beta + N_\beta), \tag{16.22}$$

where D_α, D_β are diagonal, and N_α, N_β are straightly upper triangular.

From the QR decomposition in (16.5), we have $Q_2^\top B = 0$ and $B^\dagger = R_B^{-1} Q_1^\top$. Pre-multiplying the equations in (16.22), respectively, by $Q_2^\top$ and $B^\dagger$, we obtain

$$Q_2^\top AX - Q_2^\top ZD_\alpha - Q_2^\top ZN_\alpha = 0, \quad Q_2^\top EX - Q_2^\top ZD_\beta - Q_2^\top ZN_\beta = 0; \tag{16.23}$$

$$F = R_B^{-1} Q_1^\top [Z(D_\alpha + N_\alpha)X^{-1} - A], \quad G = R_B^{-1} Q_1^\top [Z(D_\beta + N_\beta)X^{-1} - E]. \tag{16.24}$$

For a given eigenvalue pairs $\{D_\alpha, D_\beta\}$, we can select Z, X from (16.23) then obtain the solution to the pole assignment problem using (16.24).

Let $Y \equiv \left[Y_1^\top, Y_2^\top\right]^\top \equiv [F^\top, G^\top]^\top X$, when X and Y are chosen, the feedback matrices can be obtained through $H \equiv [F^\top, G^\top]^\top = YX^{-1}$. Furthermore, minimizing the norm of Y is equivalent to minimizing the feedback gain $\|H\|$.

16.4.1 Real Eigenvalues

Let us first consider the case when all the closed-loop eigenvalues are real, with the closed-loop system matrix pencil $(A_c, E_c) = (A + BF, E + BG) = (Z\Lambda_\alpha X^\top, Z\Lambda_\beta X^\top)$ in Schur form. Here we have $(\Lambda_\alpha, \Lambda_\beta) = (D_\alpha + N_\alpha, D_\beta + N_\beta)$, with $D_\alpha = \mathrm{diag}\{\alpha_1, \ldots, \alpha_n\}, D_\beta = \mathrm{diag}\{\beta_1, \ldots, \beta_n\}$ being real, $N_\alpha = [\hat{\eta}_1, \hat{\eta}_2, \ldots, \hat{\eta}_n]$, $N_\beta = \left[\hat{\zeta}_1, \hat{\zeta}_2, \ldots, \hat{\zeta}_n\right]$ being straightly upper triangular and nilpotent, and $\eta_j = [\eta_{1,j}, \ldots, \eta_{j-1,j}]^\top$, $\zeta_j = [\zeta_{1,j}, \ldots, \zeta_{j-1,j}]^\top$ are the vectors constructed from $\hat{\eta}_j$ and $\hat{\zeta}_j$ with

the zeroes at the bottom deleted (thus η_1, ζ_1 are degenerate and $\eta_j, \zeta_j \in \mathbb{R}^{j-1}$). The Schur vector matrix X is orthogonal.

Similar to Sect. 16.3.1, we seek $X,\ Y_1,\ Y_2,\ Z,\ N_\alpha$ and N_β such that

$$AX + BY_1 - Z(D_\alpha + N_\alpha) = 0, EX + BY_2 - Z(D_\beta + N_\beta) = 0, \quad X^\top X = I.$$

Considering the jth column, we have

$$M_j \left[x_j^\top | y_{1j}^\top, y_{2j}^\top, \eta_j^\top, \zeta_j^\top | z_j^\top\right]^\top = 0 \tag{16.25}$$

with

$$M_j \equiv \left[\begin{array}{c|cccc|c} A & B & 0 & -Z_{-j} & 0 & -\alpha_j I \\ E & 0 & B & 0 & -Z_{-j} & -\beta_j I \\ X_{-j}^\top & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & Z_{-j}^\top \end{array}\right]$$

With QR to extract the unitary basis of the null space, we have

$$\left[x_j^\top | y_{1j}^\top, y_{2j}^\top, \eta_j^\top, \zeta_j^\top | z_j^\top\right]^\top = \left[S_{1j}^\top, S_{2j}^\top, S_{3j}^\top\right]^\top u_j$$

We are then required to choose u_j such that

$$\min_{u_j} = \frac{u_j^\top \hat{S}_{2j}^\top \hat{S}_{2j} u_j}{u_j^\top S_{1j}^\top S_{1j} u_j}, \quad \hat{S}_{2j} = \begin{bmatrix} \omega_1 I & 0 \\ 0 & \omega_2 I \end{bmatrix} S_{2j}$$

where ω_k are some weights in the objective function. The minimization can then be achieved by the GSVD of $\mathcal{S} \equiv (S_{1j}, \hat{S}_{2j})^\top$, by choosing u_j to be the generalized singular vector of $\mathcal{S}$ corresponding to the smallest generalized singular value.

Remarks In Definition 2.1, Z is only required to be nonsingular, but this will be inconvenient to achieve in practice. If it is unconstrained, an ill-conditioned Z may cause problems in the Schur–Newton refinement in the next section. Consequently, we require in the calculations in (16.25) and (16.15) that Z is unitary with perfect condition.

From the initial point, we use Newton's algorithm to optimize the problem.

Optimization Problem 3

$$\min \omega_1^2 \|Y\|_F^2 + \omega_2^2 \|[N_\alpha, N_\beta]\|_F^2 \quad \textbf{s.t.} \quad \begin{cases} AX + BY_1 - Z(D_\alpha + N_\alpha) = 0 \\ EX + BY_2 - Z(D_\beta + N_\beta) = 0 \\ X^\top X - I = 0 \end{cases}$$

where N_α, N_β are $n \times n$ strictly upper triangular, X is $n \times n$ orthogonal and Z is nonsingular.

Optimization Problem 3 is equivalent to:

$$\min \omega_1^2 v(F)^\top v(F) + \omega_2^2 \Big[\mathrm{Stk}(N_\alpha)^\top \mathrm{Stk}(N_\alpha) + \mathrm{Stk}(N_\beta)^\top \mathrm{Stk}(N_\beta)\Big]$$
$$s.t. \begin{cases} (I \otimes A)v(X) + (I \otimes B)v(Y_1) - (D_\alpha^\top \otimes I)v(Z) - (N_\alpha^\top \otimes I)v(Z) = 0 \\ (I \otimes E)v(X) + (I \otimes B)v(Y_2) - (D_\beta^\top \otimes I)v(Z) - (N_\beta^\top \otimes I)v(Z) = 0 \\ d_0(X)^\top v(X) - \mathrm{Stk}(I) = 0 \end{cases}$$

where $\mathrm{Stk}(N_\alpha) \equiv \Big[\eta_{12}^\top | \eta_{13}^\top, \eta_{23}^\top | \cdots | \eta_{1n}^\top, \ldots, \eta_{n-1,n}^\top\Big]^\top \in \mathbb{R}^{n(n-1)/2 \equiv p}$, $\mathrm{Stk}(N_\beta) \equiv \Big[\zeta_{12}^\top | \zeta_{13}^\top, \zeta_{23}^\top | \cdots | \zeta_{1n}^\top, \ldots, \zeta_{n-1,n}^\top\Big]^\top \in \mathbb{R}^{n(n-1)/2 \equiv p}$.

We then consider the Lagrangian function of Optimization Problem 3

$$\begin{aligned} &L\big(\gamma, \varepsilon, \delta, v(X), v(Y_1), v(Y_2), v(Z), \mathrm{Stk}(N_\alpha), \mathrm{Stk}(N_\beta)\big) \\ &\quad = \omega_1^2 v(Y)^\top v(Y) + \omega_2^2 \Big[\mathrm{Stk}(N_\alpha)^\top \mathrm{Stk}(N_\alpha) + \mathrm{Stk}(N_\beta)^\top \mathrm{Stk}(N_\beta)\Big] \\ &\qquad + \gamma^\top \big[(I \otimes A)v(X) + (I \otimes B)v(Y_1) - (D_\alpha^\top \otimes I)v(Z) - (N_\alpha^\top \otimes I)v(Z)\big] \\ &\qquad + \varepsilon^\top \Big[(I \otimes E)v(X) + (I \otimes B)v(Y_2) - (D_\beta^\top \otimes I)v(Z) - (N_\beta^\top \otimes I)v(Z)\Big] \\ &\qquad + \delta^\top \Big[d_0(X)^\top v(X) - \mathrm{Stk}(I)\Big] \end{aligned}$$

where

$$\varepsilon = [\underbrace{\varepsilon_1^\top}_{n} | \underbrace{\varepsilon_2^\top}_{n} | \cdots | \underbrace{\varepsilon_n^\top}_{n}]^\top \in \mathbb{R}^{n^2}, \quad W = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n .$$

The derivatives of L satisfy

$$f_1 \equiv \frac{\partial L}{\partial \gamma} = (I \otimes A)v(X) + (I \otimes B)v(Y_1) - (D_\alpha^\top \otimes I)v(Z) - (N_\alpha^\top \otimes I)v(Z) = 0,$$

$$f_2 \equiv \frac{\partial L}{\partial \varepsilon} = (I \otimes E)v(X) + (I \otimes B)v(Y_2) - (D_\beta^\top \otimes I)v(Z) - (N_\beta^\top \otimes I)v(Z) = 0,$$

$$f_3 \equiv \frac{\partial L}{\partial \delta} = d_0(X)^\top v(X) - \mathrm{Stk}(I) = 0,$$

$$f_4 \equiv \frac{\partial L}{\partial v(X)} = (I \otimes A^\top)\gamma + (I \otimes E^\top)\varepsilon + v(X\Delta) = 0, \tag{16.26}$$

$$f_5 \equiv \frac{\partial L}{\partial v(Y_1)} = (I \otimes B^\top)\gamma + 2\omega_1^2 v(Y_1) = 0, \tag{16.27}$$

$$f_6 \equiv \frac{\partial L}{\partial v(Y_2)} = (I \otimes B^\top)\varepsilon + 2\omega_1^2 v(Y_2) = 0, \tag{16.28}$$

$$f_7 \equiv \frac{\partial L}{\partial v(Z)} = -[(D_\alpha \otimes I) + (N_\alpha \otimes I)]\gamma - [(D_\beta \otimes I) + (N_\beta \otimes I)]\varepsilon = 0, \quad (16.29)$$

$$f_8 \equiv \frac{\partial L}{\partial \mathrm{Stk}(N_\alpha)} = 2\omega_2^2\,\mathrm{Stk}(N_\alpha) - d_1(Z)^\top \gamma = 0, \quad (16.30)$$

$$f_9 \equiv \frac{\partial L}{\partial \mathrm{Stk}(N_\beta)} = 2\omega_2^2\,\mathrm{Stk}(N_\beta) - d_1(Z)^\top \varepsilon = 0. \quad (16.31)$$

We apply Newton's method to $f \equiv (f_1^\top, f_2^\top, \ldots, f_9^\top)^\top$ with respect to the variables in $\left[\gamma^\top, \varepsilon^\top, \delta^\top, v(X)^\top, v(Y_1)^\top, v(Y_2)^\top, v(Z)^\top, \mathrm{Stk}(N_\alpha)^\top, \mathrm{Stk}(N_\beta)^\top\right]^\top$, where the symmetric

$$J_f = \begin{bmatrix} 0 & 0 & 0 & I \otimes A & I \otimes B & 0 & -D_\alpha^\top \otimes I - N_\alpha^\top \otimes I & -d_1(Z) & 0 \\ 0 & 0 & I & \otimes E & 0 & I \otimes B & -D_\beta^\top \otimes I - N_\beta^\top \otimes I & 0 & -d_1(Z) \\ 0 & d_{02} & & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta \otimes I & & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 2\omega_1^2 I_{mn} & 0 & 0 & 0 & 0 & 0 \\ & & & & 2\omega_1^2 I_{mn} & 0 & 0 & 0 & 0 \\ & & & & & & 0 & -d_{3,\gamma} & -d_{3,\varepsilon} \\ & & & * & & & & 2\omega_2^2 I_p & 0 \\ & & & & & & & & 2\omega_2^2 I_p \end{bmatrix}$$

with $d_{02} \equiv d_0(X)^\top + d_2(X^\top)$, $d_{3,\gamma} \equiv d_3([\gamma_2, \ldots, \gamma_n])$ and $d_{3,\varepsilon} \equiv ([\varepsilon_2, \ldots, \varepsilon_n])$.

Applying Newton's method to Optimization Problem 3, we obtain Z, X then by using (16.24), the feedback matrices F, G. Now, we can write down the Schur–Newton Algorithm for the RPAP_DS with real eigenvalues:

Algorithm 3 (*Real Schur–Newton*)

1. Use `SCHUR` to find the initial $X_0, Z_0, Y_{10}, Y_{20}, N_{\alpha 0}$ and $N_{\beta 0}$.
2. Substitute $X_0, Z_0, Y_{10}, Y_{20}, N_{\alpha 0}$ and $N_{\beta 0}$ into (16.26–16.29) to construct:

$$\begin{bmatrix} I \otimes A^\top & I \otimes E^\top & d_{02}^\top \\ I \otimes B^\top & 0 & 0 \\ 0 & I \otimes B^\top & 0 \\ \Phi_3 & \Phi_4 & 0 \\ d_1(Z)^\top & 0 & 0 \\ 0 & d_1(Z)^\top & 0 \end{bmatrix} \begin{bmatrix} \gamma_0 \\ \varepsilon_0 \\ \delta_0 \end{bmatrix} = \begin{bmatrix} 0 \\ -2\omega_1^2 v(Y_1) \\ -2\omega_1^2 v(Y_2) \\ 0 \\ 2\omega_2^2 \mathrm{Stk}(N_\alpha) \\ 2\omega_2^2 \mathrm{Stk}(N_\beta) \end{bmatrix} \quad (16.32)$$

   with $\Phi_3 \equiv -(D_\alpha \otimes I + N_\alpha \otimes I)$ and $\Phi_4 \equiv -(D_\beta \otimes I + N_\beta \otimes I)$. Solve the over-determined system (16.32) in the least squares for γ_0, ε_0 and δ_0.
3. Choose $\{\gamma_0, \varepsilon_0, \delta_0, v(X_0), v(Z_0), Y_{10}, Y_{20}, \mathrm{Stk}(N_{\alpha 0}), \mathrm{Stk}(N_{\beta 0})\}$ as the starting values, run Newton's iteration for f until convergence to X, Z, N_α and N_β.
4. Substitute the X, Z, Y_1, Y_2, N_α and N_β into (16.24) to obtain F, G.

16.4.2 Complex Eigenvalues

Let $\{(\alpha_1,\beta_1),\ldots,(\alpha_{n-2s},\beta_{n-2s});\ (\mu_1\pm i\nu_1,\kappa_1\pm i\tau_1),\ldots,(\mu_s\pm i\nu_s,\kappa_s\pm i\tau_s)\}$ be the prescribed eigenvalues, where s is the number of complex eigenvalue pairs. We seek feedback matrices $F,G\in\mathbb{R}^{m\times n}$ such that $\sigma(A+BF,E+BG)=\sigma(D_\alpha,D_\beta)$ where $\alpha_j,\beta_j,\mu_l,\nu_l,\kappa_l,\tau_l\in\mathbb{R},\alpha_j^2+\beta_j^2=1=\mu_l^2+\nu_l^2+\kappa_l^2+\tau_l^2$ $(j=1,\ldots,n-2s;\ l=1,\ldots,s)$, $D_\alpha\equiv\operatorname{diag}\{\alpha_1,\ldots,\alpha_{n-2s};\mu_1\pm i\nu_1,\ldots,\mu_s\pm i\nu_s\}$ and $D_\beta\equiv\operatorname{diag}\{\beta_1,\ldots,\beta_{n-2s};\ \kappa_1\pm i\tau_1,\ldots,\kappa_s\pm i\tau_s\}$.

Consequently, we are required to choose X,Y_1,Y_2,Z,N_α and N_β such that

$$AX+BY_1-Z(D_\alpha+N_\alpha)=0,\quad EX+BY_2-Z(D_\beta+N_\beta)=0,\quad X^\top X=I.$$

Using a modified real Schur form, the real vectors $x_j,\ x_{j+1},z_j,z_{j+1},\eta_j,\eta_{j+1},\zeta_j$ and ζ_{j+1} are chosen via

$$\begin{aligned}
&A\left[x_j,x_{j+1}\right]+B\left[Y_{1j},Y_{1j+1}\right]-\left[z_j,z_{j+1}\right]D_{\alpha j}-Z_{-j}\left[\eta_j,\eta_{j+1}\right]=0,\\
&E\left[x_j,x_{j+1}\right]+B\left[Y_{2j},Y_{2j+1}\right]-\left[z_j,z_{j+1}\right]D_{\beta j}-Z_{-j}\left[\zeta_j,\zeta_{j+1}\right]=0,\\
&X_{-j}^T\left[x_j,x_{j+1}\right]=0;
\end{aligned}$$

where $D_{\alpha j}=\Phi(\mu_j,\nu_j)$ and $D_{\beta j}=\Phi(\kappa_j,\tau_j)$ [using $\Phi(\cdot,\cdot)$ defined in (16.14)]. So

$$M_j\left[x_j^\top,x_{j+1}^\top,y_{1j}^\top,y_{1j+1}^\top,y_{2j}^\top,y_{2j+1}^\top,\eta_j^\top,\eta_{j+1}^\top,\zeta_j^\top,\zeta_{j+1}^\top,z_j^\top,z_{j+1}^\top\right]^\top=0,$$

where

$$M_j\equiv\left[\begin{array}{c|cccc|c}
I_2\otimes A & I_2\otimes B & 0 & -I_2\otimes Z_{-j} & 0 & -D_{\alpha j}^\top\otimes I\\
I_2\otimes E & 0 & I_2\otimes B & 0 & -I_2\otimes Z_{-j} & -D_{\beta j}^\top\otimes I\\
I_2\otimes X_{-j}^\top & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & I_2\otimes Z_{-j}^\top
\end{array}\right]$$

With QR to extract the unitary basis of the null space, we have

$$\left[x_j^\top,x_{j+1}^\top,y_{1j}^\top,y_{1j+1}^\top,y_{2j}^\top,y_{2j+1}^\top,\eta_j^\top,\eta_{j+1}^\top,\zeta_j^\top,\zeta_{j+1}^\top,z_j^\top,z_{j+1}^\top\right]^\top=\left[S_{1j}^\top,S_{2j}^\top,S_{3j}^\top\right]^\top u_j$$

We are then required to choose u_j such that

$$\min_{u_j}=\frac{u_j^\top\hat{S}_{2j}^\top\hat{S}_{2j}u_j}{u_j^\top S_{1j}^\top S_{1j}u_j},\quad \hat{S}_{2j}=\begin{bmatrix}\omega_1 I & 0\\ 0 & \omega_2 I\end{bmatrix}S_{2j}$$

where ω_k are some weights in the objective function. The minimization can be achieved by the GSVD of $\mathcal{S}\equiv(S_{1j},\hat{S}_{2j})^\top$, by choosing u_j to be the generalized singular vector of $\mathcal{S}$ corresponding to the smallest generalized singular value.

In the Real Schur Decomposition (16.22), with $a_j, b_j, c_j, d_j \in \mathbb{R}, a_{2j-1} + a_{2j} = 2\mu_j, a_{2j-1}a_{2j} - b_{2j-1}b_{2j} = \mu_j^2 + v_j^2; c_{2j-1} + c_{2j} = 2\kappa_j, c_{2j-1}c_{2j} - d_{2j-1}d_{2j} = \kappa_j^2 + \tau_j^2$ for $j = 1, \ldots, s$; we have

$$D_\alpha = \begin{bmatrix} \alpha_1 \oplus \cdots \oplus \alpha_{n-2s} \oplus \begin{bmatrix} a_1 & b_2 \\ b_1 & a_2 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} a_{2s-1} & b_{2s} \\ b_{2s-1} & a_{2s} \end{bmatrix} \end{bmatrix} \text{ and}$$

$$D_\beta = \begin{bmatrix} \beta_1 \oplus \cdots \oplus \beta_{n-2s} \oplus \begin{bmatrix} c_1 & d_2 \\ d_1 & c_2 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} c_{2s-1} & d_{2s} \\ d_{2s-1} & c_{2s} \end{bmatrix} \end{bmatrix}.$$

The upper triangular N_α is identical to N in (16.16) and N_β shares the same structure with ζ replacing all the η's.

From the initial point, which are calculated from a modified real Schur form and is usually infeasible, we can use Newton's algorithm to optimize the problem. We arrive at the optimization problem for complex eigenvalues:

Optimization Problem 4

$$\min \omega_1^2 \|Y\|_F^2 + \omega_2^2 \|[N_\alpha, N_\beta]\|_F^2 \ \mathbf{s.t.} \begin{cases} AX + BY_1 - Z(D_\alpha + N_\alpha) = 0 \\ EX + BY_2 - Z(D_\beta + N_\beta) = 0 \\ X^\top X - I = 0 \end{cases}$$

where $D_\alpha, D_\beta, N_\alpha$ and N_β are as defined before, X is $n \times n$ orthogonal and Z is $n \times n$ nonsingular.

Optimization Problem 4 is equivalent to:

$$\min \omega_1^2 v(Y)^\top v(Y) + \omega_2^2 [\mathrm{Stk}(N_\alpha)^\top \mathrm{Stk}(N_\alpha) + \mathrm{Stk}(N_\beta)^\top \mathrm{Stk}(N_\beta)]$$

$$\mathbf{s.t.} \begin{cases} (I \otimes A)v(X) + (I \otimes B)v(Y_1) - (D_\alpha^\top \otimes I)v(Z) - (N_\alpha^\top \otimes I)v(Z) = 0 \\ (I \otimes E)v(X) + (I \otimes B)v(Y_2) - (D_\beta^\top \otimes I)v(Z) - (N_\beta^\top \otimes I)v(Z) = 0 \\ d_0(X)^\top v(X) - \mathrm{Stk}(I) = 0 \\ \begin{bmatrix} a_1 + a_2 - 2\mu_1 = 0 \\ \vdots \\ a_{2s-1} + a_{2s} - 2\mu_s = 0 \end{bmatrix}, \begin{bmatrix} c_1 + c_2 - 2\kappa_1 = 0 \\ \vdots \\ c_{2s-1} + c_{2s} - 2\kappa_s = 0 \end{bmatrix} \\ \begin{bmatrix} a_1 a_2 - b_1 b_2 - (\mu_1^2 + v_1^2) = 0 \\ \vdots \\ a_{2s-1} a_{2s} - b_{2s-1} b_{2s} - (\mu_s^2 + v_s^2) = 0 \end{bmatrix}, \begin{bmatrix} c_1 c_2 - d_1 d_2 - (\kappa_1^2 + \tau_1^2) = 0 \\ \vdots \\ c_{2s-1} c_{2s} - d_{2s-1} d_{2s} - (\kappa_s^2 + \tau_s^2) = 0 \end{bmatrix} \end{cases}$$

for which the Lagrangian function equals

$$
\begin{aligned}
&L(\gamma,\varepsilon,\delta,\omega,\theta,\xi,\sigma,v(X),v(Y_1),v(Y_2),v(Z),a,b,c,d,\mathrm{Stk}(N_\alpha),\mathrm{Stk}(N_\beta))\\
&\quad=\omega_1^2 v(Y)^\top v(Y)+\omega_2^2[\mathrm{Stk}(N_\alpha)^\top \mathrm{Stk}(N_\alpha)+\mathrm{Stk}(N_\beta)^\top \mathrm{Stk}(N_\beta)]\\
&\qquad+\gamma^\top[(I\otimes A)v(X)+(I\otimes B)v(Y_1)-(D_\alpha^\top\otimes I)v(Z)-(N_\alpha^\top\otimes I)v(Z)]\\
&\qquad+\varepsilon^\top[(I\otimes E)v(X)+(I\otimes B)v(Y_2)-(D_\beta^\top\otimes I)v(Z)-(N_\beta^\top\otimes I)v(Z)]\\
&\qquad+\delta^\top[d_0(X)^\top v(X)-\mathrm{Stk}(I)]+\sum_{j=1}^{s}\omega_j(a_{2j-1}+a_{2j}-2\mu_j)+\sum_{j=1}^{s}\theta_j(c_{2j-1}+c_{2j}-2\kappa_j)\\
&\qquad+\sum_{j=1}^{s}\xi_j[a_{2j-1}a_{2j}-b_{2j-1}b_{2j}-(\mu_j^2+v_j^2)]+\sum_{j=1}^{s}\sigma_j[c_{2j-1}c_{2j}-d_{2j-1}d_{2j}-(\kappa_j^2+\tau_j^2)]
\end{aligned}
$$

where $\theta=[\theta_1,\theta_2,\ldots,\theta_s]^\top,\sigma=[\sigma_1,\sigma_2,\ldots,\sigma_s]^\top,c=[c_1,c_2,\ldots,c_{2s}]^\top$ and $d=[d_1,d_2,\ldots,d_{2s}]^\top$.

The derivatives of L are

$$f_1\equiv\frac{\partial L}{\partial\gamma}=(I\otimes A)v(X)+(I\otimes B)v(Y_1)-(D_\alpha^\top\otimes I)v(Z)-(N_\alpha^\top\otimes I)v(Z)=0,$$

$$f_2\equiv\frac{\partial L}{\partial\varepsilon}=(I\otimes E)v(X)+(I\otimes B)v(Y_2)-(D_\beta^\top\otimes I)v(Z)-(N_\beta^\top\otimes I)v(Z)=0,$$

$$f_3\equiv\frac{\partial L}{\partial\delta}=d_0(X)^\top v(X)-\mathrm{Stk}(I)=0,$$

$$f_4\equiv\frac{\partial L}{\partial\omega}=\begin{bmatrix}a_1+a_2-2\mu_1\\ \vdots\\ a_{2s-1}+a_{2s}-2\mu_s\end{bmatrix},\quad f_5\equiv\frac{\partial L}{\partial\theta}=\begin{bmatrix}c_1+c_2-2\kappa_1\\ \vdots\\ c_{2s-1}+c_{2s}-2\kappa_s\end{bmatrix},$$

$$f_6\equiv\frac{\partial L}{\partial\xi}=\begin{bmatrix}a_1a_2-b_1b_2-(\mu_1^2+v_1^2)=0\\ \vdots\\ a_{2s-1}a_{2s}-b_{2s-1}b_{2s}-(\mu_s^2+v_s^2)=0\end{bmatrix},$$

$$f_7\equiv\frac{\partial L}{\partial\sigma}=\begin{bmatrix}c_1c_2-d_1d_2-(\kappa_1^2+\tau_1^2)=0\\ \vdots\\ c_{2s-1}c_{2s}-d_{2s-1}d_{2s}-(\kappa_s^2+\tau_s^2)=0\end{bmatrix},$$

$$f_8\equiv\frac{\partial L}{\partial v(X)}=(I\otimes A^\top)\gamma+(I\otimes E^\top)\varepsilon+v(X\Delta)=0,\tag{16.33}$$

$$f_9\equiv\frac{\partial L}{\partial v(Y_1)}=(I\otimes B^\top)\gamma+2\omega_1^2 v(Y_1)=0,\tag{16.34}$$

$$f_{10} \equiv \frac{\partial L}{\partial v(Y_2)} = (I \otimes B^{\top})\varepsilon + 2\omega_1^2 v(Y_2) = 0, \tag{16.35}$$

$$f_{11} \equiv \frac{\partial L}{\partial v(Z)} = -[(D_\alpha \otimes I) + (N_\alpha \otimes I)]\gamma - \left[(D_\beta \otimes I) + (N_\beta \otimes I)\right]\varepsilon = 0, \tag{16.36}$$

$$f_{12} \equiv \frac{\partial L}{\partial a} = \begin{bmatrix} \omega_1 \\ \omega_1 \\ \vdots \\ \omega_s \\ \omega_s \end{bmatrix} + \begin{bmatrix} \xi_1 a_2 \\ \xi_1 a_1 \\ \vdots \\ \xi_s a_{2s} \\ \xi_s a_{2s-1} \end{bmatrix} - \begin{bmatrix} \gamma_{n-2s+1}^{\top} & & \\ & \ddots & \\ & & \gamma_n^{\top} \end{bmatrix} v([z_{n-2s+1}, \ldots, z_n]) = 0, \tag{16.37}$$

$$f_{13} \equiv \frac{\partial L}{\partial b} = -\begin{bmatrix} \xi_1 b_2 \\ \xi_1 b_1 \\ \vdots \\ \xi_s b_{2s} \\ \xi_s b_{2s-1} \end{bmatrix} - \begin{bmatrix} \gamma_{n-2s+1}^{\top} & & \\ & \ddots & \\ & & \gamma_n^{\top} \end{bmatrix} \Pi_s \, v([z_{n-2s+1}, \ldots, z_n]) = 0, \tag{16.38}$$

$$f_{14} \equiv \frac{\partial L}{\partial c} = \begin{bmatrix} \theta_1 \\ \theta_1 \\ \vdots \\ \theta_s \\ \theta_s \end{bmatrix} + \begin{bmatrix} \sigma_1 c_2 \\ \sigma_1 c_1 \\ \vdots \\ \sigma_s c_{2s} \\ \sigma_s c_{2s-1} \end{bmatrix} - \begin{bmatrix} \varepsilon_{n-2s+1}^{\top} & & \\ & \ddots & \\ & & \varepsilon_n^{\top} \end{bmatrix} v([z_{n-2s+1}, \ldots, z_n]) = 0, \tag{16.39}$$

$$f_{15} \equiv \frac{\partial L}{\partial d} = -\begin{bmatrix} \sigma_1 d_2 \\ \sigma_1 d_1 \\ \vdots \\ \sigma_s d_{2s} \\ \sigma_s d_{2s-1} \end{bmatrix} - \begin{bmatrix} \varepsilon_{n-2s+1}^{\top} & & \\ & \ddots & \\ & & \varepsilon_n^{\top} \end{bmatrix} \Pi_s \, v([z_{n-2s+1}, \ldots, z_n]) = 0, \tag{16.40}$$

$$f_{16} \equiv \frac{\partial L}{\partial \mathrm{Stk}(N_\alpha)} = 2\omega_2^2 \, \mathrm{Stk}(N_\alpha) - \widehat{d_1}(Z, s)^{\top}\gamma = 0, \tag{16.41}$$

$$f_{17} \equiv \frac{\partial L}{\partial \mathrm{Stk}(N_\beta)} = 2\omega_2^2 \, \mathrm{Stk}(N_\beta) - \widehat{d_1}(Z, s)^{\top}\varepsilon = 0. \tag{16.42}$$

We can obtain the symmetric gradient matrix $J_f = \begin{bmatrix} 0 & J_2 \\ J_2^\top & J_3 \end{bmatrix}$, where

$$J_2 = \begin{bmatrix} \Omega_1 & \Psi & 0 & \Phi_1 & -\widehat{d}_4(Z,s) & -d_4(Z,s) & 0 & 0 & -\widehat{d}_1(Z,s) & 0 \\ \Omega_2 & 0 & \Psi & \Phi_2 & 0 & 0 & -\widehat{d}_4(Z,s) & -d_4(Z,s) & 0 & -\widehat{d}_1(Z,s) \\ \Xi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_5(e) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_5(e) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_5(a) & -d_5(b) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_5(c) & -d_5(d) & 0 & 0 \end{bmatrix},$$

$$J_3 = \begin{bmatrix} \Delta \otimes I & 0 & 0 \\ 0 & 2\omega_1^2 I & 0 \\ 0 & 0 & 2\omega_1^2 I \end{bmatrix} \oplus J_4,$$

$$J_4 = \begin{bmatrix} 0 & -\widehat{d}_4(R,s) & -d_6(R,s) & -\widehat{d}_4(W,s) & -d_6(W,s) & -d_8(R,s)^\top & -d_8(W,s)^\top \\ & d_7(\xi) & 0 & 0 & 0 & 0 & 0 \\ & & -d_7(\xi) & 0 & 0 & 0 & 0 \\ & & & d_7(\sigma) & 0 & 0 & 0 \\ & & & & -d_7(\sigma) & 0 & 0 \\ & & * & & & 2\omega_2^2 I_{\widehat{p}} & 0 \\ & & & & & & 2\omega_2^2 I_{\widehat{p}} \end{bmatrix},$$

with $\Omega_1 = I \otimes A, \Omega_2 = I \otimes E, \Phi_1 = -D_\alpha^\top \otimes I - N_\alpha^\top \otimes I$ and $\Phi_2 = -D_\beta^\top \otimes I - N_\beta^\top \otimes I$. Similar to Algorithm 3, we solve Optimization Problem 4 by Newton's iteration for real F, G:

Algorithm 4 (*Complex Schur–Newton*)

1. Use SCHUR to find the initial $X_0, Y_{10}, Y_{20}, Z_0, N_{\alpha 0}, N_{\beta 0}$.
2. Substitute $X_0, Y_{10}, Y_{20}, Z_0, N_{\alpha 0}, N_{\beta 0}$ into (16.33–16.42), we obtain $\{\gamma_0, \varepsilon_0, \delta_0, \omega_0, \theta_0, \xi_0, \sigma_0, a_0, b_0, c_0, d_0\}$.
3. With $\{\gamma_0, \varepsilon_0, \delta_0, \omega_0, \theta_0, \xi_0, \sigma_0, a_0, b_0, c_0, d_0, v(X_0), v(Y_{10}), v(Y_{20}), v(Z_0), \mathrm{Stk}(N_{\alpha 0}), \mathrm{Stk}(N_{\beta 0})\}$ as starting values, run Newton's iteration until convergence to X, Y_1, Y_2, Z and N_α, N_β.
4. Substitute X, Y_1, Y_2, Z and N_α, N_β into (16.24) to obtain F, G.

Remarks

- At step 2, we set a, b, c, d being the same as the given eigenvalues, and obtain γ_0 by substituting $Z_0, N_{\alpha 0}$ into (16.41) and ε_0 by substituting $Z_0, N_{\beta 0}$ into (16.42). Then from (16.33)–(16.40) we obtain $\delta_0, \omega_0, \theta_0, \xi_0$ and σ_0.
- The starting point $\{\gamma_0, \varepsilon_0, \delta_0, \omega_0, \theta_0, \xi_0, \sigma_0, a_0, b_0, c_0, d_0, v(X_0), v(Y_{10}), v(Y_{20}), v(Z_0), \mathrm{Stk}(N_{\alpha 0}), \mathrm{Stk}(N_{\beta 0})\}$ is often far away from being optimal. In such an

event, we apply the GBB Gradient method [10] to decrease the objective function sufficiently, before Newton's iteration is applied.
- At step 4, since the matrix X is orthogonal, we can use $X^\top$ in place of X^{-1}.

16.5 Numerical Examples

The examples are quoted from [7], some modified with the addition of a singular E. The first two examples are for ordinary systems, with the second one (Ex 2) containing a complex conjugate pair of closed-loop poles. The last two are for descriptor systems with Ex4 containing a complex conjugate pair of poles. The computations were performed using MATLAB [14] on a PC with accuracy `eps` $\approx 2.2(-16)$ (denoting 2.2×10^{-16}). Various weights $(\omega_1, \omega_2) = (0,1), (1,0)$ and $(1,1)$ have been tried to illustrate the feasibility of the algorithms. To save space, only the solution matrices corresponding to the last sets of weights (ω_1, ω_2), with both weights being distinct and nonzero, are included here. For each set of weights, $\texttt{Obj}_{\texttt{SCHUR}}$ and $\texttt{Obj}_{\texttt{Newton}}$, the values of the objective function from `SCHUR` at the starting point and after Newton refinement respectively, are presented.

Example 1: $n = 4, m = 2, \lambda = \{-2, -3, -4, -1\}; E = I_4$,

$$A = \begin{bmatrix} -65 & 65 & -19.5 & 19.5 \\ 0.1 & -0.1 & 0 & 0 \\ 1 & 0 & -0.5 & -1 \\ 0 & 0 & 0.4 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 65 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.4 \end{bmatrix};$$

$$\begin{aligned} &\omega_1 = 0, \omega_2 = 1, \quad \texttt{Obj}_{\texttt{SCHUR}} = 35.41, \quad \texttt{Obj}_{\texttt{Newton}} = 20.67; \\ &\omega_1 = 1, \omega_2 = 0, \quad \texttt{Obj}_{\texttt{SCHUR}} = 13.02, \quad \texttt{Obj}_{\texttt{Newton}} = 6.049; \\ &\omega_1 = 1, \omega_2 = 1, \quad \texttt{Obj}_{\texttt{SCHUR}} = 73.35, \quad \texttt{Obj}_{\texttt{Newton}} = 32.16. \end{aligned}$$

Example 2: $n = 4, m = 2, \lambda = \{-29.4986, -10.0922, 2.5201 \pm 6.8910i\}; E = I_4$,

$$A = \begin{bmatrix} 5.8765 & 9.3456 & 4.5634 & 9.3520 \\ 6.6526 & 0.5867 & 3.5829 & 0.6534 \\ 0.0000 & 9.6738 & 7.4876 & 4.7654 \\ 0.0000 & 0.0000 & 6.6784 & 2.5678 \end{bmatrix}, \quad B = \begin{bmatrix} 3.9878 & 0.5432 \\ 0 & 2.7650 \\ 0 & 0 \\ 0 & 0 \end{bmatrix};$$

$$\begin{aligned} &\omega_1 = 0, \omega_2 = 1, \quad \texttt{Obj}_{\texttt{SCHUR}} = 13.07, \quad \texttt{Obj}_{\texttt{Newton}} = 49.04; \\ &\omega_1 = 1, \omega_2 = 0, \quad \texttt{Obj}_{\texttt{SCHUR}} = 15.79, \quad \texttt{Obj}_{\texttt{Newton}} = 14.71; \\ &\omega_1 = 1, \omega_2 = 1, \quad \texttt{Obj}_{\texttt{SCHUR}} = 20.65, \quad \texttt{Obj}_{\texttt{Newton}} = 58.77. \end{aligned}$$

Example 3: $n = 3, m = 2, \lambda_\alpha = \{1, 1, 1\}, \lambda_\beta = \{-1, -2, -3\}$;

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 0 & 100 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix};$$

$$\omega_1 = 0, \omega_2 = 1, \quad \texttt{Obj}_{\texttt{SCHUR}} = 186.9, \quad \texttt{Obj}_{\texttt{Newton}} = 6.380;$$
$$\omega_1 = 1, \omega_2 = 0, \quad \texttt{Obj}_{\texttt{SCHUR}} = 1.825, \quad \texttt{Obj}_{\texttt{Newton}} = 0.833;$$
$$\omega_1 = 1, \ \omega_2 = 1, \quad \texttt{Obj}_{\texttt{SCHUR}} = 23.36, \quad \texttt{Obj}_{\texttt{Newton}} = 8.767.$$

Example 4: $n = 5, m = 2, \lambda_\alpha = \{1, 1, 1, 1, 1\}, \lambda_\beta = \{-0.2, -0.5, -1, -1 \pm i\}$;

$$A = \begin{bmatrix} -0.1094 & 0.0628 & 0 & 0 & 0 \\ 1.306 & -2.132 & 0.9807 & 0 & 0 \\ 0 & 1.595 & -3.149 & 1.547 & 0 \\ 0 & 0.0355 & 2.632 & -4.257 & 1.855 \\ 0 & 0.0023 & 0 & 0.1636 & -0.1625 \end{bmatrix},$$

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0.0638 & 0 \\ 0.0838 & -0.1396 \\ 0.1004 & -0.206 \\ 0.0063 & -0.0128 \end{bmatrix};$$

$$\omega_1 = 0, \omega_2 = 1, \quad \texttt{Obj}_{\texttt{SCHUR}} = 4.609, \quad \texttt{Obj}_{\texttt{Newton}} = 4.012;$$
$$\omega_1 = 1, \omega_2 = 0, \quad \texttt{Obj}_{\texttt{SCHUR}} = 8968, \quad \texttt{Obj}_{\texttt{Newton}} = 66.84;$$
$$\omega_1 = 1, \omega_2 = 1, \quad \texttt{Obj}_{\texttt{SCHUR}} = 1253, \quad \texttt{Obj}_{\texttt{Newton}} = 78.46.$$

Comments

1. For Examples 2 and 3 with real eigenvalues, the starting vectors from SCHUR fall within the domain of convergence for the Real SCHUR-NEWTON algorithm. This coincides with our experience with other RPAP and RPAP_DS with real eigenvalues. Newton refinement produces a local minimum which improves the robustness measure substantially.
2. For the RPAP_DS with complex eigenvalues like Example 4, the starting vectors from SCHUR are usually infeasible. Preliminary correction by Newton's iteration can be applied to the constraints in Optimization Problem 4, with gradient J_2. This produces a feasible starting vector for the Complex SCHUR-NEWTON algorithm. However, the improvement in the robustness measure is usually limited, as shown in Example 6. Apart from having an infeasible starting vector far from a local minimum, the main difficulty lies in the choice of finding accurate starting values for the Lagrange multipliers. However, improvements are still possible theoretically and achieved in practice.
3. Similarly for the RPAP with complex eigenvalues like Example 2, the starting vectors from SCHUR are often infeasible. The feasible Newton refined solutions may then have objective function values greater than those for the infeasible starting points from SCHUR.
4. The Newton method has been applied in his paper to optimize the robustness measures under constraints. Other methods, such as the augmented Lagrange method, may also be applied. Alternatives will be investigated elsewhere.

5. The convergence of the Newton refinement is expected to be fast, assuming that the associated Jacobian is well behaved or when the starting vector is near the solution. Otherwise, the convergence will be slower. When the starting vector is poor, the GBB Gradient method has been applied.

16.6 Epilogue

The algorithm SCHUR [7] for state-feedback pole assignment, based on the Schur form, has been improved and extended for descriptor systems, minimizing a weighted sum of the departure from normality and feedback gain. Similar to the original SCHUR algorithm, the method appears to be efficient and numerically robust, controlling the conditioning of the closed-loop eigensystem and the feedback gain. The method can be generalized further for second-order and periodic systems, as well as systems with output feedback.

Acknowledgements We would like to thank Professor Shu-Fang Xu (Beijing University, China) for various interesting discussions and much encouragement.

References

1. Beattie C, Ipesn ICF (2003) Inclusion regions for matrix eigenvalues. Lin Alg Appl 358:281–291
2. Braconnier T, Saad Y (1998) Eigenvalue bounds from the Schur form. Research report. University of Minnesota Supercomputing Institute UMSI 98/21
3. Cho GE, Ipsen ICF (1997) If a matrix has only a single eigenvalue, how sensitive is this eigenvalue? CRSC Technical report, North Carolina State University, Raleigh NC, TR97-20
4. Cho GE, Ipsen ICF (1998) If a matrix has only a single eigenvalue, how sensitive is this eigenvalue? II, CRSC technical report, North Carolina State University, Raleigh NC, TR98-8
5. Chu K-WE (1988) A controllability condensed form and a state feedback pole assignment algorithm for descriptor systems. IEEE Trans Autom Control 33:366–370
6. Chu EK-W (2001) Optimization and pole assignment in control system. Int J Appl Maths Comp Sci 11:1035–1053
7. Chu EK-W (2007) Pole assignment via the Schur form. Syst Control Lett 56:303–314
8. Duan G-R, Patton RJ (1999) Robust pole assignment in descriptor systems via proportional plus partial derivative state feedback. Int J Control 72:1193–1203
9. Golub GH, Van Loan CF (1989) Matrix Computations. 2nd edn. Johns Hopkins University Press, Baltimore, MD
10. Grippo L, Sciandrone M (2002) Nonmonotone globalization techniques for the Barzilai-Borwein gradient method. Comput Optim Applics 23:143–169
11. Henrici P (1962) Bounds for iterates, inverses, spectral variation and the field of values of nonnormal matrcies. Numer Math 4:24–40
12. Kautsky J, Nichols NK, Chu K-WE (1989) Robust pole assignment in singular control systems. Lin Alg Applic 121:9–37
13. Kautsky J, Nichols NK, Van Dooren P (1985) Robust pole assignment via in linear state feedback. Int J Control 41:1129–1155

14. Mathworks (2002) MATLAB User's Guide
15. Stewart GW, Sun J-G (1990) Matrix Perturbation Theory. Academic Press, New York
16. Sun J-G (2001) Perturbation Theory of Matrices (in Chinese), 2nd ed., Science Press
17. White BA (1995) Eigenstructure assignment: a survey. Proc Instn Mech Engrs 209:1–11
18. Varga A (2003) A numerical reliable approach to robust pole assignment for descriptor systems. Future Generation Comp Syst 19:1221–1230

Chapter 17
Synthesis of Fixed Structure Controllers for Discrete Time Systems

Waqar A. Malik, Swaroop Darbha and S. P. Bhattacharyya

Abstract In this paper, we develop a linear programming approach to the synthesis of stabilizing fixed structure controllers for a class of linear time invariant discrete-time systems. The stabilization of this class of systems requires the determination of a real controller parameter vector (or simply, a controller), K, so that a family of real polynomials, affine in the parameters of the controllers, is Schur. An attractive feature of the paper is the systematic approximation of the set of all such stabilizing controllers, K. This approximation is accomplished through the exploitation of the interlacing property of Schur polynomials and a systematic construction of sets of linear inequalities in K. The union of the feasible sets of linear inequalities provides an approximation of the set of all controllers, K, which render $P(z, K)$ Schur. Illustrative examples are provided to show the applicability of the proposed methodology. We also show a related result, namely, that the set of rational proper stabilizing controllers for single-input single-output linear time invariant discrete-time plants will form a bounded set in the controller parameter space *if and only if* the order of the stabilizing cannot be reduced any further. Moreover, if the order of the controller is increased, the set of higher order controllers will necessarily be unbounded.

W. A. Malik (✉) · S. Darbha
Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA
e-mail: waqar_am@tamu.edu

S. Darbha
e-mail: dswaroop@tamu.edu

S. P. Bhattacharyya
Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA
e-mail: bhatt@tamu.edu

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_17,

17.1 Introduction

There is renewed interest in the synthesis of fixed-order stabilization of a linear time invariant dynamical system. Surveys by Syrmos et al. [18], show that this problem has attracted significant attention over the last four decades. Application of fixed-order stabilization problem can be found in the work of Buckley [7], Zhu et al. [19], and Bengtsson and Lindahl [2]. This problem may be simply stated as follows: Given a finite-dimensional LTI dynamical system, is there a stabilizing proper, rational controller of a given order (a causal controller of a given state-space dimension)? The set of all the stabilizing controllers of fixed order is the *basic* set in which all design must be carried out.

Given the widespread use of fixed-order controllers in various applications (see Ref. [10], Chap. 6), it is important to understand whether fixed-order controllers that achieve a specified performance exist and if so, how one can compute the set of all such stabilizing controllers that achieve a specified performance. Unfortunately, the standard optimal design techniques result in controllers of higher order, and provide no control over the order or the structure of the controller. Moreover, the set of all fixed order/structure stabilizing controllers maybe *non-convex* and in general, *disconnected* in the space of controller parameters, see Ref. [1]. This is a major source of difficulty in its computation.

A good survey of the attempts to solve the fixed order control problem and the related static output feedback (SOF) problem is given in Syrmos et al. [18], Blondel et al. [5], and Bernstein [3]. Henrion et al. [11] combine ideas from strict positive realness (SPRness), positive polynomials written as sum of squares (SOS) and LMIs to solve the problem of robust stabilization with fixed order controllers. The LMI approach for synthesizing a static output feedback (SOF) controller is also explored in Ghaoui et al. [9], and Iwasaki and Skelton [12].

Datta et al. [8] used the Hermite–Biehler theorem for obtaining the set of all stabilizing PID controllers for SISO plants. Discrete-time PID controllers have been designed by Keel et al. [13] using Chebyshev representation and the interlacing property of Schur polynomial. They use root counting formulas and carry out search for the separating frequencies by exploiting the structure of the PID control problem. The interlacing property of real and complex Hurwitz polynomials was used by Malik et al. [14, 16] to construct the set of stabilizing fixed order controllers that achieve certain specified criterion.

In this paper, we focus on the problem of determining the set of all real controller parameters, $K = (k_1, k_2, \ldots, k_l)$, which render a real polynomial Schur, where each member of the set is of the form:

$$P(z, K) = P_o(z) + \sum_{i=1}^{l} k_i P_i(z).$$

The paper is organized as follows: In Sect. 17.2, we describe the Chebyshev representation of polynomials and we provide the characterization for a polynomial,

$P(z)$, to be Schur in terms of its Chebyshev representation. Section 17.3, deals with the generation of outer approximation $\mathcal{S}_o$ and inner approximation $\mathcal{S}_i$ of the set of controllers $\mathcal{S}$, of a given structure, that stabilize a given linear time invariant discrete-time system. It is seen that $\mathcal{S}_i \subset \mathcal{S} \subset \mathcal{S}_o$. Illustrative examples provided here show how the inner and outer approximations of the set of fixed structure stabilizing controllers may be constructed. Section 17.4 describes the boundedness property of the set of stabilizing controllers. In Sect. 17.5, we provide concluding remarks.

17.2 Chebyshev Representation and Condition for a Polynomial to be Schur

Let $P(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0$ denote a real polynomial, that is the coefficients, a_i are real numbers. We are interested in determining the root distribution of $P(z)$ with respect to the unit circle. The root distribution of $P(z)$ is necessary in characterizing the set of stabilizing controllers for a discrete-time control system. In such systems, $P(z)$ could denote the characteristic polynomial of the given discrete-time control system. Stability would require that all roots of $P(z)$ lie in the interior of the unit circle, i.e. $P(z)$ must be *Schur*.

17.2.1 Chebyshev Representation of Polynomials

We need to determine the image of the boundary of the unit circle under the action of the real polynomial $P(z)$.

$$\left\{P(z) : z = e^{j\theta}, 0 \leq \theta \leq 2\pi\right\}.$$

As the coefficients, a_i, of the polynomial $P(z)$ are real, $P(e^{j\theta})$ and $P(e^{-j\theta})$ are conjugate complex numbers. Hence, it is sufficient to determine the image of the upper half of the unit circle:

$$\left\{P(z) : z = e^{j\theta}, 0 \leq \theta \leq \pi\right\}.$$

By using, $z^k\big|_{z=e^{j\theta}} = \cos k\theta + j \sin k\theta$, we have

$$P(e^{j\theta}) = (a_n \cos n\theta + \cdots + a_1 \cos\theta + a_0) + j(a_n \sin n\theta + \cdots + a_1 \sin\theta).$$

$\cos k\theta$ and $\sin k\theta / \sin\theta$ can be written as polynomials in $\cos\theta$ using Chebyshev polynomials. Using $u = -\cos\theta$, if $\theta \in [0, \pi]$ then, $u \in [-1, 1]$. Now,

$$e^{j\theta} = \cos\theta + j \sin\theta = -u + j\sqrt{1 - u^2}.$$

Let $\cos k\theta = c_k(u)$ and $\sin k\theta / \sin\theta = s_k(u)$, where $c_k(u)$ and $s_k(u)$ are real polynomials in u and are known as the *Chebyshev polynomials* of the first and second kind, respectively. It is easy to show that,

$$s_k(u) = -\frac{1}{k}\frac{dc_k(u)}{du}, \quad k = 1, 2, \ldots \tag{17.1}$$

and that the Chebyshev polynomials satisfy the recursive relation,

$$c_{k+1}(u) = -uc_k(u) - (1-u^2)s_k(u), \quad k = 1, 2, \ldots \tag{17.2}$$

Using (17.1) and (17.2), we can determine $c_k(u)$ and $s_k(u)$ for all k.

From the above development, we see that

$$P(e^{j\theta})\big|_{\theta=\cos^{-1}(-u)} = R(u) + j\sqrt{1-u^2}T(u) =: P_c(u).$$

We refer to $P_c(u)$ as the Chebyshev representation of $P(z)$. $R(u)$ and $T(u)$ are real polynomials of degree n and $n-1$ respectively, with leading coefficients of opposite sign and equal magnitude. More explicitly,

$$\begin{aligned} R(u) &= a_n c_n(u) + \cdots + a_1 c_1(u) + a_0, \\ T(u) &= a_n s_n(u) + a_{n-1} s_{n-1}(u) + \cdots + a_1 s_1(u). \end{aligned}$$

The complex plane image of $P(z)$ as z traverses the upper half of the unit circle can be obtained by evaluating $P_c(u)$ as u runs from -1 to $+1$.

17.2.2 Root Distribution

Let $\phi_P(\theta) := \arg[P(e^{j\theta})]$ denote the phase of $P(z)$ evaluated at $z = e^{j\theta}$ and let $\Delta_{\theta_1}^{\theta_2}[\phi_P(\theta)]$ denote the net change in phase of $P(e^{j\theta})$ as θ increases from θ_1 to θ_2. Similarly, let $\phi_{P_c}(\theta) := \arg[P_c(u)]$ denote the phase of $P_c(u)$ and $\Delta_{u_1}^{u_2}[\phi_{P_c}(u)]$ denote the net change in phase of $P_c(u)$ as u increases for u_1 to u_2.

Lemma 1 *Let the real polynomial $P(z)$ have i roots in the interior of the unit circle, and no roots on the unit circle. Then*

$$\Delta_0^{\pi}[\phi_P(\theta)] = \pi i = \Delta_{-1}^{+1}[\phi_{P_c}(u)].$$

Proof From geometric considerations it is easily seen that each interior root contributes 2π to $\Delta_0^{2\pi}[\phi_P(\theta)]$ and therefore because of symmetry of roots about the real axis the interior roots contribute $i\pi$ to $\Delta_0^{\pi}[\phi_P(\theta)]$. The second equality follows from the Chebyshev representation described above. □

17.2.3 Characterization of a Schur Polynomial in Terms of Its Chebyshev Representation

Let $P(z)$ be a real polynomial of degree n. This polynomial will be said to be *Schur* if all n roots lie within the unit circle. In this section, we characterize the Schur property of a polynomial in terms of its Chebyshev representation, $P(e^{j\theta}) = \tilde{R}(\theta) + j\tilde{T}(\theta) = R(u) + j\sqrt{1-u^2}T(u)$, where $u = -\cos\theta$.

Theorem 1 *$P(z)$ is Schur if and only if,*

1. *$R(u)$ has n real distinct zeros r_i, $i = 1, 2, \ldots, n$ in $(-1, +1)$,*
2. *$T(u)$ has $n-1$ real distinct zeros t_j, $j = 1, 2, \ldots, n-1$ in $(-1, +1)$,*
3. *the zeros r_i and t_j interlace, i.e*

$$-1 < r_1 < t_1 < r_2 < t_2 < \cdots < t_{n-1} < r_n < +1.$$

Proof Let

$$t_j = -\cos\alpha_j, \quad \alpha_j \in (0, \pi), \quad j = 1, 2, \ldots, n-1$$

or

$$\begin{aligned} &\alpha_j = \cos^{-1}(-t_j), \quad j = 1, 2, \ldots, n-1, \\ &\alpha_0 = 0, \quad \alpha_n = \pi \end{aligned}$$

and let

$$\beta_i = \cos^{-1}(-r_i), \quad i = 1, 2, \ldots, n, \quad \beta_i \in (0, \pi).$$

Then $(\alpha_0, \alpha_1, \ldots, \alpha_n)$ are the $n+1$ zeros of $\tilde{T}(\theta)$ and $(\beta_1, \beta_2, \ldots, \beta_n)$ are the n zeros of $\tilde{R}(\theta)$, the third condition means that α_i and β_j satisfy

$$0 = \alpha_0 < \beta_1 < \alpha_1 < \cdots < \beta_{n-1} < \alpha_n = \pi.$$

This condition means that the plot of $P(e^{j\theta})$ for $\theta \in [0, \pi]$ turns counter-clockwise through exactly $2n$ quadrants. Therefore,

$$\Delta_0^{\pi}[\phi_P(\theta)] = 2n\frac{\pi}{2} = n\pi.$$

and this condition is equivalent to $P(z)$ having n zeros inside the unit circle. □

17.3 Synthesis of a Set of Stabilizing Controllers

In this section, we seek to exploit the Interlacing Property (IP) of Schur polynomials to systematically generate inner and outer approximation of the set of stabilizing controllers, $\mathcal{S}$. This approach leads to sets of linear programs (LPs).

Let $P(z, K)$ be a *real* closed loop characteristic polynomial whose coefficients are affinely dependent on the design parameters K; one can define the Chebyshev representation through $P(e^{j\theta}, K) = R(u, K) + j\sqrt{1-u^2}T(u, K)$, where $u = -\cos\theta$. $R(u, K)$ and $T(u, K)$ are real polynomials of degree n and $n-1$, respectively and are affine in the controller parameter K. The leading coefficients of $R(u, K)$ and $T(u, K)$ are of opposite sign and are of equal magnitude.

17.3.1 Inner Approximation

The stabilizing set of controllers, $\mathcal{S}$ is the set of all controllers, K, that simultaneously satisfy the conditions of Theorem 1. The problem of rendering $P(z, K)$ Schur can be posed as a search for $2n-2$ values of u. By way of notation, we represent the polynomials $R(u, K)$ and $T(u, K)$ compactly in the following form:

$$R(u, K) = \begin{bmatrix} 1 & u & \cdots & u^n \end{bmatrix} \Delta_R \begin{bmatrix} 1 \\ K \end{bmatrix}, \tag{17.3}$$

$$T(u, K) = \begin{bmatrix} 1 & u & \cdots & u^{n-1} \end{bmatrix} \Delta_T \begin{bmatrix} 1 \\ K \end{bmatrix}. \tag{17.4}$$

In (17.3) and (17.4), Δ_R and Δ_T are real constant matrices that depend on the plant data and the structure of the controller sought; they are respectively of dimensions $(n+1) \times (l+1)$ and $(n) \times (l+1)$, where, n is the degree of the characteristic polynomial and l is the size of the controller parameter vector. For $i = 1, 2, 3, 4$, let C_i and S_i be diagonal matrices of size $2n$; for an integer m, the $(m+1)$st diagonal entry of C_i is $\cos\left(\frac{(2i-1)\pi}{4} + \frac{m\pi}{2}\right)$ and the corresponding entry for S_i is $\sin\left(\frac{(2i-1)\pi}{4} + \frac{m\pi}{2}\right)$. For any given set of $2n-2$ distinct values of u,

$$-1 = u_0 < u_1 < \cdots < u_{2n-2} < u_{2n-1} = 1,$$

and for any integer m define a Vandermonde-like matrix,

$$V(u_0, u_1, \ldots, u_{2n-1}, m) := \begin{bmatrix} 1 & u_0 & \ldots & u_0^m \\ 1 & u_1 & \ldots & u_1^m \\ 1 & u_2 & \ldots & u_2^m \\ \vdots & \vdots & \vdots & \vdots \\ 1 & u_{2n-1} & \ldots & u_{2n-1}^m \end{bmatrix}.$$

We are now ready to characterize the set of stabilizing controllers K.

Theorem 2 *There exists a real control parameter vector $K = (k_1, k_2, \ldots, k_l)$ so that the real polynomial $P(z, K)$*

$$P(z,K) := P_0(z) + k_1 P_1(z) + \cdots + k_l P_l(z)$$
$$= p_n(K)z^n + p_{n-1}(K)z^{n-1} + \cdots + p_0(K)$$

is Schur if and only if there exists a set of $2n-2$ *values,* $-1 = u_0 < u_1 < u_2 < \cdots < u_{2n-2} < u_{2n-1} = 1$, *so that one of the following two linear programs (LPs) is feasible:*

LPs:

$$C_k V(u_0, u_1, \ldots, u_{2n-1}, n)\Delta_R \begin{bmatrix} 1 \\ K \end{bmatrix} > 0,$$

$$S_k V(u_0, u_1, \ldots, u_{2n-1}, n-1)\Delta_T \begin{bmatrix} 1 \\ K \end{bmatrix} > 0, \quad \textit{for } k = 1, 3.$$

Proof The three conditions of Theorem 1 is equivalent to the existence of $2n-2$ values of $u, -1 < u_1 < u_2 < \cdots < u_{2n-2} < 1$ such that the roots of the Chebyshev polynomial $R(u,K)$ lie in

$$(-1, u_1), (u_2, u_3), (u_4, u_5), \ldots$$

while the roots of the other Chebyshev polynomial $T(u,K)$ lie in

$$(u_1, u_2), (u_3, u_4), (u_5, u_6), \ldots$$

If $R(-1,K) > 0, T(-1,K) > 0$, then the placement of roots will require

$$R(u_1,K) < 0, R(u_2,K) < 0, R(u_3,K) > 0, \ldots$$

and

$$T(u_1,K) > 0, T(u_2,K) < 0, T(u_3,K) < 0, \ldots$$

In other words, the signs of $R(u_i,K)$ and $T(u_i,K)$ are the same as that of $\cos(\frac{\pi}{4} + i\frac{\pi}{2})$ and $\sin(\frac{\pi}{4} + i\frac{\pi}{2})$ respectively. This corresponds to the **LP** for $k = 1$. Similarly for $R(-1,K) < 0$ and $T(-1,K) < 0$ we have the **LP** corresponding to $k = 3$. □

The essential idea is that the plot of the polynomial $P(e^{j\theta})$ must go through $2n$ quadrants in the counterclockwise direction as θ increases from 0 to π. The conditions given above correspond to the plot starting in the kth quadrant at $\theta = 0^+$.

The procedure to find the inner approximation is to partition the interval $(-1,1)$ using more than $(2n-2)$ points (either uniformly or by using appropriate Chebyshev polynomial) and systematically searching for the feasibility of the obtained set of linear inequalities. Every feasible LP, yields a controller K which makes the polynomial $P(z,K)$ Schur. The union of all the feasible sets of the LPs described above, for all possible sets of $(2n-2)$ points in $(-1,1)$ is the set of all stabilizing controllers. With partitioning $(-1,1)$, however, one will be able to capture only finitely many of the possible sets of $(2n-2)$ points, $u_1, \ldots, u_{2n-2}$. The feasible sets of the LPs corresponding to these finitely many possible sets will provide an *inner approximation* of the set of all stabilizing controllers. This approximation can be made more accurate by refining the partition—i.e., if K is a

stabilizing controller not in the approximate set, then there is refinement [which will separate the roots of $R(u, K)$ and $T(u, K)$] of the partition from which one can pick $2n - 2$ points so that one of the two LPs corresponding to these points is feasible. This is the basic procedure for finding the inner approximation.

17.3.2 Outer Approximation

In the previous section, we outlined a procedure to construct LPs whose feasible set is contained in $\mathcal{S}$. Their union $\mathcal{S}_i$ is an inner approximation to $\mathcal{S}$. For computation, it is useful to develop an outer approximation, $\mathcal{S}_o$ that contains $\mathcal{S}$. In this section, we present a procedure to construct an arbitrarily tight outer approximation $\mathcal{S}_o$ as a union of the feasible sets of LPs. We propose to use the Poincare's generalization of Descartes' rule of signs.

Theorem 3 (Poincare's Generalization) *Let $P(x)$ be a polynomial with real coefficients. The number of sign changes in the coefficients of $Q_k(x) := (x + 1)^k P(x)$ is a non-increasing function of k; for a sufficiently large k, the number of sign changes in the coefficients equals the number of real, positive roots of $P(x)$.*

The proof of the generalization due to Poincare is given in Polya and Szego [17].

For the discussion on outer approximation, we will treat the polynomials, $\hat{R}(\lambda, K)$ and $\hat{T}(\lambda, K)$, as polynomials in λ obtained through the bijective mapping $\lambda = \frac{1+u}{1-u}$. This maps the interval $(-1, +1)$ into the interval $(0, \infty)$. This mapping is applied in the following way:

$$(1 + \lambda)^n Q\left(\frac{\lambda - 1}{1 + \lambda}\right) = \hat{Q}(\lambda).$$

The ith roots of $\hat{R}(\lambda, K)$ and $\hat{T}(\lambda, K)$ be represented as $\lambda_{r,i}$ and $\lambda_{t,i}$ respectively. Since the polynomials $\hat{R}$ and $\hat{T}$ must have respectively n and $n - 1$ real, positive roots, an application of Poincare's result to the polynomials $\hat{R}$ and $\hat{T}$ yields the following:

Lemma 2 *If K is a stabilizing control vector, then $(\lambda + 1)^{k-1}\hat{R}(\lambda, K)$ and $(\lambda + 1)^{k-1}\hat{T}(\lambda, K)$ have exactly n and $n - 1$ sign changes in their coefficients respectively for every $k \geq 1$.*

The procedure in [4] corresponds to $k = 1$ of the above lemma.

The following lemma takes care of the interlacing of the roots of two polynomials:

Lemma 3 *Let K render a polynomial $P(z, K)$ Schur. Then the polynomial,*

$$\tilde{Q}(\lambda, K, \eta) = \lambda\hat{T}(\lambda, K) - \eta(1 + \lambda)\hat{R}(\lambda, K),$$

has exactly n real positive roots for all $\eta \in \mathcal{R}$.

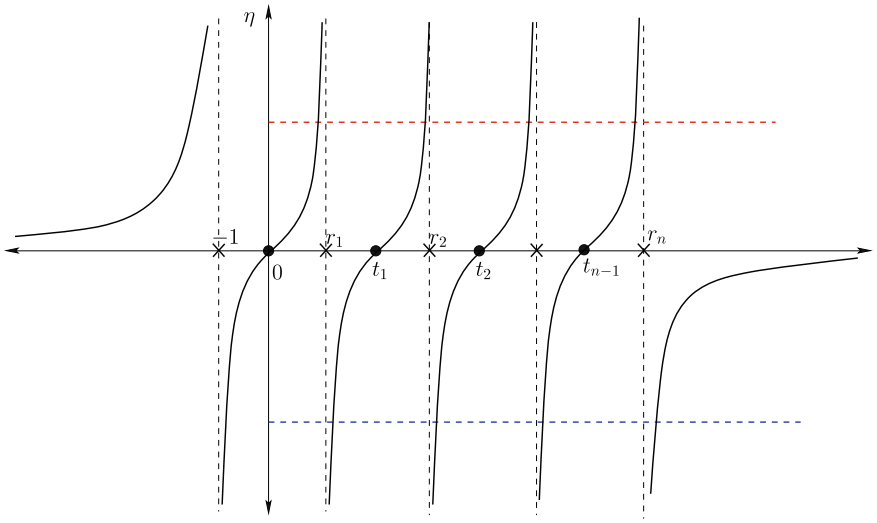


Fig. 17.1 Graph of the rational function $y := \frac{\lambda \hat{T}(\lambda)}{(1+\lambda)\hat{R}(\lambda)}$

Proof The roots of $\hat{T}(\lambda, K)$ and $\hat{R}(\lambda, K)$ are real and positive and they interlace *if and only if* $\tilde{Q}(\lambda, K, \eta)$ has exactly n real positive roots for all $\eta \in \mathcal{R}$. To prove sufficiency, we consider the graph of the rational function $y := \frac{\lambda \hat{T}(\lambda)}{(1+\lambda)\hat{R}(\lambda)}$ and consider the intersections with $y = \eta$ (see Fig. 17.1). To prove necessity, we argue, via a root locus argument, that if the interlacing of real roots condition is violated, then for some value of $\eta \in \Re$, polynomial $\tilde{Q}(\lambda, K, \eta)$ will have at least a pair of complex conjugate roots. □

Lemmas 2 and 3 can be put together to show that an arbitrarily tight outer approximation can be constructed.

Example 1 Consider the plant

$$G(z) = \frac{z^2 - 2z + 1}{1.9z^2 + 2.1}.$$

It is desired to calculate the complete set of first order controllers of the form

$$C(z) = \frac{k_1(z-1)}{z + k_2}.$$

The characteristic equation is given by

$$(1.9 + k_1)z^3 + (1.9k_2 - 3k_1)z^2 + (2.1 + 3k_1)z + (2.1k_2 - k_1).$$

The Chebyshev polynomials are found to be:

$$R(u,K) = -(7.6 + 4k_1)u^3 + (3.8k_2 - 6k_1)u^2 + 3.6u + 2k_1 + 0.2k_2,$$
$$T(u,K) = (7.6 + 4k_1)u^2 + (6k_1 - 3.8k_2)u + (2k_1 + 0.2).$$

These can be written in the compact form of (3) and (4) as:

$$R(u,K) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 0 & 2 & 0.2 \\ 3.6 & 0 & 0 \\ 0 & -6 & 3.8 \\ -7.6 & -4 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ k_1 \\ k_2 \end{bmatrix},$$
$$T(u,K) = \begin{bmatrix} 1 & u & u^2 \end{bmatrix} \begin{bmatrix} 0.2 & 2 & 0 \\ 0 & 6 & -3.8 \\ 7.6 & 4 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ k_1 \\ k_2 \end{bmatrix}.$$

The above compact form allows the LPs to be formulated quite easily and the interval $[-1, 1]$ is partitioned and a systematic search for $2n - 1$ points in the interval is carried out.

Figure 17.2 displays the inner and outer approximation of the set of stabilizing controllers. The difference between outer approximation and inner approximation is the black colored region. The inner approximation is an excellent approximation of the complete set of stabilizing controllers.

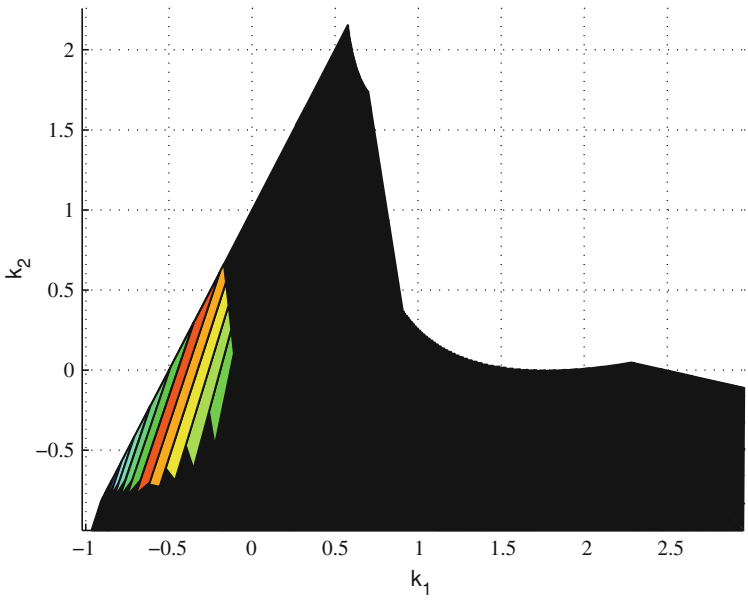


Fig. 17.2 Set of stabilizing controllers—an inner and outer approximation

Example 2 Consider the plant:

$$G(z) = \frac{1}{z^2 - 0.25}.$$

The controller is considered to be of the following PID structure:

$$C(z) = \frac{k_3 z^2 + k_2 z + k_1}{z^2 - z}.$$

The characteristic polynomial is

$$z^4 - z^3 + (k_3 - 0.25)z^2 + (k_2 + 0.25)z + k_1.$$

The Chebyshev polynomials are:

$$R(u, K) = 8u^4 + 4u^3 + (2k_3 - 8.5)u^2 - (k_2 + 3.25)u - k_3 + 1.25 + k_1,$$
$$T(u, K) = -8u^3 - 4u^2 - (2k_3 - 4.5)u + k_2 + 1.25.$$

In compact form, the Chebyshev polynomials can be represented as:

$$R(u, K) = \begin{bmatrix} 1 & u & u^2 & u^3 & u^4 \end{bmatrix} \begin{bmatrix} 1.25 & 1 & 0 & -1 \\ -3.25 & 0 & -1 & 0 \\ -8.5 & 0 & 0 & 2 \\ 4 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ k_1 \\ k_2 \\ k_3, \end{bmatrix}$$

$$T(u, K) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1.25 & 0 & 1 & 0 \\ 4.5 & 0 & 0 & -2 \\ -4 & 0 & 0 & 0 \\ -8 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ k_1 \\ k_2 \\ k_3. \end{bmatrix}$$

An inner approximation of the set of controllers is shown in Fig. 17.3. An inner and outer approximation of the set of controllers are shown in Fig. 17.4. The inner approximation obviously lies inside the outer approximation, which is depicted using the lighter color.

17.4 On the Boundedness of the Set of Stabilizing Controllers

It is a known fact that an nth order plant can be stabilized by a $(n-1)$th order controller [6]. The poles of the closed loop system can be freely assigned with such a controller which can be obtained by using the inversion of the eliminant matrix

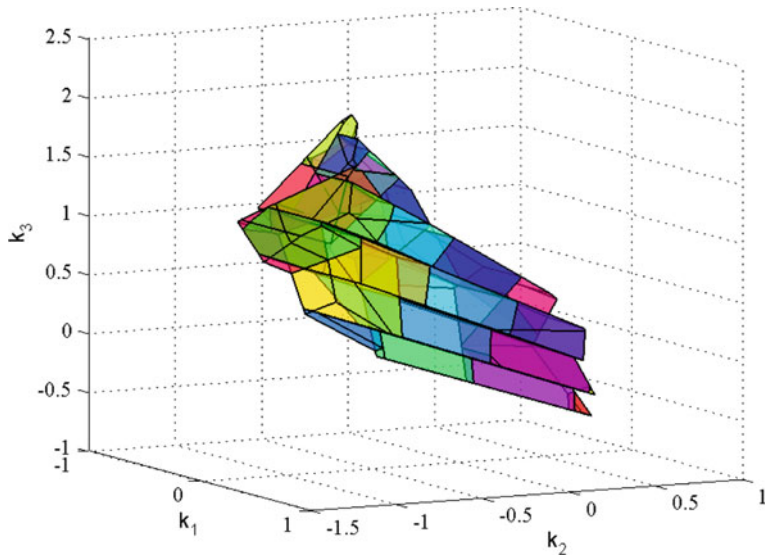


Fig. 17.3 Solution for Example 2: an inner approximation

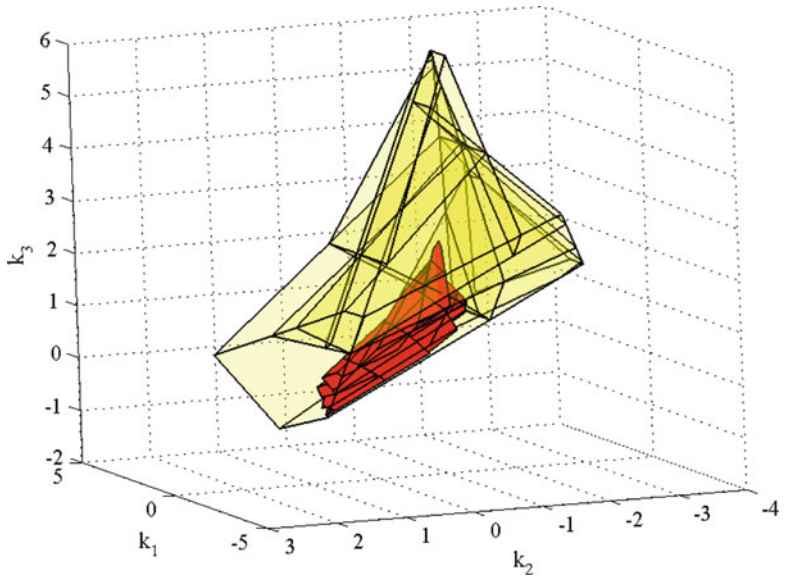


Fig. 17.4 Set of stabilizing controllers—an inner and outer approximation

of the plant. However for a given controller order r it is not clear if stabilization is possible. A related question is the minimal order of a stabilizing controller for a plant. In this section, we provide the first such characterization for discrete-time

LTI system through the boundedness of the set of controllers of a given order. In the previous section, we provided a systematic approach for synthesizing the inner and outer approximations of the set of stabilizing controllers and such approximations can be brought to bear in the verification of the minimal order of stabilization. We show that the set of rational proper stabilizing controllers for single-input single-output linear time-invariant discrete-time plants will form a bounded set in the controller parameter space *if and only if* the order of the stabilizing cannot be reduced any further. Moreover, if the order of the controller is increased, the set of higher order controllers will necessarily be unbounded. The continuous-time counterpart of these results are presented in Malik et al. [15].

The following lemmas provide the key basis for the proposed characterization of stabilizing controllers.

Lemma 4 *If* $C_r(z) = \frac{N_r(z)}{D_r(z)}$ *is a rth order rational, proper controller that stabilizes* $P(z) = \frac{N_p(z)}{D_p(z)}$, *then given any polynomials* $\tilde{N}_r(z)$ *and* $\tilde{D}_r(z)$ *of degree r, there is a* $\epsilon^* > 0$ *such that the* $(r+1)th$ *order proper, rational controller* $C_{r+1}(z) = \frac{\frac{1}{\epsilon}zN_r(z)+\tilde{N}_r(z)}{\frac{1}{\epsilon}zD_r(z)+\tilde{D}_r(s)}$ *also stabilizes* $\frac{N_p(z)}{D_p(z)}$ *for every* $0 < \epsilon \leq \epsilon^*$.

Proof Let $\Delta(z) := N_p(z)N_r(z) + D_p(z)D_r(z)$. The characteristic polynomial, $\Delta_{\text{pert}}(z, \epsilon)$, associated with the perturbed controller, $C_{r+1}(z)$, is $\frac{1}{\epsilon}z\Delta(z) + \left(\tilde{N}_r(z)N_p(z) + \tilde{D}_r(z)D_p(z)\right)$. If ϵ is treated as a variable in the following root locus problem,

$$1 + \epsilon\frac{\tilde{N}_r(z)N_p(z) + \tilde{D}_r(z)D_p(z)}{z\Delta(z)},$$

it follows that there is a $\epsilon^* > 0$ such that for all $0 < \epsilon \leq \epsilon^*$, the polynomial, $\Delta_{\text{pert}}(z, \epsilon)$, is Schur. □

The following are consequences of Lemma 4:

1. If there is a *r*th order stabilizing controller, then there is a stabilizing controller of order $r + 1$. Therefore, there is no gap in the order of stabilization. Hence, minimal order compensators can be synthesized by recursively reducing the order of stabilizing controllers by one.
2. Let us consider a *class* of rational proper controllers of the form

$$C_r(z) = \frac{k_0 + k_1 z + \cdots + k_r z^r}{1 + k_{r+2}z + \cdots + k_{2r}z^{r-1} + k_{2r+1}z^r},$$

and associate it with a vector

$$K = (k_0, k_1, \ldots, k_r, 1, \ldots, k_{2r}, k_{2r+1}).$$

Clearly, there is a one-to-one correspondence with $K \in \Re^{2r+2}$ and a rational, proper rth order controller $C_r(z)$. Note that we have fixed the k_{r+1} entry to be one. Without any loss of generality, we will use K and $C_r(z)$ interchangeably.

Let $\tilde{N}_r(z) = \tilde{k}_0 + \tilde{k}_1 z + \cdots + \tilde{k}_r z^r$, and $\tilde{D}_r(z) = 1 + \tilde{k}_{r+2} z + \cdots + \tilde{k}_{2r} z^{r-1} + \tilde{k}_{2r+1} z^r$, so that, by Lemma 4, there is a ϵ^* such that for all $0 < \epsilon \leq \epsilon^*$, the following $(r+1)$th order controller, $\tilde{C}_{r+1}(z)$, is also stabilizing:

$$\tilde{C}_{r+1}(z) = \frac{\tilde{k}_0 + \left(\tilde{k}_1 + \frac{1}{\epsilon}k_0\right)z + \cdots + \left(\tilde{k}_r + \frac{1}{\epsilon}k_{r-1}\right)z^r + \frac{1}{\epsilon}k_r z^{r+1}}{1 + \left(\tilde{k}_{r+2} + \frac{1}{\epsilon}\right)z + \cdots + \left(\tilde{k}_{2r+1} + \frac{1}{\epsilon}k_{2r}\right)z^r + \frac{1}{\epsilon}k_{2r+1}z^{r+1}}.$$

Hence the associated vector, $\tilde{K} \in \Re^{2r+4}$,

$$\tilde{K}(\epsilon) = \left(\tilde{k}_0, \tilde{k}_1 + \frac{1}{\epsilon}k_0, \ldots, \tilde{k}_r + \frac{1}{\epsilon}k_{r-1}, \frac{1}{\epsilon}k_r, 1, \tilde{k}_{r+2} + \frac{1}{\epsilon}k_{r+1}, \ldots, \tilde{k}_{2r+1} + \frac{1}{\epsilon}k_{2r}, \frac{1}{\epsilon}k_{2r+1}z^{r+1}\right).$$

Define $K_0 := \tilde{K}(\epsilon^*), \lambda := \frac{1}{\epsilon} - \frac{1}{\epsilon^*}$, and let K_1 be

$$K_1 := (0, k_0, k_1, \ldots, k_r, 0, 1, k_{r+2}, \ldots, k_{2r}, k_{2r+1}).$$

Then, $\tilde{K} = K_0 + \lambda K_1$ is stabilizing for every $\lambda \geq 0$, by Lemma 4. Thus, $\tilde{K}$ is a ray originating at K_0 and is in the direction of K_1 in the space of parameters of $(r+1)$th order proper stabilizing controllers. Two things can be inferred from the above:

(a) If an rth order stabilizing compensator exists, the set of $(r+1)$th order proper stabilizing controller parameters is unbounded. In particular, the set of $(r+1)$th order proper stabilizing controllers contains a ray of the form $K_0 + \lambda K_1$ in $\Re^{2r+4}$ that is stabilizing for every $\lambda \geq 0$. The converse is in general not true.
(b) If, by some means, one were to find a ray, $\{K_0 + \lambda K_1,\ \lambda \geq 0\}$, of proper $(r+1)$st order stabilizing controllers, with K_1 having the first and $(r+2)$nd entry to be zero and $k_{r+1} = 1$, then it seems likely to recover a lower order controller from K_1 considering the correspondence between K_1 and $C(z)$.

Note that the *class* of stabilizing controllers we consider, are controllers whose denominators have a constant term (k_{r+1} entry) to be unity. If a stabilizing controller $C_r(z)$ has a pole at 0, then one can always construct a stabilizing controller of the same order without a pole at zero by a slight perturbation. For this reason, there is no loss of generality in assuming that $C_r(z)$ has no poles at 0.

The following theorem provides the conditions for the existence of a lower-order controller from the unboundedness of the set of higher-order controllers.

Theorem 4 *A proper controller of order r stabilizing $P(z)$ exists if there exists a $\rho \in (0,1)$ and a ray of proper stabilizing controllers of order $r+1$, namely $\{K_0 + \lambda K_1,\ \lambda > 0\}$, that place the closed loop poles inside the disk, D_ρ.*

Proof A controller of order $n-1$ always exists for a SISO plant of order n. Hence, we will assume that $r \leq n-2$.

Necessity: Suppose an rth-order proper controller, $C(z)$ stabilizes the plant, $P(z)$ and let p_i be the roots of the closed loop polynomial. Define $\rho_0 := \max(|p_i|)$ and define $\rho := \frac{1+\rho_0}{2}$. By Lemma 4, there exists an ϵ^*, such that for all $0 < \epsilon \leq \epsilon^*$, the $(r+1)$th order controller,

$$C_{r+1}(z) = \frac{z}{z+\epsilon} C(z),$$

stabilizes the plant $P(z)$ and places the poles of the closed loop inside the disk D_ρ.

If $C(z)$ is of the form

$$C(z) = \frac{c_0 + c_1 z + \cdots + c_r z^r}{1 + d_1 z + \cdots + d_{r-1} z^{r-1} + d_r z^r},$$

then

$$C_{r+1}(z) = \frac{\frac{z}{\epsilon}(c_0 + c_1 z + \cdots + c_r z^r)}{\frac{z}{\epsilon}(1 + d_1 z + \cdots + d_{r-1} z^{r-1} + d_r z^r) + (1 + d_1 z + \cdots + d_r z^r)}.$$

In the parameter space of $(r+1)$th order controller, it is of the form, $K_0 + \lambda K_1$, where

$$\begin{aligned}
\lambda &:= \frac{1}{\epsilon} - \frac{1}{\epsilon^*}, \\
K_1 &= (0, c_0, c_1, \ldots, c_r, 0,\ 1,\ d_1,\ \ldots, d_{r-1},\ d_r), \\
K_0 &= \frac{1}{\epsilon^*} K_1 + (\underbrace{0, \ldots, 0}_{r+2 \text{ zeros}}, 1, d_1, \ldots, d_{r-1}, d_r, 0),
\end{aligned}$$

and this ray of controllers, $\{K_0 + \lambda K_1,\ \lambda > 0\}$ stabilizes the plant $P(z)$ and places the closed loop poles inside the disk D_ρ.

Sufficiency: Consider a ray of controllers, of order $r+1$ as given below:

$$C(z, \lambda) = \frac{\lambda z N_c(z) + N_c^*(z)}{\lambda z D_c(z) + D_c^*(z)},$$

where,

$$\begin{aligned}
N_c(z) &= c_0 + c_1 z + c_2 z^2 + \cdots + c_r z^r, \\
N_c^*(z) &= e_0 + e_1 z + e_2 z^2 + \cdots + e_{r+1} z^{r+1}, \\
D_c(z) &= d_0 + d_1 z + d_2 z^2 + \cdots + d_r z^r, \\
D_c^*(z) &= f_0 + f_1 z + f_2 z^2 + \cdots + f_{r+1} z^{r+1}.
\end{aligned}$$

As shown above, we require $D_c(z)$ to be of order r. Suppose this ray of controllers $C(z, \lambda)$ of order $r+1$ stabilize the plant $P(z)$ and place the closed loop poles inside the disk D_ρ. for some $\rho \in (0, 1)$. If $P(z) = \frac{N_p(z)}{D_p(z)}$, then, the closed loop characteristic

polynomial for the plant $P(z)$ with a controller from the ray (identified by λ) may be written as:

$$\Delta(P(z), \lambda) = \lambda z \Delta_0(z) + \Delta_1(z),$$

where $\Delta_0(z) = N_p(z)N_c(z) + D_p(z)D_c(z)$ and $\Delta_1(z) = N_p(z)N_c^*(z) + D_p(z)D_c^*(z)$. Since $\Delta(P(z), \lambda)$ is Schur for all $\lambda > 0$, it must be true that $\Delta_0(z)$ must be Schur. Hence, $C(z) = \frac{N_c(z)}{D_c(z)}$ is a lower order controller stabilizing the plant $P(z)$. □

Example 3 Consider the plant:

$$G(z) = \frac{z-2}{z^3 - 8z^2 + 19z - 12}.$$

A first order controller of the following form is considered:

$$C(z) = \frac{k_1 z + k_2}{k_3 z + 1}.$$

The characteristic polynomial is

$$k_3 z^4 + (1 - 8k_3)z^3 + (k_1 - 8 + 19k_3)z^2 + (-2k_1 + 19 + k_2 - 12k_3)z - 12 - 2k_2.$$

The Chebyshev polynomials are:

$$\begin{aligned} R(u) &= 8k_3u^4 + (-4 + 32k_3)u^3 + (2k_1 - 16 + 30k_3)u^2 \\ &\quad + (2k_1 - k_2 - 16 - 12k_3)u + (-2k_2 - k_1 - 18k_3 - 4), \\ rT(u) &= -8k_3u^3 + (4 - 32k_3)u^2 + (16 - 2k_1 - 34k_3)u + 18 - 4k_3 - 2k_1 + k_2. \end{aligned}$$

In compact form, the Chebyshev polynomials can be represented as:

$$R(u, K) = \begin{bmatrix} 1 & u & u^2 & u^3 & u^4 \end{bmatrix} \begin{bmatrix} -4 & -1 & -2 & -18 \\ -16 & 2 & -1 & -12 \\ -16 & 2 & 0 & 30 \\ -4 & 0 & 0 & 32 \\ 0 & 0 & 0 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ k_1 \\ k_2 \\ k_3 \end{bmatrix},$$

$$T(u, K) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 18 & -2 & 1 & -4 \\ 16 & -2 & 0 & -34 \\ 4 & 0 & 0 & -32 \\ 0 & 0 & 0 & -8 \end{bmatrix} \begin{bmatrix} 1 \\ k_1 \\ k_2 \\ k_3 \end{bmatrix}.$$

Outer and inner approximations of the set of controllers are shown in Figs. 17.5 and 17.6 respectively. The outer approximation is bounded and hence the minimal order of stabilizing controller for the given plant is one.

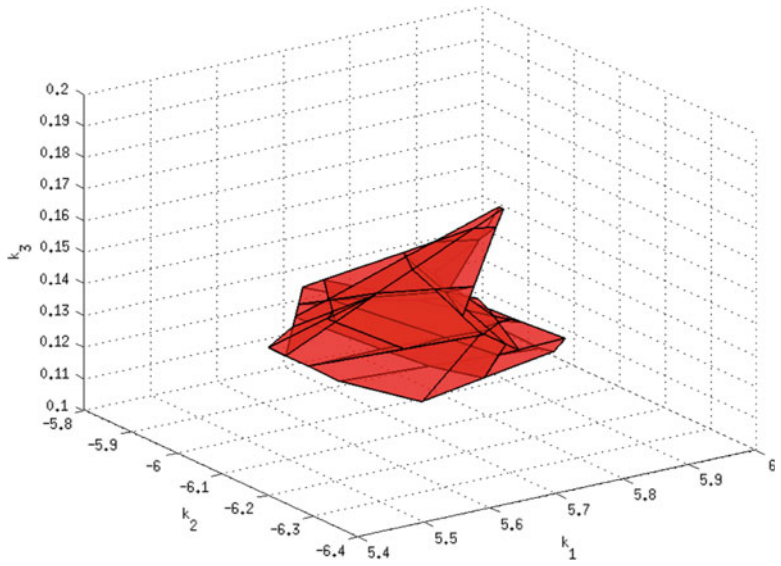


Fig. 17.5 Solution for Example 3: an outer approximation

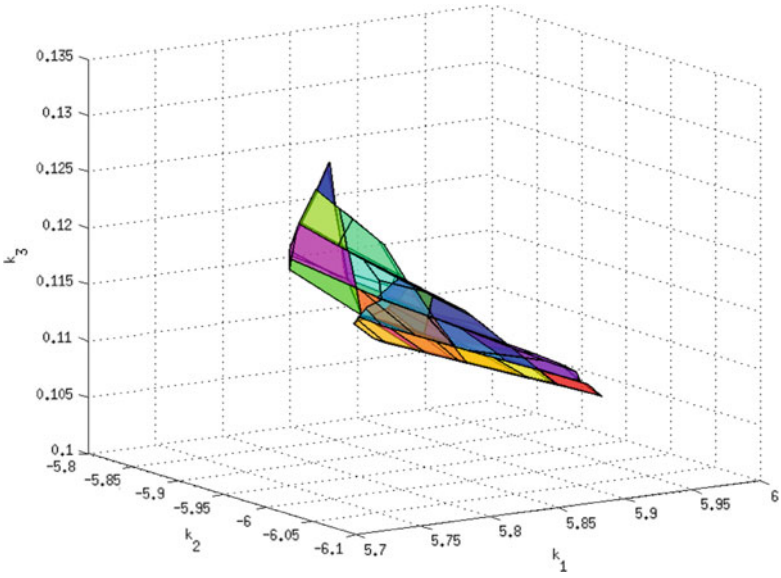


Fig. 17.6 Solution for Example 3: an inner approximation

17.5 Conclusions

In this paper, we considered the problem of synthesis of fixed order and structure controllers, where the coefficients of the closed loop characteristic polynomial are linear in the parameters of the controller. A novel feature of this paper is the systematic exploitation of the interlacing property of Schur polynomials and the use of Poincare's generalization of Descartes' rule of signs to generate LPs in the parameters of a fixed order controller. The feasible set of any LP generated for an inner approximation of the set of all stabilizing controllers, can be indexed by a set of $2n-2$ increasing values, $-1 = u_0 < u_1 < u_2 < \cdots < u_{2n-2} < u_{2n-1} = 1$; in particular, any controller in the feasible set of LPs places the roots of the Chebyshev polynomials of $P(z, K)$ alternately in the intervals (u_i, u_{i+1}), $i = 0, \ldots, 2n-1$. The problem of inner approximation of the set of stabilizing controllers is then posed as the search for all sets of ordered $2n-2$-tuples of points for which the associated LP is feasible; the union of all feasible LPs is an inner approximation for the set of all stabilizing controllers. The proposed methodology naturally extends to the computation of the set of simultaneously stabilizing controllers. We also show that the set of proper stabilizing controllers of order r is not empty and is bounded *iff* r is the minimal order of stabilization for the plant. We provide examples to illustrate some of the results.

References

1. Ackermann J (1993) Robust control systems with uncertain physical parameters. Springer, Berlin
2. Bengtsson G, Lindahl S (1974) A design scheme for incomplete state or output feedback with applications to boiler and power system control. Automatica 10:15–30
3. Bernstein D (1992) Some open problems in matrix theory arising in linear systems and control. Linear Algebra Appl 162–164:409–432
4. Bhattacharyya SP, Keel LH, Howze J (1988) Stabilizability conditions using linear programming. IEEE Trans Automat Contr 33:460–463
5. Blondel V, Gevers M, Lindquist A (1995) Survey on the state of systems and control. European J Contr 1:5–23
6. Brasch FM, Pearson JB (1970) Pole placement using dynamic compensator. IEEE Trans Automat Contr AC-15:34–43
7. Buckley A (1995) Hubble telescope pointing control system design improvement study. J Guid Control Dyn 18:194–199
8. Datta A, Ho MT, Bhattacharyya SP (2000) Structure and synthesis of PID controllers. Springer, London
9. El Ghaoui L, Oustry F, AitRami M (1997) A cone complementarity linearization algorithm for static output feedback and related problems. IEEE Trans Automat Contr 42–48:1171–1176
10. Goodwin GC, Graebe SF, Salgado ME (2001) Control system design. Prentice-Hall, Upper Saddle River
11. Henrion D, Sebek M, Kucera V (2003) Positive polynomials and robust stabilization with fixed-order controllers. IEEE Trans Automat Contr 48(7):1178–1186

12. Iwasaki T, Skelton RE (1995) The XY-centering algorithm for the dual LMI problem: a new approach to fixed order control design. Int J Contr 62(6):1257–1272
13. Keel LH, Rego JI, Bhattacharyya SP (2003) A new approach to digital PID controller design. IEEE Trans Automat Contr 48(4):687–692
14. Malik WA, Darbha S, Bhattacharya SP (2004) On the synthesis of fixed structure controllers satisfying given performance criteria. In: 2nd IFAC Symposium on System, Structure and Control
15. Malik WA, Darbha S, Bhattacharyya SP (2007a) On the boundedness of the set of stabilizing controllers. Int J Robust Nonlinear Contr, page in print
16. Malik WA, Darbha S, Bhattacharyya SP (2007b) A linear programming approach to the synthesis of fixed structure controllers. IEEE Trans Automat Contr, page in print
17. Polya G, Szego G (1998) Problems and theorems in analysis II—theory of functions, zeros, polynomials, determinants, number theory, geometry. Springer, Berlin
18. Syrmos VL, Abdullah CT, Dorato P, Grigoriadis K (1997) Static output feedback—a survey. Automatica 33(2):125–137
19. Zhu G, Grigoriadis K, Skelton R (1995) Covariance control design for the Hubble space telescope. J Guid Control Dyn 18(2):230–236

Chapter 18
A Secant Method for Nonlinear Matrix Problems

Marlliny Monsalve and Marcos Raydan

Abstract Nonlinear matrix equations arise in different scientific topics, such as applied statistics and control theory, among others. Standard approaches to solve them include and combine some variations of Newton's method, matrix factorizations, and reduction to generalized eigenvalue problems. In this paper we explore the use of secant methods in the space of matrices, that represent a new approach with interesting features. For the special problem of computing the inverse or the pseudoinverse of a given matrix, we propose a specialized secant method for which we establish stability and q-superlinear convergence, and for which we also present some numerical results. In addition, for solving quadratic matrix equations, we discuss several issues, and present preliminary and encouraging numerical experiments.

18.1 Introduction

The aim of this paper is to present a secant method for solving the following matrix nonlinear problem:

$$\text{given } F : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n} \text{ find } X_* \in \mathbb{C}^{n\times n} \text{ such that } F(X_*) = 0, \tag{18.1}$$

Dedicated with friendship to Biswa Datta for his scientific contributions.

M. Monsalve (✉)
Departamento de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela
e-mail: marlliny.monsalve@ciens.ucv.ve

M. Raydan
Departamento de Cómputo Científico y Estadística, Universidad Simón Bolívar (USB), Ap. 89000, Caracas 1080-A, Venezuela
e-mail: marcos.raydan@ciens.ucv.ve

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_18,

where F is a Fréchet differentiable map. In what follows, we denote by F' the Fréchet derivative of F.

This problem appears in different applications. For instances, given the matrices A,B, and C, the quadratic matrix equation $AX^2 + BX + C = 0$ arises in control theory [1, 2, 5, 9]. Another application is to compute the inverse or the pseudo-inverse of any given matrix A. Indeed, if we find the root of $F(X) = X^{-1} - A$, we obtain the inverse of A, and iterative schemes based on Newton's method can be applied for finding the inverse or the pseudoinverse of any matrix [13, 15]. For additional applications and further results concerning nonlinear matrix problems see [8, 11].

A useful tool for solving Eq. (18.1) is the well-known Newton's method:

Algorithm 1: Newton's method
Given $X_0 \in \mathbb{C}^{n\times n}$;
For $k = 0, 1, \cdots$
Solve $F'(X_k)S_k = -F(X_k)$ /*for S_k*/
Set $X_{k+1} = X_k + S_k$
End For

Note that we need F' to find S_k in each step of Algorithm 1 and in order to obtain F' we can use the Taylor series for F about $X, F(X+S) = F(X) + F'(X)S + R(S)$, where $R(S)$ is such that

$$\lim_{\|S\|\to 0} \frac{\|R(S)\|}{\|S\|} = 0,$$

The Taylor series allows us to identify the application of $F'(X)$ on S which is required to solve the linear equation of step 1 in Algorithm 1. In many cases, solving that linear equation is computationally expensive (see, e.g., [16]).

As a general principle, whenever a Newton's method is applicable, a suitable secant method can be obtained, that hopefully has interesting features to exploit. For example, in the well-known scalar case, the secant method does not require the derivative, and only uses function evaluations. In that case ($f : \mathbb{C} \to \mathbb{C}$), the secant method can be written as follow:

$$x_{k+1} = x_k - \frac{f(x_k)}{a_k},$$

where a_k satisfies that $f(x_k) = f(x_{k-1}) + a_k(x_k - x_{k-1})$ for $k \geq 0$, and $x_{-1}, x_0 \in \mathbb{C}$ are given.

Moreover, an extension for nonlinear systems of equations ($F : \mathbb{C}^n \to \mathbb{C}^n$), can be written as

$$x_{k+1} = x_k - A_k F(x_k)$$

where $A_k \in \mathbb{C}^{n\times n}$ satisfies that $F(x_k) = F(x_{k-1}) + A_k(x_k - x_{k-1})$ for $k \geq 1$, and the vector x_0 and the matrix A_0 are given. There are infinitely many options for building the matrix A_k at every iteration. In particular, the Broyden's family of quasi-Newton methods avoids the knowledge of the Jacobian and has produced a significant body of research for many different problems (see, e.g., [4, 12]).

In this work we develop secant methods for nonlinear matrix problems that inherit, as much as possible, the features of the classical secant methods in previous scenarios (e.g., scalar equations, nonlinear algebraic systems of equations). The rest of this document is organized as follows. In Sect. 18.2 we propose a general secant method for matrix problems which is based on the standard secant method, and we also describe some of its variations. In Sect. 18.3 we propose a specialized secant method for approximating the inverse or the pseudoinverse of a matrix. The global convergence and the stability are proved for this specialized secant method. We present numerical experiments for computing the inverse of some given nonsingular matrices, and for computing the pseudoinverse of a singular matrix. In Sect. 18.4 we consider the application of the general secant algorithms for solving quadratic matrix equations, and we also present some encouraging preliminary numerical results. Finally, in Sect. 18.5, we present some conclusions and perspectives.

18.2 A Secant Equation for Matrix Problems

A general secant method for solving (18.1) should be given by the following iteration

$$X_{k+1} = X_k - A_k^{-1}F(X_k), \tag{18.2}$$

where $X_{-1} \in \mathbb{C}^{n\times n}$ and $X_0 \in \mathbb{C}^{n\times n}$ are given, and A_{k+1} is a suitable linear operator that satisfies

$$A_{k+1}S_k = Y_k, \tag{18.3}$$

where $S_k = X_{k+1} - X_k$ and $Y_k = F(X_{k+1}) - F(X_k)$. Equation (18.3) is known as the *secant equation*.

Once X_{k+1} has been obtained, we observe in (18.3) that A_{k+1} can be computed at each iteration by solving a linear system of n^2 equations. Therefore, there is a resemblance with the scalar case, in which one equation is required to find one unknown. Similarly, we notice that one $n \times n$ matrix is enough to satisfy the matrix secant Eq. (18.3). Hence, we force the operator A_k to be a matrix of the same dimension of the step S_k and the map-difference Y_k, as in the scalar case. The proposed algorithm, and some important variants, can be summarized as follows:

Algorithm 2: **General secant method for matrix problems**

Given $X_{-1} \in \mathbb{C}^{n\times n}$, $X_0 \in \mathbb{C}^{n\times n}$
Set $S_{-1} = X_0 - X_{-1}$
Set $Y_{-1} = F(X_0) - F(X_{-1})$
Solve $A_0 S_{-1} = Y_{-1}$ /*for A_0*/
For $k = 0, 1, \cdots$ until convergence
Solve $A_k S_k = -F(X_k)$ /*for S_k*/
Set $X_{k+1} = X_k + S_k$
Set $Y_k = F(X_{k+1}) - F(X_k)$
Solve $A_{k+1} S_k = Y_k$ /*for A_{k+1}*/
End For

We can generate the sequence $B_k = A_k^{-1}$, instead of A_k, and obtain an inverse version that solves only one linear system of equations per iteration:

Algorithm 3: **Inverse secant method**

Given $X_{-1} \in \mathbb{C}^{n\times n}$, $X_0 \in \mathbb{C}^{n\times n}$
Set $S_{-1} = X_0 - X_{-1}$
Set $Y_{-1} = F(X_0) - F(X_{-1})$
Solve $B_0 Y_{-1} = S_{-1}$ /*for B_0*/
For $k = 0, 1, \cdots$ until convergence
Set $S_k = -B_k F(X_k)$
Set $X_{k+1} = X_k + S_k$
Set $Y_k = F(X_{k+1}) - F(X_k)$
Solve $B_{k+1} Y_k = S_k$ /*for B_{k+1}*/
End For

Solving a secant method that deals with $n \times n$ matrices is the most attractive feature of our proposal, and represents a sharp contrast with the standard extension of quasi-Newton methods for general Hilbert spaces, (see e.g. [6, 14]), that in this context would involve $n^2 \times n^2$ linear operators to approximate the derivative of F. Clearly, dealing with $n \times n$ matrices for solving the related linear systems significantly reduces the computational cost associated with the linear algebra of the algorithm.

In order to discuss some theoretical issues of the proposed general secant methods, let us consider the standard assumptions for problem (18.1): $F : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is continuously differentiable in an open and convex set $D \subseteq \mathbb{C}^{n\times n}$. There exists $X_* \in \mathbb{C}^{n\times n}$ and $r > 0$, such that $N(X_*, r) \subset D$ is an open neighborhood of radius r around X_*, $F(X_*) = 0$, and $F'(X_*)$ is nonsingular, and $F'(X) \in \mathrm{Lip}_\gamma(N(X_*, r))$, i.e., $F'(X)$ is a Lipschitz continuous function with constant $\gamma > 0$ in $N(X_*, r)$.

We begin by noticing that the operator A_k does not approximate $F'(X_k)$ as in previous scenarios due to dimensional discrepancies. Indeed, $F'(X_k) \in \mathbb{C}^{n^2\times n^2}$ and $A_k \in \mathbb{C}^{n\times n}$. However, fortunately, $F'(X_k)S_k$ and $A_k S_k$ both live in $\mathbb{C}^{n\times n}$, which turns out to be the suitable approximation since, using the secant equation (18.3), we have that

$$A_{k+1}S_k = Y_k = F(X_{k+1}) - F(X_k) = F'(X_k)S_k + R(S_k). \tag{18.4}$$

Subtracting $F'(X_*)S_k$ in both sides of (18.4), and taking norms we obtain

$$\|A_{k+1}S_k - F'(X_*)S_k\| \leq \|F'(X_k) - F'(X_*)\|\|S_k\| + \|R(S_k)\|,$$

for any subordinate norm $\|\cdot\|$. Using now that $F'(X) \in \text{Lip}_\gamma(N(X_*, r))$, and dividing by $\|S_k\|$ we have

$$\frac{\|A_{k+1}S_k - F'(X_*)S_k\|}{\|S_k\|} \leq \gamma\|E_k\| + \frac{\|R(S_k)\|}{\|S_k\|}, \tag{18.5}$$

where $E_k = X_k - X_*$ represents the error matrix.

From this inequality we observe that, if convergence is attained, the left hand side tends to zero when k goes to infinity, and so the sequence $\{A_k\}$, generated by Algorithm 2, tends to the Fréchet derivative, $F'(X_*)$, when they are both applied to the direction of the step S_k. Concerning local convergence, we have from Step 7 in Algorithm 2 that

$$\begin{aligned} E_{k+1} &= E_k - A_k^{-1}F(X_k) \\ &= E_k - A_k^{-1}F'(X_*)E_k - O(E_k^2), \end{aligned}$$

which implies that

$$\|E_{k+1}\| \leq \|E_k - A_k^{-1}(F'(X_*)E_k)\| + O(\|E_k\|^2). \tag{18.6}$$

Consequently, if A_k in our secant algorithms is such that $A_k^{-1}(F'(X_*)E_k)$ approximates E_k in a neighborhood of X_*, as expected, then $\|E_{k+1}\|$ is reduced with respect to $\|E_k\|$. Inequalities (18.5) and (18.6), somehow, explain the convergence behavior we have observed in our numerical results. In our next section, though, we will establish formally the stability and also the local and q-superlinear convergence of the proposed secant methods for the special case of computing the inverse or the pseudoinverse of a given matrix.

18.3 Special Case: Inverse or Pseudoinverse of a Matrix

For computing the inverse of a given matrix A we will consider iterative methods to find the root of

$$F(X) = X^{-1} - A, \tag{18.7}$$

and for the sake of clarity let us assume, for a while, that A is nonsingular.

Newton's method from an initial guess X_0, for solving (18.7), also known as Schulz method [15], is given by

$$X_{k+1} = 2X_k - X_k A X_k. \tag{18.8}$$

It has been established that if $X_0 = \frac{A^T}{\|A\|_2^2}$, then Schulz method possesses global convergence [7, 17]. Moreover, if A does not have an inverse, it converges to the pseudoinverse (also known as the generalized inverse) of A [7, 8, 17].

First, let us consider the general secant method applied to (18.7)

$$\begin{aligned} X_{k+1} &= X_k - S_{k-1}(F(X_k) - F(X_{k-1}))^{-1} F(X_k) \\ &= X_k - (X_k - X_{k-1})(X_k^{-1} - X_{k-1}^{-1})^{-1}(X_k^{-1} - A). \end{aligned} \tag{18.9}$$

Let us assume that A is *diagonalizable*, that is, there exists a nonsingular matrix V such that

$$V^{-1} A V = \Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of A, and let us define $D_k = V^{-1} X_k V$. From (18.9) we have that

$$\begin{aligned} D_{k+1} &= D_k - (V^{-1}X_k - V^{-1}X_{k-1})VV^{-1}(X_k^{-1} - X_{k-1}^{-1})^{-1}VV^{-1}(X_k^{-1}V - AV) \\ &= D_k - (D_k - D_{k-1})(D_k^{-1} - D_{k-1}^{-1})^{-1}(D_k^{-1} - \Lambda). \end{aligned} \tag{18.10}$$

Note that if we choose X_{-1} and X_0 such that $D_{-1} = V^{-1}X_{-1}V$ and $D_0 = V^{-1}X_0 V$ are diagonal matrices, then all successive D_k are diagonal too, and in this case $D_i D_j = D_j D_i$ for all i, j. Therefore (18.10) can be written as

$$\begin{aligned} D_{k+1} &= D_k - (D_k - D_{k-1})(D_k^{-1}D_{k-1}^{-1}D_{k-1} - D_{k-1}^{-1}D_k^{-1}D_k)^{-1}(D_k^{-1} - \Lambda) \\ &= D_k - (D_k - D_{k-1})(D_k^{-1}D_{k-1}^{-1}D_{k-1} - D_k^{-1}D_{k-1}^{-1}D_k)^{-1}(D_k^{-1} - \Lambda) \\ &= D_k - (D_k - D_{k-1})((D_{k-1}D_k)^{-1}(D_{k-1} - D_k))^{-1}(D_k^{-1} - \Lambda) \\ &= D_k + (D_k - D_{k-1})(D_k - D_{k-1})^{-1}(D_{k-1}D_k)(D_k^{-1} - \Lambda) \\ &= D_{k-1} + D_k - D_{k-1}\Lambda D_k. \end{aligned} \tag{18.11}$$

Motivated by (18.11) we now consider the specialized secant method for (18.7),

$$X_{k+1} = X_{k-1} + X_k - X_{k-1} A X_k, \tag{18.12}$$

that avoids the inverse matrix calculations per iteration associated with iteration (18.9). Notice the resemblance between (18.12) and Schulz method for solving the same problem. Therefore, in what follows (18.12) will be denoted as the

secant-Schulz method. Our next result establishes that if A is diagonalizable and the two initial guesses are chosen properly, then the secant-Schulz method converges locally and q-superlinearly to the inverse of A.

Theorem 1.1 *Let $A \in \mathbb{C}^{n\times n}$ be a nonsingular diagonalizable matrix, that is, there exists a nonsingular matrix V such that*

$$V^{-1}AV = \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of A. Let X_{-1} and X_0 be such that $V^{-1}X_{-1}V$ and $V^{-1}X_0V$ are diagonal matrices. Then the secant-Schulz method converges locally and q-superlinearly to the inverse of A.

Proof Let us define $D_k = V^{-1}X_kV$ for all $k \geq -1$. From (18.12) we have that

$$D_{k+1} = D_{k-1} + D_k - D_{k-1}\Lambda D_k. \tag{18.13}$$

Since D_{-1} and D_0 are diagonal matrices, then all successive D_k are diagonal too, and in this case $D_iD_j = D_jD_i$ for all i,j. Moreover, since $D_k = \mathrm{diag}(d_k^1, d_k^2, \ldots, d_k^n)$ we see from (18.13) that

$$d_{k+1}^i = d_{k-1}^i + d_k^i - d_{k-1}^i d_k^i \lambda_i, \quad \text{for all } 1 \leq i \leq n, \tag{18.14}$$

where (18.14) represents n uncoupled scalar secant iterations converging to $1/\lambda_i, 1 \leq i \leq n$. Indeed, subtracting $1/\lambda_i$ in both sides of (18.14) and letting $e_k^i = d_k^i - 1/\lambda_i$ we have that

$$\begin{aligned} e_{k+1}^i &= d_k^i + d_{k-1}^i - d_{k-1}^i d_k^i \lambda_i - 1/\lambda_i \\ &= -\lambda_i(d_k^i d_{k-1}^i - d_k^i/\lambda_i - d_{k-1}^i/\lambda_i + 1/\lambda_i^2) \\ &= -\lambda_i(d_k^i - 1/\lambda_i)(d_{k-1}^i - 1/\lambda_i) \\ &= -\lambda_i e_k^i e_{k-1}^i. \end{aligned} \tag{18.15}$$

From (18.15) we conclude that each scalar secant iteration (18.14) converges locally and q-superlinearly to $1/\lambda_i$. Therefore, equivalently [4], there exists a sequence $\{c_k^i\}$, for each $1 \leq i \leq n$, such that $c_k^i > 0$ for all k, $\lim_{k\to\infty} c_k^i = 0$, and

$$|e_{k+1}^i| \leq c_k^i |e_k^i|. \tag{18.16}$$

Using (18.16) we now obtain in the Frobenius norm

$$\begin{aligned} \|D_{k+1} - \Lambda^{-1}\|_F^2 &= \sum_{i=1}^n (e_{k+1}^i)^2 \leq \sum_{i=1}^n (c_k^i)^2 (e_k^i)^2 \\ &\leq n\widehat{c}_k^2 \sum_{i=1}^n (e_k^i)^2 \leq n\widehat{c}_k^2 \|D_k - \Lambda^{-1}\|_F^2, \end{aligned} \tag{18.17}$$

where $\widehat{c}_k = \max_{1\leq i\leq n}\{c_k^i\}$.

Finally, we have that

$$\begin{aligned}\|X_{k+1} - A^{-1}\|_F &= \|VV^{-1}(X_{k+1} - A^{-1})VV^{-1}\|_F \\ &= \|V(D_{k+1} - \Lambda^{-1})V^{-1}\|_F \\ &\leq \kappa_F(V)\|D_{k+1} - \Lambda^{-1}\|_F \\ &\leq \kappa_F(V)\sqrt{n}\widehat{c}_k\|D_k - \Lambda^{-1}\|_F \\ &= \kappa_F(V)\sqrt{n}\widehat{c}_k\|V^{-1}V(D_k - \Lambda^{-1})V^{-1}V\|_F \\ &\leq \kappa_F(V)^2\sqrt{n}\widehat{c}_k\|X_k - A^{-1}\|_F, \end{aligned} \tag{18.18}$$

where $\kappa_F(V)$ is the Frobenius condition number of V. Hence, the secant-Schulz method converges locally and q-superlinearly to the inverse of A. □

When A has no inverse, we can prove that the secant-Schulz method converges locally and q-superlinearly to the pseudoinverse of A, denoted by $A^{\dagger}$. For this case, let $A \in \mathbb{C}^{m\times n}$ be a matrix of rank r, and let us assume that its singular value decomposition is given by

$$A = U\begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix}V^*, \tag{18.19}$$

where $U \in \mathbb{C}^{m\times m}, V \in \mathbb{C}^{n\times n}$ are unitary matrices and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ where $\sigma_1, \sigma_2, \ldots, \sigma_r$ are the singular values of A.

Corollary 1.2 *Let $A \in \mathbb{C}^{m\times n}$ be a matrix of rank r, and let X_{-1} and X_0 be such that $V^*X_{-1}U = \begin{pmatrix} D_{-1} & 0 \\ 0 & 0 \end{pmatrix}$ and $V^*X_0U = \begin{pmatrix} D_0 & 0 \\ 0 & 0 \end{pmatrix}$ where V^*, U are defined in (18.19) and $D_{-1}, D_0 \in \mathbb{C}^{r\times r}$ are diagonal matrices. Then the secant-Schulz method converges locally and q-superlinearly to the pseudoinverse of A.*

Proof From iteration (18.12) and defining $\begin{pmatrix} D_k & 0 \\ 0 & 0 \end{pmatrix} = V^*X_kU$, with $D_k \in \mathbb{C}^{r\times r}$, we have that

$$\begin{pmatrix} D_{k+1} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} D_{k-1} + D_k - D_{k-1}\Sigma D_k & 0 \\ 0 & 0 \end{pmatrix}. \tag{18.20}$$

Since X_{-1} and X_0 are such that D_{-1} and D_0 are diagonal matrices then, using the same arguments as in the proof of Theorem 1.1, we obtain that

$$D_{k+1} = D_{k-1} + D_k - D_{k-1}\Sigma D_k,$$

represents r uncoupled scalar secant iterations that converges locally and q-superlinearly to $1/\sigma_i, 1 \leq i \leq r$, that is,

$$\|D_{k+1} - \Sigma^{-1}\|_F^2 \leq r\widehat{c}_k^2\|D_k - \Sigma^{-1}\|_F^2, \tag{18.21}$$

where $\widehat{c}_k = \max_{1 \le i \le r}\{c_k^i\}$ and the sequences $\{c_k^i\}$ are such that $c_k^i > 0$ and $\lim_{k\to\infty} c_k^i = 0$ for each $0 \le i \le r$. Finally using the same arguments used for obtaining (18.18) we have that

$$\begin{aligned}
\|X_{k+1} - A^\dagger\|_F &= \|VV^*(X_{k+1} - A^\dagger)UU^*\|_F \\
&= \|V\begin{pmatrix} D_k - \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix}U^*\|_F \\
&\le \sqrt{mn}\|D_{k+1} - \Sigma^{-1}\|_F \\
&\le \sqrt{mn}\sqrt{r}\widehat{c}_k\|D_k - \Sigma^{-1}\|_F \\
&= \sqrt{mn}\sqrt{r}\widehat{c}_k\|V^*V\begin{pmatrix} D_k - \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix}U^*U\|_F \\
&\le mn\sqrt{r}\widehat{c}_k\|X_k - A^\dagger\|_F. \qquad \square
\end{aligned}$$

It is important to note that Theorem 1.1 implies the well-known Dennis–Moré condition [3, 4]

$$\lim_{k\to\infty} \frac{\|A_k S_k - F'(X_*)S_k\|}{\|S_k\|} = 0,$$

that establishes the most important property of the sequence $\{A_k\}$ generated by the secant-Schulz method.

We now discuss the stability of our specialized secant method for the inverse matrix. First, let us recall the suitable definition from [8]. The fixed point iteration $Y_{k+1} = G(Y_k)$ is stable in a neighborhood of a fixed point Y_* if the Fréchet derivative $G'(Y_*)$ has bounded powers.

Theorem 1.3 *The secant-Schulz method generates a stable iteration.*

Proof The secant-Schulz method, as a fixed point iteration, can be obtained setting

$$Y_{k+1} = \begin{pmatrix} X_{k+1} \\ X_k \end{pmatrix}, \quad Y_* = \begin{pmatrix} A^{-1} \\ A^{-1} \end{pmatrix}$$

and

$$G(Y_k) = G\begin{pmatrix} X_k \\ X_{k-1} \end{pmatrix} = \begin{pmatrix} X_{k-1} + X_k - X_{k-1}AX_k \\ X_k \end{pmatrix}.$$

Therefore, the map we need to study is given by

$$G\begin{pmatrix} W \\ Z \end{pmatrix} = \begin{pmatrix} Z + W - ZAW \\ W \end{pmatrix},$$

for W and Z in $\mathbb{C}^{n\times n}$. Now we will use Taylor series for identifying G':

$$G(Y+P) = G(Y) + G'(Y)P + R(P), \tag{18.22}$$

where $P = (E_1, E_2)^T$, E_1 and E_2 are perturbation matrices, and R is such that

$$\lim_{\|P\|\to 0} \frac{\|R(P)\|}{\|P\|} = 0.$$

We have that

$$\begin{aligned} G\begin{pmatrix} W+E_1 \\ Z+E_2 \end{pmatrix} &= \begin{pmatrix} ((Z+E_2)+(W+E_1)) - (Z+E_2)A(W+E_1) \\ W+E_1 \end{pmatrix} \\ &= \begin{pmatrix} (Z+W-ZAW) + (E_1+E_2-ZAE_1-E_2AW) - E_2AE_1 \\ W+E_1 \end{pmatrix}. \end{aligned} \tag{18.23}$$

Comparing Eqs. (18.22) and (18.23) we conclude that

$$G'(Y)P = \begin{pmatrix} E_1+E_2-ZAE_1-E_2AW \\ E_1 \end{pmatrix}. \tag{18.24}$$

When $Y = Y_* = (A^{-1}, A^{-1})^T$ from (18.24) we obtain that

$$G'(Y_*)P = \begin{pmatrix} 0 \\ E_1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ I & 0 \end{pmatrix}\begin{pmatrix} E_1 \\ E_2 \end{pmatrix}.$$

Therefore, $G'(Y_*)$ is an idempotent matrix, and the iteration is stable. □

We now present a set of experiments to compute the inverse of a given matrix using the secant-Schulz iterative method. We choose as initial guesses $X_{-1} = \alpha A^T/\|A\|_2^2$ and $X_0 = \beta A^T/\|A\|_2^2$, with $\alpha, \beta \in (0, 1]$. When A is symmetric and positive definite we can also choose $X_{-1} = \alpha I$ and $X_0 = \beta I$ with $\alpha > 0$ and $\beta > 0$. For these initial choices global convergence can be established. Indeed, from (18.12) we obtain that

$$\|I - AX_{k+1}\|_2 \le \|I - AX_{-1}\|_2^{\alpha_k} \|I - AX_0\|_2^{\beta_k} \tag{18.25}$$

where $\alpha_k > 0$ and $\beta_k > 0$. Note that the matrices $I - AX_{-1}$ and $I - AX_0$ are symmetric, and for this case (18.25) can be written as

$$\|I - AX_{k+1}\|_2 \le \rho(I - AX_{-1})^{\alpha_k} \rho(I - AX_0)^{\beta_k}$$

where $\rho(B)$ represents the spectral radius of the matrix B. Finally using similar arguments to the ones used to prove the global convergence of Schulz method [7, 17], we can prove that $\rho(I - AX_{-1}) < 1$ and $\rho(I - AX_0) < 1$.

In our implementation we stop all considered algorithms when

$$\|X_k - X_*\|/\|X_*\| \leq 0.5D - 14.$$

All experiments were run on a Pentium Centrino Duo, 2.0 GHz, using Matlab 7. We report the number of required iterations (Iter) and the relative error ($\|X_k - X_*\|/\|X_*\|$) when the process is stopped. For our first experiment we consider the symmetric and positive definite matrix `poisson` from the Matlab gallery with $n = 400$. We compare the performance of the secant-Schulz method with the Newton-Schulz method described in (18.8). For the secant-Schulz method we choose $X_{-1} = 0.5 * I$, and $X_0 = A^T/\|A\|_2^2$, and for the Newton-Schulz we choose the same X_0. We report the results in Table 18.1, and the semilog of the relative error in Fig. 18.1.

For our second experiment we consider the nonsymmetric matrix `grcar` from the Matlab gallery with $n = 200$. For the secant-Schulz method we choose

Table 18.1 Performance of secant-Schulz and Newton-Schulz for finding the inverse of $A =$ `gallery('poisson',20)` when $n = 400$, $X_{-1} = 0.5 * I$, and $X_0 = A^T/\|A\|_2^2$

Method	Iter	$\|X_k - X_*\|/\|X_*\|$
Secant-Schulz	18	1.95e−15
Newton-Schulz	22	1.87e−15

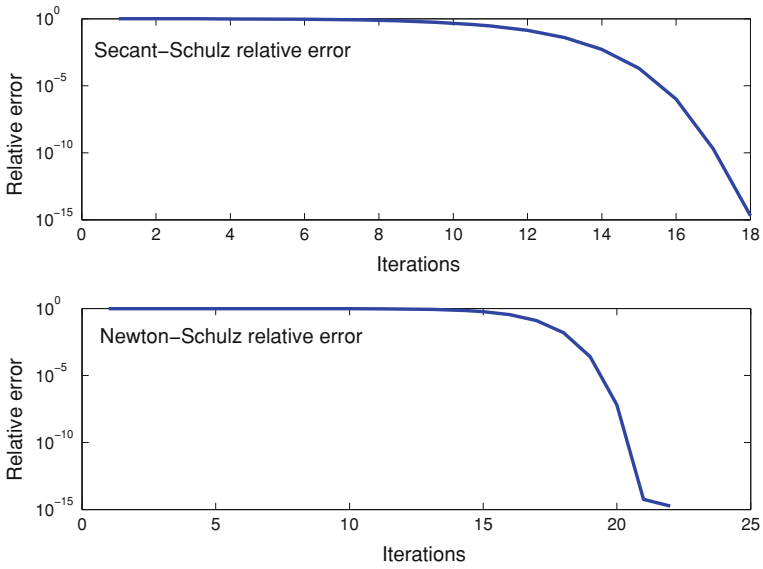


Fig. 18.1 Semilog of the relative error for finding the inverse of $A =$ `gallery('poisson',20)` when $n = 400, X_{-1} = 0.5 * I$, and $X_0 = A^T/\|A\|_2^2$

Table 18.2 Performance of secant-Schulz and Newton-Schulz for finding the inverse of $A =$ `gallery('grcar',200)` with $X_{-1} = 0.2 * A^T/\|A\|_2^2$ and $X_0 = A^T/\|A\|_2^2$

Method	Iter	$\|X_k - X_*\|/\|X_*\|$
Secant-Schulz	14	2.69e−15
Newton-Schulz	10	4.32e−16

Table 18.3 Performance of secant-Schulz and Newton-Schulz for finding the pseudo-inverse of $A =$ `gallery('cycol',n,8)` with $X_{-1} = 0.2 * A^T/\|A\|_2^2$ and $X_0 = A^T/\|A\|_2^2$

Method	Iter	$\|X_k - X_*\|/\|X_*\|$
Secant-Schulz	9	1.85e−15
Newton-Schulz	8	1.86e−15

$X_{-1} = 0.2 * A^T/\|A\|_2^2$, and $X_0 = A^T/\|A\|_2^2$, whereas for the Newton-Schulz we choose $X_0 = A^T/\|A\|_2^2$. We compare the performance of the secant-Schulz method with the Newton-Schulz. We report the results in Table 18.2.

As in the Newton-Schulz method, the secant-Schulz method also converges to the pseudo-inverse of any given matrix. For our next experiment we consider the rectangular matrix `cycol` from the Matlab gallery with $n = [100 \;\; 10]$ to compute its pseudoinverse, starting from the same initial choices of the previous experiment. We report the results in Table 18.3.

In all the experiments we observe the typical q-superlinear behavior of the proposed secant-Schulz method as compared with the q-quadratic behavior associated with the Newton-Schulz method.

18.4 Quadratic Matrix Equation

We now consider the application of Algorithms 2 and 3 for solving quadratic matrix equations of the form $AX^2 + BX + C = 0$, where A, B, and C are $n \times n$ matrices. For a recent globalized implementation of Newton's method see [10]. For our secant algorithms we set $F(X) = AX^2 + BX + C$ and seek roots of F. For this special case, the general secant algorithm can be simplified as follows:

Algorithm 4: **Secant method for quadratic problems**

Given $X_{-1} \in \mathbb{C}^{n\times n}$, $X_0 \in \mathbb{C}^{n\times n}$
Set $S_{-1} = X_0 - X_{-1}$
Solve $W_0 S_{-1} = A(X_0^2 - X_{-1}^2)$ /*for W_0*/
Set $A_0 = W_0 + B$
For $k = 0, 1, \cdots$ until convergence
Solve $A_k S_k = -F(X_k)$ /*for S_k*/
Set $X_{k+1} = X_k + S_k$
Solve $W_{k+1} S_k = A(X_{k+1}^2 - X_k^2)$ /*for W_{k+1}*/
Set $A_{k+1} = W_{k+1} + B$
End For

and the inverse version of the secant algorithm can be written as follows:

Algorithm 5: Inverse secant method for quadratic problems
Given $X_{-1} \in \mathbb{C}^{n\times n}$, $X_0 \in \mathbb{C}^{n\times n}$
Set $S_{-1} = X_0 - X_{-1}$
Set $Y_{-1} = A(X_0^2 - X_{-1}^2) + B(X_0 - X_{-1})$
Solve $B_0 Y_{-1} = S_{-1}$ /*for B_0*/
For $k = 0, 1, \cdots$ until convergence
Set $S_k = -B_k F(X_k)$
Set $X_{k+1} = X_k + S_k$
Set $Y_k = A(X_{k+1}^2 - X_k^2) + BS_k$
Solve $B_{k+1} Y_k = S_k$ /*for B_{k+1}*/
End For

We now present some experiments to illustrate the advantages of using Algorithms 4 and 5 for solving quadratic matrix equations. For that we choose two examples already studied and described in [9, 10]. We choose as initial guesses $X_{-1} = 0.1I$ and $X_0 = \beta I$, as in [2] for Newton's method, where $\beta = \left(\|B\|_F + \sqrt{\|B\|_F^2 + 4\|A\|_F\|C\|_F} \right) / (2\|A\|_F)$.

In our implementation we stop the algorithms when $\mathrm{Res}(X_k) \le n * \mathrm{eps}$, where $\mathrm{Res}(X_k) = \|F(X_k)\|_F / (\|A\|_F \|X_k\|_F^2 + \|B\|_F \|X_k\|_F + \|C\|_F)$ and $\mathrm{eps} = 2.2D - 16$. This stopping criterion is also suggested in [10]. These experiments were also run on a Pentium Centrino Duo, 2.0 GHz, using Matlab 7. We report the number of required iterations (Iter) and the value of $\mathrm{Res}(X_k)$ when the process is stopped. For our first experiment we consider the problem described by the following matrices $A = I$,

$$B = \begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix}, C = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \tag{18.26}$$

Problem (18.26) has a solvent at $X_* = I$.

Table 18.4 Performance of secant and inverse secant for solving problem (18.26)

X_0	Method	Iter	$\mathrm{Res}(X_k)$
βI	Secant	10	4.15e−17
βI	Inverse secant	11	2.22e−17
$10I$	Secant	13	2.22e−17
$10I$	Inverse secant	14	3.14e−17
$10^5 I$	Secant	15	1.57e−17
$10^5 I$	Inverse secant	16	5.02e−19
$10^{10} I$	Secant	15	2.74e−19
$10^{10} I$	Inverse secant	16	2.22e−17

Table 18.5 Performance of secant and inverse secant for solving our second quadratic experiment

X_0	Method	Iter	Res(X_k)
βI	Secant	12	1.62e−14
βI	Inverse secant	18	9.93e−15
$10^2 I$	Secant	15	3.76e−15
$10^2 I$	Inverse secant	18	1.23e−14
$10^5 I$	Secant	17	1.92e−15
$10^5 I$	Inverse secant	17	2.2e−14
$10^{10} I$	Secant	18	1.71e−15
$10^{10} I$	Inverse secant	16	7.55e−15
$10^{20} I$	Secant	15	1.62e−14
$10^{20} I$	Inverse secant	17	2.05e−14

We compare the performance of the direct secant method (Algorithm 3) with the inverse secant method (Algorithm 3). We report the results in Table 18.4.

For our second experiment we consider the problem described in [9] which is given by the following matrices $A = I, B = \text{tridiag}[-10, 30, -10]$ except $B(1,1) = B(n,n) = 20$, and $C = \text{tridiag}[-5, 15, -5]$, for $n = 100$. We compare the performance of the direct secant method (Algorithm 4) with the inverse secant method (Algorithm 5). We report the results in Table 18.5 and Fig. 18.2. We observe in both experiments that the secant algorithms show a robust

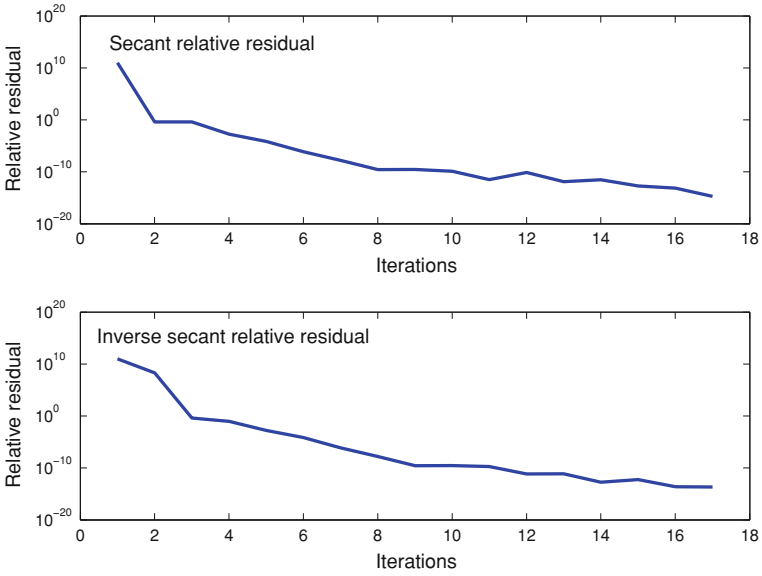


Fig. 18.2 Semilog of the relative residual for solving our second quadratic experiment when $n = 100$ and $X_0 = 10^5 I$

behavior converging from initial guesses either close or far away from the solution. In contrast, as reported in [10], Newton's method requires an exact line search globalization strategy to avoid the increase in number of iterations for convergence.

18.5 Conclusions and Perspectives

Whenever a Newton's method is applicable to a general nonlinear problem, a suitable secant method should be obtained for the same problem. In this work we present an interpretation of the classical secant method for solving nonlinear matrix problems. In the special case of computing the inverse of a given matrix, we present and fully analyze a specialized version, the secant-Schulz method, that resembles the well-known Schulz method which is a specialized version of Newton's method.

For solving quadratic matrix problems, we explore the use of the direct and also the inverse secant method. Our preliminary numerical experiments show the expected q-superlinear convergence, and indicate that these secant schemes seems to have interesting properties that remain to be established.

Finally, we hope that our specialized secant methods, for solving some simple cases, stimulate further extensions and research for solving additional and more complicated nonlinear matrix problems.

Acknowledgements Marlliny Monsalve was supported by the Scientific Computing Center at UCV, and CDCH-UCV project 03.00.6640.2008; and Marcos Raydan was partially supported by USB, the Scientific Computing Center at UCV, and CDCH-UCV project 03.00.6640.2008.

References

1. Datta B (2003) Numerical methods for linear control systems design and analysis. Elsevier, New York
2. Davis GJ (1981) Numerical solution of a quadratic matrix equation. SIAM J Sci Stat Comput 2:164–175
3. Dennis JE, Moré JJ (1974) A characterization of superlinear convergence and its application to quasi-Newton methods. Math Comp 28:549–560
4. Dennis JE, Schnabel R (1996) Numerical methods for unconstrained optimization and nonlinear equations, classics in applied mathematics. SIAM Publications, Philadelphia
5. Gao YH (2006) Newton's method for the quadratic matrix equation. Appl Math Comput 2:1772–1779
6. Gomes-Ruggiero MA, Martínez JM (1992) The column-updating method for solving nonlinear equations in Hilbert space. RAIRO Math Model Numer Anal 26:309–330
7. Héron B, Issard-Roch F, Picard C (1999) Analyse numérique, Dunod
8. Higham NJ (2008) Functions of matrices: theory and computation. SIAM, Philadelphia
9. Higham NJ, Kim HM (2000) Numerical analysis of a quadratic matrix equation. IMA J Numer Anal 20:499–519

10. Higham NJ, Kim HM (2001) Solving a quadratic matrix equation by Newton's method with exact line searches. SIAM J Matrix Anal 23:303–316
11. Horn RA, Johnson CR (1991) Topics in matrix analysis. Cambridge University Press, Cambridge
12. Martínez JM (1990) A family of quasi-Newton methods with direct secant updates of matrix factorizations. SIAM J Numer Anal 27:1034–1049
13. Pan V, Schreiber R (1991) An improved Newton iteration for the generalized inverse of a matrix, with applications. SIAM J Sci Stat Comput 12:1109–1130
14. Sachs E (1986) Broyden's method in Hilbert space. Math Program 35:71–81
15. Schulz G (1933) Iterative berechnung del reziproken matrix. Z Angew Math Mech 13:57–59
16. Smith MI (2003) A Schur algorithm for computing matrix *p*th roots. SIAM J Matrix Anal Appl 24:971–989
17. Söderström T, Stewart GW (1974) On the numerical properties of an iterative method for computing the Moore-Penrose generalized inverse. SIAM J Numer Anal 11:61–74

Chapter 19
On FastICA Algorithms and Some Generalisations

Hao Shen, Knut Hüper and Martin Kleinsteuber

Abstract The FastICA algorithm, a classical method for solving the one-unit linear ICA problem, and its generalisations are studied. Two interpretations of FastICA are provided, a scalar shifted algorithm and an approximate Newton method. Based on these two interpretations, two natural generalisations of FastICA on a full matrix are proposed to solve the parallel linear ICA problem. Specifically, these are a matrix shifted parallel ICA method and an approximate Newton-like parallel ICA method.

19.1 Introduction

In recent years, there has been an increasing interest in applying numerical linear algebra tools to either analyse existing methods or develop new methods in signal processing. In this paper, we present our recent results in analysing and generalising a classic algorithm for doing linear Independent Component Analysis (ICA), which is now a standard statistical tool for the problem of Blind Source Separation

H. Shen (✉) · M. Kleinsteuber
Geometric Optimization and Machine Learning Group, Technische Universität München, Munich, Germany
e-mail: hao.shen@tum.de

M. Kleinsteuber
e-mail: kleinsteuber@tum.de

K. Hüper
Department of Mathematics, Julius-Maximilians-Universität Würzburg, Würzburg, Germany
e-mail: hueper@mathematik.uni-wuerzburg.de

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_19,

(BSS), a challenging problem in signal processing. The tools we utilise in our analysis are similar to the techniques in analysing the Rayleigh quotient iteration (RQI), which is a well known method to the numerical linear algebra community for computing an eigenvalue eigenvector pair of a real symmetric matrix.

Since the seminal paper by Comon [6], many ICA algorithms have been developed by researchers from various communities. The FastICA algorithm, proposed by the Finnish school, is a classic linear ICA algorithm with both good accuracy and fast speed of convergence, see [12]. It solves the so-called one-unit linear ICA problem, which extracts one signal per pass. Originally, FastICA was developed as a Newton type method using a Lagrange multiplier approach together with a heuristic approximation of Hessians.

The first contribution of this work is to give two new rigorous interpretations of FastICA. First of all, FastICA can be easily considered as a scalar shifted version of a simpler linear ICA algorithm. It can be shown that the scalar shift strategy utilised in FastICA accelerates the simpler algorithm to achieve local quadratic convergence. Alternatively, by using geometric optimisation techniques, we develop an approximate Newton one-unit linear ICA method with a sensible approximation of Hessians, which has a rigorous justification by statistical features of the linear ICA problem. It can be shown that FastICA is indeed a special case of our proposed approximate Newton ICA method.

Due to the great success of FastICA, a natural question one may raise is whether it is possible to generalise FastICA to solve the problem of extracting all sources in parallel. On one hand, by generalising the concept of a scalar shift strategy to a matrix shift strategy, we develop a matrix shifted parallel linear ICA algorithm, which is indeed a natural generalisation of FastICA to a full matrix. On the other hand, following the same idea of approximating Hessians as in developing the approximate Newton one-unit linear ICA method, we formulate an approximate Newton-like parallel linear ICA method, which shares the significant property of local quadratic convergence, in this case to a correct separation of all sources.

This paper is organised as follows. In Sect. 19.2, we give a brief statement of the linear ICA problem. In Sect. 19.3, the FastICA algorithm is interpreted as special cases of a scalar shifted one-unit linear ICA algorithm and an approximate Newton one-unit linear ICA method. As natural generalisations of FastICA to a full matrix, Sect. 19.4 presents two parallel linear ICA methods in the frameworks of matrix shifted algorithm and approximate Newton-type method. Performance of all presented linear ICA methods is demonstrated and compared by several numerical experiments in Sect. 19.5. Finally, a conclusion is made in Sect. 19.6.

19.2 Problem Statement

In the literature, blind source separation (BSS) refers to the problem of recovering signals only from linear mixtures of sources, in the absence of prior information about either the sources or the mixing process. It has enormous applications in

bioinformatics, telecommunications, speech recognition systems, and so on. We refer to [5, 14, 20] and references therein for further details.

A noiseless linear BSS model is generally formulated as follows, refer to [12] for more details,

$$z = As, \tag{19.1}$$

where $s = [s_1, \ldots, s_m]^\top \in \mathbb{R}^m$ denotes an m-dimensional random vector representing m source signals, $A \in \mathbb{R}^{m\times m}$ is the mixing matrix of full rank and $z \in \mathbb{R}^m$ represents m observed linear mixtures. The task of linear BSS problem (19.1) is to recover the source signals s by estimating the mixing matrix A or its inverse A^{-1} based only on the observations z via the demixing model

$$y = Bz, \tag{19.2}$$

where $B \in \mathbb{R}^{m\times m}$ is the demixing matrix, an estimation of A^{-1} and $y \in \mathbb{R}^m$ represents the corresponding estimated source signals.

It is clear that, without further constraints, there are infinitely many possibilities of B for the demixing BSS model (19.2). Therefore, to allow a solution of the BSS problem, one would have to impose certain prior assumptions about either the mixing process or the features of sources. One most common assumption is the concept of *statistical independence*, i.e.,

Assumption 1 All individual components of the sources $s \in \mathbb{R}^m$ as in model (19.1) are *mutually statistically independent*.

It then leads to the standard linear ICA approach for solving the linear BSS problem (19.1), refer to [6] for details.

It is obvious that, for the linear model (19.1), any scaling factors of both the columns of A and the components of the sources s are indeed interchangeable, i.e., the sign and amplitude of each source cannot be uniquely identified. Moreover, the mean value of each source signal is irrelevant to the concept of statistical independence. Therefore, without loss of generality, each component of the sources s can be assumed to have *zero mean* and *unit variance*.

In practice, the observations z can always be whitened to simplify the problem further. Usually by using principal component analysis (PCA), one computes a matrix $C \in \mathbb{R}^{m\times m}$ such that

$$w = Cz = CAs, \quad \text{where } \mathbb{E}\left[ww^\top\right] = I_m, \tag{19.3}$$

i.e., all components of the whitened mixtures w are *uncorrelated* and each component of w has *zero mean* and *unit variance* as well. By denoting the product $CA =: V \in \mathbb{R}^{m\times m}$ in (19.3), one observes

$$\begin{aligned} \mathbb{E}\left[ww^\top\right] &= V\mathbb{E}\left[ss^\top\right]V^\top \\ &\Longleftrightarrow \\ VV^\top &= I_m, \end{aligned} \tag{19.4}$$

i.e., $V \in \mathbb{R}^{m\times m}$ is indeed an orthogonal matrix. The linear relation (19.3) is often referred to as the *whitened linear ICA model*. Let $O(m)$ denote the *orthogonal group*, i.e.,

$$O(m) := \{X \in \mathbb{R}^{m\times m} | \, X^\top X = I_m\}. \tag{19.5}$$

The *whitened linear ICA demixing model* is simply stated as follows

$$y = X^\top w, \tag{19.6}$$

where $X \in O(m)$ is called the demixing matrix as an estimation of $V \in O(m)$ and y now represents an estimation of the whitened sources s in (19.3).

In some applications, one might prefer to extract only one desired source, rather than all sources. Let $X = [x_1, \ldots, x_m] \in O(m)$. Then a single source can be recovered by simply the following

$$y_i = x_i^\top w. \tag{19.7}$$

We refer to such a problem as the *one-unit linear ICA problem*, while we refer to the *parallel linear ICA problem* for problems of simultaneous extraction of all sources in the form of (19.6).

Now let $V = [v_1, \ldots, v_m] \in O(m)$ be the mixing matrix in the whitened mixing ICA model (19.3). Following Theorem 11 in [6], all correct solutions of the parallel linear ICA problem (19.6) are given by

$$\Theta := \{VDP \in O(m)\}, \tag{19.8}$$

where $D = \mathrm{diag}(\varepsilon_1, \ldots, \varepsilon_m)$ with $\varepsilon_i \in \{\pm 1\}$ and $P = \{e_{\tau(1)}, \ldots, e_{\tau(m)}\} \in O(m)$ a permutation matrix with $\tau: \{1, \ldots, m\} \to \{1, \ldots, m\}$ being a permutation. Straightforwardly, all correct solutions for the one-unit linear ICA problem (19.7) form the set

$$\Upsilon := \{\varepsilon v_i \,|\, V = [v_1, \ldots, v_m] \in O(m) \quad \text{and}\, \varepsilon \in \{\pm 1\}\}. \tag{19.9}$$

One popular category of ICA methods is the so-called *contrast-based ICA method*. A general scheme of such methods involves an optimisation procedure of a *contrast function*, which measures the statistical independence between recovered signals. Usually, correct separations of sources are expected to be obtained at certain optimal points of contrast functions. A simple approach to designing ICA contrasts is to construct their statistical independence measure by using some parameterised families of functions, in accordance with certain hypothesis about the distributions (probability density functions) of sources. The corresponding ICA methods are usually referred to as *parametric ICA methods*.

Let us define the $m-1$ dimensional unit sphere by

$$S^{m-1} := \{x \in \mathbb{R}^m | \, \|x\| = 1\}. \tag{19.10}$$

A generic parametric one-unit linear ICA contrast for the model (19.7) is usually formulated as follows

$$f: S^{m-1} \to \mathbb{R}, \quad f(x) := \mathbb{E}[G(x^\top w)], \tag{19.11}$$

where $\mathbb{E}[\cdot]$ denotes the expectation over the observation w and the function $G: \mathbb{R} \to \mathbb{R}$ is a smooth non-linear function, which is chosen according to the specific application. Additionally, G is often assumed to be *even* [12]. We stick to this assumption throughout this paper. Simply, for the parallel linear ICA problem (19.6), the corresponding parametric contrast function is given as follows, cf. [8],

$$F: O(m) \to \mathbb{R}, \quad F(X) := \sum_{i=1}^{m} \mathbb{E}[G(x_i^\top w)]. \tag{19.12}$$

The performance of corresponding linear ICA methods developed via optimising the above contrast functions f and F is significantly dependent on choices of the function G and statistical properties of the source signals s.

Note that in this paper, all computations regarding the one-unit ICA problem, i.e., optimising the contrast function f, are performed using coordinate functions of $\mathbb{R}^m$, which is the embedding space of S^{m-1}. The tangent space $T_x S^{m-1}$ of S^{m-1} at $x \in S^{m-1}$ is given by

$$T_x S^{m-1} = \left\{\xi \in \mathbb{R}^m | \, x^\top \xi = 0\right\}. \tag{19.13}$$

For the parallel ICA problem (19.12), we consider the orthogonal group $O(m)$ as an embedded submanifold of $\mathbb{R}^{m\times m}$. The tangent space $T_X O(m)$ of $O(m)$ at $X \in O(m)$ is given by

$$T_X O(m) = \left\{\Xi \in \mathbb{R}^{m\times m} | \, X^\top \Xi + \Xi^\top X = 0\right\}. \tag{19.14}$$

Therefore, S^{m-1} and $T_x S^{m-1}$ are considered as submanifolds of the embedding space $\mathbb{R}^m$, similarly, $O(m) \subset \mathbb{R}^{m\times m}$ and $T_X O(m) \subset \mathbb{R}^{m\times m}$. For an introduction to differential geometry, we refer to [4, 26]. For an introduction to differential geometry in relation to optimisation we refer to [1].

19.3 Two Interpretations of FastICA

The original FastICA algorithm was developed as an approximate Newton method with a heuristic approximation of Hessians, which optimises the one-unit ICA contrast function f (19.11) by using a standard Lagrange multiplier approach [11].

Algorithm 1 The FastICA algorithm

Step 1: Given an initial guess $x^{(0)} \in S^{m-1}$ and set $k = 0$.
Step 2: Compute $x^{(k+1)} = \mathbb{E}[G'(x^{(k)\top} w)w] - \mathbb{E}[G''(x^{(k)\top} w)]x^{(k)}$.
Step 3: Update $x^{(k+1)} \leftarrow \frac{x^{(k+1)}}{\|x^{(k+1)}\|}$.
Step 4: If $\|x^{(k+1)} - x^{(k)}\|$ is small enough, stop.
Otherwise, set $k = k + 1$ and go to Step 2.

Here, G', G'' are the first and second derivatives of the nonlinear function G. Each iteration of FastICA can be considered as the map

$$\phi_{\mathrm{f}}: S^{m-1} \to S^{m-1}, \quad x \mapsto \frac{\mathbb{E}[G'(x^\top w)w] - \mathbb{E}[G''(x^\top w)]x}{\|\mathbb{E}[G'(x^\top w)w] - \mathbb{E}[G''(x^\top w)]x\|}. \tag{19.15}$$

In this section, we will investigate two interpretations of FastICA, i.e. the algorithmic map ϕ_{f} (19.15), in the framework of a scalar shift strategy and an approximate Newton method.

19.3.1 Critical Point Analysis of the One-Unit ICA Contrast

We start with the analysis of the one-unit ICA contrast function f, which plays an important role in the subsequent development and analysis.

First of all, recall the definition of a great circle γ_x of S^{m-1} at $x \in S^{m-1}$ as

$$\begin{aligned} \gamma_x: \mathbb{R} \to S^{m-1}, \quad \gamma_x(t) &:= \exp\big(t(\xi x^\top - x\xi^\top)\big)x \\ &= \begin{cases} x, & \|\xi\| = 0; \\ x\cos t\|\xi\| + \xi \frac{\sin t\|\xi\|}{\|\xi\|}, & \text{otherwise}, \end{cases} \end{aligned} \tag{19.16}$$

where $\xi \in T_x S^{m-1}$ and $\exp(\cdot)$ denotes matrix exponentiation. Obviously, $\gamma_x(0) = x$ and $\dot{\gamma}_x(0) = \xi$. As a matter of fact, great circles are geodesics on the unit sphere with respect to the Riemannian metric induced by the Euclidean metric of the embedding space $\mathbb{R}^m$. Now let us compute the first derivative of f

$$\mathrm{D}f(x): T_x S^{m-1} \to \mathbb{R}, \tag{19.17}$$

which assigns to an arbitrary tangent vector $\xi \in T_x S^{m-1}$ the value

$$\begin{aligned} \mathrm{D}f(x)\xi &= \frac{\mathrm{d}}{\mathrm{d}t}(f \circ \gamma_x)(t)\bigg|_{t=0} \\ &= \mathbb{E}[G'(x^\top w)\xi^\top w]. \end{aligned} \tag{19.18}$$

Thus, critical points of f can be characterised as solutions of

$$\xi^\top \cdot \mathbb{E}[G'(x^\top w)w] = 0 \tag{19.19}$$

for all $\xi \in T_x S^{m-1}$. This is further equivalent to saying, that

$$\mathbb{E}[G'(x^\top w)w] = \lambda x, \tag{19.20}$$

with $\lambda \in \mathbb{R}$.

It is worthwhile to notice that the critical point condition for f depends significantly on both the function G and the statistical features of the sources s. It therefore appears to be hardly possible to fully characterise all critical points respecting all, possibly unknown, statistical features. Here we at least show that any $x^* \in \Upsilon$, which corresponds to a correct separation of a single source, fulfills the critical point condition (19.19).

Let us denote the left-hand side of (19.20) as follows

$$k\colon S^{m-1} \to \mathbb{R}^m, \quad k(x) := \mathbb{E}[G'(x^\top w)w]. \tag{19.21}$$

For any $x^* = \varepsilon v_i \in \Upsilon$, one computes

$$\begin{aligned} k(x^*) &= \mathbb{E}[G'(\varepsilon v_i^\top Vs)Vs] \\ &= V\mathbb{E}[G'(\varepsilon s_i)s] \\ &= \varepsilon V\mathbb{E}[G'(s_i)s], \end{aligned} \tag{19.22}$$

following the fact that G is even, i.e., G' is odd. Using the mutual statistical independence and centering property (zero mean) of sources, the entries of the expression $\mathbb{E}[G'(s_i)s]$ are computed as

$$\mathbb{E}[G'(s_i)s_j] = \begin{cases} \mathbb{E}[G'(s_i)s_i], & j = i, \\ \mathbb{E}[G'(s_i)s_j] = \mathbb{E}[G'(s_i)]\mathbb{E}[s_j] = 0, & j \neq i, \end{cases} \tag{19.23}$$

i.e.,

$$\mathbb{E}[G'(s_i)s] = \mathbb{E}[G'(s_i)s_i]e_i, \tag{19.24}$$

where e_i denotes the i-th standard basis vector of $\mathbb{R}^m$. It is then clear that the critical point condition for f (19.19) holds true at any $x^* \in \Upsilon$, i.e.

$$k(x^*) = \mathbb{E}[G'(s_i)s_i]x^*. \tag{19.25}$$

Note, that there might exist more critical points of the contrast f, which do not correspond to a correct separation of sources.

Now, we compute the Riemannian Hessian of f, i.e. the symmetric bilinear form

$$\mathcal{H}f(x)\colon T_x S^{m-1} \times T_x S^{m-1} \to \mathbb{R}, \tag{19.26}$$

given by

$$\begin{aligned} \mathcal{H}f(x)(\xi,\xi) &= \frac{\mathrm{d}^2}{\mathrm{d}t^2}(f \circ \gamma_x)(t)\Big|_{t=0} \\ &= \xi^\top \big(\mathbb{E}[G''(x^\top w)ww^\top] - \mathbb{E}[G'(x^\top w)(x^\top w)]I_m\big)\xi. \end{aligned} \tag{19.27}$$

Note that $\gamma_x(t)$ is a geodesic through x as defined in (19.16). Let us denote the first summand of (19.27) by

$$H\colon S^{m-1} \to \mathbb{R}^{m\times m}, \quad H(x) := \mathbb{E}[G''(x^\top w)ww^\top]. \tag{19.28}$$

For $x^* \in \Upsilon$, we get

$$\begin{aligned} H(x^*) &= V\mathbb{E}[G''(\varepsilon s_i)ss^\top]V^\top \\ &= V\mathbb{E}[G''(s_i)ss^\top]V^\top, \end{aligned} \tag{19.29}$$

by the fact of G'' being even. Again, by applying mutual statistical independence and whitening properties of the sources, the (p,q)-th entry, for all $p,q = 1,\ldots,m$, of the expression $\mathbb{E}[G''(s_i)ss^\top]$ can be computed as

(i) if $p = q \neq i$, then

$$\begin{aligned} \mathbb{E}[G''(s_i)s_ps_q] &= \mathbb{E}[G''(s_i)s_p^2] \\ &= \mathbb{E}[G''(s_i)]\mathbb{E}[s_p^2] \\ &= \mathbb{E}[G''(s_i)]; \end{aligned} \tag{19.30}$$

(ii) if $p = q = i$, then

$$\mathbb{E}[G''(s_i)s_ps_q] = \mathbb{E}[G''(s_i)s_i^2]; \tag{19.31}$$

(iii) if $p \neq q$, without loss of generality, we can assume that $q \neq i$. Then

$$\begin{aligned} \mathbb{E}[G''(s_i)s_ps_q] &= \mathbb{E}[G''(s_i)s_p]\mathbb{E}[s_q] \\ &= 0. \end{aligned} \tag{19.32}$$

Thus the expression $\mathbb{E}[G''(s_i)ss^\top]$ is indeed a diagonal matrix with the i-th diagonal entry equal to $\mathbb{E}[G''(s_i)s_i^2]$ and all others being equal to $\mathbb{E}[G''(s_i)]$. A direct calculation leads to

$$H(x^*) = \mathbb{E}[G''(s_i)]I_m + \big(\mathbb{E}[G''(s_i)s_i^2] - \mathbb{E}[G''(s_i)]\big)(x^*x^{*\top}). \tag{19.33}$$

Hence, we compute

$$\begin{aligned} \frac{\mathrm{d}^2}{\mathrm{d}t^2}(f\circ\gamma_{x^*})(t)\bigg|_{t=0} &= \xi^\top H(x^*)\xi - \mathbb{E}[G'(\varepsilon v_i^\top w)(\varepsilon v_i^\top w)]\xi^\top\xi \\ &= \mathbb{E}[G''(s_i)]\xi^\top\xi - \mathbb{E}[G'(\varepsilon s_i)(\varepsilon s_i)]\xi^\top\xi \\ &= (\mathbb{E}[G''(s_i)] - \mathbb{E}[G'(s_i)(s_i)])\xi^\top\xi, \end{aligned} \tag{19.34}$$

i.e., the Hessian $\mathcal{H}f(x)$ of f at the critical point x^* acts on a tangent vector ξ simply by scalar multiplication

$$\mathcal{H}f(x^*)\xi = (\mathbb{E}[G''(s_i)] - \mathbb{E}[G'(s_i)s_i])\xi. \tag{19.35}$$

From a statistical point of view, refer to Theorem 1 in [11], it is usually assumed that

Assumption 2 The nonlinear function $G\colon \mathbb{R} \to \mathbb{R}$ is even and chosen such that the following inequality holds true for all sources in the parallel linear ICA problem (19.3), i.e.,

$$\mathbb{E}[G''(s_i)] - \mathbb{E}[G'(s_i)s_i] \neq 0, \tag{19.36}$$

for all $i = 1, \ldots, m$.

Thus, the inverse of the Hessian of f can be ensured to exist at any $x^* \in \Upsilon$. We then conclude

Corollary 1.1 *Let $\Upsilon \subset S^{m-1}$, defined as in (19.9), be the set of solutions of the one-unit linear ICA problem (19.7) and $G\colon \mathbb{R} \to \mathbb{R}$ a smooth function satisfying Assumption 2. Then any $x^* \in \Upsilon$ is a non-degenerated critical point of the one-unit linear ICA contrast f.*

19.3.2 FastICA as a Scalar Shifted Algorithm

In this subsection, we will generalise the FastICA algorithm ϕ_{f} (19.15) in the framework of a scalar shift strategy. The scalar shift strategy utilised in FastICA accelerates a simpler one-unit linear ICA algorithm to converge locally quadratically fast to a correct separation.

Now, let $x^* = \varepsilon v_i \in \Upsilon$, i.e. x^* correctly recovers the i-th source signal s_i. Then following Eq. 19.15, we get

$$\phi_{\mathrm{f}}(x^*) = \operatorname{sign}\big(\mathbb{E}[G'(s_i)s_i] - \mathbb{E}[G''(s_i)]\big)x^*. \tag{19.37}$$

It is easily seen that, if $\mathbb{E}[G'(s_i)s_i] - \mathbb{E}[G''(s_i)] < 0$, i.e., $\phi_{\mathrm{f}}(x^*) = -x^*$ and $\phi_{\mathrm{f}}(-x^*) = x^*$, the FastICA algorithm oscillates between neighborhoods of two antipodes $\pm x^* \in S^{m-1}$, both of which recover the same single source up to sign. A closer look at the FastICA map ϕ_{f} (19.15) suggests that the second term, i.e. the expression $\mathbb{E}[G''(x^\top w)]$, in the numerator of ϕ_{f} can be treated as a scalar shift, i.e., $x \mapsto \mathbb{E}[G''(x^\top w)]x$. In other words, the FastICA map ϕ_{f} is simply a scalar shifted version of the following one-unit ICA algorithmic map proposed in [19]

$$\phi\colon S^{m-1} \to S^{m-1}, \quad x \mapsto \frac{\mathbb{E}[G'(x^\top w)w]}{\|\mathbb{E}[G'(x^\top w)w]\|}. \tag{19.38}$$

Note that, any $x^* \in \Upsilon$ is a fixed point of ϕ simply because of Eq. 19.25. By replacing the scalar term $\mathbb{E}[G''(x^\top w)]$ in (19.15) by another smooth real-valued function $\rho\colon S^{m-1} \to \mathbb{R}$, we construct a more general scalar shifted version of the simple ICA map ϕ as follows

$$\phi_{\mathrm{s}}: S^{m-1} \to S^{m-1}, \quad x \mapsto \frac{\mathbb{E}[G'(x^\top w)w] - \rho(x)x}{\|\mathbb{E}[G'(x^\top w)w] - \rho(x)x\|}. \tag{19.39}$$

Let us denote the numerator in (19.39) as

$$\pi: S^{m-1} \to \mathbb{R}^m, \quad \pi(x) := \mathbb{E}[G'(x^\top w)w] - \rho(x)x, \tag{19.40}$$

and let $x^* \in \Upsilon$, i.e, x^* is a correct separation point of the one-unit linear ICA problem (19.7). Following the result (19.25), one gets

$$\pi(x^*) = \big(\mathbb{E}[G'(s_i)s_i] - \rho(x^*)\big)x^*. \tag{19.41}$$

Clearly, the map ϕ_{s} will still suffer from the sign flipping phenomenon as the FastICA iteration does, when the expression $\mathbb{E}[G'(s_i)s_i] - \rho(x^*)$ is negative. This discontinuity of the map ϕ_{s} can be removed by considering the unique mapping on the real projective space $\mathbb{RP}^{m-1}$ induced by ϕ_{s}, refer to [23] for more details. In this paper, however, we take an alternative approach of introducing a proper sign correction term to tackle the discontinuity issue.

Let us define a scalar function by

$$\begin{aligned} \beta: S^{m-1} \to \mathbb{R}, \quad \beta(x) &:= x^\top \pi(x) \\ &= \mathbb{E}[G'(x^\top w)x^\top w] - \rho(x), \end{aligned} \tag{19.42}$$

which indicates the angle between two consecutive iterates produced by the map ϕ_{s}. We construct the following algorithmic map

$$\widetilde{\phi}_{\mathrm{s}}: S^{m-1} \to S^{m-1}, \quad x \mapsto \frac{\beta(x)\pi(x)}{\|\beta(x)\pi(x)\|}, \tag{19.43}$$

and further denote the numerator of $\widetilde{\phi}_{\mathrm{s}}$ by

$$\sigma: S^{m-1} \to \mathbb{R}^m, \quad \sigma(x) := \beta(x)\pi(x). \tag{19.44}$$

The equation (19.41) easily leads to

$$\sigma(x^*) = (\beta(x^*))^2 x^*, \tag{19.45}$$

thus $\widetilde{\phi}_{\mathrm{s}}(x^*) = x^*$. Obviously, to make the map $\widetilde{\phi}_{\mathrm{s}}$ (19.43) well defined around x^*, it requires that $\beta(x^*) \neq 0$. Thus we just showed

Corollary 1.2 *Let $\Upsilon \subset S^{m-1}$, defined as in (19.9), be the set of solutions of the one-unit linear ICA problem (19.7). Let the smooth scalar shift $\rho: S^{m-1} \to \mathbb{R}$ satisfy the condition*

$$\rho(x^*) \neq \mathbb{E}[G'(s_i)s_i], \tag{19.46}$$

for any $x^* = \varepsilon v_i \in \Upsilon$. Then each point $x^* \in \Upsilon$ is a fixed point the algorithmic map $\widetilde{\phi}_{\mathrm{s}}$.

In the following, we will investigate the conditions on the scalar shift ρ more closely, so that the resulting scalar shifted algorithm converges locally quadratically fast to a correct separation point. Later on we will give an example for ρ to show that such a scalar shift actually exists. According to Lemmas 2.7–2.9 in [15], we only need to show under which conditions on ρ the first derivative of $\widetilde{\phi}_{\mathrm{s}}$

$$\mathrm{D}\widetilde{\phi}_{\mathrm{s}}(x)\colon T_x S^{m-1} \to T_{\widetilde{\phi}_{\mathrm{s}}(x)} S^{m-1} \tag{19.47}$$

will vanish at x^*. Let $\|\sigma\|^2 = \sigma^\top \sigma$, one computes

$$\begin{aligned} \mathrm{D}\widetilde{\phi}_{\mathrm{s}}(x)\xi\Big|_{x=x^*} &= \frac{\mathrm{D}\sigma(\mathrm{x})\xi|_{x=x^*}}{\|\sigma(x^*)\|} - \frac{\sigma(x^*)\sigma^\top(x^*)\mathrm{D}\sigma(\mathrm{x})\xi|_{x=x^*}}{\|\sigma(x^*)\|^3} \\ &= \frac{1}{\|\sigma(x^*)\|} \underbrace{\left(I_m - \frac{\sigma(x^*)\sigma^\top(x^*)}{\|\sigma(x^*)\|^2}\right)}_{=:\Pi(x^*)} \mathrm{D}\sigma(x)\xi|_{x=x^*}. \end{aligned} \tag{19.48}$$

By Corollary 1.2, we know that $\widetilde{\phi}_{\mathrm{s}}(x^*) = x^*$. Therefore the expression $\Pi(x^*)$ is an orthogonal projection operator onto the complement of $\mathrm{span}(x^*)$. Consequently, the algorithmic mapping $\widetilde{\phi}_{\mathrm{s}}$ converges locally quadratically fast to x^*, if and only if the expression $\mathrm{D}\sigma(x)\xi|_{x=x^*}$ is equal to a scalar multiple of x^*. A simple computation shows

$$\mathrm{D}\sigma(x)\xi|_{x=x^*} = \mathrm{D}\beta(x)\xi\Big|_{x=x^*}\pi(x^*) + \beta(x^*)\mathrm{D}\pi(x)\xi\Big|_{x=x^*}. \tag{19.49}$$

Following the fact that $\mathrm{D}\beta(x)\xi\Big|_{x=x^*}$ is a scalar and using Eq. 19.41, the first summand in (19.49) is equal to a scalar multiple of x^*. Now we compute

$$\mathrm{D}\pi(x)\xi|_{x=x^*} = \mathrm{D}\mathbb{E}[G'(x^\top w)w]\xi\Big|_{x=x^*} - \rho(x^*)\xi - \mathrm{D}\rho(x)\xi|_{x=x^*}x^*. \tag{19.50}$$

Following Eq. 19.33, the first summand in (19.50) is computed as

$$\begin{aligned} \mathrm{D}k(x)\xi|_{x=x^*} &= \mathrm{D}\mathbb{E}[G'(x^\top w)w]\xi\Big|_{x=x^*} \\ &= \mathbb{E}[G''(x^{*\top} w)ww^\top]\xi \\ &= \mathbb{E}[G''(s_i)]\xi. \end{aligned} \tag{19.51}$$

Then, due to the fact that the third summand in (19.50) is already a scalar multiple of x^*, the expression $\mathrm{D}\pi(x)\xi|_{x=x^*}$, as well as $\mathrm{D}\sigma(x)\xi|_{x=x^*}$, are both equal to scalar multiples of x^*, if and only if the following equality holds true

$$\mathbb{E}[G''(s_i)]\xi - \rho(x^*)\xi = \lambda x^*, \tag{19.52}$$

where $\lambda \in \mathbb{R}$, i.e., $\rho(x^*) = \mathbb{E}[G''(s_i)]$. Thus the convergence properties of the map $\widetilde{\phi}_{\mathrm{s}}$ can be summarised in the following theorem.

Theorem 1.3 *Let* $\Upsilon \subset S^{m-1}$, *defined as in* (19.9), *be the set of solutions of the one-unit linear ICA problem* (19.7). *Let the smooth scalar shift* $\rho: S^{m-1} \to \mathbb{R}$ *satisfy the condition* (19.46). *Then the algorithmic map* $\widetilde{\phi}_{\mathrm{s}}$ *is locally quadratically convergent to* $x^* = \varepsilon v_i \in \Upsilon$ *if and only if*

$$\rho(x^*) = \mathbb{E}[G''(s_i)]. \tag{19.53}$$

Remark 1 Consequently, the scalar shift utilised in FastICA, i.e.,

$$\rho_{\mathrm{f}}: S^{m-1} \to \mathbb{R}, \quad \rho_{\mathrm{f}}(x) := \mathbb{E}[G''(x^\top w)] \tag{19.54}$$

is actually a simple choice of a scalar shift strategy, satisfying the condition (19.53). It therefore accelerates the map $\widetilde{\phi}_{\mathrm{s}}$ to converge locally quadratically fast to a correct separation.

19.3.3 FastICA as an Approximate Newton Algorithm

As pointed out before, the original FastICA algorithm was developed by using some heuristic approximation of the Hessian. In this subsection, we will develop an approximate Newton one-unit linear ICA method by using geometric optimisation techniques. The approximation of Hessians we employ here is rigorously justified using the statistical features of the linear ICA problem. FastICA is shown to be just a special case of our proposed method.

Let $S^{m-1} \subset \mathbb{R}^m$ be endowed with the Riemannian metric induced from the standard scalar product of the embedding $\mathbb{R}^m$. The geodesics with respect to this metric are precisely the great circles γ_x (19.16). Thus, according to Eq. 19.18, the Riemannian gradient of f at $x \in S^{m-1}$ can be computed as

$$\nabla f(x) = (I_m - xx^\top)\mathbb{E}[G'(x^\top w)w]. \tag{19.55}$$

Here, $I_m - xx^\top$ is the orthogonal projection operator onto the tangent space $T_x S^{m-1}$. Computing the second derivative of f along a great circle, a Newton direction $\xi \in T_x S^{m-1}$ can be computed by solving the following linear system

$$\begin{aligned} &\left(I_m - xx^\top\right)\left(\mathbb{E}[G''(x^\top w)ww^\top] - \mathbb{E}[G'(x^\top w)x^\top w]I_m\right)\xi \\ &\quad = \left(I_m - xx^\top\right)\mathbb{E}[G'(x^\top w)w]. \end{aligned} \tag{19.56}$$

Thus, a single iteration of a Riemannian Newton method for optimising the contrast f can be easily completed by projecting the above Newton step ξ back onto S^{m-1} along a great circle (19.16) uniquely defined by the direction $\xi \in T_x S^{m-1}$.

However, there exists a serious drawback of such a standard approach. The expression $\mathbb{E}[G''(x^\top w)ww^\top]$ at the left-hand side of (19.56) is an $m \times m$ dense matrix, often expensive to evaluate. In order to avoid computing the true Hessian,

one would prefer to have certain approximations of the true Hessian with low computational cost. Suggested by the property of the Hessian of f at critical points $x^* \in \Upsilon$ being scalar, it is sensible to approximate the expression $\mathbb{E}[G''(x^\top w)ww^\top] - \mathbb{E}[G'(x^\top w)x^\top w]I_m$ at *arbitrary* $x \in S^{m-1}$ within an open neighborhood $U_{x^*} \subset S^{m-1}$ around x^* by the following scalar matrix,

$$\big(\mathbb{E}[G''(x^\top w)] - \mathbb{E}[G'(x^\top w)x^\top w]\big)I_m, \tag{19.57}$$

which obviously gives the correct expression at x^*. Thus, an approximate Newton direction $\xi_a \in T_x S^{m-1}$ for optimising the one-unit ICA contrast f can be computed by solving the following linear system

$$\begin{aligned}&\big(I_m - xx^\top\big)\big(\mathbb{E}[G''(x^\top w)] - \mathbb{E}[G'(x^\top w)x^\top w]\big)\,\xi_a \\ &= \big(I_m - xx^\top\big)\mathbb{E}[G'(x^\top w)w].\end{aligned} \tag{19.58}$$

Note that by construction, $(\mathbb{E}[G''(x^\top w)] - \mathbb{E}[G'(x^\top w)x^\top w])\xi_a$ lies in the tangent space $T_x S^{m-1}$. The projection $I_m - xx^\top$ on the left-hand side of (19.58) is therefore redundant. Thus, the solution of (19.58) in terms of ξ_a is simply given by

$$\xi_a = -\frac{(I_m - xx^\top)\mathbb{E}[G'(x^\top w)w]}{\mathbb{E}[G''(x^\top w)] - \mathbb{E}[G'(x^\top w)x^\top w]} \in T_x S^{m-1}. \tag{19.59}$$

Although evaluating the great circle (19.16) is not too costly, we prefer to use the orthogonal projection instead, which costs slightly less computations than γ_x (19.16). Here, we specify the following curve on S^{m-1} through $x \in S^{m-1}$

$$\mu_x \colon (-\kappa, \kappa) \to S^{m-1}, \quad t \mapsto \frac{x + t\xi}{\|x + t\xi\|}, \tag{19.60}$$

with $\kappa > 0$ and $\xi \in T_x S^{m-1}$ arbitrary. Obviously, $\mu_x(0) = x$ and $\dot{\mu}_x(0) = \xi$. A similar idea of replacing geodesics (great circles) by certain smooth curves on manifolds has already been explored in [3, 10, 24]. By substituting the approximate Newton direction ξ_a into (19.60), we end up with an approximate Newton method for optimising the one-unit ICA contrast f. This leads to

$$\widetilde{\phi}_{\mathrm{n}} \colon S^{m-1} \to S^{m-1}, \quad x \mapsto \frac{\frac{1}{\eta(x)}(\mathbb{E}[G'(x^\top w)w] - \mathbb{E}[G''(x^\top w)]x)}{\|\frac{1}{\eta(x)}(\mathbb{E}[G'(x^\top w)w] - \mathbb{E}[G''(x^\top w)]x)\|}, \tag{19.61}$$

where

$$\eta \colon S^{m-1} \to \mathbb{R}, \quad \eta(x) := \mathbb{E}[G'(x^\top w)x^\top w] - \mathbb{E}[G''(x^\top w)]. \tag{19.62}$$

Remark 2 *If* $\eta(x) > 0$ holds always, it is easy to see that the map $\widetilde{\phi}_{\mathrm{n}}$ is actually the FastICA map ϕ_{f}.

Moreover, the map $\widetilde{\phi}_{\mathrm{n}}$ is identical to the scalar shifted ICA map $\widetilde{\phi}_{\mathrm{s}}$ (19.43) when employing the FastICA shift ρ_{f} (19.54). Straightforwardly following

Theorem 1.3 and Remark 1, the local convergence properties of the algorithmic map $\widetilde{\phi}_{\mathrm{n}}$ can be summarised as follows

Corollary 1.4 *Let* $\Upsilon \subset S^{m-1}$, *defined as in* (19.9), *be the set of solutions of the one-unit linear ICA problem* (19.7) *and* $G\colon \mathbb{R} \to \mathbb{R}$ *a smooth function satisfying the condition* (19.36). *Then the algorithmic map* $\widetilde{\phi}_{\mathrm{n}}$ *is locally quadratically convergent to* $x^* \in \Upsilon$.

19.4 Generalised FastICA for Parallel Linear ICA

In Sect. 19.3, FastICA is reinterpreted as special cases of a scalar shifted algorithm and an approximate Newton method, respectively. A quite natural question is whether it is possible to generalise FastICA to solve the parallel linear ICA problem, i.e., to extract all sources simultaneously. Although, there exist several other parallel linear ICA algorithms, e.g., fixed point Cayley ICA algorithm [8], retraction based ICA algorithm [9], gradient based ICA algorithm [13], and geodesic flow based ICA algorithm [18], a thorough discussion about these algorithms is certainly out of scope for this paper. We refer to [1, 3, 7, 10, 25] for further reading on optimisation on manifolds and comparisons of different algorithms in terms of performance and implementation. In this section, we generalise FastICA using a matrix shift strategy and an approximate Newton method to solve the parallel linear ICA problem.

19.4.1 Matrix Shifted QR-ICA Algorithm

First of all, let $Z = Z_{\mathrm{Q}} Z_{\mathrm{R}}$ be the unique QR decomposition of an invertible matrix $Z \in \mathbb{R}^{m\times m}$ into an orthogonal matrix $Z_{\mathrm{Q}} \in O(m)$ and an upper triangular matrix $Z_{\mathrm{R}} \in \mathbb{R}^{m\times m}$ with positive diagonal entries. We denote the two factors of the QR decomposition by

$$\begin{aligned} Q\colon \mathbb{R}^{m\times m} &\to O(m), \\ X &\mapsto X_{\mathrm{Q}}, \end{aligned} \tag{19.63}$$

and

$$\begin{aligned} R\colon \mathbb{R}^{m\times m} &\to \left\{X \in \mathbb{R}^{m\times m} \,|\, x_{ii} > 0, x_{ij} = 0 \text{ for all } i > j\right\}, \\ X &\mapsto X_{\mathrm{R}}. \end{aligned} \tag{19.64}$$

We then construct a simple and direct generalisation of the basic ICA map (19.38) to a full matrix as

$$\Phi\colon O(m) \to O(m), \quad X \mapsto (K(X))_{\mathrm{Q}}, \tag{19.65}$$

where

$$K\colon O(m) \to \mathbb{R}^{m\times m}, \quad K(X) := [k(x_1), \ldots, k(x_m)], \tag{19.66}$$

with k defined in (19.21).

Let $X^* = [x_1^*, \ldots, x_m^*] \in \Theta$ be a correct separation point of the parallel linear ICA problem, i.e., $x_i^* = \varepsilon_i v_{\tau(i)} \in S^{m-1}$. Using Eq. 19.25, one gets

$$\begin{aligned} K(X^*) &= \big[k(x_1^*), \ldots, k(x_m^*)\big] \\ &= X^*\Lambda, \end{aligned} \tag{19.67}$$

where

$$\Lambda = \operatorname{diag}\big(\mathbb{E}[G'(s_{\tau(1)})s_{\tau(1)}]), \ldots, \mathbb{E}[G'(s_{\tau(m)})s_{\tau(m)}]\big). \tag{19.68}$$

Due to the fact that every diagonal entry of Λ is positive, one has $(K(X^*))_{\mathrm{Q}} = X^*$, i.e., any correct separation point $X^* \in \Theta$ is a fixed point of the map Φ (19.65).

Now, by generalising the concept of a scalar shift strategy to a matrix shift strategy, we construct a matrix shifted version of Φ as follows,

$$\Phi_{\mathrm{s}}\colon O(m) \to O(m), \quad X \mapsto (K(X) - XL(X))_{\mathrm{Q}}, \tag{19.69}$$

where

$$L\colon O(m) \to \mathbb{R}^{m\times m}, \quad X \mapsto \operatorname{diag}\big(\mathbb{E}[G''(x_1^\top w)], \ldots, \mathbb{E}[G''(x_m^\top w)]\big). \tag{19.70}$$

Here we choose a special shift, namely a diagonal matrix L (19.70). To discuss more general (dense) matrix shifts is certainly a challenge. We consider such an issue as an open problem for our future research.

Now each column of the matrix $K(X) - XL(X)$ can be computed individually by

$$\pi_{\mathrm{f}}\colon S^{m-1} \to \mathbb{R}^m, \quad \pi_{\mathrm{f}}(x) := \mathbb{E}[G'(x^\top w)w] - \mathbb{E}[G''(x^\top w)]x, \tag{19.71}$$

i.e., the algorithmic map Φ_{s} can be restated as

$$\Phi_{\mathrm{s}}\colon O(m) \to O(m), \quad X \mapsto ([\pi_{\mathrm{f}}(x_1), \ldots, \pi_{\mathrm{f}}(x_m)])_{\mathrm{Q}}. \tag{19.72}$$

Note that each column π_{f} is indeed the numerator of the original FastICA map ϕ_{f} (19.15) applied to each x_i individually. In other words, the algorithm by iterating Φ_{s} is essentially a natural generalisation of FastICA to a full matrix. Unfortunately, such a map Φ_{s} is still not immune from the phenomenon of column-wise sign flipping, which is caused by the same reason as for the original FastICA. Thus, by using the function η (19.62) as a column-wise sign correction term, we construct

$$\widetilde{\Phi}_{\mathrm{s}}\colon O(m) \to O(m), \quad X \mapsto ([\sigma_{\mathrm{f}}(x_1), \ldots, \sigma_{\mathrm{f}}(x_m)])_{\mathrm{Q}}, \tag{19.73}$$

where

$$\sigma_{\mathrm{f}}\colon S^{m-1} \to \mathbb{R}^m, \quad \sigma_{\mathrm{f}}(x) := \eta(x)\pi_{\mathrm{f}}(x). \tag{19.74}$$

Note that the first column of $X \in O(m)$ under the map $\widetilde{\Phi}_{\mathrm{s}}$ evolves exactly as the column $x \in S^{m-1}$ under the algorithmic map $\widetilde{\phi}_{\mathrm{n}}$ (19.61). We refer to the algorithm induced by iterating the map $\widetilde{\Phi}_{\mathrm{s}}$ as matrix shifted *QR*-ICA algorithm. This algorithm is summarised as follows.

Algorithm 2 Matrix shifted *QR*-ICA algorithm

Step 1 Given an initial guess $X^{(0)} = [x_1^{(0)}, \ldots, x_m^{(0)}] \in O(m)$ and set $k = 0$.
Step 2 For $i = 1, \ldots, m$, compute

$$X^{(k+1)} = \left[\sigma_{\mathrm{f}}(x_1^{(k)}), \ldots, \sigma_{\mathrm{f}}(x_m^{(k)})\right],$$

where $\sigma_{\mathrm{f}}(x) = \mathrm{sign}(x^\top \pi_{\mathrm{f}}(x))\pi_{\mathrm{f}}(x)$, and

$$\pi_{\mathrm{f}}(x) = \mathbb{E}[G'(x^\top w)w] - \mathbb{E}[G''(x^\top w)]x.$$

Step 3 Update $X^{(k+1)} \leftarrow \left(X^{(k+1)}\right)_{\mathrm{Q}}$.
Step 4 If $\|X^{(k+1)} - X^{(k)}\|$ is small enough, stop.
Otherwise, set $k = k + 1$ and go to Step 2

Here, $\|\cdot\|$ is an arbitrary norm of matrices.

The convergence properties of the algorithmic map $\widetilde{\Phi}_{\mathrm{s}}$ are then summarised.

Lemma 1.5 *Let $\Theta \subset O(m)$, as defined in (19.8), be the set of solutions of the parallel linear ICA problem (19.6) and $G: \mathbb{R} \to \mathbb{R}$ a smooth function satisfying Assumption 2. Then any $X^* \in \Theta$ is a fixed point of the algorithmic map $\widetilde{\Phi}_{\mathrm{s}}$.*

Proof Let us denote

$$K_{\mathrm{s}}: O(m) \to \mathbb{R}^{m \times m}, \quad K_{\mathrm{s}}(X) := [\sigma_{\mathrm{f}}(x_1), \ldots, \sigma_{\mathrm{f}}(x_m)]. \tag{19.75}$$

According to the result (19.25), one computes that

$$\sigma_{\mathrm{f}}(x^*) = (\eta(x^*))^2 x^*, \tag{19.76}$$

i.e.,

$$K_{\mathrm{s}}(X^*) = X^* \Delta, \tag{19.77}$$

where

$$\Delta = \mathrm{diag}\left((\eta(x_1^*))^2, \ldots, (\eta(x_m^*))^2\right). \tag{19.78}$$

The result follows from the fact that $\widetilde{\Phi}_{\mathrm{s}}(X^*) = X^*$. □

To show the local convergence properties of the algorithmic map $\widetilde{\Phi}_{\mathrm{s}}$, we need the following lemma.

Lemma 1.6 *Let $Z \in \mathbb{R}^{m\times m}$ be invertible and consider the unique QR decomposition of Z. Then the derivative of $(Z)_{\mathrm{Q}}$ in direction $Y \in \mathbb{R}^{m\times m}$ is given by*

$$\mathrm{D}(Z)_{\mathrm{Q}}Y = (Z)_{\mathrm{Q}}\big(((Z)_{\mathrm{Q}})^{\top}Y\big((Z)_{\mathrm{R}}\big)^{-1}\big)_{\mathrm{skew}}, \tag{19.79}$$

where T_{skew} denotes the skew-symmetric part from the unique additive decomposition of $T \in \mathbb{R}^{m\times m}$ into skew-symmetric and upper triangular part, i.e.,

$$T = T_{\mathrm{skew}} + T_{\mathrm{upper}}, \tag{19.80}$$

where $T_{\mathrm{skew}} = -(T_{\mathrm{skew}})^{\top}$ and $\big(T_{\mathrm{upper}}\big)_{ij} = 0$ for all $i > j$.

Proof See Lemma 1 in [16] for a proof. □

Theorem 1.7 *Let $\Theta \subset O(m)$, as defined in (19.8), be the set of solutions of the parallel linear ICA problem (19.6) and $G\colon \mathbb{R} \to \mathbb{R}$ a smooth function satisfying Assumption 2. Then the algorithmic map $\widetilde{\Phi}_{\mathrm{s}}$ is locally quadratically convergent to $X^* \in \Theta$.*

Proof We will show that the linear map

$$\mathrm{D}\widetilde{\Phi}_{\mathrm{s}}(X) : T_X O(m) \to T_{\widetilde{\Phi}_{\mathrm{s}}(X)} O(m), \tag{19.81}$$

at X^* is indeed the zero map. By exploiting the fact that $\widetilde{\Phi}_{\mathrm{s}}(X^*) = (K_{\mathrm{s}}(X^*))_{\mathrm{Q}} = X^*$ and $(K_{\mathrm{s}}(X^*))_{\mathrm{R}} = \Delta$, one gets

$$\begin{aligned}
\mathrm{D}\widetilde{\Phi}_{\mathrm{s}}(X)\Xi|_{X=X^*} &= \mathrm{D}(K_{\mathrm{s}}(X))_{\mathrm{Q}}\Xi|_{X=X^*} \\
&= (K_{\mathrm{s}}(X^*))_{\mathrm{Q}}\Big(\Big((K_{\mathrm{s}}(X^*))_{\mathrm{Q}}\Big)^{\top}(\mathrm{D}K_{\mathrm{s}}(X)\Xi|_{X=X^*})\big((K_{\mathrm{s}}(X^*))_{\mathrm{R}}\big)^{-1}\Big)_{\mathrm{skew}} \\
&= X^*\big(X^{*\top}(\mathrm{D}K_{\mathrm{s}}(X)\Xi|_{X=X^*})\Delta^{-1}\big)_{\mathrm{skew}}.
\end{aligned} \tag{19.82}$$

Following Theorem 1.3, i.e., for all $i = 1, \ldots, m$,

$$\mathrm{D}\sigma_{\mathrm{f}}(x)\xi|_{x=x_i^*} = \delta_i x_i^*, \tag{19.83}$$

with some $\delta_i \in \mathbb{R}$ and denoting $\widehat{\Delta} = \mathrm{diag}(\delta_1, \ldots, \delta_m) \in \mathbb{R}^{m\times m}$, we have

$$\mathrm{D}K_{\mathrm{s}}(X)\Xi|_{X=X^*} = X^*\widehat{\Delta}, \tag{19.84}$$

i.e.,

$$\begin{aligned}
\mathrm{D}\widetilde{\Phi}_{\mathrm{s}}(X)\Xi|_{X=X^*} &= X^*\Big(\widehat{\Delta}\Delta^{-1}\Big)_{\mathrm{skew}} \\
&= 0.
\end{aligned} \tag{19.85}$$

Therefore the result follows. □

19.4.2 Approximate Newton-Like Parallel ICA Method

Following the same idea of approximating Hessians as in developing the approximate Newton one-unit linear ICA method, in this subsection, we develop an approximate Newton type method for minimising the parallel contrast function F (19.12).

19.4.2.1 Critical Point Analysis of Parallel ICA Contrast

Let $\mathfrak{so}(m) = \left\{\Omega \in \mathbb{R}^{m\times m} | \Omega = -\Omega^\top\right\}$ be the vector space of skew-symmetric matrices and denote $\Omega = [\omega_1, \ldots, \omega_m] = (\omega_{ij})_{i,j=1}^m \in \mathfrak{so}(m)$. By means of the matrix exponential map, a local parameterisation μ_X of $O(m)$ around $X \in O(m)$ is given by

$$\mu_X \colon \mathfrak{so}(m) \to O(m), \quad \mu_X(\Omega) := X \exp(\Omega). \tag{19.86}$$

By using the chain rule, the first derivative of contrast function F (19.12) at $X \in O(m)$ in direction $\Xi = X\Omega = [\xi_1, \ldots, \xi_m] \in T_X O(m)$, i.e., $\xi_i = X\omega_i$, is computed by

$$\begin{aligned} \mathrm{D}F(X)\Xi &= \frac{\mathrm{d}}{\mathrm{d}t}(F \circ \mu_X)(t\Omega)\Big|_{t=0} \\ &= \sum_{i=1}^m \mathbb{E}[G'(x_i^\top w)(\xi_i^\top w)] \\ &= \sum_{i=1}^m \omega_i^\top X^\top k(x_i) \\ &= \sum_{1\le i<j\le m}^m \omega_{ij}\Big(x_j^\top k(x_i) - k(x_j)^\top x_i\Big). \end{aligned} \tag{19.87}$$

The critical points $X \in O(m)$ of F are therefore characterised by

$$\sum_{1\le i<j\le m}^m \omega_{ij}\Big(x_j^\top k(x_i) - k(x_j)^\top x_i\Big) = 0. \tag{19.88}$$

Analogously to the one-unit linear ICA contrast f in (19.20), we are able to show that any $X^* \in \Theta$, which corresponds to a correct separation of all sources, is a critical point of F. Let $X^* = [x_1^*, \ldots, x_m^*] \in \Theta$, i.e., $x_i^* = \varepsilon_i v_{\tau(i)} \in S^{m-1}$. Recalling Eq. 19.25, we compute

$$x_j^{*\top} k(x_i^*) - k(x_j)^{*\top} x_i^* = 0, \tag{19.89}$$

for all $i \neq j$.

Since the geodesics through $X \in O(m)$ with respect to the Riemannian metric $\langle X\Omega_1, X\Omega_2\rangle := -\mathrm{tr}\Omega_1\Omega_2$ are given by $\mu_X(t\Omega)$, (left translated one parameter subgroups), the Riemannian Hessian of F can be computed from

$$\begin{aligned}\mathcal{H}F(X)(X\Omega, X\Omega) &= \frac{\mathrm{d}^2}{\mathrm{d}t^2}(F\circ\mu_X)(t\Omega)\Big|_{t=0}\\ &= \sum_{i=1}^m \mathbb{E}\Big[G''(x_i^\top w)(\omega_i^\top X^\top w)^2 - G'(x_i^\top w)(\xi_i^\top \Xi X w)\Big]\\ &= \sum_{i=1}^m \omega_i^\top X^\top \mathbb{E}\big[G''(x_i^\top w)ww^\top\big]X\omega_i - \sum_{i=1}^m \omega_i^\top \Omega X^\top \mathbb{E}\big[G'(x_i^\top w)w\big]\\ &= \sum_{i=1}^m \omega_i^\top X^\top H(x_i)X\omega_i + \mathrm{tr}\,\Omega^2 X^\top K(X). \end{aligned} \tag{19.90}$$

Following Eq. 19.33, the first summand in (19.90) evaluated at $X^* \in \Theta$ is computed as

$$\begin{aligned}&\sum_{i=1}^m \omega_i^\top X^{*\top} H\big(\varepsilon_i v_{\tau(i)}\big)X^*\omega_i\\ &\quad= \sum_{i=1}^m \mathbb{E}[G''(s_{\tau(i)})]\omega_i^\top\omega_i\\ &\quad= \sum_{1\le i<j\le m} \omega_{ij}^2\big(\mathbb{E}[G''(s_{\tau(i)})] + \mathbb{E}[G''(s_{\tau(j)})]\big),\end{aligned} \tag{19.91}$$

and the second summand in (19.90) is evaluated as

$$\begin{aligned}\mathrm{tr}\,\Omega^2 X^{*\top}K(X^*) &= \mathrm{tr}\,\Omega^2 X^{*\top}X^*\Lambda\\ &= \sum_{1\le i<j\le m} -\omega_{ij}^2\big(\mathbb{E}[G'(s_{\tau(i)})s_{\tau(i)}] + \mathbb{E}[G'(s_{\tau(j)})s_{\tau(j)}]\big).\end{aligned} \tag{19.92}$$

Hence,

$$\begin{aligned}\mathcal{H}F(X^*)(X^*\Omega, X^*\Omega) &= \sum_{1\le i<j\le m} \omega_{ij}^2\big(\big(\mathbb{E}[G''(s_{\tau(i)})] - \mathbb{E}[G'(s_{\tau(i)})s_{\tau(i)}]\big)\\ &\quad + \big(\mathbb{E}[G''(s_{\tau(j)})] - \mathbb{E}[G'(s_{\tau(j)})s_{\tau(j)}]\big)\big)\\ &= \sum_{1\le i<j\le m} -\omega_{ij}^2\Big(\eta(x_i^*) + \eta(x_j^*)\Big).\end{aligned} \tag{19.93}$$

Let $\Omega_{ij} \in \mathfrak{so}(m)$ be defined by $\omega_{ij} = -\omega_{ji} = 1$ and zeros anywhere else. We denote the standard basis of $\mathfrak{so}(m)$ by

$$\mathcal{B} = \left\{\Omega_{ij} \in \mathfrak{so}(m) \,\middle|\, 1 \le i < j \le m\right\}. \tag{19.94}$$

It is obvious that $\mathcal{H}F(X^*)(X^*\Omega_{ij}, X^*\Omega_{pq}) = 0$ for any distinct pair of basis matrices $\left(\Omega_{ij}, \Omega_{pq}\right) \subset \mathcal{B}$, i.e., the Hessian of F at $X^* \in \Theta$ is indeed diagonal with respect to the standard basis $\mathcal{B}$. Moreover, to ensure that such a Hessian is invertible, it will require the expression $\eta(x_i^*) + \eta(x_j^*)$ to be nonzero. According to specific applications, such a condition can be fulfilled by choosing the function G carefully. A special example is given and discussed in Remark 3. We conclude our results as follows.

Theorem 1.8 *Let $\Theta \subset O(m)$, as defined in (19.8), be the set of solutions of the parallel linear ICA problem (19.6) and let $G\colon \mathbb{R} \to \mathbb{R}$ be a smooth even function such that*

$$\mathbb{E}[G''(s_i)] - \mathbb{E}[G'(s_i)s_i] \neq -\left(\mathbb{E}[G''(s_j)] - \mathbb{E}[G'(s_j)s_j]\right), \tag{19.95}$$

for all $i, j = 1, \ldots, m$ and $i \neq j$. Then any $X^* \in \Theta$ is a non-degenerated critical point of F (19.12).

Remark 3 As pointed out before, the value of the expression $\mathbb{E}[G''(s_i)] - \mathbb{E}[G'(s_i)s_i]$ depends on both the function G and source signals s. It is hardly possible to characterise the set of critical points in full generality. As a special case, let us specify G by the following popular choice

$$G\colon \mathbb{R} \to \mathbb{R}, \quad G(x) := \frac{1}{\zeta}\log(\cosh(\zeta x)), \tag{19.96}$$

where $\zeta \in \mathbb{R}$ is positive. According to the fact that the absolute values of entries of whitened sources are bounded by the pre-whitening process, it can be shown that, by choosing ζ carefully, the value of $\mathbb{E}[G''(s_i)] - \mathbb{E}[G'(s_i)s_i]$ is ensured to be positive for any signal. In other words, every point $X^* \in \Theta$ is a strict local minimum of the parallel ICA contrast F.

19.4.2.2 Approximate Newton-Like Parallel ICA Algorithm

Following the same strategy of developing the approximate Newton one-unit ICA algorithm $\widetilde{\phi}_{\mathrm{n}}$ (19.61) in Sect. 19.3, we now apply the idea of approximating the Hessian of F to develop a Newton-like parallel ICA method. The approximation is given as follows

$$\mathcal{H}F(X)(X\Omega, X\Omega) \approx \sum_{1 \le i < j \le m}^{m} -\omega_{ij}^2\left(\eta(x_i) + \eta(x_j)\right). \tag{19.97}$$

Clearly, this choice may seriously differ from the true Hessian. However, coinciding at the critical points $X^* \in \Theta$, it locally yields a good approximation.

Thus, by recalling Eq. 19.87, an approximate Newton direction $X\Omega \in T_X O(m)$ with $\Omega = (\omega_{ij})_{i,j=1}^m \in \mathfrak{so}(m)$ is explicitly computed by

$$\big(\eta(x_i) + \eta(x_j)\big)\omega_{ij} = x_i^\top k(x_j) - k(x_i)^\top x_j, \tag{19.98}$$

for all $1 \le i < j \le m$. According to the condition (19.95), one computes directly

$$\omega_{ij} = \frac{x_i^\top k(x_j) - k(x_i)^\top x_j}{\eta(x_i) + \eta(x_j)}. \tag{19.99}$$

The projection of $X\Omega$ onto $O(m)$ by using μ_X (19.86) completes an iteration of an approximate Newton-like parallel ICA method. It is known that the calculation of the matrix exponential of an arbitrary $\Omega \in \mathfrak{so}(m)$ requires an expensive iterative process [17]. To overcome this computational issue, one can utilise a first order approximation of the matrix exponential via a *QR* decomposition, which preserves orthogonality. A similar idea has been explored in [3]. For a given $\Omega \in \mathfrak{so}(m)$, let us compute the unique *QR* decomposition of the matrix $I_m + \Omega$. Since the determinant of $I_m + \Omega$ is always positive, the *QR* decomposition is unique with $\det(I_m + \Omega)_{\mathrm{Q}} = 1$. We then define a second local parameterisation on $O(m)$ as

$$v_X \colon \mathfrak{so}(m) \to O(m), \quad v_X(\Omega) := X(I_m + \Omega)_{\mathrm{Q}}. \tag{19.100}$$

Substituting the approximate Newton step as computed in (19.99) into v_X leads to the following algorithmic map on $O(m)$

$$\Phi_{\mathrm{n}} \colon O(m) \to O(m), \quad X \mapsto X(I_m + \widetilde{\Omega}(X))_{\mathrm{Q}}, \tag{19.101}$$

where $\widetilde{\Omega} \colon O(m) \to \mathfrak{so}(m)$ is the smooth map consisting of the following entry-wise maps

$$\widetilde{\omega}_{ij} \colon O(m) \to \mathbb{R}, \quad \widetilde{\omega}_{ij}(X) := \frac{k(x_i)^\top x_j - x_i^\top k(x_j)}{\eta(x_i) + \eta(x_j)}, \tag{19.102}$$

for all $1 \le i < j \le m$. Iterating the map Φ_{n} gives an approximate Newton-like parallel ICA algorithm as follows.

Algorithm 3 Approximate Newton-like parallel ICA algorithm

Step 1 Given an initial guess $X^{(0)} = [x_1^{(0)}, \ldots, x_m^{(0)}] \in O(m)$ and set $k = 0$.
Step 2 Compute $\Omega^{(k)} = (\omega_{ij}^{(k)})_{i,j=1}^m \in \mathfrak{so}(m)$ with

$$\omega_{ij}^{(k)} = \frac{\zeta(x_i, x_j) - \zeta(x_j, x_i)}{\eta(x_i) + \eta(x_j)}, \quad \text{for } 1 \le i < j \le m,$$

where $\zeta(x_i, x_j) = \mathbb{E}[G'(x_i^\top w)(x_j^\top w)]$, and

$$\eta(x) = \mathbb{E}[G'(x^\top w)(x^\top w)] - \mathbb{E}[G''(x^\top w)].$$

Step 3 Update $X^{(k+1)} \leftarrow X^{(k)}(I_m + \Omega^{(k)})_{\mathrm{Q}}$.

Step 4 If $\|X^{(k+1)} - X^{(k)}\|$ is small enough, stop.
Otherwise, set $k = k + 1$ and go to Step 2.

Remark 4 For the one-unit linear ICA problem, the scalar shifted algorithm and the approximate Newton method are indeed identical as shown in Sect. 19.3. For the parallel linear ICA problem, however, the two resulting algorithms, i.e. matrix shifted *QR* -ICA algorithm (Algorithm 2) and approximate Newton-like parallel ICA algorithm (Algorithm 3), are different. Experimental results can be found in Sect. 19.5.

19.4.2.3 Local Convergence Properties

Finally the local convergence properties of the algorithmic map Φ_{n} are characterised by the following two results.

Lemma 1.9 *Let $\Theta \subset O(m)$, as defined in (19.8), be the set of solutions of the parallel linear ICA problem (19.6) and $G: \mathbb{R} \to \mathbb{R}$ be a smooth function satisfying the condition (19.95). Then any $X^* \in \Theta$ is a fixed point of the algorithmic map Φ_{n}.*

Proof Following Eq. 19.95, one knows that the denominator in (19.102) is non-zero. Therefore Eq. 19.89 yields

$$\widetilde{\omega}_{ij}(X^*) = 0, \tag{19.103}$$

for all $1 \leq i < j \leq m$. The result follows from $(I_m)_{\mathrm{Q}} = I_m$. □

Theorem 1.10 *Let $\Theta \subset O(m)$, as defined in (19.8), be the set of solutions of the parallel linear ICA problem (19.6) and $G: \mathbb{R} \to \mathbb{R}$ be a smooth function satisfying the condition (19.95). Then the algorithmic map Φ_{n} is locally quadratically convergent to $X^* \in \Theta$.*

Proof Following the argument in [15], Lemmas 2.7–2.9, we will show that the first derivative of the algorithmic map Φ_{n}

$$\mathrm{D}\,\Phi_{\mathrm{n}}(X): T_X O(m) \to T_{\Phi_{\mathrm{n}}(X)} O(m) \tag{19.104}$$

vanishes at a fixed point X^*.

Recall the results of Lemma 1 in [16], we compute

$$\mathrm{D}\,\Phi_{\mathrm{n}}(X^*)\Xi = \Xi + X^* \mathrm{D}\widetilde{\Omega}(X^*)\Xi, \tag{19.105}$$

where $\Xi = X^*\Omega \in T_{X^*} O(m)$. Thus, to show that the first derivative of Φ_{n} vanishes at X^* is equivalent to showing

$$\mathrm{D}\widetilde{\Omega}(X^*)\Xi = -\Omega, \tag{19.106}$$

which indeed is equivalent to

$$\mathrm{D}\widetilde{\omega}_{ij}(X^*)\Xi = -\omega_{ij}, \tag{19.107}$$

for all $1 \le i < j \le m$. We compute the first derivative of $\widetilde{\omega}_{ij}$ (19.102)

$$\begin{aligned}\mathrm{D}\widetilde{\omega}_{ij}(X^*)\Xi &= \frac{\mathrm{d}}{\mathrm{d}t}\,\widetilde{\omega}_{ij}(X^*\exp(t\Omega))\Big|_{t=0} \\ &= \mathrm{D}\left(\frac{1}{\eta(x_i^*)+\eta(x_j^*)}\right)\Xi \cdot \left(k(x_i^*)^\top x_j^* - x_i^{*\top}k(x_j^*)\right) \\ &\quad + \frac{1}{\eta(x_i^*)+\eta(x_j^*)} \cdot \mathrm{D}\left(k(x_i^*)^\top x_j^* - x_i^{*\top}k(x_j^*)\right)\Xi. \end{aligned} \tag{19.108}$$

It is clear that the first summand vanishes following the previous results, see (19.89). By using (19.25) and (19.51), we compute

$$\begin{aligned}&\mathrm{D}\left(k(x_i^*)^\top x_j^* - x_i^{*\top}k(x_j^*)\right)\Xi \\ &= \left(\mathrm{D}k(x_i^*)\Xi\right)^\top x_j^* + k(x_i^*)^\top \xi_j - \xi_i^\top k(x_j^*) - x_i^{*\top}\mathrm{D}k(x_j^*)\Xi \\ &= \mathbb{E}[G''(s_{\tau(i)})]\omega_i^\top X^{*\top} x_j^* + \mathbb{E}[G'(s_{\tau(i)})s_{\tau(i)}]x_i^{*\top}X^*\omega_j \\ &\quad - \mathbb{E}[G'(s_{\tau(j)})s_{\tau(j)}]\omega_i^\top X^{*\top}x_j^* - \mathbb{E}[G''(s_{\tau(i)})]x_i^*X^*\omega_j \\ &= \mathbb{E}[G''(s_{\tau(i)})]\omega_{ij} + \mathbb{E}[G'(s_{\tau(i)})s_{\tau(i)}]\omega_{ji} \\ &\quad - \mathbb{E}[G'(s_{\tau(j)})s_{\tau(j)}]\omega_{ij} - \mathbb{E}[G''(s_{\tau(i)})]\omega_{ji} \\ &= \left(\mathbb{E}[G''(s_{\tau(i)})] - \mathbb{E}[G'(s_{\tau(i)})s_{\tau(i)}] + \mathbb{E}[G''(s_{\tau(i)})] - \mathbb{E}[G'(s_{\tau(j)})s_{\tau(j)}]\right)\omega_{ij} \\ &= -\left(\eta(x_i^*)+\eta(x_j^*)\right)\omega_{ij}. \end{aligned} \tag{19.109}$$

Applying now Eq. 19.108 yields Eq. 19.107. The result follows. □

19.5 Numerical Experiments

In this section, the performance of all presented methods is demonstrated and compared by using several numerical experiments. We apply all algorithms to an audio signal separation dataset provided by the Brain Science Institute, RIKEN, see http://www.bsp.brain.riken.jp/data. The dataset consists of 200 natural speech signals sampled at 4 kHz with 20,000 samples per signal.

We know that the FastICA algorithm only recovers one single source. In order to extract all source signals, a deflationary FastICA approach using the Gram–Schmidt orthogonalisation has already been developed in [11]. In our experiment, to avoid dealing with the sign flipping issue, we implement the map $\widetilde{\phi}_{\mathrm{n}}$ (19.61) instead of the FastICA map ϕ_{f} (19.15). Nevertheless, we still refer to this method as FastICA-Defl. It is important to notice that, for a linear ICA problem of extracting m sources, after computing the second last column x_{m-1} of the demixing matrix $X \in O(m)$, the last column x_m is already uniquely determined up to sign by the Gram–Schmidt process. In other words, one only needs to apply $m-1$ deflationary FastICA procedures to extract m signals. By the same reason, in implementing the matrix shifted QR-ICA algorithm $\widetilde{\Phi}_{\mathrm{s}}$ (19.73), one can skip the update on the last column, i.e., for each iteration, we do the following

$$\widehat{\Phi}_{\mathrm{f}}\colon O(m) \to O(m), \quad X \mapsto \left([\sigma_{\mathrm{f}}(x_1), \ldots, \sigma_{\mathrm{f}}(x_{m-1}), x_m]\right)_{\mathrm{Q}}. \tag{19.110}$$

In the sequel, we call it QR-ICA, and refer to the approximate Newton-like parallel ICA method Φ_{n} (19.101) as ANPICA.

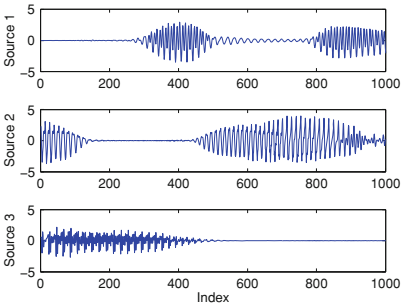
All methods developed in this paper involve evaluations of the functions k (19.21) and ρ_{f} (19.54) at certain columns of the demixing matrix $X \in O(m)$. It is then sensible to utilise the number of column-wise evaluations of k and ρ_{f} as a basic computational unit to compare these methods. Let us denote the sample size of each mixture by n, which is usually much greater than the number of signals m. In all our experiments, we fix the sample size to $n = 10^4$. Obviously, the computational burden of these methods is mainly due to the evaluations of k and ρ_{f}, which depend on a huge data matrix $W \in \mathbb{R}^{m\times n}$. For a given $x \in S^{m-1}$, a reconstruction of the corresponding estimated signal firstly involves $m \cdot n$ scalar multiplications and $(m-1)\cdot n$ scalar additions. Consequently, computing $k(x)$ and $\rho_{\mathrm{f}}(x)$ requires n evaluations of G' and G'' on the estimated signal together with another $m \cdot n$ scalar multiplications and $m \cdot (n-1) + n$ scalar additions. Finally, the major computational cost for evaluating $k(x)$ and $\rho_{\mathrm{f}}(x)$ can be summarised in the following Table 19.1.

Here the notations $\otimes$ and $\oplus$ represent the scalar multiplication and addition, respectively. Thus by counting in the factor of the number of columns, the complexity of all three methods is roughly of the same order, i.e., of $\mathcal{O}(m^2 n)$.

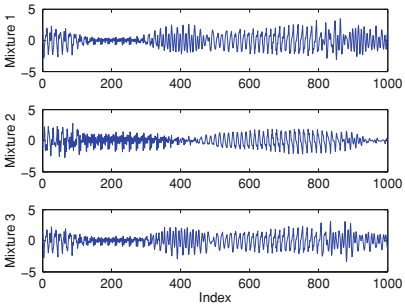
The task of our first experiment is to separate $m = 3$ signals which are randomly chosen out of 200 source speeches. All three methods are initialised by the same point on $O(m)$. Figure 19.1 demonstrates the results from a single experiment. It shows that all methods can successfully recover all source signals up to a

Table 19.1 Major computational costs of evaluating $k(x)$ (19.21) and $\rho_{\mathrm{f}}(x)$ (19.54).

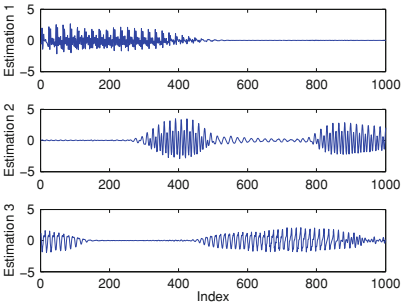
Operation	$\otimes$	$\oplus$	G'	G''
Times	$2m \cdot n$	$2m \cdot n - m$	n	n



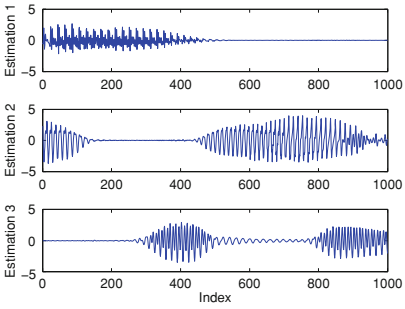
(a) Source signals



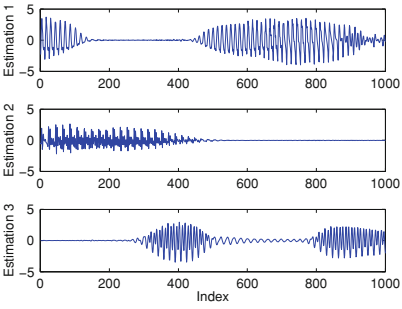
(b) Mixed signals



(c) FastICA-Defl



(d) *QR*-ICA



(e) ANPICA

Fig. 19.1 A simple example of three audio signals. **c–e** are estimated signals by FastICA-Defl, *QR*-ICA and ANPICA

sign and an arbitrary permutation Fig. 19.1c–e. It is worthwhile to notice that the first extracted signal by FastICA-Defl and *QR*-ICA, respectively, are always identical, since the first column of *QR*-ICA evolves exactly as the first column of FastICA-Defl does.

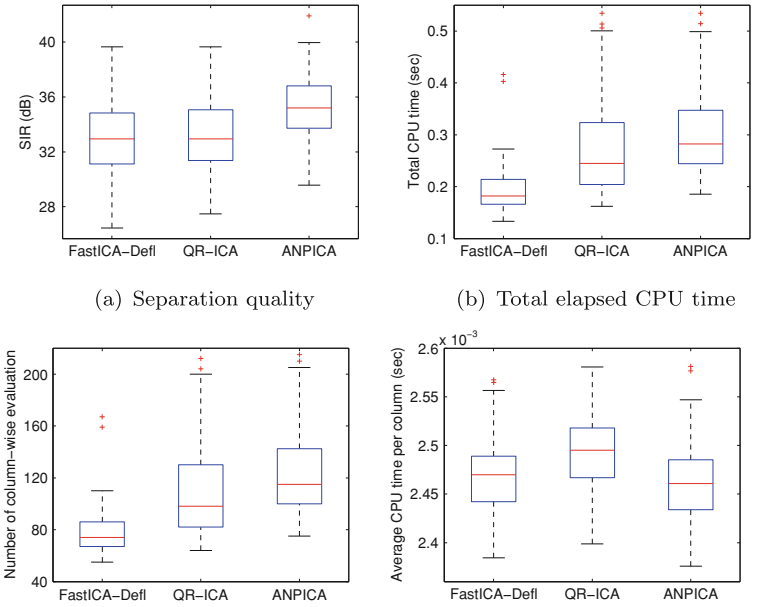
Finally, we will compare the performance of all presented methods in terms of both separation quality and convergence speed. The separation performance is measured by the average signal-to-interference-ratio (SIR) index [21], i.e.,

$$SIR(Z) := \frac{10}{m}\sum_{i=1}^{m}\log_{10}\frac{\max_j z_{ij}^2}{\sum_{j=1}^{m} z_{ij}^2 - \max_j z_{ij}^2}, \tag{19.111}$$

where $Z = (z_{ij})_{i,j=1}^{m} = X^\top V$, and $V, X \in O(m)$ are the mixing matrix and the computed demixing matrix, respectively. In general, the greater the SIR index, the better the separation. The convergence speed is measured by comparing the elapsed CPU time required by each algorithm to reach the same level of error $\|X^{(k)} - X^*\|_{\mathrm{F}} < \epsilon$. Here $\|X^{(k)} - X^*\|_{\mathrm{F}}$ denotes the Frobenius norm of the difference between the terminated demixing matrix $X^* \in O(m)$ and the k-th iterate $X^{(k)} \in O(m)$. Since

$$\left\|X^{(k)} - X^*\right\|_{\mathrm{F}}^2 = \sum_{i=1}^{m}\left\|x_i^{(k)} - x_i^*\right\|^2, \tag{19.112}$$

the column-wise stop criterion for FastICA-Defl is chosen to be $\|x^{(k)} - x^*\|^2 < \epsilon/m$. In our experiments, we set $\epsilon = 10^{-10}$.



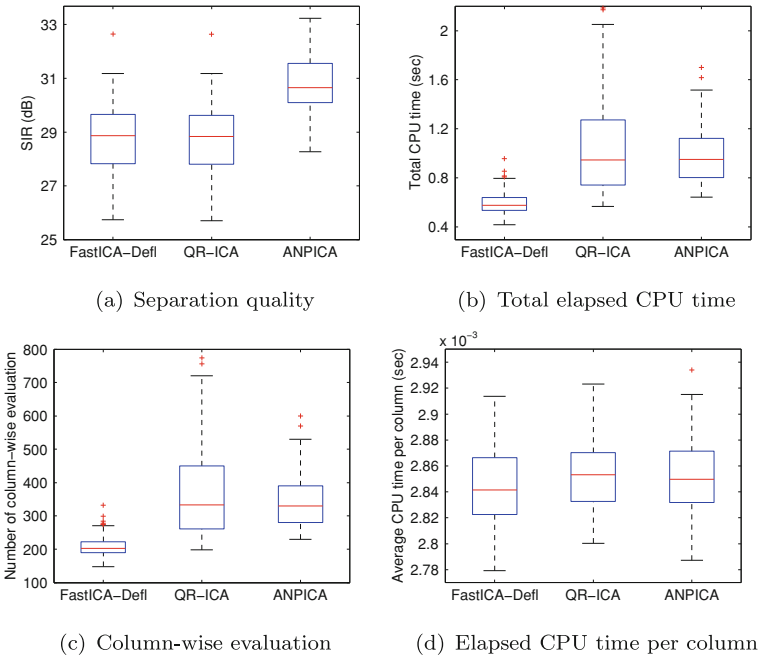
(a) Separation quality (b) Total elapsed CPU time

(c) Column-wise evaluation (d) Elapsed CPU time per column

Fig. 19.2 Comparison of performance ($m = 5$ and $n = 10^4$)

We firstly apply all three algorithms to extract $m = 5$ randomly chosen speech signals. By replicating the experiment 100 times, the boxplots of the SIR index and the elapsed CPU time (seconds) required by each algorithm are drawn in Fig. 19.2a, b, respectively. Figure 19.2a shows that ANPICA outperforms both FastICA-Defl and *QR*-ICA in terms of separation quality, and that both FastICA-Defl and *QR*-ICA perform equally well. In terms of convergence speed shown in Fig. 19.2b, it indicates that FastICA-Defl is the fastest algorithm to reach the pre-chosen error level, while *QR*-ICA takes the longest time to converge. By counting the total number of column-wise evaluations for each method until convergence as shown in Fig. 19.2c, the average elapsed CPU time for evaluating each column in each method are computed in Fig. 19.2d. It shows that ANPICA requires the most amount of column-wise evaluations.

Finally, we further apply all methods to extract $m = 10$, then 15, and finally 20 speech signals. Similar boxplots are produced in Figs. 19.3, 19.4, and 19.5. One can see that, in term of separation quality, ANPICA outperforms the other two methods consistently and the outperformance becomes more significant when the number of singles increase. Both FastICA-Defl and *QR*-ICA perform constantly equally well. By comparing the convergence speed, there is no surprise that FastICA-Defl is still the fastest as well as the cheapest algorithm among all the



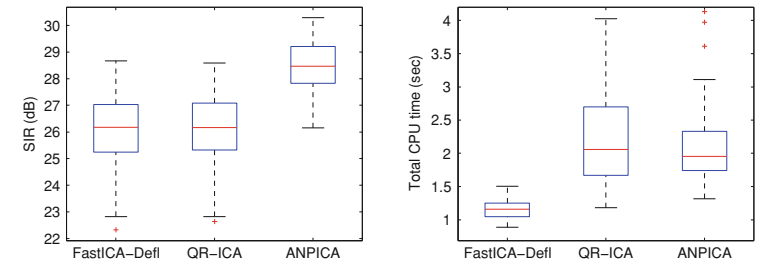
(a) Separation quality

(b) Total elapsed CPU time

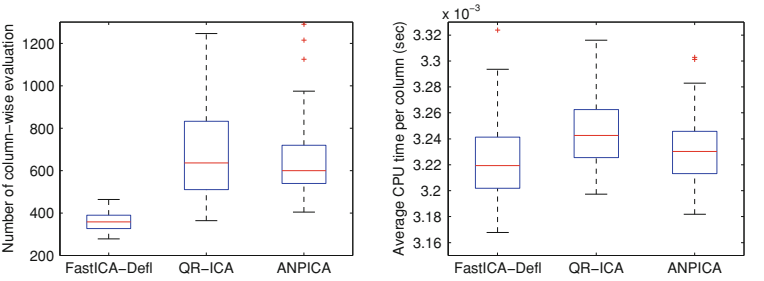
(c) Column-wise evaluation

(d) Elapsed CPU time per column

Fig. 19.3 Comparison of performance ($m = 10$ and $n = 10^4$)

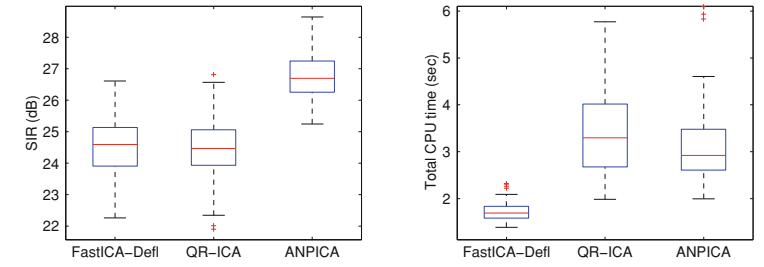


(a) Separation quality (b) Total elapsed CPU time

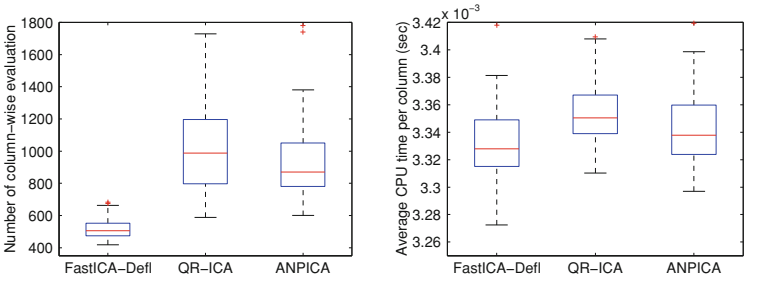


(c) Column-wise evaluation (d) Elapsed CPU time per column

Fig. 19.4 Comparison of performance ($m = 15$ and $n = 10^4$)



(a) Separation quality (b) Total elapsed CPU time



(c) Column-wise evaluation (d) Elapsed CPU time per column

Fig. 19.5 Comparison of performance ($m = 20$ and $n = 10^4$)

three algorithms. We also observe that, when the number of signals is big enough, *QR*-ICA requires more column-wise evaluations than ANPICA does, i.e., *QR*-ICA becomes the slowest algorithm among all three, see Fig. 19.5c.

To summarise, FastICA-Defl is the most simple and fastest algorithm to solve a linear ICA problem. However, it is not able to extract signals simultaneously. Nevertheless ANPICA is the most superior method for solving the parallel linear ICA problem in terms of separation quality at any circumstance.

19.6 Conclusions

In this paper, we study the FastICA algorithm, a classic method for solving the one-unit linear ICA problem. After reinterpreting FastICA in the framework of a scalar shift strategy and an approximate Newton method, we generalise FastICA to solve the parallel linear ICA problem by using a matrix shift strategy and an approximate Newton method. The work presented in this paper also demonstrates similarities in terms of analysis and generalisations between the FastICA algorithms and the RQI method in numerical linear algebra.

Recently, the present authors have successfully generalised FastICA to solve the problem of independent subspace analysis (ISA) [22]. The key ideas are again similar to developing the so-called Graßmann-RQI algorithm [2], a generalisation of RQI for computing invariant subspaces.

References

1. Absil P-A, Mahony R, Sepulchre R (2008) Optimization algorithms on matrix manifolds. Princeton University Press, Princeton
2. Absil P-A, Mahony R, Sepulchre R, Dooren PV (2002) A Grassmann–Rayleigh quotient iteration for computing invariant subspaces. SIAM Rev 44(1):57–73
3. Adler R, Dedieu J-P, Margulies J, Martens M, Shub M (2002) Newton's method on Riemannian manifolds and a geometric model for the human spine. IMA J Numer Anal 22:359–390
4. Boothby WM (2002) An introduction to differentiable manifolds and Riemannian geometry, revised, 2nd edn. Academic Press, Oxford
5. Cichocki A, Amari S-I (2002) Adaptive blind signal and image processing: learning algorithms and applications. John Wiley & Sons Ltd., Chichester
6. Comon P (1994) Independent component analysis, a new concept?. Signal Process 36(3):287–314
7. Edelman A, Arias TA, Smith ST (1998) The geometry of algorithms with orthogonality constraints. SIAM J Matrix Anal Appl 20(2):303–353
8. Fiori S (2006) Fixed-point neural independent component analysis algorithms on the orthogonal group. Future Generation Comput Syst 22(4):430–440
9. Fiori S (2007) Learning independent components on the orthogonal group of matrices by retractions. Neural Process Lett 25(3):187–198

10. Hüper K, Trumpf J (2004) Newton-like methods for numerical optimisation on manifolds. In: Proceedings of thirty-eighth Asilomar conference on signals, systems and computers, pp 136–139
11. Hyvärinen A (1999) Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans Neural Netw 10(3):626–634
12. Hyvärinen A, Karhunen J, Oja E (2001) Independent component analysis. Wiley, New York
13. Journée M, Absil P-A, Sepulchre R (2007) Gradient-optimization on the orthogonal group for independent component analysis. In: Proceedings of the 7th international conference on independent component analysis and source separation (ICA 2007). Lecture notes in computer science, vol 4666. Springer, Berlin, pp 57–64
14. Journée M, Teschendorff AE, Absil P-A, Sepulchre R (2007) Geometric optimization methods for independent component analysis applied on gene expression data. In: Proceedings of the 32nd IEEE international conference on acoustics, speech, and signal processing (ICASSP 2007), pp IV1413–IV1416, Hawaii, USA
15. Kleinsteuber M (2006) Jacobi-type methods on semisimple Lie algebras—a Lie algebraic approach to the symmetric eigenvalue problem. PhD thesis, Bayerische Julius-Maximilians-Universität Würzburg
16. Kleinsteuber M, Hüper K (2007) An intrinsic CG algorithm for computing dominant subspaces. In: Proceedings of the 32nd IEEE international conference on acoustics, speech, and signal processing (ICASSP 2007), pp IV1405–IV1408, Hawaii, USA
17. Moler C, van Loan C (2003) Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. SIAM Rev 45(1):3–49
18. Nishimori Y (1999) Learning algorithm for ICA by geodesic flows on orthogonal group. In: The International Joint Conference on Neural Networks (IJCNN'99), pp 933–938, Washington, DC, USA
19. Regalia P, Kofidis E (2003) Monotonic convergence of fixed-point algorithms for ICA. IEEE Trans Neural Netw 14(4):943–949
20. Roberts S, Everson R (2001) Independent component analysis: principles and practice. Cambridge University Press, Cambridge
21. Schobben D, Torkkola K, Smaragdis P (1999) Evaluation of blind signal separation methods. In: Proceedings of the 1st international workshop on independent component analysis and blind source separation (ICA 1999), pp 261–266
22. Shen H, Hüper K (2007) Generalised fastICA for independent subspace analysis. In: Proceedings of the 32nd IEEE international conference on acoustics, speech, and signal processing (ICASSP 2007), pp IV1409–IV1412, Hawaii, USA
23. Shen H, Kleinsteuber M, Hüper K (2008) Local convergence analysis of FastICA and related algorithms. IEEE Trans Neural Netw 19(6):1022–1032
24. Shub M (1986) Some remarks on dynamical systems and numerical analysis. In: Dynamical systems and partial differential equations (Caracas, 1984): Proceedings of the VII ELAM pp 69–92, Universidad Simon Bolivar, Caracas
25. Smith ST (1994) Optimization techniques on Riemannian manifolds. In: Bloch A (ed) Hamiltonian and gradient flows, algorithms and control, Fields Institute Communications. American Mathematical Society, Providence, pp 113–136
26. Spivak M (1999) A comprehensive introduction to differential geometry, vols 1–5, 3rd edn. Publish or Perish, Inc, Houston

Chapter 20
On Computing Minimal Proper Nullspace Bases with Applications in Fault Detection

Andras Varga

Abstract We discuss computationally efficient and numerically reliable algorithms to compute minimal proper nullspace bases of a rational or polynomial matrix. The underlying main computational tool is the orthogonal reduction to a Kronecker-like form of the system matrix of an equivalent descriptor system realization. A new algorithm is proposed to compute a simple minimal proper nullspace basis, starting from a non-simple one. Minimal dynamic cover based computational techniques are used for this purpose. The discussed methods allow a high flexibility in addressing several fault detection related applications.

20.1 Introduction

Consider a $p \times m$ rational matrix $G(\lambda)$, where the indeterminate λ is generally a complex variable. If we interpret $G(\lambda)$ as the *transfer-function matrix* (TFM) of a (generalized) linear time-invariant system, then according to the system type, λ is the s variable in the Laplace transform in the case of a continuous-time system or λ is the z variable in the Z-transform in the case of a discrete-time system. This interpretation of λ is relevant when system stability aspects are considered.

In this paper we address the following computational problem: For a given $p \times m$ rational or polynomial matrix $G(\lambda)$ with normal rank r, determine a $(p - r) \times p$ rational basis matrix $N_l(\lambda)$ of the left nullspace of $G(\lambda)$ such that

$$N_l(\lambda)G(\lambda) = 0.$$

A. Varga (✉)
German Aerospace Center, DLR, Oberpfaffenhofen, Institute of Robotics and Mechatronics, 82234 Wessling, Germany
e-mail: Andras.Varga@dlr.de

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_20,

Of special importance are *minimal bases* having the least achievable McMillan degree. Moreover, depending on the underlying application, further properties may be desirable, as for example, determining $N_l(\lambda)$ as a polynomial matrix or as a proper rational matrix with specified poles.

The rigorous study of polynomial bases started with the theoretical works of Forney [4], and followed by initial algorithmic developments by Kailath [7]. For the computation of a minimal polynomial bases of a polynomial matrix $G(\lambda)$ there are many algorithms, see [1] and the literature cited therein. Two main classes of methods are the *resultant methods*, which determine the solution by solving directly polynomial equations involving appropriate resultant matrices [1], and *pencil methods*, which rely on matrix pencil reduction algorithms [2]. While resultant methods can be a real alternative to the unreliable polynomial manipulation based methods proposed in [7], their application to rational matrices defined implicitly via state space system realizations requires, as a supplementary step, bringing the system model into a polynomial representation. This involve factoring $G(\lambda)$ as $G(\lambda) = N(\lambda)M^{-1}(\lambda)$, where $N(\lambda)$ and $M(\lambda)$ are polynomial matrices, and applying the method to $N(\lambda)$. The converse operation (e.g., proper rational factoring of a polynomial matrix) could be also necessary, if the desired basis must be a proper rational basis. Such computational detours are generally considered highly unreliable for TFMs of large scale systems (which usually arise in a state-space form).

The pencil methods works directly on the state space realization of $G(\lambda)$, and are applicable to both polynomial and rational matrices. The main computational tool is the reduction of a matrix pencil to a Kronecker-like form using orthogonal transformations. The left Kronecker structure provides the complete information to compute a polynomial basis via straightforward matrix and polynomial matrix manipulations [2].

For many applications, proper rational bases are required. Such bases can be immediately obtained from polynomial bases. However, to avoid potentially unstable polynomial manipulations, it is of interest to compute proper rational bases directly, without the unnecessary detour of determining first polynomial bases. The theory of proper rational bases has been developed in [13], where the main concepts have been also defined. Of special importance are proper bases which are *simple* (see the exact definition in the next section), representing a direct generalization of minimal polynomial bases. A first reliable numerical method to compute proper rational bases has been proposed by the author in [20]. This method belongs to the class of pencil methods and its main advantage is that a minimal proper rational basis can be computed by using exclusively orthogonal transformations. Note however, that the resulting basis is generally not simple.

In this paper we extend the algorithm of [20] to compute simple minimal proper rational bases. The new algorithm can be seen as a post-processing method by determining a simple basis starting from a non-simple one. Minimal dynamic covers techniques are used for this purpose. The proposed new algorithm allows to perform easily operations with the resulting basis, which are of importance to

solve applications as those encountered in fault detection. For example, computing linear combinations of basis vectors immediately leads to candidate solutions of the fault detection problem with a least order detector. Several applications in solving fault detection problems are discussed in a separate section.

20.2 Nullspace Bases

Since polynomial bases represent an important tool in defining the corresponding concepts for the more general *rational bases*, we will recall shortly some of the main results of [4]. Assume that $N_l(\lambda)$ is a polynomial basis of the left nullspace of $G(\lambda)$. Let denote by n_i, the ith *index* (*or degree*), representing the greatest degree of the ith row of $N_l(\lambda)$. Then, the *order* of $N_l(\lambda)$ is defined as $n_d = \sum_{i=1}^{p-r} n_i$, (i.e., the sum of row degrees). A *minimal basis* is one which has least order among all polynomial bases. The indices of a minimal basis are called *minimal indices*. The order of a minimal polynomial basis $N_l(\lambda)$ is equal to the McMillan degree of $N_l(\lambda)$.

Some properties of a minimal bases are summarized below [4, 7]:

Theorem 1 *Let $N_l(\lambda)$ be a minimal polynomial basis of the left nullspace of $G(\lambda)$ with row indices n_i, $i = 1, \ldots, p - r$. Then the following holds:*

1. *The row indices are unique up to permutations (i.e., if $\widetilde{N}_l(\lambda)$ is another minimal basis, then $N_l(\lambda)$ and $\widetilde{N}_l(\lambda)$ have the same minimal indices).*
2. *The minimal indices are the left Kronecker indices of $G(\lambda)$.*
3. *$N_l(\lambda)$ is irreducible, i.e., has full row rank for all $\lambda \in \mathbb{C}$ ($N_l(\lambda)$ has no finite or infinite zeros).*
4. *$N_l(\lambda)$ is row reduced, i.e., the leading row coefficient matrix (formed from the coefficients of the highest row degrees) has full row rank.*

If $M_l(\lambda)$ is a non-singular rational matrix, then $\widetilde{N}_l(\lambda) := M_l(\lambda)N_l(\lambda)$ is also a left nullspace basis. Frequently the matrices $M_l(\lambda)$ originate from appropriate *left coprime factorizations* of an original basis $N_l(\lambda)$ in the form

$$N_l(\lambda) = M_l(\lambda)^{-1}\widetilde{N}_l(\lambda), \tag{20.1}$$

where the factors $M_l(\lambda)$ and $\widetilde{N}_l(\lambda)$ can be chosen to satisfy special requirements (e.g., have only poles in a certain "good" region of the complex plane).

The main advantage of minimal polynomial bases is the possibility to easily build *proper minimal rational bases*. These are proper rational bases having the least McMillan degree n_d. A proper minimal rational basis with arbitrary poles can be simply constructed by taking

$$M_l(\lambda) = \operatorname{diag}\left(\frac{1}{m_1(\lambda)}, \ldots, \frac{1}{m_{p-r}(\lambda)}\right), \tag{20.2}$$

where $m_i(\lambda)$ is a polynomial of degree n_i, and forming $\widetilde{N}_l(\lambda) := M_l(\lambda)N_l(\lambda)$. The resulting basis $\widetilde{N}_l(\lambda)$ has the additional property that the order of any minimal state space realization of $\widetilde{N}_l(\lambda)$ is equal to the sum of orders of the minimal state space realizations of the rows of $\widetilde{N}_l(\lambda)$. Furthermore, $D_l := \lim_{\lambda\to\infty} \widetilde{N}_l(\lambda)$ has full row rank. Such a proper basis is termed *simple* [13] and is the natural counterpart of minimal polynomial basis introduced in [4].

20.3 Computation of Minimal Proper Bases

For the computation of a rational nullspace basis $N_l(\lambda)$ a pencil method based on a state space representation of $G(\lambda)$ has been proposed in [20]. In this section we review this algorithm and give some of the properties of the resulting basis. Although minimal, it appears that the resulting minimal basis is not simple. An approach to obtain simple bases is presented in the next section.

The $p \times m$ rational matrix $G(\lambda)$ can be realized as a descriptor system

$$G(\lambda) := \left[\begin{array}{c|c} A-\lambda E & B \\ \hline C & D \end{array}\right] \tag{20.3}$$

which is an equivalent notation for

$$G(\lambda) = C(\lambda E - A)^{-1}B + D$$

We call this realization *irreducible* if the pair $(A - \lambda E, B)$ is controllable (i.e., $\text{rank}[A - \lambda E\ B] = n$ for all $\lambda \subset \mathbb{C}$) and the pair $(A - \lambda E, C)$ is observable (i.e., $\text{rank}[A^T - \lambda E^T\ C^T] = n$ for all $\lambda \subset \mathbb{C}$) [12], where n is the order of the square matrix A.

The computational method described in [20] exploits the simple fact that $N_l(\lambda)$ is a left nullspace basis of $G(\lambda)$ if and only if for a suitable $M_l(\lambda)$

$$Y_l(\lambda) := [M_l(\lambda)\ N_l(\lambda)] \tag{20.4}$$

is a left nullspace basis of the system matrix

$$S(\lambda) = \begin{bmatrix} A-\lambda E & B \\ C & D \end{bmatrix}. \tag{20.5}$$

Thus, to compute $N_l(\lambda)$ we can determine first a left nullspace basis $Y_l(\lambda)$ for $S(\lambda)$ and then $N_l(\lambda)$ simply results as

$$N_l(\lambda) = Y_l(\lambda)\begin{bmatrix} 0 \\ I_p \end{bmatrix}.$$

$Y_l(\lambda)$ and thus also $N_l(\lambda)$ can be computed by employing linear pencil reduction algorithms based on orthogonal transformations. The main advantage of this approach is that the computation of the nullspace can entirely be done by

manipulating state space matrices instead of manipulating polynomial models. The resulting nullspace is obtained in a descriptor system representation which can be immediately used in applications. In what follows we give some details of this approach.

Let Q and Z be orthogonal matrices (for instance, determined by using the algorithms of [2, 17]) such that the transformed pencil $\widetilde{S}(\lambda) := QS(\lambda)Z$ is in the Kronecker-like staircase form

$$\widetilde{S}(\lambda) = \left[\begin{array}{cc} A_r - \lambda E_r & A_{r,l} - \lambda E_{r,l} \\ \hline 0 & A_l - \lambda E_l \\ \hline 0 & C_l \end{array}\right] \tag{20.6}$$

where the descriptor pair $(A_l - \lambda E_l, C_l)$ is observable, E_l is non-singular, and $A_r - \lambda E_r$ has full row rank excepting possibly a finite set of values of λ (i.e., the invariant zeros of $S(\lambda)$). It follows that we can choose the nullspace $\widetilde{Y}_l(\lambda)$ of $\widetilde{S}(\lambda)$ in the form

$$\widetilde{Y}_l(\lambda) = \left[\, 0 \mid C_l(\lambda E_l - A_l)^{-1} \mid I \,\right]. \tag{20.7}$$

Then the left nullspace of $S(\lambda)$ is $Y_l(\lambda) = \widetilde{Y}_l(\lambda)Q$ and can be obtained easily after partitioning suitably Q as

$$Q = \left[\begin{array}{cc} \widehat{B}_{r,l} & B_{r,l} \\ \widehat{B}_l & B_l \\ \widehat{D}_l & D_l \end{array}\right]$$

where the row partitioning corresponds to the column partitioning of $\widetilde{Y}_l(\lambda)$ in (20.7), while the column partitioning corresponds to the row partitioning of $S(\lambda)$ in (20.5). We obtain

$$Y_l(\lambda) = \left[\begin{array}{c|cc} A_l - \lambda E_l & \widehat{B}_l & B_l \\ \hline C_l & \widehat{D}_l & D_l \end{array}\right] \tag{20.8}$$

and the nullspace of $G(\lambda)$ is

$$N_l(\lambda) = \left[\begin{array}{c|c} A_l - \lambda E_l & B_l \\ \hline C_l & D_l \end{array}\right] \tag{20.9}$$

To obtain this representation of the nullspace basis, we performed exclusively orthogonal transformations on the system matrices. We can prove that all computed matrices are exact for a slightly perturbed original system matrix. It follows that the algorithm to compute the nullspace basis is *numerically backward stable*.

For an irreducible realization (20.3) of $G(\lambda)$, the full column rank subpencil $\left[\begin{array}{c} A_l - \lambda E_l \\ \hline C_l \end{array}\right]$ defines also the left Kronecker structure of $G(\lambda)$ [12]. In our case, for $p > m$ this result can be relaxed asking only for controllability of the realization (20.3). Indeed, it can be easily verified that all unobservable eigenvalues of $A - \lambda E$ appear either as invariant zeros or in the right Kronecker structure and thus

do not affect the left Kronecker structure of the system pencil $S(\lambda)$. This is not anymore true in the case when the realization (20.3) is not controllable. In this case, a part of uncontrollable eigenvalues may appear as invariant zeros, while the rest of them enters in $A_l - \lambda E_l$, thus affecting the left Kronecker structure.

It is possible to obtain the subpencil characterizing the left structure in an observability staircase form

$$\left[\begin{array}{c} A_l - \lambda E_l \\ \hline\hline C_l \end{array}\right] = \left[\begin{array}{c|c|c|c} A_{\ell,\ell+1} & A_{\ell,\ell} - \lambda E_{\ell,\ell} & \cdots & A_{\ell,1} - \lambda E_{\ell,1} \\ \hline & A_{\ell-1,\ell} & \ddots & \vdots \\ \hline & & \ddots & A_{1,1} - \lambda E_{1,1} \\ \hline\hline & & & A_{0,1} \end{array}\right] \tag{20.10}$$

where $A_{i,i+1} \in \mathbb{R}^{\mu_i \times \mu_{i+1}}$, with $\mu_{\ell+1} = 0$, are full column rank upper triangular matrices, for $i = 0, \ldots, \ell$. Note that this form is automatically obtained by using the pencil reduction algorithms described in [2, 17]. The left (or row) Kronecker indices result as follows: there are $\mu_{i-1} - \mu_i$ Kronecker blocks of size $i \times (i-1)$, for $i = 1, \ldots, \ell+1$. The row dimension of $N_l(\lambda)$ (i.e., the number of linearly independent basis vectors) is given by the total number of Kronecker indices, thus $\sum_{i=1}^{\ell+1}(\mu_{i-1} - \mu_i) = \mu_0$. Applying standard linear algebra results, it follows that $\mu_0 := p - r$.

We give now some properties of the computed rational basis.

Theorem 2 *If the realization* (20.3) *of* $G(\lambda)$ *is controllable, then the rational matrix* $N_l(\lambda)$ *defined in* (20.9) *is a minimal proper rational basis of the left nullspace of* $G(\lambda)$.

Proof According to the definition of a minimal proper rational basis [4, 13], its McMillan degree is given by the sum of row indices of a minimal polynomial basis. The order of the computed basis in (20.9) is

$$n_l := \sum_{i=1}^{\ell} \mu_i$$

We have to show that this order is the same as that of an equivalent minimal polynomial basis.

The controllability of the realization (20.3) ensures that the left Kronecker structure of $G(\lambda)$ and of $S(\lambda)$ are characterized by the same Kronecker indices. Instead of the rational basis $\widetilde{Y}_l(\lambda)$ in (20.7), we can directly compute a minimal polynomial basis of the form

$$\widehat{Y}_l(\lambda) = \left[0 \mid \widehat{N}_l(\lambda)\right], \tag{20.11}$$

where $\widehat{N}_l(\lambda)$ is a minimal polynomial basis for the left nullspace of $\left[\begin{array}{c} A_l - \lambda E_l \\ \hline\hline C_l \end{array}\right]$. For this purpose, we can exploit the staircase form (20.10). Using the staircase

form (20.10), it is shown in [2] in a dual context that a minimal polynomial basis can be computed by selecting $\mu_{i-1} - \mu_i$ polynomial basis vectors of degree $i - 1$, for $i = 1, \ldots, \ell + 1$. This basis can be used to construct a minimal rational basis by making each row proper with appropriate order denominators (as shown in Sect. 20.2). The total order of such a basis is

$$\overline{n}_l = \sum_{i=1}^{\ell+1} (\mu_{i-1} - \mu_i)(i-1)$$

But this is exactly n_l, since

$$\begin{aligned} \overline{n}_l &= \sum_{i=1}^{\ell+1} \mu_{i-1}(i-1) - \sum_{i=1}^{\ell+1} \mu_i (i-1) \\ &= \sum_{i=1}^{\ell} \mu_i i - \sum_{i=1}^{\ell} \mu_i (i-1) = \sum_{i=1}^{\ell} \mu_i \end{aligned}$$

To finish this part of the proof, we need to show additionally that the realization (20.9) is minimal. The pair $(A_l - \lambda E_l, C_l)$ is observable, by the construction of the Kronecker-like form (20.6). To show the pair $(A_l - \lambda E_l, B_l)$ is controllable, observe that due to the controllability of the pair $(A - \lambda E, B)$, the sub-pencil $[A - \lambda E \; B]$ has full row rank, and thus the reduced pencil

$$Q \left[\begin{array}{cc|c} A - \lambda E & B & 0 \\ C & D & I_p \end{array}\right] \left[\begin{array}{c|c} Z & 0 \\ 0 & I_p \end{array}\right] = \left[\begin{array}{cc|c} A_r - \lambda E_r & A_{r,l} - \lambda E_{r,l} & B_{r,l} \\ 0 & A_l - \lambda E_l & B_l \\ 0 & C_l & D_l \end{array}\right]$$

has full row rank as well. It follows that

$$\operatorname{rank}[A_l - \lambda E_l \, B_l] = n_l$$

and thus the pair $(A_l - \lambda E_l, B_l)$ is controllable.

Since, we also have that

$$\operatorname{rank} \begin{bmatrix} A_l - \lambda E_l & B_l \\ C_l & D_l \end{bmatrix} = n_l + p - r$$

for all λ, it follows that $N_l(\lambda)$ has no finite or infinite zeros. Thus, D_l has full row rank $p - r$ and the computed basis is column reduced at $\lambda = \infty$ [13]. □

In the case, when the realization of (20.3) of $G(\lambda)$ is not controllable, the realization of $N_l(\lambda)$ is not guaranteed to be controllable. The uncontrollable eigenvalues of $A - \lambda E$ enters partly either as invariant zeros (i.e., part of the sub-pencil $A_r - \lambda E_r$) or are part of the sub-pencil $A_l - \lambda E_l$. Therefore, in this case, the resulting nullspace basis has not the least possible McMillan degree.

Additionally the following important result holds:

Proposition 1 *If the realization* (20.3) *of* $G(\lambda)$ *is controllable, then the realization of* $N_l(\lambda)$ *defined in* (20.9) *is maximally controllable.*

Proof According to a dual formulation of [10], we have to show that for an arbitrary output injection matrix K, the pair $(A_l + KC_l - \lambda E_l, B_l + KD_l)$ remains controllable. Consider the transformation matrix

$$U = \begin{bmatrix} I & 0 & 0 \\ 0 & I & K \\ 0 & 0 & I \end{bmatrix} \tag{20.12}$$

and compute $\widehat{S}(\lambda) := UQS(\lambda)Z$, which, due to the particular form of C_l, is still in the Kronecker-like staircase form

$$\widehat{S}(\lambda) = \left[\begin{array}{cc} A_r - \lambda E_r & A_{r,l} - \lambda E_{r,l} \\ \hline 0 & A_l + KC_l - \lambda E_l \\ \hline 0 & C_l \end{array}\right] \tag{20.13}$$

If we form also

$$UQ \begin{bmatrix} 0 \\ I_p \end{bmatrix} = \begin{bmatrix} B_{r,l} \\ B_l + KD_l \\ D_l \end{bmatrix}$$

we obtain an alternative minimal proper rational basis in the form

$$\widetilde{N}_l(\lambda) = \left[\begin{array}{c|c} A_l + KC_l - \lambda E_l & B_l + KD_l \\ \hline C_l & D_l \end{array}\right] \tag{20.14}$$

We already have proven in Theorem 2 that such a nullspace basis is a minimal realization. Thus, the pair $(A_l + KC_l - \lambda E_l, B_l + KD_l)$ is controllable. □

Even if the above computed rational basis has the least possible McMillan degree, and thus is minimal, still in general, this basis is not simple. In the next section, we consider a postprocessing approach permitting to obtain a simple basis from a non-simple one.

20.4 Computation of Simple Bases

The most obvious approach to determine a simple minimal proper rational basis has been sketched in Sect. 20.2 and consists in computing first a minimal polynomial basis $N_l(\lambda)$ and then to determine the rational basis as $\widetilde{N}_l(\lambda) := M_l(\lambda)N_l(\lambda)$, where $M_l(\lambda)$ has the form (20.2).

We discuss shortly the method to compute a polynomial basis proposed in [2]. This method determines first a minimal polynomial basis $W(\lambda)$ for the left nullspace of the sub-pencil $\begin{bmatrix} A_l - \lambda E_l \\ C_l \end{bmatrix}$ in (20.6). This computation can be done by fully exploiting the staircase structure (20.10) of this pencil. The details for a dual algorithm (for right basis) are presented in [2]. The degrees of the resulting left basis vectors are equal to the left Kronecker indices, and this information can be simply read out from the staircase structure. As already mentioned, there are $p - r$ basis vectors, of which there are $\mu_{i-1} - \mu_i$ vectors of degree $(i - 1)$.

The minimal polynomial nullspace basis of $G(\lambda)$ results as

$$N_l(\lambda) = [0\, W(\lambda)]Q\begin{bmatrix} 0 \\ I_p \end{bmatrix}$$

Note that $W(\lambda)$ and $N_l(\lambda)$ have the same row degrees. Furthermore, it is shown in [2] that the resulting $N_l(\lambda)$ is *row reduced*.

The approach to compute a simple proper minimal basis has been sketched in Sect. 20.2 and additionally involves to determine $M(\lambda)$ of the form (20.2), where $m_i(\lambda)$ is an arbitrary polynomial of degree n_i. The resulting simple proper minimal basis is $\widetilde{N}_l(\lambda) := M(\lambda)N_l(\lambda)$ and has arbitrarily assignable poles. A state-space realization of the resulting basis $\widetilde{N}_l(\lambda)$ can be simply built by inspection, exploiting the simpleness property. This realization is obtained by simply stacking $p - r$ minimal realizations of orders n_i, $i = 1, \ldots, p - r$ of each row of $\widetilde{N}_l(\lambda)$. The resulting state matrix has a block diagonal structure. Although simple, this approach is not always well suited for applications (e.g., in fault detection) for reasons which will become apparent in Sect. 20.7.

We propose an alternative to this method which is based on minimum cover techniques and, as will be shown later, directly supports the design of least order fault detectors. Consider the proper minimal left nullspace (20.9) and denote with $c_{l,i}$ and $d_{l,i}$ the ith rows of matrices C_l and D_l, respectively.

Theorem 3 *For each $i = 1, \ldots, p - r$, let K_i be an output injection matrix such that*

$$v_i(\lambda) := c_{l,i}(\lambda E_l - A_l - K_i C_l)^{-1}(B_l + K_i D_l) + d_{l,i} \tag{20.15}$$

has the least possible McMillan degree. Then, $\widetilde{N}_l(\lambda)$ formed from the $p - r$ rows $v_i(\lambda)$ is a simple proper minimal left nullspace basis.

Proof According to Proposition 1, the realization (20.9) of $N_l(\lambda)$ is maximally controllable, i.e., the pair $(A_l + K_i C_l - \lambda E_l, B_l + K_i D_l)$ is controllable for arbitrary K_i. Therefore, the maximal order reduction of the McMillan degree of $v_i(\lambda)$ can be achieved by making the pair $(A_l + K_i C_l - \lambda E_l, c_{l,i})$ maximally unobservable via an

appropriate choice of K_i. For each $i = 1, \ldots, p-r$, the achievable least McMillan degree of $v_i(\lambda)$ is the corresponding minimal index n_i, representing, in a dual setting, the dimension of the least order controllability subspace of the standard pair $(E_l^{-T}A_l^T, E_l^{-T}C_l^T)$ containing $\mathrm{span}(E_l^{-T}c_{l,i}^T)$. This result is the statement of Lemma 6 in [29]. It is easy to check that $v_i(\lambda)G(\lambda) = 0$, thus $\widetilde{N}_l(\lambda)$ is a left annihilator of $G(\lambda)$. Furthermore, the set of vectors $\{v_1(\lambda), \ldots, v_{p-r}(\lambda)\}$ is linearly independent since the realization of $\widetilde{N}_l(\lambda)$ has the same full row rank matrix D_l as that of $N_l(\lambda)$. It follows that $\widetilde{N}_l(\lambda)$ is a proper left nullspace basis of least dimension $\sum_{i=1}^{p-r} n_i$, with each row $v_i(\lambda)$ of McMillan degree n_i. It follows that $N_l(\lambda)$ is simple. □

The poles of the nullspace basis can be arbitrarily placed by performing left coprime rational factorizations

$$v_i(\lambda) = m_i(\lambda)^{-1}\hat{v}_i(\lambda)$$

The basis $\widehat{N}_l(\lambda) := [\hat{v}_1^T(\lambda), \ldots, \hat{v}_{p-r}^T(\lambda)]^T$ obtained in this way, can have arbitrarily assigned poles.

Simple bases are the direct correspondents of polynomial bases, and therefore each operation on a polynomial basis has a direct correspondent operation on the corresponding simple rational basis. An important operation (with applications in fault detection) is building linear combinations of basis vectors up to a certain McMillan degree.

Consider the proper left nullspace basis $N_l(\lambda)$ constructed in (20.9). By looking to the details of the resulting staircase form (20.10) of the pair $(A_l - \lambda E_l, C_l)$, recall that the full column rank matrices $A_{i-1,i} \in \mathbb{R}^{\mu_{i-1} \times \mu_i}$ have the form

$$A_{i-1,i} = \begin{bmatrix} R_{i-1,i} \\ 0, \end{bmatrix}$$

where $R_{i-1,i}$ is an upper-triangular invertible matrix of order μ_i. The row dimension $\mu_{i-1} - \mu_i$ of the zero block of $A_{i-1,i}$ gives the number of polynomial vectors of degree $i-1$ in a minimal polynomial basis [2, Section 4.6] and thus, also the number of vectors of McMillan degree $i-1$ in a simple basis. It is straightforward to show the following result.

Corollary 1 *For a given left nullspace basis$N_l(\lambda)$ in the form (20.9), let $1 \le i < p-r$be a given index and lethbe a $(p-r)$-dimensional row vector having only the lasticomponents non-zero. Then, a linear combination of the simple basis vectors not exceeding McMillan degree n_i can be generated as*

$$v(\lambda) := hC_l(\lambda E_l - A_l - KC_l)^{-1}(B_l + KD_l) + hD_l \tag{20.16}$$

where K is an output injection matrix such that $v(\lambda)$ has the least possible McMillan degree.

This result shows that the determination of a linear combination of vectors of a simple basis up to a given order n_i is possible directly from a proper basis determined in the form (20.9). As it will be shown in the next section, the matrix K together with a minimal realization of $v(\lambda)$ can be computed efficiently using minimal dynamic cover techniques. The same approach can be applied repeatedly to determine the basis vectors $v_i(\lambda)$, $i = 1, \ldots, p - r$, of a simple basis by using the particular choices $h = e_i^T$, where e_i is the ith column of the $(p - r)$th order identity matrix.

20.5 Minimal Dynamic Cover Techniques

Let $N_l(\lambda)$ be the $(p - r) \times p$ minimal proper left nullspace basis of $G(\lambda)$ constructed in (20.9). In this section we will address the following computational problem encountered when computing simple proper bases or when computing linear combination of basis vectors with least McMillan degree: given a row vector h, determine the output injection matrix K such that the vector $v(\lambda)$ in (20.16) has least McMillan degree. As already mentioned, minimal dynamic cover techniques can be employed to perform this computation.

Computational procedures of minimal dynamic covers are presented in [22] (see also Appendix A). The general idea of the cover algorithms is to perform a similarity transformation on the system matrices in (20.9) to bring them in a *special form* which allows to cancel the maximum number of unobservable eigenvalues. In a dual setting, for the so-called *Type I* dynamic covers [8], two nonsingular transformation matrices L and V result such that

$$\left[\begin{array}{c} N_l(\lambda) \\ hN_l(\lambda) \end{array}\right] = \left[\begin{array}{c|c} L(A_l - \lambda E_l)V & LB_l \\ \hline C_lV & D_l \\ hC_lV & hD_l \end{array}\right] = \left[\begin{array}{cc|c} \widehat{A}_{11} - \lambda E_{11} & \widehat{A}_{12} - \lambda E_{12} & \widehat{B}_1 \\ \widehat{A}_{21} & \widehat{A}_{22} - \lambda E_{22} & \widehat{B}_2 \\ \hline \widehat{C}_{11} & \widehat{C}_{12} & D_l \\ 0 & \widehat{c}_{22} & hD_l \end{array}\right], \tag{20.17}$$

where the pairs $(\widehat{A}_{11} - \lambda E_{11}, \widehat{C}_{11})$ and $(\widehat{A}_{22} - \lambda E_{22}, \widehat{c}_{22})$ are observable, and the submatrices $\widehat{C}_{11}$ and $\widehat{A}_{21}$ have the particular structure

$$\left[\begin{array}{c} \widehat{A}_{21} \\ \widehat{C}_{11} \end{array}\right] = \left[\begin{array}{cc} 0 & A_{21} \\ 0 & C_{11} \end{array}\right]$$

with C_{11} having full column rank. By taking

$$K = V \left[\begin{array}{c} 0 \\ \widehat{K} \end{array}\right]$$

with $\widehat{K}$ satisfying $\widehat{K}C_{11} + A_{21} = 0$, we annihilate $\widehat{A}_{21}$, and thus make the pair $(A_l + KC_l - \lambda E_l, hC_l)$ maximally unobservable by making all eigenvalues of $\widehat{A}_{11} - \lambda E_{11}$ unobservable. The resulting vector $v(\lambda)$ of least McMillan degree, obtained by deleting the unobservable part, has the minimal state space realization

$$v(\lambda) = \left[\begin{array}{c|c} \widehat{A}_{22} + \widehat{K}\widehat{C}_{12} - \lambda E_{22} & \widehat{B}_2 + \widehat{K}D_l \\ \hline \widehat{c}_{22} & hD_l \end{array}\right] \tag{20.18}$$

This is also the typical form of achieved realizations for the basis vectors (20.15) of a simple basis. To obtain the above realization, the computation of the transformation matrices L and V is not necessary, provided all transformations which are performed during the reductions in the minimal cover algorithm are applied to the input matrix B_l as well. In Appendix A we present a detailed algorithm for the computation of *Type I* dynamic covers.

20.6 Computation of Proper Coprime Factorizations

We present a straightforward application of minimal proper nullspaces in determining proper fractional factorizations of improper rational matrices. This computation is often a preliminary preprocessing step when designing residual generator filters for solving the optimal fault detection problem involving improper systems [25]. Let $G(\lambda)$ be a given $p \times m$ improper rational matrix for which we want to determine a fractional representation in the form

$$G(\lambda) = M^{-1}(\lambda)N(\lambda), \tag{20.19}$$

where both $M(\lambda)$ and $N(\lambda)$ are proper. In applications, the stability of the factors is frequently imposed as an additional requirement. For this computation, state space techniques have been proposed in [18], based on stabilization and pole assignment methods for descriptor systems. We show, that alternatively a conceptually simple and numerically reliable approach can be used to obtain the above factorization.

The relation (20.19) can be rewritten as

$$[M(\lambda)\, N(\lambda)] \begin{bmatrix} G(\lambda) \\ -I_m \end{bmatrix} = 0.$$

It follows that the $p \times (p + m)$ rational matrix $[M(\lambda)\ N(\lambda)]$ can be determined as a minimal proper left nullspace basis of the full column rank matrix

$$G_e(\lambda) = \begin{bmatrix} G(\lambda) \\ -I_m \end{bmatrix}.$$

The invertibility of $M(\lambda)$ is guaranteed by Lemma 2 of [28] by observing that $[M(\lambda)\ N(\lambda)]$, as a nullspace basis, has full row rank.

Using the state-space realizations based algorithm described in Sect. 20.3, we obtain the left nullspace basis $[M(\lambda)\ N(\lambda)]$ of $G_e(\lambda)$ in the form (20.9) with the matrices B_l and D_l partitioned accordingly

$$\left[\, M(\lambda)\ N(\lambda) \,\right] = \left[\begin{array}{c|cc} A_l - \lambda E_l & B_{M,l} & B_{N,l} \\ \hline C_l & D_{M,l} & D_{N,l} \end{array}\right]. \tag{20.20}$$

Since E_l is invertible, the resulting factors are proper. An important aspect of this simple approach is that the state space realizations (20.20) of the factors of the proper factorization (20.19) have been obtained using exclusively orthogonal transformations to reduce the system matrix of $G_e(\lambda)$ to a Kronecker-like form as that in (20.6). This contrasts with the algorithms of [18] which involve also some non-orthogonal manipulations. The stability of the resulting factors can be enforced, using instead (20.9), a representation of the form (20.14) for the left nullspace. Here, K is determined to fulfill the stability requirements.

20.7 Operations Involving Nullspace Bases

Assume that besides the $p \times m$ rational matrix $G(\lambda)$, we have given also a $p \times q$ rational matrix $F(\lambda)$, and the compound matrix $[G(\lambda)\ F(\lambda)]$ has the state space realization

$$[\, G(\lambda)\ F(\lambda) \,] = \left[\begin{array}{c|cc} A - \lambda E & B & B_f \\ \hline C & D & D_f \end{array}\right]. \tag{20.21}$$

Observe that the realizations of $G(\lambda)$ and $F(\lambda)$ share the same state, descriptor and output matrices A, E, and C respectively. Let $N_l(\lambda)$ be a proper left nullspace basis of $G(\lambda)$ which can be non-simple in the form in (20.9) or a simple basis formed with vectors of the form (20.15). In several applications, besides the computation of the nullspace basis, operations with the basis matrix are necessary. For example, the left multiplications $N_l(\lambda)F(\lambda)$ or $\widetilde{N}_l(\lambda)F(\lambda)$, where $N_l(\lambda) = M_l^{-1}(\lambda)\widetilde{N}_l(\lambda)$ is a left coprime factorization, are often necessary in fault detection applications. Important are also operations involving a linear combination of the basis vectors, i.e., the computation of $v(\lambda)F(\lambda)$, where $v(\lambda)$ has the form (20.16) or is in a minimal form (20.18) as resulted from the application of the minimal cover algorithm. This last operation is also important when computing $N_l(\lambda)F(\lambda)$ with $N_l(\lambda)$ a simple proper left nullspace basis formed from row vectors of the form (20.15).

The determination of state space realizations of products like $N_l(\lambda)F(\lambda)$, $\widetilde{N}_l(\lambda)F(\lambda)$ or $v(\lambda)F(\lambda)$ can be done by computing minimal realizations of the state

space realizations of these rational matrix products. The computation of a minimal realization relies on numerically stable algorithms for standard or descriptor systems as those proposed in [14, 15]. However, these algorithms depend on intermediary rank determinations and thus can produce results which critically depend on the choice of threshold values used to detect zero elements. Since it is always questionable that the resulting order is the correct one, this computational approach can be categorized as a difficult numerical computation. Alternative ways relying on balancing related model reduction are primarily intended for standard stable systems. The application of this approach in the case when $F(\lambda)$ is unstable or not proper leads to other types of numerical difficulties. For example, by assuming that the unstable/improper part cancels completely out, a preliminary spectral splitting of eigenvalues must be performed first, which is often associated with unnecessary accuracy losses. For polynomial nullspace bases the only alternative to the above approach is to manipulate polynomial matrices. However, as already mentioned, in some applications this leads to unavoidable detours (state-space to polynomial model conversions) which involve delicate rank decisions as well.

In what follows, we show that all these numerical difficulties to evaluate the above products can be completely avoided and explicit state space realizations for these products can be obtained as a natural byproduct of the nullspace computation procedure. An important aspect of the developed explicit realizations is that both $N_l(\lambda)$ and $N_l(\lambda)F(\lambda)$ share the same state, descriptor and output matrices. Thus, the developed formulas are also useful for performing nullspace updating and two important applications of this techniques in the context of fault detection are presented in the next section.

20.7.1 Left Multiplication with a Non-simple Basis

Let $N_l(\lambda)$ be a proper left nullspace basis of $G(\lambda)$ computed in the form (20.9) and let $Y_l(\lambda)$ be the left nullspace basis of $S(\lambda)$ in (20.4). It is easy to show that

$$Y_l(\lambda)\left[\begin{array}{c|c} A-\lambda E & B_f \\ C & D_f \end{array}\right] = \left[\begin{array}{c|c} 0 & N_l(\lambda)F(\lambda) \end{array}\right]$$

and thus

$$N_l(\lambda)F(\lambda) = Y_l(\lambda)\begin{bmatrix} B_f \\ D_f \end{bmatrix} = \widetilde{Y}_l(\lambda)Q\begin{bmatrix} B_f \\ D_f \end{bmatrix},$$

where Q is the orthogonal transformation matrix used in computing the Kronecker-like form (20.6) and $\widetilde{Y}_l(\lambda)$ is defined in (20.7). We compute now

$$Q \begin{bmatrix} B_f \\ D_f \end{bmatrix} = \left[\begin{array}{c} * \\ \hline \widetilde{B}_f \\ \hline \widetilde{D}_f \end{array}\right], \tag{20.22}$$

where the row partitioning of the right hand side corresponds to the column partitioning of $\widetilde{Y}_l(\lambda)$ in (20.7). The realization of $N_l(\lambda)F(\lambda)$ results as

$$N_l(\lambda)F(\lambda) = \left[\begin{array}{c|c} A_l - \lambda E_l & \widetilde{B}_f \\ \hline C_l & \widetilde{D}_f \end{array}\right]. \tag{20.23}$$

Note that to compute this realization, only orthogonal transformations have been employed.

In assessing the properties of the resulting realization (20.23), two aspects are relevant. The realizations of $N_l(\lambda)$ and $N_l(\lambda)F(\lambda)$ are observable since they share the same A_l, E_l and C_l matrices. However, the realization in (20.23) may not be minimal, because its controllability also depends on the involved B_f and D_f. The second aspect concerns the minimality of the proper left nullspace basis $N_l(\lambda)$ itself. When computing $N_l(\lambda)$, we can freely assume that that the overall realization of $[G(\lambda)\ F(\lambda)]$ is irreducible. According to Proposition 2, to obtain a minimal proper basis for the left nullspace of $G(\lambda)$ using the proposed rational nullspace procedure, the corresponding realization in (20.21) must be controllable. Although this condition is usually fulfilled in fault detection applications (see Sect. 20.8.1), still the realization of $G(\lambda)$ can be in general uncontrollable, and therefore the resulting left nullspace basis $N_l(\lambda)$ may not have the least possible McMillan degree. This can be also the case for the resulting realization (20.23) of $N_l(\lambda)F(\lambda)$. These two aspects are the reasons why the order of the resulting realization of $N_l(\lambda)F(\lambda)$ in (20.23) may exceed the least possible one (which can be obtained by working exclusively with minimal realizations and employing the already mentioned minimal realization techniques).

20.7.2 Left Coprime Factorization

Assume $N_l(\lambda)$ be the left nullspace basis in (20.9). In several applications, this rational basis must be stable, that is, to have in a continuous-time setting only poles with negative real parts, or in a discrete-time setting poles inside the unit circle of the complex plane. As already mentioned, instead $N_l(\lambda)$ we can freely use as left nullspace basis $\widetilde{N}_l(\lambda)$, the denominator factor of the left fractional representation

$$N_l(\lambda) = M_l^{-1}(\lambda)\widetilde{N}_l(\lambda), \tag{20.24}$$

where $M_l(\lambda)$ and $\widetilde{N}_l(\lambda)$ are rational matrices with poles in appropriate stability domains. A state space realization of $[\widetilde{N}_l(\lambda)\ M_l(\lambda)]$ is given by well known formulas [32]

$$[\,\widetilde{N}_l(\lambda)\ M_l(\lambda)\,] = \left[\begin{array}{c|cc} A_l + KC_l - \lambda E_l & B_l + KD_l & K \\ \hline C_l & D_l & I \end{array}\right], \tag{20.25}$$

where K is an appropriate output injection matrix which assigns the eigenvalues of $A_l + KC_l - \lambda E_l$ in desired positions or in a suitable stability domain. Recall that this is always possible, since the pair $(A_l - \lambda E_l, C_l)$ is observable. Numerically reliable algorithms to determine a suitable K can be used based on pole assignment or stabilization techniques [16]. Alternatively, recursive factorization techniques as those proposed in [18] can be employed.

With U of the form (20.12), we can compute

$$UQ\begin{bmatrix} B_f \\ D_f \end{bmatrix} = \begin{bmatrix} * \\ \widetilde{B}_f + K\widetilde{D}_f \\ \widetilde{D}_f \end{bmatrix}$$

and in a completely similar way as in the previous subsection, we can obtain the realization of $\widetilde{N}_l(\lambda)F(\lambda)$ as

$$\widetilde{N}_l(\lambda)F(\lambda) = \left[\begin{array}{c|c} A_l + KC_l - \lambda E_l & \widetilde{B}_f + K\widetilde{D}_f \\ \hline C_l & \widetilde{D}_f \end{array}\right].$$

When employing the algorithms in [18], a supplementary orthogonal similarity transformation is also implicitly applied to the resulting system matrices, such that the resulting pencil $A_l + KC_l - \lambda E_l$ is in a quasi-triangular (generalized real Schur) form. This computation can be seamlessly integrated into the evaluation of $\widetilde{N}_l(\lambda)F(\lambda)$ if we perform the left coprime factorization algorithm directly to the compound matrix realization

$$[\,N_l(\lambda)\ N_l(\lambda)F(\lambda)\,] = \left[\begin{array}{c|cc} A_l - \lambda E_l & B_l & \widetilde{B}_f \\ \hline C_l & D_l & \widetilde{D}_f \end{array}\right].$$

In the case of a simple basis, this technique can be employed by considering fractional representations of the form (20.24) with $M_l(\lambda)$ diagonal. In this case the same algorithm can be applied to each row of $[N_l(\lambda)\ N_l(\lambda)F(\lambda)]$, by exploiting the block-diagonal structure of the underlying $A_l - \lambda E_l$ to increase the efficiency of computations.

20.7.3 Left Multiplication with a Simple Nullspace Basis

We show first how to compute $v(\lambda)F(\lambda)$, where $v(\lambda)$ is given in (20.16). The same formula applies for a vector of the form (20.15), with obvious replacements. By observing that $v(\lambda) = h\widetilde{N}_l(\lambda)$ with $\widetilde{N}_l(\lambda)$ having the form (20.25), it follows immediately

$$v(\lambda)F(\lambda) = \left[\begin{array}{c|c} A_l + KC_l - \lambda E_l & \widetilde{B}_f + K\widetilde{D}_f \\ \hline hC_l & h\widetilde{D}_f \end{array}\right].$$

In the case when K has been obtained from the cover algorithm, the minimal realization of $v(\lambda)$ (after eliminating the unobservable part) is given in (20.18). The corresponding realization of $v(\lambda)F(\lambda)$ is

$$v(\lambda)F(\lambda) = \left[\begin{array}{c|c} \widehat{A}_{22} + \widehat{K}\widehat{C}_{12} - \lambda E_{22} & \widehat{B}_{f,2} + \widehat{K}\widetilde{D}_f \\ \hline \widehat{c}_{22} & h\widetilde{D}_f \end{array}\right]$$

where we used

$$L\widetilde{B}_f = \left[\begin{array}{c} \widehat{B}_{f,1} \\ \widehat{B}_{f,2} \end{array}\right]$$

with the transformation matrix L employed in (20.17) and the row partition corresponding to that of the input matrix LB_l in (20.17). Note that the explicit computation of transformation matrix L is not necessary, because the minimal realization of the product $v(\lambda)F(\lambda)$ can be directly obtained by applying the performed transformations in the minimal cover algorithm (see Appendix A) to the input matrices of the compound realization

$$\left[\begin{array}{cc} N_l(\lambda) & N_l(\lambda)F(\lambda) \\ hN_l(\lambda) & hN_l(\lambda)F(\lambda) \end{array}\right] = \left[\begin{array}{c|cc} A_l - \lambda E_l & B_l & \widetilde{B}_f \\ \hline C_l & D_l & \widetilde{D}_f \\ hC_l & hD_l & h\widetilde{D}_f \end{array}\right].$$

To compute the products $v_i(\lambda)F(\lambda)$, for $i = 1, \ldots, p - r$, the same approach can be used taking into account the particular form of $h = e_i^T$.

If $N_l(\lambda)$ is a simple basis formed from row vectors of the form (20.16), then the resulting state space realization for $N_l(\lambda)F(\lambda)$ is obtained by stacking the realizations of $v_i(\lambda)F(\lambda)$ for $i = 1, \ldots, p - r$. Also in this case, $N_l(\lambda)$ and $N_l(\lambda)F(\lambda)$ will share the same state, descriptor and output matrices (i.e., A_l, E_l, C_l), and the pole pencil $A_l - \lambda E_l$ will have a block diagonal form, where the dimensions of the diagonal blocks are the minimal indices n_i.

20.8 Applications to Fault Detection

We consider the linear time-invariant system described by input–output relations of the form

$$\mathbf{y}(\lambda) = G_u(\lambda)\mathbf{u}(\lambda) + G_d(\lambda)\mathbf{d}(\lambda) + G_f(\lambda)\mathbf{f}(\lambda), \tag{20.26}$$

where $\mathbf{y}(\lambda)$, $\mathbf{u}(\lambda)$, $\mathbf{d}(\lambda)$, and $\mathbf{f}(\lambda)$ are Laplace- or Z-transformed vectors of the p-dimensional system output vector $y(t)$, m_u-dimensional control input vector $u(t)$, m_d-dimensional disturbance vector $d(t)$, and m_f-dimensional fault signal vector $f(t)$, respectively, and where $G_u(\lambda)$, $G_d(\lambda)$ and $G_f(\lambda)$ are the TFMs from the control inputs to outputs, disturbances to outputs, and fault signals to outputs, respectively.

In what follows we will address three applications of the techniques developed in the previous sections.

20.8.1 Solving Fault Detection Problems with Least Order Detectors

The following is the standard formulation of the *Fault Detection Problem* (FDP): Determine a proper and stable linear residual generator (or fault detector) having the general form

$$\mathbf{r}(\lambda) = R(\lambda)\begin{bmatrix}\mathbf{y}(\lambda)\\ \mathbf{u}(\lambda)\end{bmatrix} \tag{20.27}$$

such that: (i) $r(t) = 0$ when $f(t) = 0$ for all $u(t)$ and $d(t)$; and (ii) $r(t) \neq 0$ when $f_i(t) \neq 0$, for $i = 1, \ldots, m_f$. Besides the above requirements it is often required for practical use that the TFM of the detector $R(\lambda)$ has the least possible McMillan degree. Note that as fault detector, we can always choose $R(\lambda)$ as a rational row vector.

The requirements (i) and (ii) can be easily transcribed into equivalent algebraic conditions. The (decoupling) condition (i) is equivalent to

$$R(\lambda)G(\lambda) = 0, \tag{20.28}$$

where

$$G(\lambda) = \begin{bmatrix} G_u(\lambda) & G_d(\lambda) \\ I_{m_u} & 0 \end{bmatrix}, \tag{20.29}$$

while the (detectability) condition (ii) is equivalent to

$$R_{f_i}(\lambda) \neq 0, \quad i = 1, \ldots, m_f, \tag{20.30}$$

where $R_{f_i}(\lambda)$ is the ith column of

$$R_f(\lambda) := R(\lambda)\begin{bmatrix} G_f(\lambda) \\ 0 \end{bmatrix}. \tag{20.31}$$

Let $G_{f_i}(\lambda)$ be the ith column of $G_f(\lambda)$. A necessary and sufficient condition for the existence of a solution is the following one [3, 11]:

Theorem 4 *For the system* (20.26) *the FDP is solvable if and only if*

$$\operatorname{rank}[G_d(\lambda)\ G_{f_i}(\lambda)] > \operatorname{rank} G_d(\lambda), \quad i = 1, \ldots, m_f \tag{20.32}$$

From (20.28) it appears that $R(\lambda)$ is a left annihilator of $G(\lambda)$, thus one possibility to determine $R(\lambda)$ is to compute first a left minimal basis $N_l(\lambda)$ for the *left nullspace* of $G(\lambda)$, and then to build a stable scalar output detector as

$$R(\lambda) = h(\lambda)N_l(\lambda), \tag{20.33}$$

representing a linear combination of the rows of $N_l(\lambda)$, such that conditions (20.30) are fulfilled. The above expression represents a parametrization of *all* possible scalar output fault detectors and is the basis of the so-called *nullspace methods*.

The first nullspace method to design residual generators for fault detection has been formally introduced in [5], where a polynomial basis based approach was used. This approach has been later extended to rational bases in [20, 24]. The main advantage of the nullspace approach is that the least order design aspect is naturally present in the formulation of the method. In a recent survey [26], it was shown that the nullspace method also provides a unifying design paradigm for most of existing approaches, which can be interpreted as special cases of this method.

Consider a *descriptor* state space realization of (20.26)

$$\begin{aligned} E\lambda x(t) &= Ax(t) + B_u u(t) + B_d d(t) + B_f f(t) \\ y(t) &= Cx(t) + D_u u(t) + D_d d(t) + D_f f(t), \end{aligned} \tag{20.34}$$

where $\lambda x(t) = \dot{x}(t)$ or $\lambda x(t) = x(t+1)$ depending on the type of the system, continuous or discrete, respectively. For convenience, in what follows we assume the pair $(A - \lambda E, C)$ is observable and the pair $(A - \lambda E, [B_u\ B_d])$ is controllable. This latter condition is typically fulfilled when considering actuator and sensor faults. In this case, B_f has partly the same columns as B_u (in the case of actuator faults) or zero columns (in the case of sensor faults).

$G(\lambda)$ defined in (20.29) has the irreducible realization

$$G(\lambda) = \left[\begin{array}{c|cc} A - \lambda E & B_u & B_d \\ \hline C & D_u & D_d \\ 0 & I_{m_u} & 0 \end{array}\right].$$

Using the method described in Sect. 20.3, we compute first a minimal proper left nullspace basis $N_l(\lambda)$ of $G(\lambda)$. The state space realization of the $(p-r) \times (p+m_u)$ TFM $N_l(\lambda)$ is given by (20.9), where r is the rank of $G_d(\lambda)$.

To check the existence conditions of Theorem 4, we use (20.23) to compute

$$N_f(\lambda) := N_l(\lambda) \begin{bmatrix} G_f(\lambda) \\ 0 \end{bmatrix} = \left[\begin{array}{c|c} A_l - \lambda E_l & \widetilde{B}_f \\ \hline C_l & \widetilde{D}_f \end{array}\right], \tag{20.35}$$

where

$$Q \begin{bmatrix} B_f \\ D_f \\ 0 \end{bmatrix} = \begin{bmatrix} * \\ \widetilde{B}_f \\ \widetilde{D}_f \end{bmatrix}.$$

Since the pair $(A_l - \lambda E_l, C_l)$ is observable, checking the condition (20.30) is equivalent to verify that

$$\begin{bmatrix} \widetilde{B}_{f_i} \\ \widetilde{D}_{f_i} \end{bmatrix} \neq 0, \quad i = 1, \ldots, m_f,$$

where $\widetilde{B}_{f_i}$ and $\widetilde{D}_{f_i}$ denote the ith columns of $\widetilde{B}_f$ and $\widetilde{D}_f$, respectively.

To address the determination of least order scalar output detectors, we can compute linear combinations of the basis vectors of a simple proper basis of increasing McMillan degrees and check the detectability condition (20.30) for the resulting vectors (seen as candidate detectors). According to Corollary 1, this comes down to choose an appropriate h and obtain the corresponding K such that the row vector $v(\lambda) = h\widetilde{N}_l(\lambda)$ in (20.16) has the least possible McMillan order. Note that in general, with a randomly generated h, one achieves a detector whose order is ℓ, the maximum degree of a minimal polynomial basis. Recall that ℓ is the number of nonzero subdiagonal blocks in the Kronecker-like form (20.10) and represents the observability index of the observable pair $(A_l - \lambda E_l, C_l)$. In the case when no disturbance inputs are present, this is a well know result in designing functional observers [9].

Lower orders detectors can be obtained using particular choices of the row vector h. Using Corollary 1, by choosing h with only the trailing i components nonzero, the corresponding linear combination of i basis vectors has McMillan degree n_i. A systematic search can be performed by generating successive candidates for h with increasing number of nonzero elements and checking for the resulting residual generator the conditions (20.30). The resulting detectors have non-decreasing orders and thus the first detector satisfying these conditions represents a satisfactory least order design. To speed up the selection, the choice of the nonzero components of h can be done such that for a given tentative order n_i a combination of all $\mu_0 - \mu_i$ intervening vectors of order less than or equal to n_i is built. In this way, repeated checks for the same order are avoided and the search is terminated in at most ℓ steps.

For the final design, the resulting dynamics of the detector can be arbitrarily assigned by choosing the detector in the form

$$R(\lambda) = m(\lambda)h\widetilde{N}_l(\lambda),$$

where $m(\lambda)$ is an appropriate scalar transfer function. Note that the resulting least order at previous step is preserved provided $m(\lambda)$ is computed using coprime factorization techniques [18].

20.8.2 Solving Fault Isolation Problems

The more advanced functionality of fault isolation (i.e., exact location of faults) can be often achieved by designing a bank of fault detectors [6] or by direct design of fault isolation filters [21]. Designing detectors which are sensitive to some faults and insensitive to others can be reformulated as a *standard* FDP, by formally redefining the faults to be rejected in the residual as fictive disturbances.

Let $R(\lambda)$ be a given detector and let $R_f(\lambda)$ be the corresponding fault-to-residual TFM in (20.31). We define the *fault signature matrix* S, with the (i, j) entry S_{ij} given by

$$\begin{aligned} S_{ij} &= 1, \text{if the } (i,j) \text{ entry of } R_f(\lambda) \text{ is nonzero;} \\ S_{ij} &= 0, \text{if the } (i,j) \text{ entry of } R_f(\lambda) \text{ is zero.} \end{aligned}$$

If $S_{ij} = 1$, then we say that the fault j is *detected* in residual i and if $S_{ij} = 0$, then the fault j is *decoupled* (not detected) in residual i.

The following *fault detection and isolation problem* (FDIP) can be now formulated: Given a $q \times m_f$ fault signature matrix S determine a bank of q stable and proper scalar output residual generator filters

$$\mathbf{r}_i(\lambda) = R^i(\lambda)\begin{bmatrix} \mathbf{y}(\lambda) \\ \mathbf{u}(\lambda) \end{bmatrix}, \quad i = 1, \ldots, q \tag{20.36}$$

such that, for all $u(t)$ and $d(t)$ we have:

(i) $r_i(t) = 0$ when $f_j(t) = 0$, $\forall\, j$ with $S_{ij} \neq 0$;
(ii) $r_i(t) \neq 0$ when $f_j(t) \neq 0$, $\forall\, j$ with $S_{ij} \neq 0$.

In this formulation of the FDIP, each scalar output detector $R^i(\lambda)$ achieves the fault signature specified by the ith row of the desired fault signature matrix S. The resulting global detector corresponding to this S can be assembled as

$$R(\lambda) = \begin{bmatrix} R^1(\lambda) \\ \vdots \\ R^q(\lambda) \end{bmatrix} \tag{20.37}$$

Let S be a given $q \times m_f$ fault signature matrix and denote by $\overline{G}_f^i(\lambda)$ the matrix formed from the columns of $G_f(\lambda)$ whose column indices j correspond to zero elements in row i of S. The solvability conditions of the FDIP build up from the solvability of q individual FDPs.

Theorem 5 *For the system* (20.26) *the FDIP with the given fault signature matrix S is solvable if and only if for each $i = 1, \ldots, q$, we have*

$$\text{rank}[G_d(\lambda)\ \overline{G}_f^i(\lambda)\ G_{f_j}(\lambda)] > \text{rank}[G_d(\lambda)\ \overline{G}_f^i(\lambda)] \tag{20.38}$$

for all j such that $S_{ij} \neq 0$.

The *standard approach* to determine $R(\lambda)$ is to design for each row i of the fault signature matrix S, a detector $R^i(\lambda)$ which generates the ith residual signal $r_i(t)$, and thus represents the ith row of $R(\lambda)$. For this purpose, the nullspace method of the previous subsection can be applied with $G(\lambda)$ in (20.29) replaced by

$$G(\lambda) = \begin{bmatrix} G_u(\lambda) & G_d(\lambda) & \overline{G}_f^i(\lambda) \\ I_{m_u} & 0 & 0 \end{bmatrix}$$

and with a redefined fault to output TFM $\widetilde{G}_f^i(\lambda)$, formed from the columns of $G_f(\lambda)$ whose indices j correspond to $S_{ij} \neq 0$. The McMillan degree of the global detector (20.37) is bounded by the sum of the McMillan degrees of the component detectors. Note that this upper bound can be effectively achieved, for example, by choosing mutually different poles for the individual detectors. It is to be expected that lower orders result when the scalar detectors share their poles.

Using the least order design techniques described in this paper, for each row of S we can design a scalar output detector of least McMillan degree. However, even if each detector has the least possible order, there is generally no guarantee that the resulting order of $R(\lambda)$ is also the least possible one. To the best of our knowledge, the determination of a detector of least global McMillan degree for a given specification S is still an open problem. A solution to this problem has been recently suggested in [24] and involves a post processing step as follows.

Assume that the resulting least order scalar detector $R^i(\lambda)$ has McMillan degree v_i, for $i = 1, \ldots, q$. We can easily ensure that for $v_i \leq v_j$, the poles of $R^i(\lambda)$ are among the poles of $R^j(\lambda)$. The resulting global detector $R(\lambda)$ according to (20.37) has a McMillan degree which is conjectured in [24] to be the least possible one.

We describe now an *improved approach* in two steps to design a bank of detectors, which for larger values of q, is potentially more efficient than the above standard approach. In a first step, we can reduce the complexity of the original problem by decoupling the influences of disturbances and control inputs on the residuals. In a second stage, a residual generation filter is determined for a system without control and disturbance inputs which achieves the desired fault signature.

Let $N_l(\lambda)$ be a minimal left nullspace basis for $G(\lambda)$ defined in (20.29) and define a new system without control and disturbance inputs as

$$\widetilde{\mathbf{y}}(\lambda) := N_f(\lambda)\mathbf{f}(\lambda), \tag{20.39}$$

where

$$N_f(\lambda) := N_l(\lambda)\begin{bmatrix} G_f(\lambda) \\ 0 \end{bmatrix}. \tag{20.40}$$

The system (20.39) has generally a reduced McMillan degree and also a reduced number of outputs $p - r$, where r is the normal rank of $G_d(\lambda)$. The state space realization of the resulting $N_f(\lambda)$ is given in (20.35). Observe that $N_l(\lambda)$ and $N_f(\lambda)$ share the same state, descriptor and output matrices in their realizations.

For the reduced system (20.39) with TFM $N_f(\lambda)$ we can determine, using the standard approach, a bank of q scalar output least order detectors of the form

$$\mathbf{r}_i(\lambda) = \widetilde{R}^i(\lambda)\widetilde{\mathbf{y}}(\lambda), \quad i = 1, \ldots, q \tag{20.41}$$

such that the same conditions are fulfilled as for the original FDIP. The TFM of the final detector can be assembled as

$$R(\lambda) = \begin{bmatrix} \widetilde{R}^1(\lambda) \\ \vdots \\ \widetilde{R}^q(\lambda) \end{bmatrix} N_l(\lambda) \tag{20.42}$$

Comparing (20.42) and (20.37) we have

$$R^i(\lambda) = \widetilde{R}^i(\lambda)N_l(\lambda), \tag{20.43}$$

which can be also interpreted as an updating formula of a preliminary (incomplete) design. The resulting order of the ith detector is the same as before, but this two steps approach has the advantage that the nullspace computation and the associated least order design involve systems of reduced orders (in the sizes of state, input and output vectors). The realization of $R^i(\lambda)$ can be obtained using the explicit formulas derived in Sect. 20.7.1.

The improved approach relies on the detector updating techniques which can be easily performed using the explicit realizations of the underlying products. This can be seen as a major advantage of rational nullspace based methods in contrast to polynomial nullspace based computations.

The above procedure has been used for the example studied in [31, Table 2], where a 18×9 fault signature matrix S served as specification. The underlying system has order 4. Each line of S can be realized by a detector of order 1 or 2 with eigenvalues $\{-1\}$ or $\{-1, -2\}$. The sum of orders of the resulting individual detectors is 32, but the resulting global detector $R(\lambda)$ has McMillan degree 6. Recall that the "least order" detector computed in [31] has order 14.

20.8.3 The Computation of Achievable Fault Signature

An aspect apparently not addressed until recently in the literature is the generation of the achievable complete fault signature specification for a FDIP. Traditionally this aspect is addressed by trying to design a bank of detectors to achieve a desired specification matrix S. The specification is achievable if the design was successful. However, it is possible to generate systematically all possible specifications using an exhaustive search. For this purpose, a recursive procedure can be devised which has as inputs the $p \times m$ and $p \times m_f$ TFMs $G(\lambda)$ and $F(\lambda)$ and as output the corresponding signature matrix S. If we denote this procedure as FDISPEC(G,F), then the fault signature matrix for the system (20.26) can be computed as

$$S = \text{FDISPEC}\left(\begin{bmatrix} G_u & G_d \\ I_{m_u} & 0 \end{bmatrix}, \begin{bmatrix} G_f \\ 0 \end{bmatrix}\right)$$

Procedure S = FDISPEC(G,F)

1. Compute a left nullspace basis $N_l(\lambda)$ of $G(\lambda)$; **exit** with empty S if $N_l(\lambda)$ is empty.
2. Compute $N_f(\lambda) = N_l(\lambda)F(\lambda)$.
3. Compute the signature matrix S of $N_f(\lambda)$; **exit** if S is a row vector.
4. **For** $i = 1, \ldots, m_f$

   4.1 Form $\widetilde{G}_i(\lambda)$ as column i of $N_f(\lambda)$.
   4.2 Form $\widetilde{F}_i(\lambda)$ from the columns $1, \ldots, i-1, i+1, \ldots, m_f$ of $N_f(\lambda)$.
   4.3 **Call** $\widetilde{S} = \text{FDISPEC}(\widetilde{G}_i, \widetilde{F}_i)$.
   4.4 Partition $\widetilde{S} = [\widetilde{S}_1 \ \widetilde{S}_2]$ such that $\widetilde{S}_1$ has $i-1$ columns.
   4.5 Define $\widehat{S} = [\widetilde{S}_1 \ 0 \ \widetilde{S}_2]$ and update $S \leftarrow \begin{bmatrix} S \\ \widehat{S} \end{bmatrix}$.

As it can be observed, the efficient implementation of this procedure heavily benefits of the state space updating techniques developed in Sect. 20.7.1. This confers an increased efficiency during the recursive calls, because the dimensions of the systems are decreasing during a full recursion. The current recursion is broken each time an empty nullspace is encountered or when the last possible recursion level has been attained (i.e., S at Step 3 is a row vector). The computation of structural information at Step 3 involves checking for zero columns in the input and feedthrough matrices $\widetilde{B}_f$ and $\widetilde{D}_f$ of the realization of $N_f(\lambda)$ in (20.23). Note that the whole recursive computations can be performed by using exclusively orthogonal transformations.

The above procedure can be easily implemented such that it performs the minimum number of nullspace computations and updating. The resulting fault signature matrix S is obtained by stacking row-wise the matrices S_i, $i = 1, \ldots, k$ computed at different recursion levels, where k denotes the number of calls of the recursive procedure. This number is given by the combinatorial formula

$$k = \sum_{i=0}^{i_{\max}} \binom{m_f}{i},$$

where $i_{\max} = \min(m_f, p - r) - 1$ and r is the rank of the initial $G(\lambda)$. As it can be observed, k depends of the number of basis vectors $p - r$ and the number of faults m_f, and, although the number of distinct specifications can be relatively low, still k can be a large number. For the already mentioned problem in [31], $k = 1 + m_f + m_f(m_f - 1)/2 = 37$, but only 18 of the determined specifications are distinct. A detailed account of the computational aspects of the procedure FDISPEC is done in [27].

20.9 Conclusions

In this paper we presented an overview of computational techniques to determine rational nullspace bases of rational or polynomial matrices. Simple proper rational bases are the direct correspondents of the polynomial bases and can be computed using the proposed numerical algorithms based on minimal cover techniques. Having in mind potential applications, we also developed explicit realizations for several operation with nullspace bases or with a linear combination of vectors of a nullspace basis. The computational techniques presented in this paper have been implemented as robust numerical software which is now part of a DESCRIPTOR SYSTEM Toolbox for MATLAB developed by the authors over the last decade [19].

The rational nullspace computation based techniques allow to solve important applications as the solution of FDP or FDIP. A main feature of rational nullspace based techniques is a full flexibility in addressing different aspects of these problems, like computing least order detectors, checking existence conditions, computing the achievable fault signature, or employing updating techniques to design a bank of detectors to solve the FDIP. The underlying computations extensively use orthogonal similarity transformations to perform the important computational steps, as for example, to determine a proper nullspace basis or to check the existence conditions of a solution. In contrast, methods based on polynomial nullspace computations are less flexible, and involve computational detours, which are highly questionable from a numerical point of view.

An interesting result of our studies is that although using simple proper rational bases leads to a straightforward solution of the FDP with least order detectors, the computation of the simple basis is not actually necessary. Since we need to compute only linear combinations of simple basis vectors when solving the FDP with least order detector, this computation can be directly performed starting with a minimal proper basis which can be obtained using exclusively orthogonal pencil manipulations. The linear combinations of basis vectors up to a given McMillan degree can be computed using numerical algorithms based on minimal cover techniques. This aspect is highly relevant for implementing robust and efficient numerical software as those available in a recent FAULT DETECTION Toolbox for MATLAB [23].

Appendix A: Computation of Minimal Dynamic Covers

The computational problem which we solve in this section is the following: given a descriptor pair $(A - \lambda E, B)$ with $A, E \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and B partitioned as $B = [B_1\ B_2]$ with $B_1 \in \mathbb{R}^{n\times m_1}$, $B_2 \in \mathbb{R}^{n\times m_2}$, determine the matrix $F \in \mathbb{R}^{m_2\times n}$ such that the pair $(A + B_2F - \lambda E, B_1)$ is *maximally uncontrollable* (i.e., $A + B_2F - \lambda E$ has maximal number of uncontrollable eigenvalues). A *dual* problem to be solved in Sect. 20.5 deals with an observable pair $(A - \lambda E, C)$ with nonsingular E and with a C matrix partitioned as

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

In this case, a matrix K is sought such that the pair $(A + KC_2 - \lambda E, C_1)$ is *maximally unobservable*. For convenience and in agreement with the assumptions of the problem to be solved in Sect. 20.5, we will describe a computational method which is suitable for a controllable pair $(A - \lambda E, B)$ with nonsingular E. However, such an algorithm can be immediately applied to solve the above dual problem by applying it to the controllable pair $(A^T - \lambda E^T, C^T)$ to determine K^T.

The problem to determine F which makes the pair $(A + B_2F - \lambda E, B_1)$ maximally uncontrollable is equivalent [30] to compute a subspace $\mathcal{V}$ of least possible dimension satisfying

$$(\overline{A} + \overline{B}_2F)\mathcal{V} \subset \mathcal{V}, \quad \text{span}(\overline{B}_1) \subset \mathcal{V}, \tag{20.44}$$

where $\overline{A} = E^{-1}A$, $\overline{B}_1 = E^{-1}B_1$, and $\overline{B}_2 = E^{-1}B_2$. This subspace is the least order $(\overline{A}, \overline{B}_2)$-invariant subspace which contains $\text{span}(\overline{B}_1)$ [30]. The condition (20.44) can be rewritten as

$$\overline{A}\mathcal{V} \subset \mathcal{V} + \text{span}(\overline{B}_2), \quad \text{span}(\overline{B}_1) \subset \mathcal{V}, \tag{20.45}$$

which is the condition defining the subspace $\mathcal{V}$ as a *Type I dynamic cover* [8].

In this appendix we describe a computational method for determining minimal dynamic covers, which relies on the reduction of the descriptor pair $(A - \lambda E, [B_1, B_2])$ to a particular condensed form, for which the solution of the problem (i.e., the choice of appropriate F) is simple. This reduction is performed in two stages. The first stage is an orthogonal reduction which represents a particular instance of the descriptor controllability staircase procedure of [15] applied to the descriptor pair $(A - \lambda E, [B_1, B_2])$. This procedure can be interpreted as a generalized orthogonal variant of the basis selection approach underlying the determination of *Type I* minimal covers in [8]. In the second stage, additional zero blocks are generated in the reduced matrices using non-orthogonal transformations. With additional blocks zeroed via a specially chosen F, the least

order $(\overline{A}, \overline{B}_2)$-invariant subspace containing span$(\overline{B}_1)$ can be identified as the linear span of the leading columns of the resulting right transformation matrix. In what follows we present in detail these two stages as well as the determination of F.

Stage I: Special Controllability Staircase Algorithm

0. Compute an orthogonal matrix Q such that $Q^T E$ is upper triangular; compute $A \leftarrow Q^T A$, $E \leftarrow Q^T E$, $B_1 \leftarrow Q^T B_1$, $B_2 \leftarrow Q^T B_2$.
1. Set $j = 1$, $r = 0$, $k = 2, \nu_1^{(0)} = m_1, \nu_2^{(0)} = m_2, A^{(0)} = A, E^{(0)} = E$, $B_1^{(0)} = B_1$, $B_2^{(0)} = B_2$, $Z = I_n$.
2. Compute an orthogonal matrix W_1 such that

$$W_1^T \left[B_1^{(j-1)} \middle| B_2^{(j-1)} \right] := \left[\begin{array}{c|c} A_{k-1,k-3} & A_{k-1,k-2} \\ 0 & A_{k,k-2} \\ 0 & 0 \end{array}\right] \begin{array}{c} \nu_1^{(j)} \\ \nu_2^{(j)} \\ \rho \end{array}$$
$$\begin{array}{cc} \nu_1^{(j-1)} & \nu_2^{(j-1)} \end{array}$$

   with $A_{k-1,k-3}$ and $A_{k,k-2}$ full row rank matrices; compute an orthogonal matrix U_1 such that $W_1^T E^{(j-1)} U_1$ is upper triangular.

3. Compute and partition

$$W_1^T A^{(j-1)} U_1 := \left[\begin{array}{ccc} A_{k-1,k-1} & A_{k-1,k} & A_{k-1,k+1} \\ A_{k,k-1} & A_{k,k} & A_{k,k+1} \\ B_1^{(j)} & B_2^{(j)} & A^{(j)} \end{array}\right] \begin{array}{c} \nu_1^{(j)} \\ \nu_2^{(j)} \\ \rho \end{array}$$
$$\begin{array}{ccc} \nu_1^{(j)} & \nu_2^{(j)} & \rho \end{array}$$

$$W_1^T E^{(j-1)} U_1 := \left[\begin{array}{ccc} E_{k-1,k-1} & E_{k-1,k} & E_{k-1,k+1} \\ O & E_{k,k} & E_{k,k+1} \\ O & O & E^{(j)} \end{array}\right] \begin{array}{c} \nu_1^{(j)} \\ \nu_2^{(j)} \\ \rho \end{array}$$
$$\begin{array}{ccc} \nu_1^{(j)} & \nu_2^{(j)} & \rho \end{array}$$

4. For $i = 1, \ldots, k-2$ compute and partition

$$A_{i,k-1} U_1 := \begin{array}{ccc} [A_{i,k-1} & A_{i,k} & A_{i,k+1}] \\ \nu_1^{(j)} & \nu_2^{(j)} & \rho \end{array}$$
$$E_{i,k-1} U_1 := \begin{array}{ccc} [E_{i,k-1} & E_{i,k} & E_{i,k+1}] \\ \nu_1^{(j)} & \nu_2^{(j)} & \rho \end{array}$$

5. $Q \leftarrow Q\,\mathrm{diag}(I_r, W_1)$, $Z \leftarrow Z\,\mathrm{diag}(I_r, U_1)$.
6. If $v_1^{(j)} = 0$ then $\ell = j-1$ and **Exit**.
7. $r \leftarrow r + v_1^{(j)} + v_2^{(j)}$; if $\rho = 0$ then $l = j$ and **Exit**;
   else, $j \leftarrow j+1$, $k \leftarrow k+2$, and go to Step 2.

At the end of this algorithm $\widehat{A} - \lambda\widehat{E} := Q^T(A - \lambda E)Z$, $\widehat{B} := Q^T B$, $\widehat{E}$ is upper triangular, and the pair $(\widehat{A}, \widehat{B})$ is in a *special staircase form*. For example, for $\ell = 3$ and $r < n$, $[\widehat{B}\ \widehat{A}]$ and $\widehat{E}$ have similarly block partitioned forms

$$\left[\widehat{B}\middle|\widehat{A}\right] = \left[\begin{array}{cc|ccccccc} A_{1,-1} & A_{1,0} & A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} & A_{17} \\ O & A_{2,0} & A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & A_{26} & A_{27} \\ O & O & A_{31} & A_{32} & A_{33} & A_{34} & A_{35} & A_{36} & A_{37} \\ O & O & O & A_{42} & A_{43} & A_{44} & A_{45} & A_{46} & A_{47} \\ O & O & O & O & A_{53} & A_{54} & A_{55} & A_{56} & A_{57} \\ O & O & O & O & O & A_{64} & A_{65} & A_{66} & A_{67} \\ O & O & O & O & O & O & O & A_{76} & A_{77} \end{array}\right]$$

$$\widehat{E} = \begin{bmatrix} E_{11} & E_{12} & \cdots & E_{17} & E_{18} \\ O & E_{22} & \cdots & E_{26} & E_{27} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & \cdots & E_{66} & E_{67} \\ O & O & \cdots & O & E_{77} \end{bmatrix}$$

In the special staircase form $[\widehat{B}\ \widehat{A}]$, $A_{2j-1,2j-3} \in \mathbb{R}^{v_1^{(j)} \times v_1^{(j-1)}}$ and $A_{2j,2j-2} \in \mathbb{R}^{v_2^{(j)} \times v_2^{(j-1)}}$ are full row rank matrices for $j = 1, \ldots, \ell$. The trailing row blocks of $[\widehat{B}\ \widehat{A}]$ and $\widehat{E}$ are empty if $r = n$. In the case when $r < n$, the trailing diagonal blocks $A_{2\ell+1,2\ell+1}, E_{2\ell+1,2\ell+1} \in \mathbb{R}^{(n-r)\times(n-r)}$, and the pair $(A_{2\ell+1,2\ell+1} - \lambda E_{2\ell+1,2\ell+1}, A_{2\ell+1,2\ell})$ is controllable.

In the second reduction stage we use non-orthogonal upper triangular left and right transformation matrices W and U, respectively, to annihilate the minimum number of blocks in $\widehat{A}$ and $\widehat{E}$ which allows to solve the minimum cover problem. Assume W and U have block structures identical to $\widehat{E}$. The following procedure exploits the full rank of submatrices $A_{2j,2j-2}$ and $E_{2j-1,2j-1}$ to introduce zero blocks in the block row $2j$ of $\widehat{A}$ and block column $2j-1$ of $\widehat{E}$, respectively.

Stage II: Special reduction for Type I Covers

Set $W = I_n$, $U = I_n$.
for $k = \ell, \ell-1, \ldots, 2$
  Comment. Annihilate blocks $A_{2k,2j-1}$, for $j = k, k+1, \ldots, \ell$.
  for $j = k, k+1, \ldots, \ell$
    Compute $U_{2k-2,2j-1}$ such that $A_{2k,2k-2}U_{2k-2,2j-1} + A_{2k,2j-1} = 0$.
    $A_{i,2j-1} \leftarrow A_{i,2j-1} + A_{i,2k-2}U_{2k-2,2j-1}$, $i = 1, 2, \ldots, 2k$.
    $E_{i,2j-1} \leftarrow E_{i,2j-1} + E_{i,2k-2}U_{2k-2,2j-1}$, $i = 1, 2, \ldots, 2k-2$.
  end
  Comment. Annihilate blocks $E_{2k-2,2j-1}$, for $j = k, k+1, \ldots, \ell$.
  for $j = k, k+1, \ldots, \ell$
    Compute $W_{2k-2,2j-1}$ such that $W_{2k-2,2j-1}E_{2j-1,2j-1} + E_{2k-2,2j-1} = 0$.
    $A_{2k-2,i} \leftarrow A_{2k-2,i} + W_{2k-2,2j-1}A_{2j-1,i}$, $i = 2j-2, 2j-1, \ldots, 2\ell$.
    $E_{2k-2,i} \leftarrow E_{2k-2,i} + W_{2k-2,2j-1}E_{2j-1,i}$, $i = 2j, 2j+1, \ldots, 2\ell$.
  end
end

For the considered example, this algorithm introduces the following zero blocks: A_{65}, E_{45}, A_{43}, A_{45}, E_{23}, E_{25} (in this order).

Let $\widetilde{A} := W\widehat{A}U$, $\widetilde{E} := W\widehat{E}U$, and $\widetilde{B} = [\widetilde{B}_1\ \widetilde{B}_2] := W\widehat{B}$ be the system matrices resulted at the end of Stage II. Define also the feedback matrix $\widetilde{F} \in \mathbb{R}^{m_2 \times n}$ partitioned column-wise compatibly with $\widehat{A}$

$$\widetilde{F} = [F_1\ O\ F_3\ \cdots\ O\ F_{2l-1}\ O]$$

where $F_{2j-1} \in \mathbb{R}^{m_2 \times v_1^{(j)}}$ are such that $A_{2,0}F_{2j-1} + A_{2,2j-1} = 0$ for $j = 1, \ldots, l$.

For the considered example, we achieved with the above choice of F that

$$\widetilde{A} + \widetilde{B}_2\widetilde{F} = \begin{bmatrix} \overline{A}_{11} & A_{12} & \overline{A}_{13} & A_{14} & \overline{A}_{15} & A_{16} & A_{17} \\ O & \overline{A}_{22} & O & \overline{A}_{24} & O & \overline{A}_{26} & \overline{A}_{27} \\ A_{31} & A_{32} & \overline{A}_{33} & A_{34} & \overline{A}_{35} & A_{36} & A_{37} \\ O & A_{42} & O & \overline{A}_{44} & O & \overline{A}_{46} & \overline{A}_{47} \\ O & O & A_{53} & A_{54} & \overline{A}_{55} & A_{56} & A_{57} \\ O & O & O & A_{64} & O & A_{66} & A_{67} \\ O & O & O & O & O & A_{76} & A_{77} \end{bmatrix},$$

$$\widetilde{E} = \begin{bmatrix} E_{11} & E_{12} & \overline{E}_{13} & E_{14} & \overline{E}_{15} & E_{16} & E_{17} \\ O & E_{22} & O & \overline{E}_{24} & O & \overline{E}_{26} & \overline{E}_{27} \\ O & O & E_{33} & E_{34} & \overline{E}_{35} & E_{36} & E_{37} \\ O & O & O & E_{44} & O & \overline{E}_{46} & \overline{E}_{47} \\ O & O & O & O & E_{55} & E_{56} & E_{57} \\ O & O & O & O & O & E_{66} & E_{67} \\ O & O & O & O & O & O & E_{77} \end{bmatrix}$$

where the elements with bars have been modified after Stage I.

Consider now the permutation matrix defined by

$$P = \left[\begin{array}{cccc|c|cc|c} I_{\nu_1^{(1)}} & O & O & O & \cdots & O & O & O \\ \hline O & O & I_{\nu_1^{(2)}} & O & \cdots & O & O & O \\ \hline \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \hline O & O & O & O & \cdots & I_{\nu_1^{(\ell)}} & O & O \\ \hline O & I_{\nu_2^{(1)}} & O & O & \cdots & O & O & O \\ \hline O & O & O & I_{\nu_2^{(2)}} & \cdots & O & O & O \\ \hline \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \hline O & O & O & O & \cdots & O & I_{\nu_2^{(\ell)}} & O \\ \hline O & O & O & O & \cdots & O & O & I_{n-r} \end{array}\right]$$

If we define $L = PWQ^T$, $V = ZUP^T$ and $F = \widetilde{F}V^{-1}$, then overall we achieved that

$$L(A + B_2F - \lambda E)V = \left[\begin{array}{c|c} \breve{A}_1 - \lambda\breve{E}_1 & * \\ \hline O & \breve{A}_2 - \lambda\breve{E}_2 \end{array}\right], \quad L\left[\begin{array}{c|c} B_1 & B_2 \end{array}\right] = \left[\begin{array}{c|c} \breve{B}_1 & * \\ \hline O & \breve{B}_2 \end{array}\right]$$

where, by construction, the pairs $(\breve{A}_1 - \lambda\breve{E}_1, \breve{B}_1)$ and $(\breve{A}_2 - \lambda\breve{E}_2, \breve{B}_2)$ are in controllable staircase form. Thus, by the above choice of F, we made $n_2 := \sum_{i=1}^{\ell} \nu_2^{(i)}$ of the n eigenvalues of the $A + B_2F - \lambda E$ uncontrollable via B_1. It is straightforward to show that the matrix V_1 formed from the first $n_1 := \sum_{i=1}^{\ell} \nu_1^{(i)}$ columns of V, satisfies

$$\overline{A}V_1 = V_1\breve{E}_1^{-1}\breve{A}_1 - \overline{B}_2FV_1, \quad \overline{B}_1 = V_1\breve{E}_1^{-1}\breve{B}_1$$

Thus, according to (20.45), $\mathcal{V} := \mathrm{span}(\mathrm{V}_1)$ is a dynamic cover of *Type I* of dimension n_1. It can be shown using the results of [8] that the resulting *Type I* dynamic cover $\mathcal{V}$ has minimum dimension.

For the considered example, we obtained the controllable staircase forms

$$\left[\breve{B}_1 \middle| \breve{A}_1 - \lambda \breve{E}_1\right] = \left[\begin{array}{c|ccc} A_{1,-1} & \overline{A}_{11} - \lambda E_{11} & \overline{A}_{13} - \lambda \overline{E}_{13} & \overline{A}_{15} - \lambda \overline{E}_{15} \\ O & A_{31} & \overline{A}_{33} - \lambda E_{33} & \overline{A}_{35} - \lambda \overline{E}_{35} \\ O & O & A_{53} & \overline{A}_{55} - \lambda E_{55} \end{array}\right]$$

$$\left[\breve{B}_2 \middle| \breve{A}_2 - \lambda \breve{E}_2\right] = \left[\begin{array}{c|cccc} A_{2,0} & \overline{A}_{22} - \lambda E_{22} & \overline{A}_{24} - \lambda \overline{E}_{24} & \overline{A}_{26} - \lambda \overline{E}_{26} & \overline{A}_{27} - \lambda \overline{E}_{27} \\ O & A_{42} & \overline{A}_{44} - \lambda E_{44} & \overline{A}_{46} - \lambda E_{46} & \overline{A}_{47} - \lambda \overline{E}_{47} \\ O & O & A_{64} & A_{66} - \lambda E_{66} & A_{67} - \lambda E_{67} \\ O & O & O & A_{76} & \overline{A}_{77} - \lambda E_{77} \end{array}\right]$$

The Stage I reduction of system matrices to the special controllability form can be performed by using exclusively orthogonal similarity transformations. It can be shown that the computed condensed matrices $\widehat{A}$, $\widehat{E}$, and $\widehat{B}$ are exact for matrices which are nearby to the original matrices A, E, and B, respectively. Thus this part of the reduction is *numerically backward stable*. When implementing the algorithm, the row compressions are usually performed using rank revealing QR-factorizations with column pivoting.

To achieve an $O(n^3)$ computational complexity in Stage I reduction, it is essential to perform the row compressions simultaneously with maintaining the upper triangular shape of E during reductions. The basic computational technique, described in details in [16], consists in employing elementary Givens transformations from left to introduce zero elements in the rows of B, while applying from right appropriate Givens transformations to annihilate the generated nonzero subdiagonal elements in E. By performing the rank revealing QR-decomposition in this way (involving also column permutations), we can show that the overall worst-case computational complexity of the special staircase algorithm is $O(n^3)$. Note that for solving the problem in Sect. 20.5, the accumulation of Z is not even necessary, since all right transformations can be directly applied to a third matrix (e.g., a system output matrix C).

The computations at Stage II reduction to determine a basis for the minimal dynamic cover and the computation of the feedback matrix F involve the solution of many, generally overdetermined, linear equations. For the computation of the basis for $\mathcal{V}$ it is important to estimate the condition numbers of the overall transformation matrices. This can be done by computing $\|V\|_F^2 = \|U\|_F^2$ and $\|L\|_F^2 = \|W\|_F^2$ as estimations of the corresponding condition numbers. If these norms are relatively small (e.g., $\leq$10,000) then practically there is no danger for a significant loss of accuracy due to nonorthogonal reductions. On contrary, large values of these norms provide a clear hint of potential accuracy losses. In practice, it suffices only to look at the largest magnitudes of elements of W and U used at Stage II to obtain equivalent information. For the computation of F, condition numbers for solving the underlying equations can be also easily estimated. A large norm of F is an indication of possible accuracy losses. For the Stage II reduction, a

simple operation count is possible by assuming all blocks 1×1 and this indicates a computational complexity of $O(n^3)$.

References

1. Antoniou EN, Vardulakis AIG, Vologiannidis S (2005) Numerical computation of minimal polynomial bases: a generalized resultant approach. Linear Algebra Appl 405:264–278
2. Beelen ThGJ (1987) New algorithms for computing the Kronecker structure of a pencil with applications to systems and control theory. Ph. D. Thesis, Eindhoven University of Technology
3. Ding X, Frank PM (1991) Frequency domain approach and threshold selector for robust model-based fault detection and isolation. In: Proceedings of IFAC symposium SAFEPROCESS'1991, Baden-Baden, Germany
4. Forney GD (1975) Minimal bases of rational vector spaces with applications to multivariable linear systems. SIAM J Control 13:493–520
5. Frisk E, Nyberg M (2001) A minimal polynomial basis solution to residual generation for fault diagnosis in linear systems. Automatica 37:1417–1424
6. Gertler J (1998) Fault detection and diagnosis in engineering systems. Marcel Dekker, New York
7. Kailath T (1980) Linear systems. Prentice Hall, Englewood Cliffs
8. Kimura G (1977) Geometric structure of observers for linear feedback control laws. IEEE Trans Automat Control 22:846–855
9. Luenberger DG (1966) Observers for multivariable systems. IEEE Trans Automat Control 11:190–197
10. Morse AS (1976) Minimal solutions to transfer matrix equations. IEEE Trans Automat Control 21:131–133
11. Nyberg M (2002) Criterions for detectability and strong detectability of faults in linear systems. Int J Control 75:490–501
12. Van Dooren P (1981) The generalized eigenstructure problem in linear systems theory. IEEE Trans Automat Control 26:111–129
13. Vardulakis AIG, Karcanias N (1984) Proper and stable, minimal MacMillan degrees bases of rational vector spaces. IEEE Trans Automat Control 29:1118–1120
14. Varga A (1981) Numerically stable algorithm for standard controllability form determination. Electron Lett 17:74–75
15. Varga A (1990) Computation of irreducible generalized state-space realizations. Kybernetika 26:89–106
16. Varga A (1995) On stabilization of descriptor systems. Syst Control Lett 24:133–138
17. Varga A (1996) Computation of Kronecker-like forms of a system pencil: applications, algorithms and software. In: Proceedings of CACSD'96 symposium, Dearborn, MI, pp 77–82
18. Varga A (1998) Computation of coprime factorizations of rational matrices. Linear Algebra Appl 271:83–115
19. Varga A (2000) A Descriptor Systems toolbox for MATLAB. In: Proceedings of CACSD'2000 symposium, Anchorage, Alaska
20. Varga A (2003) On computing least order fault detectors using rational nullspace bases. In: Proceedings of IFAC symposium SAFEPROCESS'2003, Washington, DC
21. Varga A (2004) New computational approach for the design of fault detection and isolation filters. In: Voicu M (ed) Advances in automatic control, vol 754 of The Kluwer international series in engineering and computer science, pp 367–381. Kluwer Academic Publishers
22. Varga A (2004) Reliable algorithms for computing minimal dynamic covers for descriptor systems. In: Proceedings of MTNS'04, Leuven, Belgium

23. Varga A (2006) A fault detection toolbox for MATLAB. In: Proceedings of CACSD'06, Munich, Germany
24. Varga A (2007) On designing least order residual generators for fault detection and isolation. In: Proceedings of the 16th international conference on control systems and computer science, Bucharest, Romania, pp 323–330
25. Varga A (2009) General computational approach for optimal fault detection. In: Proceedings of SAFEPROCESS'2009, Barcelona, Spain
26. Varga A (2009) The nullspace method—a unifying paradigm to fault detection. In: Proceedings of CDC'2009, Shanghai, China
27. Varga A (2009) On computing achievable fault signatures. In: Proceedings of SAFEPROCESS'2009, Barcelona, Spain
28. Verghese G, Van Dooren P, Kailath T (1979) Properties of the system matrix of a generalized state-space system. Int J Control 30:235–243
29. Warren ME, Eckberg AE (1975) On the dimension of controllability subspaces: a characterization via polynomial matrices and Kronecker invariants. SIAM J Control 13:434–445
30. Wonham WM, Morse AS (1972) Feedback invariants of linear multivariable systems. Automatica 8:93–100
31. Yuan Z, Vansteenkiste GC, Wen CY (1997) Improving the observer-based FDI design for efficient fault isolation. Int J Control 68(1):197–218
32. Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Englewood Cliffs

Chapter 21
Optimal Control of Switched System with Time Delay Detection of Switching Signal

C. Z. Wu, K. L. Teo and R. Volker

Abstract This paper deals with optimal control problems governed by switched systems with time delay detection of switching signal. We consider the switching sequence as well as the switching instants as decision variables. We present a two-level optimization method to solve it. In the first level, we fix the switching sequence, and introduce a time scaling transformation such that the switching instants are mapped into pre-assigned fixed knot points. Then, the transformed problem becomes a standard optimal parameter selection problem, and hence can be solved by many optimal control techniques and the corresponding optimal control software packages, such as MISER. In the second level, we consider the switching sequence as decision variables. We introduce a discrete filled function method to search for a global optimal switching sequence. Finally, a numerical example is presented to illustrate the efficiency of our method.

21.1 Introduction

A switched system is a special hybrid systems. It consists of several subsystems and a switching law for assigning the active subsystem at each time instant. Many real-world processes, such as chemical processes, automotive systems, and manufacturing processes, can be modeled as such systems.

C. Z. Wu (✉)
Department of Mathematics, Chongqing Normal University, Chongqing, P.R. China
e-mail: czwu@cqnu.edu.cn

K. L. Teo · R. Volker
Department of Mathematics and Statistics, Curtin University of Technology, Perth, Western Australia, Australia

P. Van Dooren et al. (eds.), *Numerical Linear Algebra in Signals, Systems and Control*, Lecture Notes in Electrical Engineering, 80, DOI: 10.1007/978-94-007-0602-6_21,

There is a considerable interest among researchers in their study of optimal control problems of switched systems. See, for example, [1–6]. In [5], a survey of several interesting optimal control problems and methods for switched systems are reported. Among them, a particularly important approach is the one based on the parametrization of switching instants in [6] where the switching instants are parameterized by some parameters. Then, the gradient formula of the cost functional with respect to these parameters is derived. Thus, the original problem can be solved by any efficient gradient-based optimization methods. This method is similar to the transformation developed in [7]. It is very effective for computing the optimal switching instants. However, the switching sequence is assumed fixed in [4]. In [6], an optimal control problem governed by a bi-modal switched system with variable switching sequence is considered. It is then shown that this switched system is embedded into a larger family of systems and the set of trajectories of the switched system is dense in the set of those of the embedded system. Then, optimal control problem governed by the larger system is considered instead of the original problem. Based on the relationship between the two optimal control problems, it is shown that if the latter optimal control problem admits a bang–bang solution, then this solution is an optimal solution of the original problem. Otherwise, a suboptimal solution can be obtained via the Chattering Lemma. This method is effective only for the bi-modal case. The extension to multi-modal cases requires further investigation.

For all the optimal control problems considered above, they are based on the assumption that the detection of the switching signal is instantaneous. However, in practice, we may not be able to detect the change of the switching signal instantly. We often realize such a change after a time period. In [8], the stabilization of such systems is considered. In this paper, we consider an optimal control problem governed by such a system. In our problem, the optimal control is considered to be of feedback form; and both the switching instants and the switching sequence are considered as decision variables. This optimal control problem is to be solved as a two-level optimization problem. In the first level, we fix the switching sequence and introduce a time scaling transformation to map the switching instants into pre-assigned fixed knot points. Then, it can be solved by some optimal control software packages, such as MISER. In the second level, we introduce a discrete filled function method to search for an optimal switching sequence.

The paper is organized as follows. We formulate the problem in Sect. 21.2 In Sect. 21.3, we decompose the original problem into a two level optimization problem. Then, a time scaling transformation is introduced to map the switching instants into pre-fixed knot points. The transformed problem can be solved by many optimal control software packages, such as MISER. In Sect. 21.4, we introduce a discrete filled function method to determine the optimal switching sequence. In Sect. 21.5, an illustrative example is presented. Section 21.6 concludes the paper.

21.2 Problem Formulation

Consider a switched system given by

$$\dot{x}(t) = A_{\delta(t)}x(t) + B_{\delta(t)}u(t), t \in [0, T] \tag{21.1}$$

with initial condition

$$x(0) = x_0, \tag{21.2}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control vector, the switching signal $\delta(t) \in \{1, 2, \ldots, N\}$ is a right continuous function and its discontinuity points are $\tau_1, \tau_2, \ldots, \tau_{M-1}$ such that

$$0 = \tau_0 < \tau_1 < \cdots < \tau_{M-1} < \tau_M = T. \tag{21.3}$$

while for each $i \in \{1, 2, \ldots, N\}, A_i \in \mathbb{R}^{n\times n}$ and $B_i \in \mathbb{R}^{n\times m}, \gamma(t)$ is the detection function of $\delta(t)$, and $\tau > 0$ is the time-delay, which is the time required to detect which subsystem is active.

The control u is of a piecewise constant feedback form, i.e.,

$$u(t) = K_{\gamma(t)}x(t), \tag{21.4}$$

where

$$\gamma(t) = \delta(t - \tau), t \in [0, T], \tag{21.5}$$

$$K_{\gamma(t)} = \tilde{K}, \ t \in [0, \tau], \tag{21.6}$$

and $K_i \in \mathbb{R}^{m\times n}, i \in \{1, 2, \ldots, N\}$, are to be designed.

To simplify the notation, let

$$\delta(t) = i_k, \text{ if } t \in [\tau_{k-1}, \tau_k], \ k = 1, \ldots, M. \tag{21.7}$$

For a given switching signal $\delta(t)$ and the control expressed as a piecewise constant feedback form given by (21.4), the evolution of the dynamical system (21.1) can be described as in Fig. 21.1.

Let $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_{M-1}]^\top$, $\boldsymbol{K} = [K_1, K_2, \ldots, K_N] \in \mathbb{R}^{m\times nN}$, $\boldsymbol{I} = [i_1, i_2, \ldots, i_M]$. The optimal control problem dealt with in this paper can be stated as follows.

Problem 1 Consider the dynamical system (21.1) with the initial condition and the piecewise constant control given by (21.4)–(21.6). Find a $(\boldsymbol{\tau}, \boldsymbol{K}, \boldsymbol{I})$ such that

$$J(\boldsymbol{\tau}, \boldsymbol{K}, \boldsymbol{I}) = x^\top(T)Px(T) + \int_0^T x^\top(t)Qx(t) + u^\top(T)Ru(T)\mathrm{d}t \tag{21.8}$$

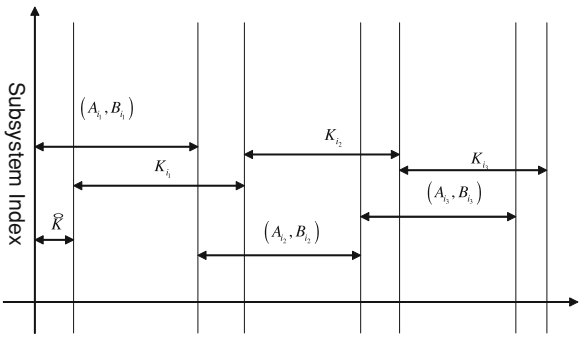


Fig. 21.1 Evolution of the system (21.1)

is minimized subject to the constraint (21.3), where P, Q, and R are matrices with appropriate dimensions.

Note that the piecewise constant control given by (21.4) is determined by the control switching points $\tau_1 + \tau, \tau_2 + \tau, \ldots, \tau_{M-1} + \tau$, where $\tau_1, \tau_2, \ldots, \tau_{M-1}$, are sub-system switching time points which are decision variables to be optimized over. Problem 1 cannot be solved directly using existing numerical optimal control techniques. However, by using a time scaling transformation developed in [7, 9, 10], we shall show in the next section that Problem 1 is, in fact, equivalent to an optimal parameter selection problem, where the varying switching points are being mapped into pre-assigned knot points in a new time scale.

To proceed further, we assume that the following condition is satisfied:

Assumption 1 All the switching durations are larger than the delay time τ, i.e.,

$$\tau_i - \tau_{i-1} > \tau, \ i = 1, \ldots, M. \tag{21.9}$$

21.3 Problem Reformulation

In this section, we will show that Problem 1 can be transformed into a standard parameter selection problem.

We re-write (21.1) as follows:

$$\dot{x}(t) = \sum_{j=1}^{N} \kappa_{i,j}\big(A_j x(t) + B_j u(t)\big), \quad \text{if } t \in [\tau_{i-1}, \tau_i], \ i = 1, \ldots, M, \tag{21.10}$$

where $\kappa_{i,j}, i = 1, \ldots, M; \ j = 1, \ldots, N$, are logical integer variables. Since only one sub-system is active at each time point, $\kappa_{i,j}, i = 1, \ldots, M; j = 1, \ldots, N$, are required to satisfy

$$\sum_{j=1}^{N} \kappa_{i,j} = 1,\ \kappa_{i,j} \in \{0,1\}, \quad i = 1,\ldots,M; \quad j = 1,\ldots,N. \tag{21.11}$$

Let $\boldsymbol{\kappa} = \left[\boldsymbol{\kappa_{1,1}},\ldots,\boldsymbol{\kappa_{1,N}},\ldots,\boldsymbol{\kappa_{M,N}}\right]^{\top}$. We introduce the following time scaling transformation:

$$\frac{\mathrm{d}t}{\mathrm{d}s} = \sum_{i=1}^{M} \xi_i \chi_{(i-1/2,i]}(s) + \sum_{i=0}^{M-1} 2\tau \chi_{(i,i+1/2]}(s),\ s \in [0,M], \tag{21.12}$$

where

$$\xi_i = 2(\tau_i - \tau),\ i = 1,\ldots,M, \tag{21.13}$$

$\chi_I(s)$ is the indicator characteristic function defined by

$$\chi_I(s) = \begin{cases} 1, & \text{if } s \in I, \\ 0, & \text{else.} \end{cases}$$

Let $\boldsymbol{\xi} = [\boldsymbol{\xi_1}, \boldsymbol{\xi_2}, \ldots, \boldsymbol{\xi_M}]^{\top}$. By (21.3) and (21.9), we have

$$\sum_{i=1}^{M} (\xi_i + \tau) = T,\ \xi_i > 0,\ \text{for } i = 1,\ldots,M. \tag{21.14}$$

After this time scaling transformation, the sub-system switching points $\tau_1, \tau_2, \ldots, \tau_{M-1}$, have been mapped into $1, 2, \ldots, M-1$, and the control switching points $\tau_1 + \tau, \tau_2 + \tau, \ldots, \tau_{M-1} + \tau$, have been mapped into $1/2, 3/2, \ldots, M - 1/2$. Then, system (21.1) is transformed into

$$\begin{aligned} \dot{x}(s) = &\sum_{j=1}^{N} \kappa_{i,j}\left(A_j x(s) + B_j K_{\gamma(s)}\right)x(s)\Big(\xi_i \chi_{(i-1/2,i]}(s) \\ &+ 2\tau \chi_{(i-1,i-1/2]}(s)\Big), \quad \text{if } s \in [i-1,i],\ i = 1,\ldots,M, \end{aligned} \tag{21.15}$$

that is,

$$\dot{x}(s) = \begin{cases} 2\tau \sum_{j=1}^{N} \kappa_{1,j}\left(A_j + B_j \tilde{K}\right)x(s), & \text{if } s \in [0,1/2], \\ \xi_1 \sum_{j=1}^{N} \kappa_{1,j}\left(A_j + B_j K_{\sum_{j=1}^{N} \kappa_{1,j}}\right)x(s), & \text{if } s \in (1/2,1], \\ \quad \ldots, & \\ \xi_M \sum_{j=1}^{N} \kappa_{M,j}\left(A_j + B_j K_{\sum_{j=1}^{N} \kappa_{M,j}}\right)x(s), & \text{if } s \in (M-1/2,M]. \end{cases} \tag{21.16}$$

with initial condition

$$x(0) = x_0, \tag{21.17}$$

and the cost functional (21.8) is transformed into

$$J(\boldsymbol{\xi},\boldsymbol{K},\boldsymbol{\kappa}) = x^{\top}(M)Px(M) + \int_0^M \left(x^{\top}(s)Qx(s) + u^{\top}(s)Ru(s)\right) \times \left(\sum_{i=1}^{M} \xi_i \chi_{(i-1/2,i]}(s) + \sum_{i=0}^{M-1} 2\tau \chi_{(i,i+1/2]}(s)\right) \mathrm{d}s, \tag{21.18}$$

that is

$$\begin{aligned} J(\boldsymbol{\xi},\boldsymbol{K},\boldsymbol{\kappa}) = {} & x^{\top}(M)Px(M) + \int_0^{1/2} 2\tau x^{\top}(s)\left(Q + \tilde{K}^{\top}R\tilde{K}\right)x(s)\mathrm{d}s \\ & + \sum_{k=1}^{M-1} \int_{k-1/2}^{k} \xi_k x^{\top}(s)\left(Q + \left(K_{\sum_{j=1}^{N}\kappa_{k,j}}\right)^{\top} R\left(K_{\sum_{j=1}^{N}\kappa_{k,j}}\right)\right)x(s)\mathrm{d}s \\ & + \sum_{k=1}^{M-1} \int_{k}^{k+1/2} 2\tau x^{\top}(s)\left(Q + \left(K_{\sum_{j=1}^{N}\kappa_{k,j}}\right)^{\top} R\left(K_{\sum_{j=1}^{N}\kappa_{k,j}}\right)\right)x(s)\mathrm{d}s. \end{aligned} \tag{21.19}$$

where for the simplicity of the notation, we write $x(s) = x(t(s)), \delta(s) = \delta(t(s)), \gamma(s) = \gamma(t(s))$. Now the transformed problem may be formally stated as:

Problem 2 Subject to system (21.16) with initial condition (21.17), find a triple $(\boldsymbol{\xi},\boldsymbol{K},\boldsymbol{\kappa})$ such that the cost functional (21.19) is minimized subject to the constraint (21.14) and (21.11).

For easy reference, the equivalence between Problem 1 and Problem 2 is stated in the following as a theorem.

Theorem 3.1 *Problem* 1 *is equivalent to Problem* 2 *in the sense that if* $(\boldsymbol{\tau}^*,\boldsymbol{K}^*,\boldsymbol{\kappa}^*)$ *is an optimal solution of Problem* 1, *then* $(\boldsymbol{\xi}^*,\boldsymbol{K}^*,\boldsymbol{\kappa}^*)$ *is an optimal solution of Problem* 2, *where the relation between* $\boldsymbol{\tau}^*$ *and* $\boldsymbol{\xi}^*$ *is determined by* (21.13), *and vice versus*.

Note that

$$\min_{(\boldsymbol{\xi},\boldsymbol{K},\boldsymbol{\kappa})} J(\boldsymbol{\xi},\boldsymbol{K},\boldsymbol{\kappa}) = \min_{\boldsymbol{\kappa}} \min_{(\boldsymbol{\xi},\boldsymbol{K})} J(\boldsymbol{\xi},\boldsymbol{K},\boldsymbol{\kappa}).$$

Thus, Problem 2 can be posed as a two-level optimization problem. The first level is

$$J(\boldsymbol{\kappa}) = \min_{(\boldsymbol{\xi},\boldsymbol{K})} J(\boldsymbol{\xi},\boldsymbol{K},\boldsymbol{\kappa}) \text{ subject to } (21.14), \tag{21.20}$$

and the second level is

$$\min_{\boldsymbol{\kappa}} J(\boldsymbol{\kappa}) \text{ subject to } (21.11). \tag{21.21}$$

For easy reference, let the optimization problem (21.20) be referred to as Problem 3 and the optimization problem (21.21) be referred to as Problem 4. For each $\boldsymbol{\kappa}$, Problem 3 is a standard optimal control problem and can be solved by gradient-based optimization methods. The required gradient formulas can be obtained from Theorem 5.2.1 in [11]. Thus, for each $\boldsymbol{\kappa}$, Problem 3 can be solved by many available control software packages, such as MISER 3.3 [12].

Problem 4 is a discrete optimization problem. We will introduce a discrete filled function method to solve it in the next section.

21.4 Determination of the Switching Sequence

In Sect. 21.3, we have shown that Problem 1 with fixed switching sequence can be reformulated as an optimal parameter selection problem and hence is solvable by many optimal control software packages, such as MISER 3.3. In this section, we will introduce a method to determine its switching sequence.

Note that the determination of a switching sequence is equivalent to the determination of the logical integer variables $\kappa_{i,j}, i = 1, \ldots, M; j = 1, \ldots, N$. It is a discrete optimization problem. Here, we will introduce a modified discrete filled function method developed in [13–15].

Let $\mathbf{e}_{i,-j}$ be an element of $\mathbb{R}^{NM}$ with the ith component 1, the jth component -1, and the remaining components 0. Similarly, let $\mathbf{e}_{-i,j}$ be an element of $\mathbb{R}^{NM}$ with the ith component -1, the jth component 1, and the remaining components 0.

Let $\mathbf{D} = \left\{\mathbf{e}_{i,-j}, \mathbf{e}_{-i,j}, i, j = 1, \ldots, NM, i \neq j\right\}$ and Π be the set of $\boldsymbol{\kappa}$ which satisfies (21.11).

Definition 4.1 For any $\boldsymbol{\kappa} \in \boldsymbol{\Pi}, \mathbf{N}(\boldsymbol{\kappa}) = \{\boldsymbol{\kappa} + \boldsymbol{d}\text{: } \boldsymbol{d} \in \boldsymbol{D}\} \cap \boldsymbol{\Pi}$ denotes the neighborhood of the integer point $\boldsymbol{\kappa}$.

Definition 4.2 A point $\boldsymbol{\kappa}^* \in \boldsymbol{\Pi}$ is said to be a discrete local minimizer of Problem 4 if $J(\boldsymbol{\kappa}^*) \leq J(\boldsymbol{\kappa})$ for any $\boldsymbol{\kappa} \in \boldsymbol{N}(\boldsymbol{\kappa}^*) \cap \boldsymbol{\Pi}$. Furthermore, if $J(\boldsymbol{\kappa}^*) \leq J(\boldsymbol{\kappa})$ for any $\boldsymbol{\kappa} \in \boldsymbol{N}(\boldsymbol{\kappa}^*) \cap \boldsymbol{\Pi}$, then $\boldsymbol{\kappa}^*$ is said to be a strict discrete local minimizer.

Definition 4.3 A point $\boldsymbol{\kappa}^* \in \boldsymbol{\Pi}$ is said to be a discrete global minimizer if $J(\boldsymbol{\kappa}^*) \leq J(\boldsymbol{\kappa})$ holds for any $\boldsymbol{\kappa} \in \boldsymbol{\Pi}$.

Definition 4.4 A sequence $\left\{\boldsymbol{\kappa}^{\boldsymbol{i}}\right\}_{i=1}^{k}$ is called a discrete path in Π between $\boldsymbol{\kappa}^{\boldsymbol{1},*} \in \boldsymbol{\Pi}$ and $\boldsymbol{\kappa}^{\boldsymbol{2},*} \in \boldsymbol{\Pi}$ if the following conditions are satisfied:

1. For any $i = 1, \ldots, k, \boldsymbol{\kappa}^{\boldsymbol{i}} \in \boldsymbol{\Pi}$
2. For any $i \neq j, \boldsymbol{\kappa}^{\boldsymbol{i}} \neq \boldsymbol{\kappa}^{\boldsymbol{j}}$
3. $\boldsymbol{\kappa}^{\boldsymbol{1}} = \boldsymbol{\kappa}^{\boldsymbol{1},*}, \boldsymbol{\kappa}^{\boldsymbol{k}} = \boldsymbol{\kappa}^{\boldsymbol{2},*}$ and
4. $\left\|\boldsymbol{\kappa}^{\boldsymbol{i+1}} - \boldsymbol{\kappa}^{\boldsymbol{i}}\right\| = 2, i = 1, \ldots, k-1.$

We note that Π is a discrete pathwise connected set. That is, for every two different points $\boldsymbol{\kappa}^{\boldsymbol{1}}, \boldsymbol{\kappa}^{\boldsymbol{2}}$, we can find a path from $\boldsymbol{\kappa}^{\boldsymbol{1}}$ to $\boldsymbol{\kappa}^{\boldsymbol{2}}$ in Π. Clearly, Π is bounded.

Algorithm 4.1 (Local search)

1. Choose a $\boldsymbol{\kappa_0} \in \boldsymbol{\Pi}$;
2. If $\boldsymbol{\kappa_0}$ is a local minimizer, then stop. Otherwise, we search the neighborhood of $\boldsymbol{\kappa_0}$ and obtain a $\boldsymbol{\kappa} \in \mathbf{N}(\boldsymbol{\kappa_0}) \cap \boldsymbol{\Pi}$ such that $J(\boldsymbol{\kappa}) < J(\boldsymbol{\kappa_0})$.
3. Let $\boldsymbol{\kappa_0} = \boldsymbol{\kappa}$, go to Step 2.

Definition 4.5 $p(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*)$ is called a discrete filled function of $J(\boldsymbol{\kappa})$ at a discrete local minimizer $\boldsymbol{\kappa}^*$ if it satisfies the following properties:

1. $\boldsymbol{\kappa}^*$ is a strict discrete local maximizer of $p(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*)$ over Π;
2. $p(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*)$ has no discrete local minimizers in the region

$$S_1 = \{\kappa : J(\boldsymbol{\kappa}) \geq J(\boldsymbol{\kappa}^*), \boldsymbol{\kappa} \in \boldsymbol{\Pi}/\boldsymbol{\kappa}^*\};$$

3. If $\boldsymbol{\kappa}^*$ is not a discrete global minimizer of $J(\boldsymbol{\kappa})$, then $p(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*)$ has a discrete minimizer in the region

$$S_2 = \{\kappa : J(\boldsymbol{\kappa}) < J(\boldsymbol{\kappa}^*), \boldsymbol{\kappa} \in \boldsymbol{\Pi}\}.$$

Now, we give a discrete filled function which is introduced in [15].

$$F(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*, \boldsymbol{q}, \boldsymbol{r}) = \frac{1}{q + \|\boldsymbol{\kappa} - \boldsymbol{\kappa}^*\|} \varphi_q(\max\{J(\boldsymbol{\kappa}) - J(\boldsymbol{\kappa}^*) + r,\ 0\}), \tag{21.22}$$

where

$$\varphi_q(t) = \begin{cases} \exp(-q/t), & \text{if } t \neq 0, \\ 0, & \text{if } t = 0, \end{cases}$$

and r satisfies

$$0 < r < \max_{\tilde{\boldsymbol{\kappa}}^*, \boldsymbol{\kappa}^* \in \boldsymbol{L}(\boldsymbol{P}), \mathrm{J}(\tilde{\boldsymbol{\kappa}}^*) > \mathrm{J}(\boldsymbol{\kappa}^*)} (J(\tilde{\boldsymbol{\kappa}}^*) - J(\boldsymbol{\kappa}^*)), \tag{21.23}$$

$L(p)$ denotes the set of discrete local minimizers of $J(\boldsymbol{\kappa})$. Next, we will show that for proper choice of $q, r, F(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*, \boldsymbol{q}, \boldsymbol{r})$ is a discrete filled function.

Theorem 4.1 *Suppose that $\boldsymbol{\kappa}^*$ is a discrete local minimizer of $J(\boldsymbol{\kappa})$. Then, for proper choices of $q > 0$ and $r > 0$, $\boldsymbol{\kappa}^*$ is a discrete local maximizer of $F(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*, \boldsymbol{q}, \boldsymbol{r})$.*

Proof The proof is similar to that given for Theorem 3.1 [15].

Let

$$J^{\mathrm{up}} = \max_{\boldsymbol{\kappa_1}, \boldsymbol{\kappa_2} \in \boldsymbol{\Pi}} \{J(\boldsymbol{\kappa_1}) - J(\boldsymbol{\kappa_2})\}.$$

Then, we have the following lemma.

Lemma 4.1 Suppose that $\boldsymbol{\kappa}^*$ is a local minimizer of $J(\boldsymbol{\kappa})$. For any $\boldsymbol{\kappa}_1, \boldsymbol{\kappa}_2 \in \Pi$, let the following conditions be satisfied:

1. $J(\boldsymbol{\kappa}_1) \geq J(\boldsymbol{\kappa}^*)$ and $J(\boldsymbol{\kappa}_2) \geq J(\boldsymbol{\kappa}^*)$,
2. $\|\boldsymbol{\kappa}_2 - \boldsymbol{\kappa}^*\| > \|\boldsymbol{\kappa}_1 - \boldsymbol{\kappa}^*\|$.

Then, when $r > 0$ and $q > 0$ are satisfactory small, $F(\boldsymbol{\kappa}_2, \boldsymbol{\kappa}^*, q, r) < F(\boldsymbol{\kappa}_1, \boldsymbol{\kappa}^*, q, r)$.

Proof The proof is similar to that given for Theorem 3.4 [15].

Theorem 4.2 *Suppose that $\boldsymbol{\kappa}^*$ is a local minimizer of $J(\boldsymbol{\kappa})$. Then, $F(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*, \boldsymbol{q}, \boldsymbol{r})$ has no discrete local minimizers in the region*

$$S_1 = \{\kappa | J(\boldsymbol{\kappa}) \geq J(\boldsymbol{\kappa}^*), \boldsymbol{\kappa} \in \boldsymbol{\Pi}/\boldsymbol{\kappa}^*\}$$

if $r > 0$ and $q > 0$ are chosen appropriately small.

Proof Suppose the conclusion was false. Then, there exists a $\tilde{\boldsymbol{\kappa}}^* \in S_1$ such that for all $\boldsymbol{\kappa} \in \Pi \cap \boldsymbol{N}(\tilde{\boldsymbol{\kappa}}^*)$, we have $J(\boldsymbol{\kappa}) \geq J(\tilde{\boldsymbol{\kappa}}^*) \geq J(\boldsymbol{\kappa}^*)$. We claim that there exists a $\boldsymbol{d} \in \boldsymbol{D}$ such that $\|\tilde{\boldsymbol{\kappa}}^* - \boldsymbol{\kappa}^* - \boldsymbol{d}\| > \|\tilde{\boldsymbol{\kappa}}^* - \boldsymbol{\kappa}^*\|$ and $\tilde{\boldsymbol{\kappa}}^* - \boldsymbol{d} \in \boldsymbol{\Pi} \cap \boldsymbol{N}(\tilde{\boldsymbol{\kappa}}^*)$. To establish this claim, we note that $\|\tilde{\boldsymbol{\kappa}}^* - \boldsymbol{\kappa}^*\|^2 = \sum_{i=1}^{NM} \left(\tilde{\boldsymbol{\kappa}}_i^* - \boldsymbol{\kappa}_i^*\right)^2$. There are two cases to be considered. (i) there exists some i, j such that $\tilde{\boldsymbol{\kappa}}_i^* - \boldsymbol{\kappa}_i^* > \boldsymbol{0}$ and $\boldsymbol{\kappa}_j^* - \tilde{\boldsymbol{\kappa}}_j^* < 0$. (ii) $\tilde{\boldsymbol{\kappa}}_i^* - \boldsymbol{\kappa}_i^* \geq \boldsymbol{0}$ or $\tilde{\boldsymbol{\kappa}}_i^* - \boldsymbol{\kappa}_i^* < \boldsymbol{0}$ for all $1 \leq i \leq NM$. For case (ii), let $i = \max_{1 \leq k \leq NM} \left|\tilde{\boldsymbol{\kappa}}_k^* - \boldsymbol{\kappa}_k^*\right|$ and $j = \min_{1 \leq k \leq NM} \left|\tilde{\boldsymbol{\kappa}}_k^* - \boldsymbol{\kappa}_k^*\right|$. Now choose $\boldsymbol{d}^* = \boldsymbol{e}_{i,-j}$. Since $\tilde{\boldsymbol{\kappa}}^* - \boldsymbol{d} \in \boldsymbol{N}(\boldsymbol{\kappa}^*)$, we have $J(\tilde{\boldsymbol{\kappa}}^* - \boldsymbol{d}) \geq J(\boldsymbol{\kappa}^*)$. By Lemma 4.1, we have $F(\tilde{\boldsymbol{\kappa}}^* - \boldsymbol{d}, \boldsymbol{\kappa}^*, \boldsymbol{q}, \boldsymbol{r}) < F(\tilde{\boldsymbol{\kappa}}^*, \boldsymbol{\kappa}^*, \boldsymbol{q}, \boldsymbol{r})$. This is a contradiction as $\tilde{\boldsymbol{\kappa}}^*$ is a discrete local minimizer.Thus, the conclusion of the theorem follows.

Theorem 4.3 *Suppose that $\boldsymbol{\kappa}^*$ is a local but not a global minimizer of $J(\boldsymbol{\kappa})$. Then, $F(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*, q, r)$ has a discrete minimizer in the region*

$$S_2 = \{\kappa | J(\boldsymbol{\kappa}) < J(\boldsymbol{\kappa}^*), \boldsymbol{\kappa} \in \boldsymbol{\Pi}\}.$$

Proof Since $\boldsymbol{\kappa}^*$ is a local but not a global minimizer of $J(\boldsymbol{\kappa})$, there exists a point $\tilde{\boldsymbol{\kappa}}^* \in \Pi$ such that $J(\tilde{\boldsymbol{\kappa}}^*) + r < J(\boldsymbol{\kappa}^*)$, where r is chosen according to (21.23). Hence, $F(\tilde{\boldsymbol{\kappa}}^*, \boldsymbol{\kappa}^*, \boldsymbol{q}, \boldsymbol{r}) = 0$. However, $F(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*, \boldsymbol{q}, \boldsymbol{r}) \geq 0$ for all $\boldsymbol{\kappa} \in \boldsymbol{\Pi}$. This implies that $\tilde{\boldsymbol{\kappa}}^*$ is a discrete minimizer and satisfies $J(\tilde{\boldsymbol{\kappa}}^*) < J(\boldsymbol{\kappa}^*)$.

By Theorem 4.1, Theorem 4.2 and Theorem 4.3, the function $F(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*, \boldsymbol{q}, \boldsymbol{r})$ is, indeed, a discrete filled function if $r > 0$ and $q > 0$ are chosen appropriately small. Now we present a numerical algorithm to search for a global minimizer of $J(\boldsymbol{\kappa})$ over Π based on the theoretical established above.

Algorithm 4.2

1. Take an initial point $\boldsymbol{\kappa_1} \in \boldsymbol{\Pi}$ and initialize the tolerance ε. Let $r = 0.1,\ q = 0.01$.

2. From $\boldsymbol{\kappa_1}$, use Algorithm 4.1 to find a local minimizer $\boldsymbol{\kappa}_1^*$ of $J(\boldsymbol{\kappa})$ over Π.
3. If $r \leq \varepsilon$, stop. Otherwise, construct a discrete filled function

$$F\left(\boldsymbol{\kappa}, \boldsymbol{\kappa}_1^*, \boldsymbol{q}, \boldsymbol{r}\right) = \frac{1}{q + \left\|\boldsymbol{\kappa} - \boldsymbol{\kappa}_1^*\right\|} \varphi_q\left(\max\left\{J(\boldsymbol{\kappa}) - J\left(\boldsymbol{\kappa}_1^*\right) + r,\ 0\right\}\right).$$

   Use Algorithm 4.1 to find its local minimizer $\boldsymbol{\kappa}^*$. In the process of minimizing the discrete filled function $F\left(\boldsymbol{\kappa}, \boldsymbol{\kappa}_1^*, \boldsymbol{q}, \boldsymbol{r}\right)$, if we find an iterative point $\boldsymbol{\kappa}$ such that $J(\boldsymbol{\kappa}) < J\left(\boldsymbol{\kappa}_1^*\right)$, then let $\boldsymbol{\kappa_1} = \boldsymbol{\kappa}^*$ and go to Step 1.
4. If $J(\boldsymbol{\kappa}^*) < J\left(\boldsymbol{\kappa}_1^*\right)$, let $\boldsymbol{\kappa_1} = \boldsymbol{\kappa}^*$ and go to Step 1. Otherwise, set $r = r/10; q = q/10$ and go to Step 3.

21.5 Numerical Example

In this section, we will apply the method developed in Sect. 21.3 and Sect. 21.4 to our test problems.

Example 5.1 The three sub-systems are given by

$$1: \begin{cases} \dot{x}_1 = x_1 + u_1 \\ \dot{x}_2 = 2x_1 + 3x_2 + u_2 \end{cases} \quad 2: \begin{cases} \dot{x}_1 = x_1 + u_1 \\ \dot{x}_2 = x_1 + x_2 + u_2 \end{cases} \quad 3: \begin{cases} \dot{x}_1 = x_1 + x_2 + u_1 \\ \dot{x}_2 = 2x_1 - x_2 + u_2 \end{cases}$$

the switching detection delay is 0.01. Let $\delta(t)$ be the switching function, $\gamma(t)$ be the detection of the switching signal, i.e., $\gamma(t) = \delta(t - 0.01)$. Let the maximum switching times be 3. Our objective is to find a feedback law $u(t) = [u_1, u_2]^\top = K_{\gamma(t)} x(t)$, where $K_{\gamma(t)} \in \mathbb{R}$, such that the cost functional

$$J\left(\delta, K_\gamma\right) = (x_1(1))^2 + (x_2(1) - 1.0)^2$$

is minimized subject to the following condition:

$$-3 \leq K_\gamma \leq 3.$$

For a given switching sequence $\boldsymbol{\kappa} = (i_1, i_2, i_3, i_4)$, we suppose that the switching points are τ_1, τ_2, τ_3. First, we use the time scaling transformation (21.12) to map the state switching points τ_1, τ_2, τ_3, and the control switching points $0.01, \tau_1 + 0.01$, $\tau_2 + 0.01, \tau_3 + 0.01$, into $1, 2, 3$, and $1/2, 1 + 1/2, 2 + 1/2, 3 + 1/2$, respectively. Then, let the initial switching sequence be $\{1, 0, 0, 0, 1, 0, 0,\ 0, 1, 1, 0, 0\}$. We incorporate MISER 3.3 as a sub-program and set the lower bound of the switching parameters to be 0.05. Using Algorithm 4.1 to search its local optimal switching sequence, the obtained local optimal switching is $\{0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0\}$, the corresponding parameters of switching instants and the constant of feedback

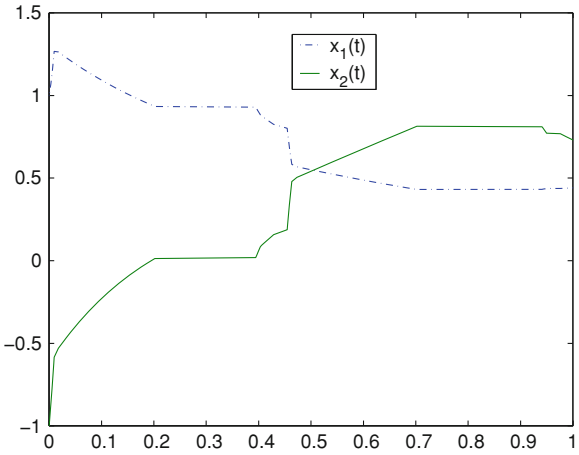


Fig. 21.2 The trajectory of the optimal state

are $0.38354, 0.05, 0.47646, 0.05, -2.603, \ -2.5069, -2.514, -2.4682$, respectively. The corresponding optimal cost is 0.1727. Thus, the 3 state switching instants are $0.39354, 0.45354, 0.94$, the control switching instants are $0.01, 0.40454, 0.46354, 0.95$. Then, we set the tolerance $\varepsilon = 10^{-5}$ and use discrete filled function to find its local minimizer. However, we cannot find any local minimizer of the corresponding discrete filled function. The program stopped because of $r = \varepsilon$ and the last obtained switching sequence is $\{0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0\}$, the corresponding value of the discrete filled function and the cost functional are 0.12497, 0.684409, respectively. Thus, the local minimizer $\{1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0\}$ is also a global optimal switching sequence. The optimal state is depicted in Fig. 21.2.

21.6 Summary

In this paper, the optimal control problem governed by switched system with time delay detection is considered. Considering the switching sequence as well as the switching instants as decision variables, we developed a two-level optimization method to solve it. In the first level, the switching sequence is fixed, thus it can be transformed into a parameter selection problem. For this problem, we use MISER 3.3 to solve it. For the second level, the switching sequence is considered as decision variables. For this discrete optimization problem, we introduced the filled function method to solve it. At last, a numerical example is presented to show that the efficiency of our method.

Acknowledgments This paper is partially supported by a research from the Australian Research Council.

References

1. Cassandras CG, Pepyne DL, Wardi Y (2001) Optimal control of a class of hybrid systems. IEEE Trans Automat Contr 46:398–415
2. Egerstedt M, Wardi Y, Delmotte F (2003) Optimal control of switching times in switched dynamical systems. In: IEEE Conference on Decision and Control, Maui, Hawaii, USA, pp 2138–2143
3. Xu X, Antsaklis PJ (2002) Optimal control of switched autonomous systems. In: IEEE Conference on Decision and Control, Las Vegas, USA
4. Xu X, Antsaklis PJ (2004) Optimal control of switched systems based on parameterization of the switching instants. IEEE Trans Automat Contr 49:2–16
5. Xu X, Antsaklis PJ (2003) Antsaklis, results and perspectives on computational methods for optimal control of switched systems. In: 6th International Workshop on Hybrid Systems: Computational and Control, Prague, The Czech Republic
6. Bengea SC, DeCarlo RA (2005) Optimal control of switching systems. Automatica 41:11–27
7. Lee HWJ, Teo KL, Jennings LS, Rehboek V (1997) Control parameterizatmn enhancing technique for optimal control problems. Dyn Syst Appl 6:243–262
8. Xie G, Wang L (2005) Stabilization of switched linear systems with time-delay in detection of switching signal. J Math Anal Appl 305:277–290
9. Wu CZ, Teo KL, Zhao Yi, Yan WY (2005) Solving an identification problem as an impulsive optimal parameter selection problem. Comput Math Appl 50:217–229
10. Teo KL, Jennmgs LS, Lee HWJ, Rehbock V (1999) The control parametrization enhancing transform for constrained optimal control problems. J Australian Math Soc B 40:314–335
11. Teo KL, Goh CJ, Wong KH (1991) A unified computational approach to optimal control problems. Longman Scientific and Technical, Longman Group UK Limited, Essex, England
12. Jennings LS, Teo KL, Fisher ME, Goh CJ (2005) MISER version 3, optimal control software, theory and user manual. http://www.maths.uwa.edu.au/∼les/miser3.3.html. Department of Mathematics, University of Western Australia
13. Gu YH, Wu ZY (2006) A new filled function method for nonlinear integer programming problem. Appl Math Comput 173:938–950
14. Zhu W (1997) An approximate algorithm for nonlinear integer programming. Appl Math Comput 93:183–193
15. Yang YJ, Liang YM (2007) A new discrete filled function algorithm for discrete global optimization. J Comput Appl Math 202:280–291