Ricardo S. Sánchez-Peña
Joseba Quevedo Casín
Vicenç Puig Cayuela
Editors

Identification and Control

The Gap between Theory and Practice

Springer

Identification and Control

Ricardo S. Sánchez Peña, Joseba Quevedo Casín
and Vicenç Puig Cayuela (Eds.)

Identification and Control

The Gap between Theory and Practice

Springer

Ricardo S. Sánchez Peña, PhD
ICREA and Advanced Control Systems (SAC)
ESAII
Universitat Politècnica de Catalunya
Terrassa, Barcelona
Spain

Joseba Quevedo Casín, PhD
Advanced Control Systems (SAC)
ESAII
Universitat Politècnica de Catalunya
Terrassa, Barcelona
Spain

Vicenç Puig Cayuela, Dr Eng
Advanced Control Systems (SAC)
ESAII
Universitat Politècnica de Catalunya
Terrassa, Barcelona
Spain

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2007929433

ISBN 978-1-84628-898-2 e-ISBN 978-1-84628-899-9 Printed on acid-free paper

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springer.com

Foreword

This book is an ambitious attempt to bridge the infamous gap between theory and practice in control. The book has ten chapters, written by seasoned researchers from all over the world. It is organized in four parts: Large-scale Problems, Aerospace, Vision and Sound and Electromechanical, and covers a very wide range of problems from chemical reactors to magnetic bearings. In spite of the wide diversity of topics and authors a coherent presentation is obtained by a uniform organization imposed by the editors. Each chapter begins with a brief summary of the theory and a description of the application, followed by results of simulations or experiments and conclusions. The problem descriptions and the discussions about the relevance of the theory are more detailed than presentations normally found in papers. The book is a nice complement to traditional textbooks and the wide range of topics illustrate the richness and ubiquity of identification and control.

Santa Barbara, California, USA *Karl Johan Åström*
March 2007

Preface

System theory, in particular, automatic control and system identification have experienced a fast evolution in the past decades. Many new methods have been developed, performance requirements in traditional engineering areas have significantly increased and new and more demanding applications in other areas of engineering and science have appeared. Many textbooks have been written that present the relevant theory and sometimes "practical" examples to which the theory has been applied. Other books focus directly on the practical issues involved, leaving the theory out. However, there is still an important gap between theory and its application to practical problems. Engineering decisions are usually made without the complete assessment of existing theoretical tools. Many times, the key to the applicability of new methods is based on such decisions. Hence, a problem appears when either theory is lacking – it trails behind the practical issues, or when engineering decisions are taken without the knowledge of recent theoretical developments.

Usually the theory/practice gap is covered by extensive simulations and/or experimental testing. There are very few (if any) situations in which the theory "fits" the practical application without any extra modification. This gap is even more important in critical situations such as nuclear plants and space applications, where the misfit between theory and practice could lead to dangerous or disastrous scenarios. Nevertheless, this gap is the motor that keeps theoreticians and practitioners active in developing new techniques and applying them in all areas of engineering. The purpose of this book it to present some examples of this gap and in this way serve as a reference, where new applied research areas are pointed out.

All chapters are written by textbook authors involved in "real world" applications from different areas of engineering: acoustics, aerospace, chemical, computer vision, electromechanical, industrial and mining. The chapters are organized with a similar format: (1) a brief theoretical background section with references to previous works; (2) the description of the application(s); (3) a presentation of simulations and/or experimental results, and (4) conclusions highlighting the discrepancy between theory and practice. The latter is what distinguishes this book from others: the gap is deliberately exposed, not hidden. Furthermore, in each chapter the complete analysis and

design process is covered, describing practical issues and decisions that accomodate the existing theory to the particular problem.

The book will be useful for engineers facing complex applications and willing to use recent methods of identification and control in solving them, in order to take sound engineering decisions. In addition, applied researchers looking for areas in which to contribute in reducing the theory/practice gap and exploring practical issues should also benefit greatly from it.

The chapters describe how different identification and control techniques are applied to a wide variety of engineering applications. The identification methods go from parametric identification of linear and nonlinear models, to robust set membership identification and model (in)validation. Control techniques range from stochastic optimal control to model predictive or robust $\mathcal{H}_\infty$ and μ-synthesis control. Also, fault tolerant and linear parameter varying (LPV) control, and the recent area of hybrid systems control techniques, have been used. Among the applications we can mention: mine planning, polymerization reactors, a copper foundry process, and the sewer network of Barcelona. Also, some electromechanical processes such as active transmission and suspension, and a magnetic-bearing rotor appear. Computer vision, aerospace applications like the NASA-JPL flexible structure or aeroservoelasticity in the F18 aircraft and acoustic-noise attenuation, are also included.

Finally, we would like to thank all the authors who contributed to this book, for producing an excellent collection of high-quality technical material. We would also like to thank Springer UK personnel, particularly Oliver Jackson, who helped us greatly in the development of this project, and our colleague Ari Ingimundarson for refining part of this text.

Terrassa, Barcelona, Spain,
March 2007

Ricardo S. Sánchez Peña
Joseba Quevedo Casín
Vicenç Puig Cayuela

Contents

Part IV Electromechanical

List of Contributors

Pedro Albertos
Universidad Politécnica de Valencia,
Dept. Systems Engineering and Control,
P.O. Box. 22012, E-46071, Valencia,
Spain
pedro@aii.upv.es

Luis Alvergue
Louisiana State University,
Department of Electrical and Computer
Engineering,
Baton Rouge, LA 70803-5901, USA
lalver1@lsu.edu

Manuel R. Arahal
Universidad de Sevilla,
Departamento de Ingeniería de
Sistemas y Automática,
Sevilla, Spain
arahal@esi.us.es

Alberto Bemporad
Università di Siena,
Dipartimento di Ingegneria
dell'Informazione
Via Roma 56, I-53100
Siena, Italy
bemporad@dii.unisi.it

Carlos Bordons
Universidad de Sevilla,
Departamento de Ingeniería de
Sistemas y Automática,
Sevilla, Spain
bordons@esi.us.es

Richard D. Braatz
University of Illinois at Urbana-
Champaign,
Department of Chemical and Biomolec-
ular Engineering,
293 Roger Adams Laboratory, Box C-3,
600 South Mathews Avenue, Urbana, IL
61801, USA
braatz@uiuc.edu

Martin J. Brenner
NASA Dryden Flight Research Center,
Aerostructures Branch, Edwards, CA,
USA
Martin.J.Brenner@nasa.gov

Eduardo F. Camacho
Universidad de Sevilla,
Departamento de Ingeniería de
Sistemas y Automática,
Sevilla, Spain
eduardo@esi.us.es

Octavia Camps
Northeastern University,
ECE Department,
Boston, MA 02115, USA
camps@ece.neu.edu

Jie Chen
University of California,
Department of Electrical Engineering,
Riverside, CA 92521, USA
jchen@ee.ucr.edu

Miquel A. Cugueró
Universitat Politècnica de Catalunya,
Sistemes Avançats de Control,
Rbla. Sant Nebridi 10,
Terrassa, 08222 Barcelona, Spain
miquel.angel.cuguero@upc.edu

Alicia Esparza
Universidad Politécnica de Valencia,
Dept. Systems Engineering and Control,
P.O. Box. 22012, E-46071, Valencia,
Spain
alespei@isa.upv.es

Alejandro S. Ghersin
Universidad Nacional de Quilmes,
Ingeniería en Automatización y Control,
and Universidad de Buenos Aires,
Facultad de Ingeniería,
Buenos Aires, Argentina
aghersin@unq.edu.ar

Graham Goodwin
The University of Newcastle,
Centre for Complex Dynamic Systems
and Control,
School of Electrical Engineering and
Computer Science,
Callaghan NSW 2308, Australia
graham.goodwin@newcastle.edu.au

Guoxiang Gu
Louisiana State University,
Department of Electrical and Computer
Engineering,
Baton Rouge, LA 70803-5901, USA
ggu@lsu.edu

Eric J. Hukkanen
Real-Time Analytics,
Mettler-Toledo AutoChem, Inc.,
14833 NE 87th Street, Redmond, WA
98052, USA
eric.hukkanen@mt.com

Tamer Inanc
University of Louisville,
ECE Department,
Louisville, KY 40292, USA
t.inanc@louisville.edu

Ari Ingimundarson
Universitat Politècnica de Catalunya,
Sistemes Avançats de Control,
Rbla. Sant Nebridi 10,
Terrassa, 08222 Barcelona, Spain
ari.ingimundarson@upc.edu

Sunil L. Kukreja
NASA Dryden Flight Research Center,
Aerostructures Branch, Edwards, CA,
USA
Sunil.Kukreja@nasa.gov

Bernardo Morcego
Universitat Politècnica de Catalunya,
Sistemes Avançats de Control,
Rbla. Sant Nebridi 10,
Terrassa, 08222 Barcelona, Spain
bernardo.morcego@upc.edu

Carlos Ocampo Martínez
Universitat Politècnica de Catalunya,
Sistemes Avançats de Control,
Rbla. Sant Nebridi 10,
Terrassa, 08222 Barcelona, Spain
carlos.ocampo@upc.edu

Vicenç Puig Cayuela
Universitat Politècnica de Catalunya,
Sistemes Avançats de Control,
Rbla. Sant Nebridi 10,
Terrassa, 08222 Barcelona, Spain
vicenc.puig@upc.edu

Cristian R. Rojas
The University of Newcastle,
Centre for Complex Dynamic Systems and Control,
School of Electrical Engineering and Computer Science,
Callaghan NSW 2308, Australia
cristian.rojas@studentmail.newcastle.edu.au

Antonio Sala
Universidad Politécnica de Valencia,
Dept. Systems Engineering and Control,
P.O. Box. 22012, E-46071, Valencia, Spain
asala@isa.upv.es

Ricardo S. Sánchez Peña
ICREA and Universitat Politècnica de Catalunya,
Sistemes Avançats de Control,
Rbla. Sant Nebridi 10,
Terrassa, 08222 Barcelona, Spain
ricardo.sanchez-pena@upc.edu

María M. Seron
The University of Newcastle,
Centre for Complex Dynamic Systems and Control,
School of Electrical Engineering and Computer Science,
Callaghan NSW 2308, Australia
maria.seron@newcastle.edu.au

Roy S. Smith
University of California,
Department of Electrical and Computer Engineering
Santa Barbara, CA 93106, USA
roy@ece.ucsb.edu

Mario Sznaier
Northeastern University,
ECE Department,
Boston, MA 02115, USA
msznaier@ece.neu.edu

José M. Tejera
Atlantic Copper,
Departamento de Servicios Generales,
Electricidad e Instrumentación,
Huelva, Spain.
Jose-Maria-Tejera@fmi.com

Jeremy G. VanAntwerp
Calvin College, Department of Engineering,
1726 Knollcrest Circle, Grand Rapids,
MI 49546, USA
jva@calvin.edu

Meimei Zhang
The University of Newcastle,
Centre for Complex Dynamic Systems and Control,
School of Electrical Engineering and Computer Science,
Callaghan NSW 2308, Australia
MeiMei.Zhang@BHPBilliton.com

Part I

Large-scale Problems

1

Identification and Control of Polymerization Reactors

Eric J. Hukkanen[1], Jeremy G. VanAntwerp[2], and Richard D. Braatz[3]

[1] Real-Time Analytics, Mettler-Toledo AutoChem, Inc., 14833 NE 87th Street, Redmond, WA 98052, USA eric.hukkanen@mt.com
[2] Calvin College, Department of Engineering, 1726 Knollcrest Circle, Grand Rapids, MI 49546, USA jva@calvin.edu
[3] University of Illinois at Urbana-Champaign, Department of Chemical and Biomolecular Engineering, 293 Roger Adams Laboratory, Box C-3, 600 South Mathews Avenue, Urbana, IL 61801, USA braatz@uiuc.edu

Summary. This chapter considers the identification and control of free-radical polymerization reactors. A discussion of the modeling and simulation of such reactors is followed by an optimal control study that demonstrates the potential of optimal control of the molecular-weight distribution based on mechanistic models. Achieving this potential in a batch reactor requires an accurate estimation of the free-radical polymerization kinetic parameters. The remainder of the chapter describes an experimental investigation of the free-radical polymerization of methyl methacrylate, in which modern sensing techniques are used to estimate kinetic parameters. The monomer conversion is measured using inline ATR-FTIR spectroscopy and robust chemometrics, and the molecular-weight distribution is measured by gas-permeation chromatography. The resulting parameter estimates and confidence intervals are used to discuss the importance of various reactions in the free-radical polymerization reaction mechanism. Discrepancies between theory and experiments are discussed.

1.1 Background

Accurate mechanistic models for predicting the molecular-weight distribution are necessary for effective design and control of free-radical polymerization reactors. Many studies have been conducted to identify mechanistic models for free-radical polymerization ([8, 9, 11–17, 20, 22, 23, 26–28]). The accuracy of the estimates of the kinetic parameters is directly related to the accuracy and reliability of the sensors available for measuring the monomer conversion and molecular-weight properties. This chapter makes use of two sensing technologies, *in situ* attenuated total reflection (ATR) Fourier transform infrared (FTIR) spectroscopy and gel-permeation chromatography, for kinetic-parameter estimation in polymerization reactors. The chapter also considers methods by which models for such processes can be simulated, as a necessary first step to model identification, and considers the use of such models for control.

The first sensing technology is enabled by the availability of midrange infrared transmitting optical fibers [21]. A significant advantage of ATR-FTIR spectroscopy over most other methods for concentration measurement is the ability to provide simultaneous measurement for multiple chemical species. This is useful, for example, for measuring the monomer concentration in a polymerization reactor independent of the concentrations of dimers, trimers, *etc*.

In ATR-FTIR spectroscopy, the infrared spectrum is characteristic of the vibrational structure of the substance in immediate contact with the ATR immersion probe. The ATR-FTIR probe, coupled with robust chemometrics, is used to monitor the conversion of monomer during experiments. The second sensing technology, gel-permeation chromatography (GPC), measures the molecular-weight distribution (MWD), which is the polymer property that determines its end-use characteristics [21]. First, a calibration curve is constructed using narrowly distributed polymer samples of known molecular-weight. Then this calibration curve is used offline to determine the MWD over the course of a reaction.

The specific polymerization under investigation is the free-radical polymerization of methyl methacrylate. Various assumptions have been used to develop mechanistic models for this process. It has been assumed that monomer transfer, k_{trm}, and/or termination by combination, k_{tc}^o, do not significantly influence the polymerization process [8, 11–16, 22, 27, 28]. The results of a literature search [8, 9, 12, 14, 17, 20, 22, 23, 26, 27] of kinetic parameters used for modeling methyl methacrylate bulk polymerization are summarized in Table 1.1. The large variability of the values of the kinetic parameters may be partly due to the inclusion/exclusion of k_{trm} and k_{tc}^o. Also, past studies usually fit kinetic parameters based only on a measurement of the monomer conversion over time for a small number of batch polymerizations, and this is very limited information for estimating the values of multiple kinetic parameters, as we will see later in this study. For control purposes, it is necessary to have an estimate of the error in the parameter estimates. Most papers do not provide any uncertainty assessment of their parameter estimates. In this chapter, monomer conversion profiles (as measured by *in situ* ATR-FTIR spectroscopy [19]) and average molecular-weight properties (determined by offline gel-permeation chromatography) are used to determine kinetic parameters for the free-radical polymerization of methyl methacrylate (MMA) with benzoyl peroxide (BPO) initiator. 95% confidence intervals are reported with the parameter estimates.

The next section describes the polymerization kinetic model, model simulation, description of the experimental system, and formulation of the parameter estimation and optimal control problems. This is followed by simulation-model verification and an evaluation of the potential of optimal control of the molecular-weight distribution in free-radical polymerization. Next, the kinetic parameters needed for optimal control design are experimentally determined. The chapter ends with a discussion of discrepancies between the experimental data and the model outputs obtained with the estimated kinetic parameters.

Table 1.1. Kinetic parameter estimates (along with 95% confidence regions) obtained from literature values

Parameters	Literature values	units
f	0.10–1.0	—
k_{trm}	1.6 ±7.8	[L mol^{-1} min^{-1}]
$k_{td}^o \times 10^{-9}$	1.8 ±0.8	[L mol^{-1} min^{-1}]
$k_{tc}^o \times 10^{-8}$	7.6 ±1.1	[L mol^{-1} min^{-1}]
$k_p^o \times 10^{-4}$	3.2 ±0.7	[L mol^{-1} min^{-1}]

1.2 Free-radical Polymerization

This section describes mechanistic modeling of a free-radical bulk polymerization reactor and an algorithm for the direct simulation of the molecular-weight distribution. It is straightforward to revise the model to treat solution, suspension, or emulsion polymerization, or anionic or cationic polymerization. Subsequent sections use this mechanistic model for parameter estimation and optimal control.

1.2.1 Kinetic Model

The reaction mechanism with reaction rates for free-radical polymerization with a single monomer and initiator are

$I_2 \xrightarrow{k_d} 2I\cdot$	$r_d = 2fk_d[I_2]$	initiator decomposition
$I\cdot + M \xrightarrow{k_i} P_1$	$r_i = k_i\,[M]\,[I\cdot]$	initiation
$P_n + M \xrightarrow{k_p} P_{n+1}$	$r_p = k_p\,[M]\,[P_n]$	propagation
$P_n + M \xrightarrow{k_{trm}} D_n + P_1$	$r_{trm} = k_{trm}\,[M]\,[P_n]$	monomer transfer
$P_n + P_m \xrightarrow{k_{tc}} D_{n+m}$	$r_{tc} = k_{tc}\,[P_m]\,[P_n]$	termination by combination
$P_n + P_m \xrightarrow{k_{td}} D_n + D_m$	$r_{td} = k_{td}\,[P_m]\,[P_n]$	termination by disproportionation

It is straightforward to generalize the mechanism to include chain-transfer and solvent-transfer reactions and multiple monomers and initiators.

The molar balances for a well-mixed batch reactor are, for initiator decomposition

$$\frac{1}{V}\frac{\mathrm{d}\left([I_2]V\right)}{\mathrm{d}t} = -k_d[I_2] \tag{1.1}$$

free-radical concentration

$$\frac{1}{V}\frac{\mathrm{d}\left([I\cdot]V\right)}{\mathrm{d}t} = -k_i[I\cdot][M] + 2fk_d[I_2] \tag{1.2}$$

monomer concentration

$$\frac{1}{V}\frac{\mathrm{d}\left([M]V\right)}{\mathrm{d}t} = -k_i[I\cdot][M] - \left(k_p + k_{trm}\right)[M]\sum_{j=1}^{\infty}[P_j] \tag{1.3}$$

live polymer chains

$$\begin{aligned}\frac{1}{V}\frac{\mathrm{d}\left([P_1]V\right)}{\mathrm{d}t} &= k_i[I\cdot][M] - k_p[M][P_1] - k_{trm}[M]\left([P_1] - \sum_{j=1}^{\infty}[P_j]\right) \\ &\quad - \left(k_{tc} + k_{td}\right)[P_1]\sum_{j=1}^{\infty}[P_j]\end{aligned} \tag{1.4}$$

$$\begin{aligned}\frac{1}{V}\frac{\mathrm{d}\left([P_m]V\right)}{\mathrm{d}t} &= k_p[M]\left([P_{m-1}] - [P_m]\right) - k_{trm}[M][P_m] \\ &\quad - \left(k_{tc} + k_{td}\right)[P_m]\sum_{j=1}^{\infty}[P_j] \text{ for } m = 2, \cdots, n\end{aligned} \tag{1.5}$$

and dead polymer chains

$$\frac{1}{V}\frac{\mathrm{d}\left([D_m]V\right)}{\mathrm{d}t} = k_{trm}[M][P_m] + k_{td}[P_m]\sum_{j=1}^{\infty}[P_j] + \frac{1}{2}k_{tc}\sum_{j=1}^{m-1}[P_j][P_{m-j}] \quad \text{for } m = 2, \cdots, n \tag{1.6}$$

The parameters $k_d, k_i, k_p, k_{trm}, k_{tc}$, and k_{td} are rate constants and f is the initiator efficiency. The short average lifetime ($<<$ 1 s) of an initiator radical molecule, $I\cdot$, implies that the quasisteady-state approximation can be applied:

$$\frac{1}{V}\frac{\mathrm{d}([I\cdot]V)}{\mathrm{d}t} = 0 \tag{1.7}$$

and

$$k_i[I\cdot][M] = 2fk_d[I_2] \tag{1.8}$$

The total concentrations of live polymer molecules and monomer are

$$[P] = \sum_{j=1}^{\infty}[P_j] \tag{1.9}$$

$$[M] = \frac{[M_o](1-x)}{1+\varepsilon x} \tag{1.10}$$

where x is the conversion and ε is the volume-contraction factor

$$\varepsilon = \frac{\rho_m(T) - \rho_p(T)}{\rho_p(T)} \tag{1.11}$$

that takes into account the density differences between the polymer and monomer. Initially, the reactor consists of pure monomer. Throughout the reaction, the volume of the reactant/product mixture changes according to

$$V = V_0(1+\varepsilon x) \tag{1.12}$$

$$\frac{\mathrm{d}V}{\mathrm{d}t} = V_0\varepsilon\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.13}$$

In this work, the gel effect or "Trommsdorf effect" is accounted for using the free-volume correlation [25]. The void volume in a monomer/polymer solution is defined as the free volume. As conversion increases, the free volume decreases. Established methods are implemented to account for the decrease in propagation and termination due to strong diffusion limitations. The overall termination rate, k_t, takes the form

$$k_t = k_{td} + k_{tc} \tag{1.14}$$

$$k_t = k_t^o g_t \tag{1.15}$$

where

$$g_t = \begin{cases} 0.10575e^{17.15\nu_f - 0.01715(T-273.15)} & \nu_f > \nu_{f,cr} \\ 0.23\times 10^{-5}e^{75\nu_f} & \nu_f \le \nu_{f,cr} \end{cases} \tag{1.16}$$

and the propagation rate takes the form

$$k_p = k_p^o g_p \tag{1.17}$$

$$g_p = \begin{cases} 1 & \nu_f > \nu_{f,cr} \\ 0.71\times 10^{-4}e^{71.53\nu_f} & \nu_f \le \nu_{f,cr} \end{cases} \tag{1.18}$$

The free volume, ν_f, is defined by

$$\nu_f = \phi_m \nu_{f,m} + \phi_p \nu_{f,p} \tag{1.19}$$

where ϕ_m and ϕ_p are the volume fractions of monomer and polymer, respectively, and the monomer $\nu_{f,m}$ and polymer $\nu_{f,p}$ free volumes are defined by

$$\nu_{f,m} = 0.025 + 0.001(T - 167) \tag{1.20}$$

$$\nu_{f,p} = 0.025 + 0.00048(T - 387) \tag{1.21}$$

The critical free volume, $\nu_{f,cr}$, is defined by

$$v_{f,cr} = 0.1856 - 2.965 \times 10^{-4}(T - 273.15) \tag{1.22}$$

This free-volume model is adapted from [25] and has been used in several investigations [11–13, 20, 26].

1.2.2 Model Simulation

This section describes how the kinetic model was simulated and presents the moment equation method that was used to verify the accuracy of the simulation results.

Moment equations

The moments are defined by

$$\lambda_i = \sum_{j=1}^{\infty} j^i [P_j] \tag{1.23}$$

for live (radical) polymer chains and

$$\mu_i = \sum_{j=2}^{\infty} j^i [D_j] \tag{1.24}$$

for dead polymer chains, where i is an integer. The infinite set of molar balance equations can be used to derive a finite number of ordinary different equations (ODEs) by using the relationships

$$\sum_{j=1}^{\infty} \frac{\mathrm{d}}{\mathrm{d}t} \left(j^i [P_j] \right) = \frac{\mathrm{d}\lambda_i}{\mathrm{d}t} \qquad \text{for } i = 0, 1, 2 \tag{1.25}$$

and

$$\sum_{j=2}^{\infty} \frac{\mathrm{d}}{\mathrm{d}t} \left(j^i [D_j] \right) = \frac{\mathrm{d}\mu_i}{\mathrm{d}t} \qquad \text{for } i = 0, 1, 2 \tag{1.26}$$

The resulting finite set of ODEs are

$$\frac{\mathrm{d}[I_2]}{\mathrm{d}t} = -\left(k_d + \frac{\varepsilon}{1 + \varepsilon x} \frac{\mathrm{d}x}{\mathrm{d}t} \right) [I_2] \tag{1.27}$$

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{2 f k_d [I_2] (1 + \varepsilon x)}{[M_o]} + (k_p + k_{trm}) (1 - x) \lambda_0 \tag{1.28}$$

$$\frac{\mathrm{d}\lambda_0}{\mathrm{d}t} = 2fk_d[I_2] - (k_{td} + k_{tc})\,\lambda_0^2 - \frac{\varepsilon\lambda_0}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.29}$$

$$\frac{\mathrm{d}\lambda_1}{\mathrm{d}t} = 2fk_d[I_2] + k_p[M]\lambda_0 + k_{trm}[M]\,(\lambda_0 - \lambda_1) - (k_{td} + k_{tc})\,\lambda_0\lambda_1 - \frac{\varepsilon\lambda_1}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.30}$$

$$\frac{\mathrm{d}\lambda_2}{\mathrm{d}t} = 2fk_d[I_2] + k_p[M]\,(2\lambda_1 + \lambda_0) + k_{trm}[M]\,(\lambda_0 - \lambda_2) - (k_{td} + k_{tc})\,\lambda_0\lambda_2 - \frac{\varepsilon\lambda_2}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.31}$$

$$\frac{\mathrm{d}\mu_0}{\mathrm{d}t} = k_{trm}[M]\lambda_0 + \left(k_{td} + \frac{1}{2}k_{tc}\right)\lambda_0^2 - \frac{\varepsilon\mu_0}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.32}$$

$$\frac{\mathrm{d}\mu_1}{\mathrm{d}t} = k_{trm}[M]\lambda_1 + (k_{td} + k_{tc})\,\lambda_0\lambda_1 - \frac{\varepsilon\mu_1}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.33}$$

$$\frac{\mathrm{d}\mu_2}{\mathrm{d}t} = k_{trm}[M]\lambda_2 + k_{td}\lambda_0\lambda_2 + k_{tc}\left(\lambda_0\lambda_2 + \lambda_1^2\right) - \frac{\varepsilon\mu_2}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.34}$$

with initial conditions

$$[I_2](0) = [I_{2o}] \tag{1.35}$$

$$x(0) = 0 \tag{1.36}$$

$$\lambda_i(0) = \mu_i(0) = 0, \quad \text{for } i = 0, 1, 2 \tag{1.37}$$

The polymer product is commonly characterized by the number-average molecular-weight, $\bar{M}_n$, weight-average molecular-weight, $\bar{M}_w$, and polydispersity, PD. These product quality variables can be computed from the moments of live and dead polymer chains:

$$\bar{M}_n = M_w\frac{\lambda_1 + \mu_1}{\lambda_0 + \mu_0} \tag{1.38}$$

$$\bar{M}_w = M_w\frac{\lambda_2 + \mu_2}{\lambda_1 + \mu_1} \tag{1.39}$$

$$\mathrm{PD} = \frac{\bar{M}_w}{\bar{M}_n} \tag{1.40}$$

where M_w is the molecular-weight of the monomer. The above system of ODEs can be solved using any stiff solver such as DASSL [24] or VODE [5].

Simulation of the Molecular-weight Distribution

Typically, the molecular-weight distribution (MWD) is simulated using stochastic codes, which are time consuming. Here, the *full* MWD is simulated in less than 10 min per simulation run. The above molar balances are directly integrated using *parallel CVODE* [1, 6, 7, 10, 31]. The maximum number of ODEs was determined experimentally by gel-permeation chromatography (GPC), which was double-checked at the end of the simulation to confirm that the live and dead polymer concentrations at the largest chain lengths were negligible. The user selects the number of processors and the maximum number of ODEs allowed on each processor, and specifies which equations are on each processor. The parallel CVODE software

handles all details of the integration including parallelization among multiple processors. The user decides how to balance the computations among the processors. All communications between processors are handling using the message-passing interface (MPI) [18], using the basic MPI_SEND, MPI_RECV, and MPI_BCAST commands. For a more indepth description of the architecture of the parallel CVODE algorithm, the reader is referred to [1, 6, 7, 10, 31].

As the average lifetime of live polymer chains is very short, the quasisteady-state approximation (QSSA) was applied, which greatly reduces the stiffness:

$$\frac{1}{V}\frac{\mathrm{d}\left([P_m]V\right)}{\mathrm{d}t} = 0, \quad \text{for } m = 1, \cdots, \infty \tag{1.41}$$

Applying the QSSA reduces the stiffness to $\sim 10^4$, which is significantly more well behaved numerically. This leads to the system of ODEs solved by parallel CVODE:

$$\frac{\mathrm{d}[I_2]}{\mathrm{d}t} = -\left(k_d + \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}\right)[I_2] \tag{1.42}$$

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{2fk_d[I_2]\left(1+\varepsilon x\right)}{[M_o]} + \left(k_p + k_{trm}\right)\left(1-x\right)[P] \tag{1.43}$$

$$\frac{\mathrm{d}[P]}{\mathrm{d}t} = 2fk_d[I_2] - \left(k_{td}+k_{tc}\right)\lambda_0^2 - \frac{\varepsilon\lambda_0}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.44}$$

$$[P_1] = \frac{2fk_d[I_2] + k_{trm}[M][P]}{\left(k_p+k_{trm}\right)[M] + \left(k_{td}+k_{tc}\right)[P] + \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}} \tag{1.45}$$

$$[P_2] = \frac{k_p[M][P_1]}{\left(k_p+k_{trm}\right)[M] + \left(k_{td}+k_{tc}\right)[P] + \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}} \tag{1.46}$$

$$\vdots$$

$$[P_n] = \frac{k_p[M][P_{n-1}]}{\left(k_p+k_{trm}\right)[M] + \left(k_{td}+k_{tc}\right)[P] + \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}} \tag{1.47}$$

$$[P] = \sum_{i=1}^{n}[P_i] = \lambda_0, \tag{1.48}$$

$$\frac{\mathrm{d}[D_2]}{\mathrm{d}t} = k_{trm}[M][P_2] + k_{td}[P_2][P] + \frac{1}{2}k_{tc}[P_1]^2 - \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}[D_2] \tag{1.49}$$

$$\vdots$$

$$\frac{\mathrm{d}[D_n]}{\mathrm{d}t} = k_{trm}[M][P_n] + k_{td}[P_n][P] + \frac{k_{tc}}{2}\sum_{j=1}^{n-1}[P_j][P_{n-j}] - \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}[D_n] \tag{1.50}$$

1.2.3 Experimental Description

Materials and Instruments

Free-radical isothermal bulk polymerization reactions were conducted in a 500-mL jacketed round-bottom flask (Lab-Glass, Vineland, NJ, see Figure 1.1). All of the experiments used

water as the solvent, methyl methacrylate (MMA) as the monomer, and benzoyl peroxide (BPO) as the initiator. Attenuated total reflection (ATR) Fourier transform infrared (FTIR) spectroscopy was used to monitor *in situ* the conversion of monomer and decomposition of the initiator during the polymerization reaction. Absorbance spectra were obtained with a DIPPER-210 ATR-FTIR immersion probe with two reflections manufactured by Axiom Analytical, and analyzed using robust chemometrics [29, 30]. ZnSe was used as the internal reflectance element. The probe was attached to a Nicolet Protege 460 FTIR spectrophotometer connected to a 333 MHz Pentium II running OMNIC 4.1a software from Nicolet Instrument Corporation.

The spectrometer was purged with N_2 gas for one hour before and during measurements to reduce the effects of CO_2 absorption in its optical path. A spectral resolution of 4 cm^{-1} was used as a compromise between scan speed and resolution. Samples were prepared using methyl methacrylate (MMA) (99%, inhibited with 10–100 ppm MEHQ, Aldrich Chemical Company, Inc.) with the inhibitor removed from the MMA using inhibitor removal columns (Scientific Polymer Products, Inc.). The temperature of the samples was controlled to within ± 0.20 °C. A single flat blade with a variable-speed IKA Eurostar Labortechnik stirrer driver was used to agitate the vessel. Measurements of the MWD were obtained by gel-permeation chromatography (GPC) applied to samples collected at intermediate times during the batch. A Viscotek Triple Detector Array (TDA) Model 300, equipped with a Waters 515 high pressure liquid chromatography (HPLC) pump, was used. A solution of 0.05 M LiBr in N,N-dimethylformamide (DMF) (99.9+% HPLC grade, Sigma-Aldrich) was used for elution in the column, as well as for sample preparation for GPC analysis. For calibration purposes, a poly(methyl methacrylate) (PMMA) Calibration Kit (Scientific Polymer Products, Inc.) was used.

Experimental Procedure

The round-bottom flask was thoroughly cleaned and rinsed with deionized water and then allowed to dry. The ATR-FTIR probe, stirrer, condenser, and thermocouple were connected to the vessel. All openings to the flask (*e.g.*, top of condenser, open-neck joints) were sealed using rubber septums, creating a closed environment. The system was purged with dry nitrogen gas for several hours. Methyl methacrylate was added *via* a syringe to the vessel after removal of inhibitor. A small amount of MMA was reserved for initiator addition. The temperature of the system was set to a constant value. The vessel was agitated for several hours. This ensured that the temperature was constant throughout the flask, the contents were well mixed, and any residual air (*i.e.* oxygen) was removed. The initiator was measured out using a ceramic spatula. The reaction was initiated when the remainder of the uninhibited MMA (kept at 5 °C) and initiator were combined and added to the system via syringe.

Concentration measurements were made in 1-min intervals with the ATR-FTIR probe and recorded to a data file for processing. A syringe was used to withdraw reaction mixtures during the experiments for molecular-weight distribution measurements. The solution of MMA/PMMA was immediately added to flask of n-hexanes (0 °C). The precipitated polymer was filtered from the solution and washed alternately with hexanes and methanol. The polymer samples were prepared for offline GPC analysis.

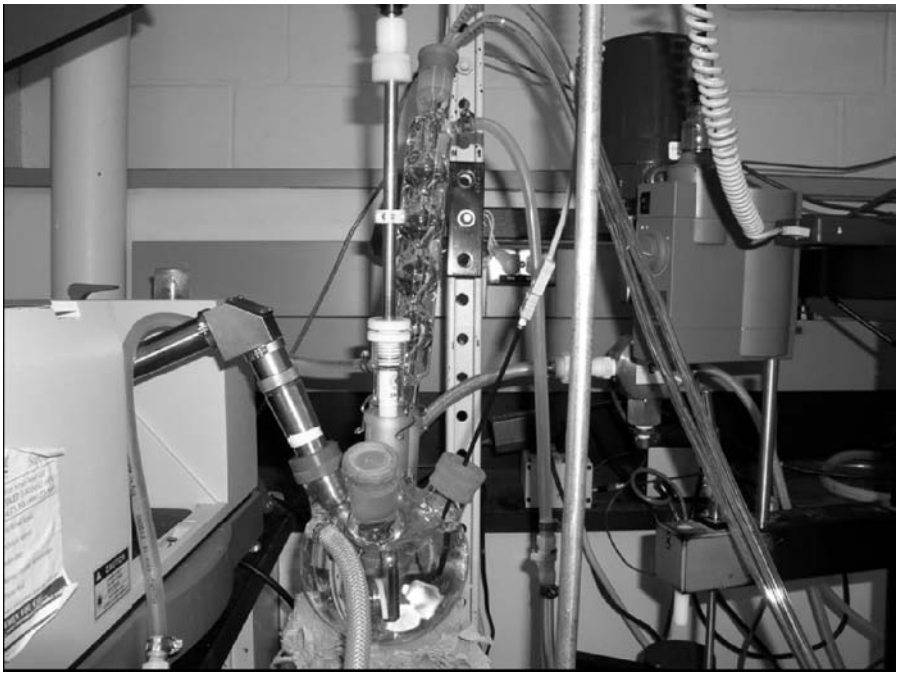


Fig. 1.1. Experimental setup for the free-radical bulk polymerization of methyl methacrylate and benzoyl peroxide. The ATR-FTIR probe is located on the left and fits directly into the reactor. GPC measurements were made offline (GPC equipment not shown in the picture).

1.2.4 Parameter-estimation Formulation

The objective function for parameter estimation is described by

$$\min_{\theta \geq 0} \Phi(\theta) \tag{1.51}$$

The parameters to be estimated are

$$\theta = \left[f, k_{trm}, k_{td}^{o}, k_{tc}^{o}, k_{p}^{o}\right]^{T} \tag{1.52}$$

A common approach in the literature is to apply parameter estimation to measurements of only the monomer conversion, which is fairly limited information and is likely one of the causes of the large confidence intervals on the parameters (see Background section). Basing parameter estimates only on monomer-conversion measurements does not consider the importance of the molecular-weight distribution, which characterizes the composition of the polymer.

In this chapter, sensitivity analysis and experimental data including the monomer conversion and MWD are incorporated into least-squares parameter estimation to determine the kinetic parameters (*e.g.*, initiator efficiency, monomer transfer, propagation, and termination) and to estimate uncertainties in their values [4]. While the parameter-estimation algorithm only utilized characteristics of the MWD, the direct method for simulating the *full* MWD was used to compare the model predictions with measurements. The specific parameter-estimation objective was

$$\Phi(\theta) = \Phi(\theta)_x + \Phi(\theta)_n + \Phi(\theta)_w \tag{1.53}$$

where

$$\Phi(\theta)_x = \sum_{i=1}^{N} \sum_{j=1}^{M_i^x} w_{ij}^x \left(\frac{x_{ij} - \tilde{x}_{ij}}{x_{ij}} \right)^2 \tag{1.54}$$

$$\Phi(\theta)_n = \sum_{i=1}^{N} \sum_{j=1}^{M_i^n} w_{ij}^n \left(\frac{\bar{M}_{n,ij} - \tilde{\bar{M}}_{n,ij}}{\bar{M}_{n,ij}} \right)^2 \tag{1.55}$$

$$\Phi(\theta)_w = \sum_{i=1}^{N} \sum_{j=1}^{M_i^w} w_{ij}^w \left(\frac{\bar{M}_{w,ij} - \tilde{\bar{M}}_{w,ij}}{\bar{M}_{w,ij}} \right)^2 \tag{1.56}$$

x_{ij}, $\bar{M}_{n,ij}$, and $\bar{M}_{w,ij}$ are the measurement and $\tilde{x}_{ij}$, $\tilde{\bar{M}}_{n,ij}$, and $\tilde{\bar{M}}_{w,ij}$ are the model predictions of the jth measurement of the ith experiment, w_{ij} is a weighting factor, N is the number of experiments, and M_i is the number of measurements in the ith experiment. The weighting factors were set based on the relative accuracy of the measurements [4].

The parameter estimates are stochastic variables because of noise associated with measurements. An approximate confidence region for the parameters can be obtained by linearizing the model near the vicinity of the estimate [4]:

$$\tilde{\mathbf{y}}_i(\theta) \approx \tilde{\mathbf{y}}_i(\theta^*) + \mathbf{F}_i(\theta^*)(\theta - \theta^*) \tag{1.57}$$

where $\tilde{\mathbf{y}}_i = [\tilde{y}_{i,1}, \ldots, \tilde{y}_{i,M_i}]^T$ is the vector of model predictions for the ith experiment, θ^* is the maximum likelihood estimate for the parameter vector, and $\mathbf{F}_i$ is the $M_i \times N_p$ matrix, where N_p is the number of parameters, given by

$$\mathbf{F}_i = \left. \frac{\partial \tilde{\mathbf{y}}_i}{\partial \theta} \right|_{\theta^*} \tag{1.58}$$

which was calculated using centered finite differences. If it is assumed that the measurement errors are normally distributed and independent, then the parameter covariance matrix $\mathbf{V}_\theta$ for the linearized problem is given by

$$\mathbf{V}_\theta{}^{-1} = \sum_{i=1}^{N} \mathbf{F}_i^T . \mathbf{V}_i^{-1} \mathbf{F}_i \tag{1.59}$$

The approximate $100(1-\alpha)\%$ confidence region is the hyperellipsoid defined by

$$(\theta - \theta^*)^T \mathbf{V}^{-1} (\theta - \theta^*) \leq \chi^2_{N_p}(\alpha) \tag{1.60}$$

where χ^2 is the chi-squared distribution that is available in statistics textbooks. For this study, $\alpha = 0.05$ was used.

The eigenvectors of $\mathbf{V}_\theta{}^{-1}$ give the direction and the eigenvalues give the length of the axes of the hyperellipsoid. It is not possible to visualize the hyperellipsoid for higher than three dimensions. Therefore, approximate confidence intervals are reported

$$\theta_j^* - \sqrt{\chi^2_{N_p}(\alpha) V_{\theta,jj}} \leq \theta_j \leq \theta_j^* + \sqrt{\chi^2_{N_p}(\alpha) V_{\theta,jj}} \tag{1.61}$$

where $V_{\theta,jj}$ is the (j,j) element of $\mathbf{V}_\theta$. It should be noted that these confidence intervals on each model parameter are not as reliable for quantifying the accuracy of the model parameters as the original confidence hyperellipsoid (Equation (1.60)). Also, note that the use of the $\chi^2_{N_p}$

distribution in Equation (1.61) results in larger confidence intervals than using the student t-distribution [4].

The linear correlation matrix $\mathbf{R}$ measures the degree of linear dependence between a pair of parameters. A magnitude of 1 represents perfect correlation and a magnitude of 0 represents no correlation. The coefficients of the correlation matrix [4] is given by

$$R_{ij} = V_{ij} \left(V_{ii} V_{jj}\right)^{-1/2} \tag{1.62}$$

The diagonal entries are 1. When all offdiagonal entries have a magnitude of 0.9 and larger, the parameters are highly correlated and considered inaccurate [4].

The sequential quadratic program (SQP), FFSQP [33], was used to solve the optimization. All simulations were run on a 1 GHz Pentium III personal computer.

1.2.5 Optimal Control Formulation

The optimal control problem is to compute the temperature profile that produces the desired number-average molecular-weight, $\bar{M}_n$, and weight-average molecular-weight, $\bar{M}_w$. As is standard in control-vector parameterization, the optimal temperature profile is discretized into N intervals during the batch time. The temperature rate is constant between each time interval, so that the overall temperature profile is a first-order spline. The initial initiator concentration and the temperature at each discrete time $k = 0, \ldots, N$ are the optimization variables:

$$\begin{aligned}
&\min_{T(t),[I_{2o}]} \left(1 - \frac{\bar{M}_n}{\bar{M}_n^*}\right)^2 + \left(1 - \frac{\bar{M}_w}{\bar{M}_w^*}\right)^2 \\
&\text{subject to} \\
&\qquad g_1(t) = T_{min}(t) - T(t) \leq 0 \\
&\qquad g_2(t) = T(t) - T_{max}(t) \leq 0 \\
&\qquad g_3(t) = R_{min}(t) - \frac{\mathrm{d}T(t)}{\mathrm{d}t} \leq 0 \\
&\qquad g_4(t) = \frac{\mathrm{d}T(t)}{\mathrm{d}t} - R_{max}(t) \leq 0 \\
&\qquad g_5 = [I_{2o}]_{min} - [I_{2o}] \leq 0 \\
&\qquad g_6 = [I_{2o}] - [I_{2o}]_{max} \leq 0
\end{aligned} \tag{1.63}$$

where T_{min}, T_{max}, R_{min}, R_{max}, $[I_{2o}]_{min}$, and $[I_{2o}]_{max}$ are the minimum and maximum temperatures, temperature ramp rates, and initial initiator concentrations, respectively, during the batch process. The temperature constraints ensure that the temperature profile stays within the operating range of the polymerization reactor while the initiator constraints ensure that a reasonable initiator concentration is used. The values of the temperature and initiator constraints are reported in Table 1.2. A fixed batch time of 150 minutes was used, although it is easy to extend the approach to variable batch time by introducing another variable. The nonlinear optimization (Equation (1.63)) was solved using sequential quadratic programming (FFSQP) [33].

1.3 Simulation and Experimental Results

1.3.1 Model-simulation Verification

The MWD evolution was simulated for a well-mixed reactor initially containing methyl methacrylate (MMA) and benzoyl peroxide (BPO) while following the four temperature pro-

Table 1.2. Constraint values for optimal temperature and initiator concentration trajectories, (Equation (1.63))

Variable	Value	Units
$T_{min}(t)$	323.15	[K]
$T_{max}(t)$	363.15	[K]
$R_{min}(t)$	-2.00	[K min^{-1}]
$R_{max}(t)$	2.00	[K min^{-1}]
$[I_{2o}]_{min}$	0.40	[g]
$[I_{2o}]_{min}$	20.00	[g]

files reported in Table 1.3. The kinetic parameters and physical constants used for this system are reported in Table 1.4. In each simulation, ∼95,000 ODEs were distributed across 20 processors. Two Linux clusters were used for all simulations. The Turing Cluster consisted of 208 dual-processor machines with two 1 GHz Pentium III processors and 1 GB of RAM each [2]. The National Center for Supercomputing Applications (NCSA) IA-32 Linux Cluster consisted of 484 dual-processor machines with two 1-GHz Pentium III processors [3]. The simulation times for each MWD simulation are reported in Table 1.5.

Table 1.3. Numerical conditions used in parallel CVODE simulations

Case	Initiator concentration (g BPO/g MMA)	Temperature profile (°C)
A	4.00	$T(t) = 50$
B	4.00	$T(t) = 4t/15 + 50$
C	4.00	$T(t) = 90$
D	4.00	$T(t) = 90 - 4t/15$

The moment equations were used to verify the full MWD simulated using parallel CVODE. Figures 1.2–1.5 show the MWD evolution, the average molecular-weights $\bar{M}_n$ and $\bar{M}_w$, and the monomer conversion, x. The average molecular-weights and monomer conversion from parallel CVODE and the moment equations are the same, confirming the validity of the QSSA used to derive the set of polymerization reactions used by parallel CVODE. Nearly complete conversion of monomer occurs at the highest operating temperature (Case C) where a non-Gaussian molecular-weight distribution is produced. The total polymer concentration increases and the average molecular-weights decrease with increasing temperature.

Figures 1.6–1.9 report the average molecular-weights for the live and dead polymer chains. As expected, at initial times the dead polymer concentrations are lower than the live polymer concentrations, with the average molecular-weights of the dead polymer chains being larger than for the live polymer chains. The average molecular-weights of the live polymer

Table 1.4. Kinetic parameters and physical constants used in simulations

Parameter	Value	Units	Reference
f	1.0	—	
k_d	$1.7 \times 10^{14} \exp\left(-3.0 \times 10^4/RT\right)$	s^{-1}	[20]
k_{trm}	0.0	[L mol^{-1} s^{-1}]	[22]
k_{td}^o	$9.8 \times 10^7 \exp\left(-701/RT\right)$	[L mol^{-1} s^{-1}]	[22]
k_{tc}^o	0.0	[L mol^{-1} s^{-1}]	[22]
k_p^o	$4.92 \times 10^5 \exp\left(-4353/RT\right)$	[L mol^{-1} s^{-1}]	[20]
R	1.987	[cal mol^{-1} K^{-1}]	—
$\rho_m(T)$	$968.0 - 1.15(T - 273.15)$	[g L^{-1}]	[27]
$\rho_p(T)$	$1.212 \times 10^3 - 0.845(T - 273.15)$	[g L^{-1}]	[27]
M_w	100.12	[g mol^{-1}]	—

Table 1.5. Simulation times using parallel CVODE

Case	Time (s)
A	484
B	913
C	480
D	513

decrease over time as the dead polymer concentrations increase. In Case C, almost 100% monomer conversion is achieved. As a result, the termination kinetic parameters decrease as a result of the gel effect and the total live polymer concentration increases. At this point the QSSA is no longer valid, but using this method results in no more than 5% error at the final time. Case D in which the temperature is decreased during the batch run is the most interesting, in that the average molecular-weights of the live polymer decreases and then increases to values larger than the molecular-weights of dead polymer. If the batch run was longer, this would eventually lead to the formation of some longer dead polymer chains (see Figure 1.5).

The above MWDs were simulated with no *prior knowledge of the shape of the MWD*. This is in contrast to alternative simulation methods that are accurate for only a range of MWDs. To more fully demonstrate the advantage of using the simulation approach, a bimodal molecular-weight distribution was simulated for the semibatch free-radical polymerization of methyl methacrylate with 2,2'-azobisisobutyronitrile (AIBN) as an initiator and dodecyl mercaptan (DDM) as a chain-transfer agent. The chain-transfer agent was added at $t = 120$ min. The reaction mechanism is identical with the addition of a chain-transfer mechanism:

$$P_n + C \xrightarrow{k_{trc}} D_n + P_1,\ r_{trc} = k_{trc}\,[C]\,[P_n],\ \text{polymer chain transfer}$$

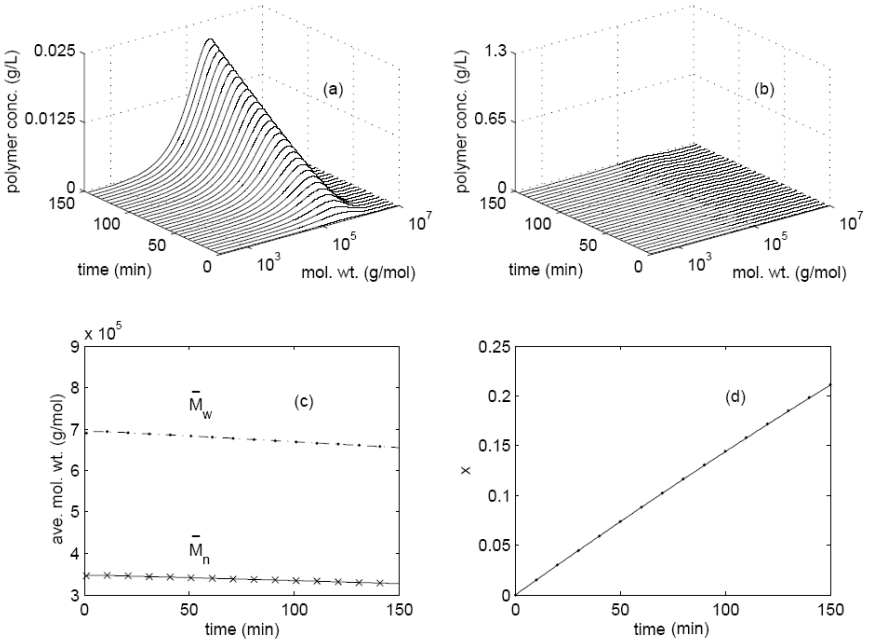


Fig. 1.2. Simulation results for Case A: (a,b) evolution of the MWD for dead polymer chains with (b) using a consistent scaling for all cases to enable different cases to be compared; (c) average molecular-weights for the sum of dead and live polymer chains as computed by the method of moments, $(\times)$ and $(\cdots)$, and parallel CVODE, (—) and $(- \cdot -)$, respectively; (d) monomer conversion computed from the method of moments $(\cdots)$ and parallel CVODE (—).

where $[C]$ is the chain-transfer agent concentration and k_{trc} is the rate constant for polymer chain transfer. The molar balances on the semibatch approach, including chain-transfer agent, is defined for initiator decomposition

$$\frac{1}{V}\frac{\mathrm{d}\left([I_2]V\right)}{\mathrm{d}t} = -k_d[I_2] \tag{1.64}$$

free-radical concentration

$$\frac{1}{V}\frac{\mathrm{d}\left([I\cdot]V\right)}{\mathrm{d}t} = -k_i[I\cdot][M] + 2fk_d[I_2] \tag{1.65}$$

monomer concentration

$$\frac{1}{V}\frac{\mathrm{d}\left([M]V\right)}{\mathrm{d}t} = -k_i[I\cdot][M] - \left(k_p + k_{trm}\right)[M]\sum_{j=1}^{\infty}[P_j] \tag{1.66}$$

chain-transfer agent

$$\frac{1}{V}\frac{\mathrm{d}\left([C]V\right)}{\mathrm{d}t} = \begin{cases} 0, & \text{for } t < 120 \\ -k_{trc}[C]\sum_{j=1}^{\infty}[P_j], & \text{for } t \geq 120 \end{cases} \tag{1.67}$$

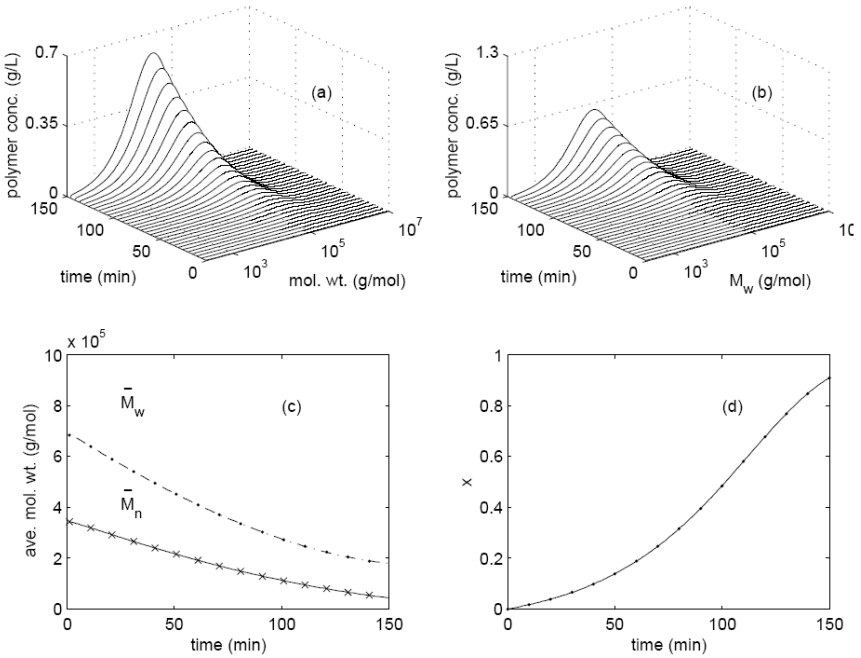


Fig. 1.3. Simulation results for Case B: (a,b) evolution of the MWD for dead polymer chains with (b) using a consistent scaling for all cases to enable different cases to be compared; (c) average molecular-weights for the sum of dead and live polymer chains as computed by the method of moments, $(\times)$ and $(\cdots)$, and parallel CVODE, (—) and $(- \cdot -)$, respectively; (d) monomer conversion computed from the method of moments $(\cdots)$ and parallel CVODE (—).

live polymer chains

$$\frac{1}{V}\frac{\mathrm{d}\left([P_1]V\right)}{\mathrm{d}t} = k_i[I\cdot][M] - k_p[M][P_1] - k_{trm}[M]\left([P_1] - \sum_{j=1}^{\infty}[P_j]\right) \\ - \left(k_{tc} + k_{td}\right)[P_1]\sum_{j=1}^{\infty}[P_j] - k_{trc}[C]\left([P_1] - \sum_{j=1}^{\infty}[P_j]\right) \tag{1.68}$$

$$\frac{1}{V}\frac{\mathrm{d}\left([P_m]V\right)}{\mathrm{d}t} = k_p[M]\left([P_{m-1}] - [P_m]\right) - k_{trm}[M][P_m] \\ - \left(k_{tc} + k_{td}\right)[P_m]\sum_{j=1}^{\infty}[P_j] - k_{trc}[C][P_m] \text{ for } m = 2, \cdots, n \tag{1.69}$$

and dead polymer chains

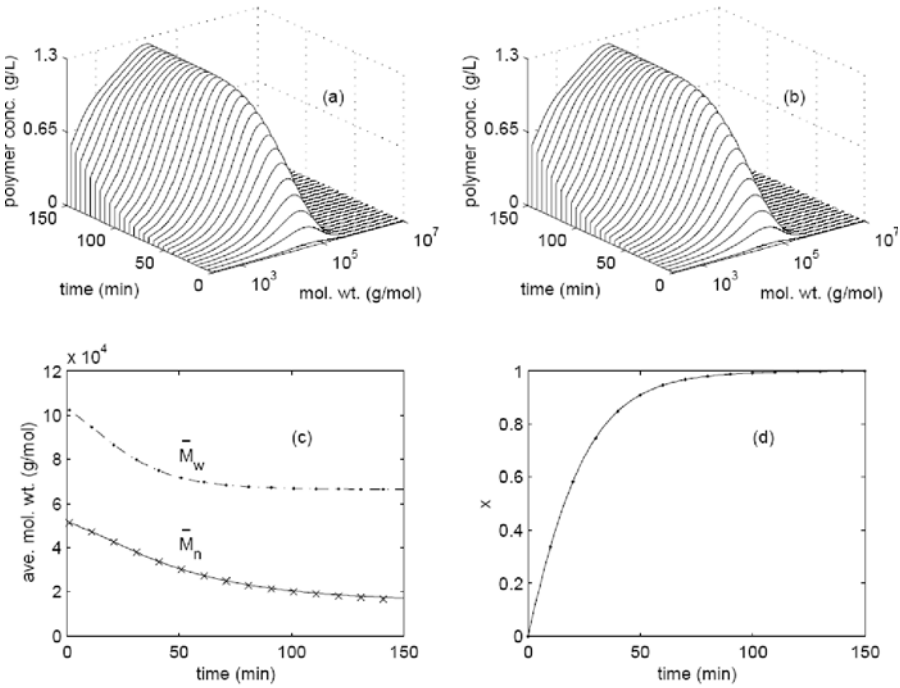


Fig. 1.4. Simulation results for Case C: (a,b) evolution of the MWD for dead polymer chains with (b) using a consistent scaling for all cases to enable different cases to be compared; (c) average molecular-weights for the sum of dead and live polymer chains as computed by the method of moments, $(\times)$ and $(\cdots)$, and parallel CVODE, (—) and $(- \cdot -)$, respectively; (d) monomer conversion computed from the method of moments $(\cdots)$ and parallel CVODE (—).

$$\frac{1}{V}\frac{\mathrm{d}\left([D_n]V\right)}{\mathrm{d}t} = k_{trm}[M][P_n] + k_{td}[P_n]\sum_{j=1}^{\infty}[P_j] + \frac{1}{2}k_{tc}\sum_{j=1}^{n-1}[P_j][P_{n-j}] \qquad (1.70)$$
$$+ k_{trc}[C][P_n] \text{ for } m = 2, \cdots, n$$

Applying the QSSA in a similar manner as earlier gives the ordinary differential equations:

$$\frac{\mathrm{d}[I_2]}{\mathrm{d}t} = -\left(k_d + \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}\right)[I_2] \qquad (1.71)$$

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{2fk_d[I_2]\,(1+\varepsilon x)}{[M_o]} + (k_p + k_{trm})\,(1-x)\,[P] \qquad (1.72)$$

$$\frac{\mathrm{d}[C]}{\mathrm{d}t} = \begin{cases} 0, & \text{for } t < 120 \\ -k_{trc}[C]\displaystyle\sum_{j=1}^{\infty}[P_j] - \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}[C], & \text{for } t \geq 120 \end{cases} \qquad (1.73)$$

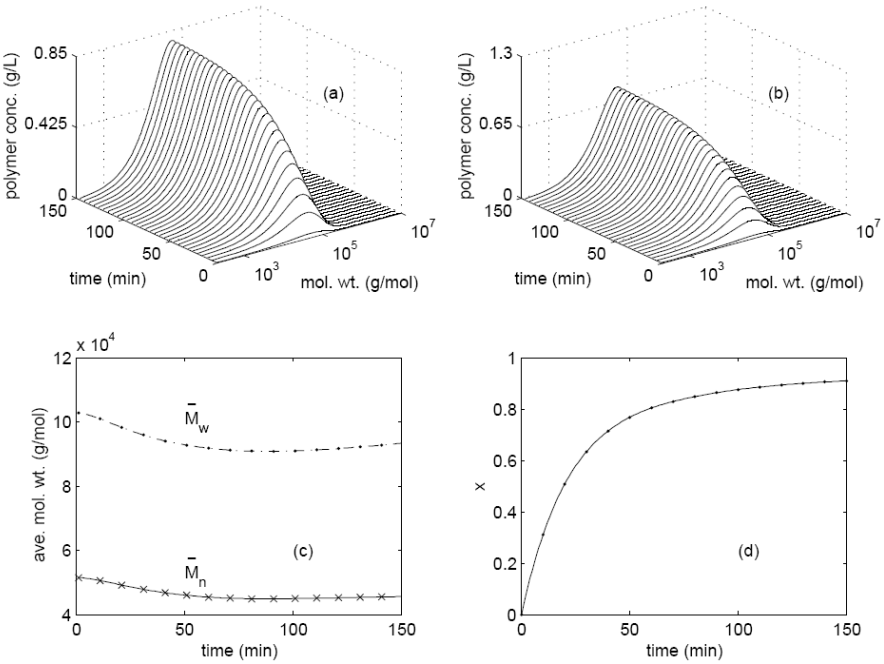


Fig. 1.5. Simulation results for Case D: (a,b) evolution of the MWD for dead polymer chains with (b) using a consistent scaling for all cases to enable different cases to be compared; (c) average molecular-weights for the sum of dead and live polymer chains as computed by the method of moments, $(\times)$ and $(\cdots)$, and parallel CVODE, (—) and $(- \cdot -)$, respectively; (d) monomer conversion computed from the method of moments $(\cdots)$ and parallel CVODE (—).

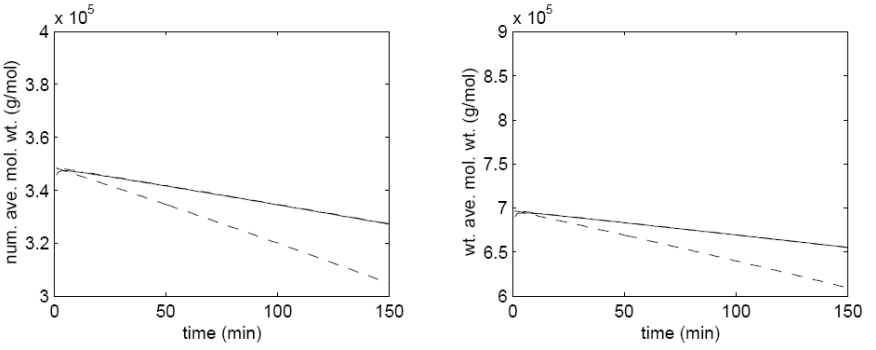


Fig. 1.6. Average molecular-weights, $\bar{M}_n$ and $\bar{M}_w$, for Case A: for live polymer chains $(--)$, dead polymer chains $(- \cdot -)$, and combined live and dead polymer chains (—)

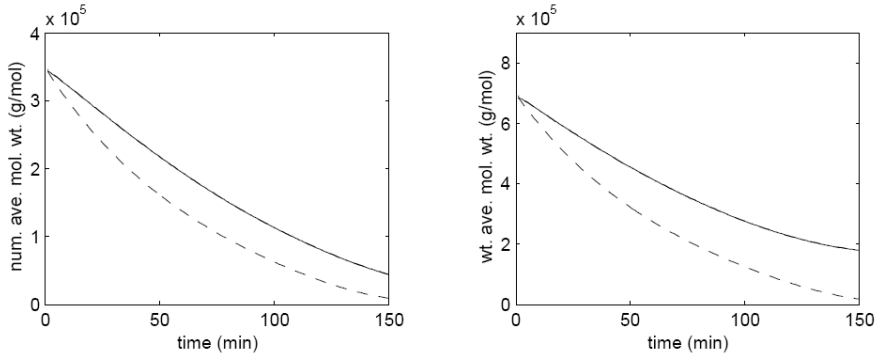


Fig. 1.7. Average molecular-weights, $\bar{M}_n$ and $\bar{M}_w$, for Case B: for live polymer chains $(--)$, dead polymer chains $(- \cdot -)$, and combined live and dead polymer chains (—)

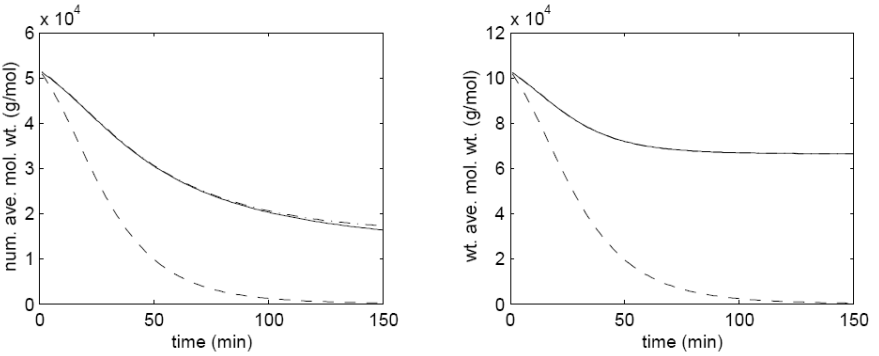


Fig. 1.8. Average molecular-weights, $\bar{M}_n$ and $\bar{M}_w$, for Case C: for live polymer chains $(--)$, dead polymer chains $(- \cdot -)$, and combined live and dead polymer chains (—)

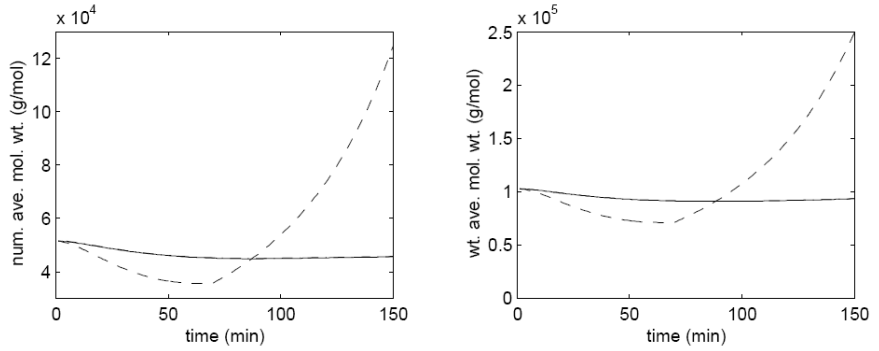


Fig. 1.9. Average molecular-weights, $\bar{M}_n$ and $\bar{M}_w$, for Case D: for live polymer chains $(--)$, dead polymer chains $(- \cdot -)$, and combined live and dead polymer chains (—)

$$\frac{\mathrm{d}[P]}{\mathrm{d}t} = 2fk_d[I_2] - (k_{td} + k_{tc})\,\lambda_0^2 - \frac{\varepsilon\lambda_0}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t} \tag{1.74}$$

$$[P_1] = \frac{2fk_d[I_2] + k_{trm}[M][P] + k_{trc}[C][P]}{(k_p + k_{trm})\,[M] + k_{trc}[C] + (k_{td} + k_{tc})\,[P] + \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}} \tag{1.75}$$

$$[P_2] = \frac{k_p[M][P_1]}{(k_p + k_{trm})\,[M] + (k_{td} + k_{tc})\,[P] + k_{trc}[C] + \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}} \tag{1.76}$$

$$\vdots$$

$$[P_n] = \frac{k_p[M][P_{n-1}]}{(k_p + k_{trm})\,[M] + (k_{td} + k_{tc})\,[P] + k_{trc}[C] + \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}} \tag{1.77}$$

$$[P] = \sum_{i=1}^{n}[P_i] = \lambda_0 \tag{1.78}$$

$$\begin{aligned}\frac{\mathrm{d}[D_2]}{\mathrm{d}t} &= k_{trm}[M][P_2] + k_{td}[P_2][P] + \frac{1}{2}k_{tc}[P_1]^2 + k_{trc}[C][P_2] \\ &\quad - \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}[D_2]\end{aligned} \tag{1.79}$$

$$\vdots$$

$$\begin{aligned}\frac{\mathrm{d}[D_n]}{\mathrm{d}t} &= k_{trm}[M][P_n] + k_{td}[P_n][P] + \frac{1}{2}k_{tc}\sum_{j=1}^{n-1}[P_j][P_{n-j}] + k_{trc}[C][P_n] \\ &\quad - \frac{\varepsilon}{1+\varepsilon x}\frac{\mathrm{d}x}{\mathrm{d}t}[D_n]\end{aligned} \tag{1.80}$$

The kinetic parameters are reported in Table 1.6. Simulations were executed for an isothermal (70 °C) semibatch polymerization with 0.3 wt% AIBN and 0.8 wt% DDM. The algorithm was implemented as discussed previously in Section 1.2.2, except now with the addition of the chain-transfer agent. The time evolution of the full molecular-weight distribution is reported in Figure 1.10. The chain-transfer agent was added at $t = 120$ min. This time was chosen to produce a bimodal distribution with equal modes. The MWD at different time instants are reported in Figure 1.11 to show how the bimodal distribution evolves with time.

The ability to simulate the full molecular-weight distribution opens many new possibilities for identification and control. It enables the estimation of the kinetic parameters from the full MWD. Simulating the full MWD also enables the estimation of kinetic parameters that are length or mass dependent, in contrast to existing polymerization models that use an average value for each kinetic parameter. Considering the MWD enables the computation of the optimal setpoint trajectories and feedback controllers to produce a desired molecular-weight distribution, including MWDs with multiple modes.

1.3.2 Potential for Optimal Control of the MWD

Before moving to the experimental parameter-estimation study, this section describes a brief study to assess the potential of optimal control based on the mechanistic model. Such initial simulation studies should be completed before doing experiments for parameter estimation, to assess whether the value of the identified model is worth the expense of doing experiments.

Table 1.6. Kinetic parameters and physical constants used in parallel CVODE simulations of semibatch free-radical polymerization of MMA, AIBN, and DDM

Parameter	Value	Units	Reference
f	0.43	—	[28]
k_d	$2.8 \times 10^{15} \exp(-15685/T)$	[s^{-1}]	[28]
k_{trm}	$2.324 \times 10^{8} \exp(-9218/T)$	[L mol^{-1} s^{-1}]	[28]
k_{trc}	771 at 70 °C	[L mol^{-1} s^{-1}]	[28]
k_{td}^o	$9.8 \times 10^{7} \exp(-353/T)$	[L mol^{-1} s^{-1}]	[28]
k_{tc}^o	0.0	[L mol^{-1} s^{-1}]	[28]
k_p^o	$4.9 \times 10^{5} \exp(-2190/T)$	[L mol^{-1} s^{-1}]	[28]
R	1.987	[cal mol^{-1} K^{-1}]	—
$\rho_m(T)$	$968.0 - 1.15(T - 273.15)$	[g L^{-1}]	[28]
$\rho_p(T)$	$1.212 \times 10^{3} - 0.845(T - 273.15)$	[g L^{-1}]	[28]
M_w	100.12	[g mol^{-1}]	—

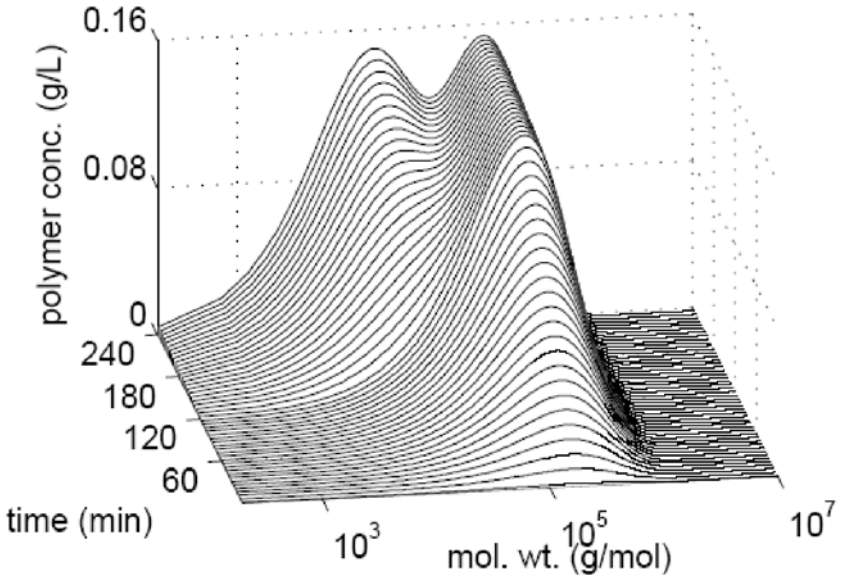


Fig. 1.10. Full molecular-weight distribution evolution for semibatch isothermal free-radical polymerization of methyl methacrylate (0.3 wt% AIBN and 0.8 wt% DDM)

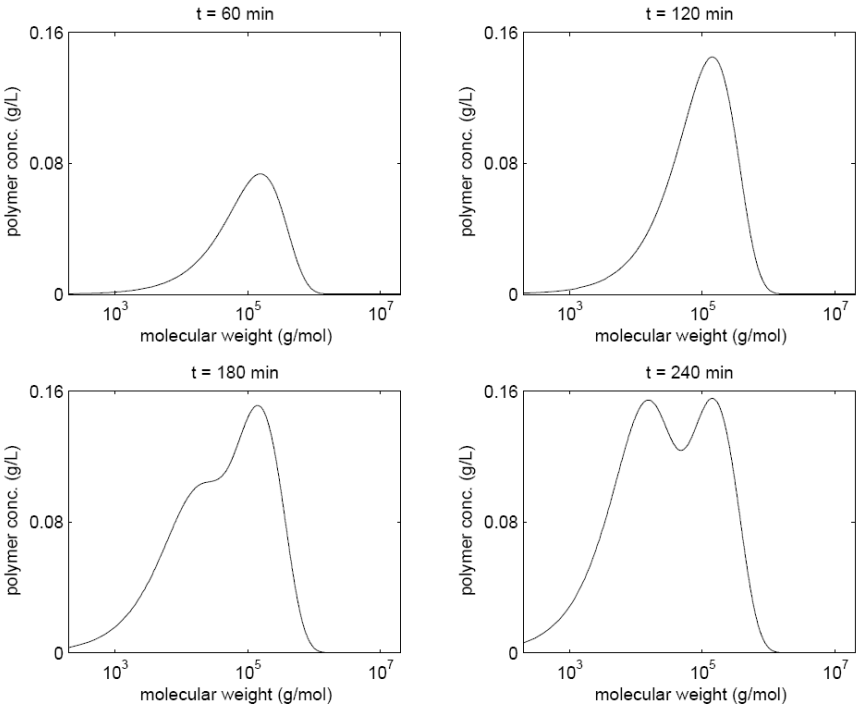


Fig. 1.11. Time instances of the full molecular-weight distribution evolution for semibatch isothermal free-radical polymerization of methyl methacrylate (0.3 wt% AIBN and 0.8 wt% DDM)

Table 1.7 reports three sets of average molecular-weight objectives investigated in this study. This is not an exhaustive list, but merely an example of optimal control trajectories for a desired set of average molecular-weights, $\bar{M}_n^*$ and $\bar{M}_w^*$.

Table 1.7. Desired average molecular-weight objectives in optimal control study

Case	$\bar{M}_n^*$ (g mol^{-1})	$\bar{M}_w^*$ (g mol^{-1})
A	100,000	200,000
B	1,000,000	2,000,000
C	40,000	80,000

The nonlinear optimization (Equation (1.63)) was solved by sequential quadratic programming to determine the optimal initial initiator concentration and temperature trajectory.

Table 1.8. Final average molecular-weight objectives for optimal control temperature trajectories and initial initiator concentration

Case	$[I_{2o}]$ (g)	T_o (K)	$\bar{M}_n$ (g mol^{-1})	$\bar{M}_w$ (g mol^{-1})	Monomer conversion x
A	9.40	346	99,997	200,005	70%
B	0.93	323	999,985	2,000,078	7.5%
C	10.95	363	39,977	80,049	92%

Table 1.8 reports these values as well as the optimal average molecular-weights. Figures 1.12–1.14 report the optimal temperature profile and resulting conversion and average molecular-weight profiles for Cases A, B, and C. Case A has a monotonically decreasing optimal cooling profile whereas the optimal temperature for Case B was nearly constant. A rapidly dropping cooling profile was optimal for Case C. In all three cases, the optimal average molecular-weights are very close to the desired values, suggesting that the experimental effort to develop an accurate mechanistic model for use in such optimal control calculations is justified.

The optimal operating conditions strongly depend on the control objective. A wide range of desired average molecular weights was considered that illustrated a wide range of temperature trajectories. Higher monomer conversion is achieved with Cases A and C. This is expected since the higher temperatures result in faster kinetics. Higher initiator concentration produces smaller molecular-weights. For large molecular-weights (*i.e.* Case B), a low temperature (almost isothermal) profile and low initiator concentration are optimal. This is consistent with the fundamental chemistry. Also worth noting is that the average molecular-weights are determined in the initial 50 min of the experiment (see Figures 1.12–1.14). The final average molecular-weight properties strongly depend on the initial part of the reaction (*e.g.*, initial temperature, initial concentration, and temperature profile).

1.3.3 Parameter Sensitivity and Estimation

The above section demonstrates the potential of free-radical polymerization models for controlling the molecular-weight distribution. To realize that potential, accurate estimates of the kinetic parameters are required. A thorough parameter-sensitivity analysis (not shown for brevity, interested readers can go to the thesis in [19]) indicated that:

- The monomer conversion is weakly sensitive to the value of the rate constant for monomer transfer, k_{trm}, indicating that it is impossible to accurately estimate k_{trm} using only monomer-conversion data. This explains the huge variation in reported values for k_{trm} in the literature (see Background section). On the other hand, the average molecular-weights are sensitive to the value of k_{trm} and so can be used to estimate its value.
- The monomer conversion and average molecular-weights are sensitive to the rate constant for termination by disproportionation, k_{td}^o, indicating that its value can be estimated.
- The monomer conversion and average molecular-weights are insensitive to the rate constant for termination by combination, k_{tc}^o, which agrees with past studies [32]. Hence, this value can be neglected ($k_{tc}^o = 0$) during parameter estimation.
- The rate constant for propagation, k_p^o, affects monomer conversion and average molecular-weights.

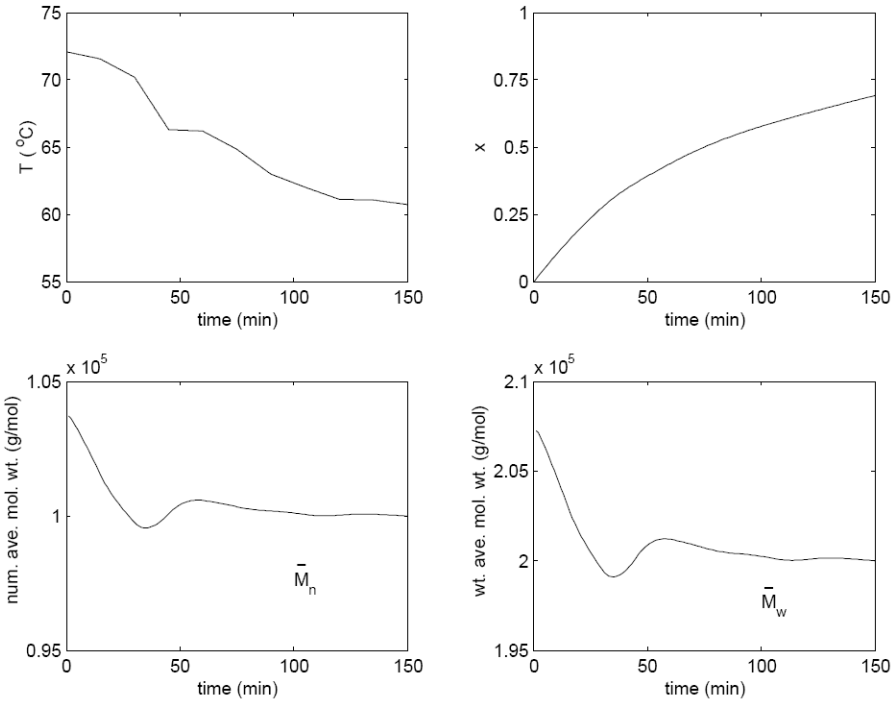


Fig. 1.12. Optimal temperature profile for Case A as determined by Equation (1.63). Simulation profiles of fractional conversion and average molecular-weight properties employing the optimal temperature profile. The initial initiator concentration and resulting average molecular-weight properties are reported in Table 1.8.

Hence, the four parameters f, k_{trm}, k_{td}^o, and k_p^o are the significant kinetic parameters to estimate in the free-radical polymerization of methyl methacrylate.

A series of free-radical bulk polymerization experiments were conducted in which the monomer concentration and molecular-weight distribution were measured, as described in the experimental description section, with the experimental conditions reported in Table 1.9. Parameter estimates and 95% confidence regions obtained from measured values for the mononer concentration and average molecular-weights are reported in Table 1.10 based on two sets of assumptions:

- Case 1 – monomer transfer does not significantly affect the conversion and molecular-weight distribution ($k_{trm} = 0$)
- Case 2 – monomer transfer is significant ($k_{trm} \neq 0$).

Plots of the monomer conversion and average molecular-weights from experiments and the nominal model for Case 1 are reported in Figures 1.15–1.18. The covariance matrix

$$\mathbf{V}_\theta = \begin{bmatrix} f & k_{td}^o & k_p^o \\ 2.00 \times 10^{-5} & -5.21 \times 10^3 & -2.49 \\ -5.21 \times 10^3 & 1.26 \times 10^{16} & 8.10 \times 10^{10} \\ -2.49 & 8.10 \times 10^{10} & 8.27 \times 10^5 \end{bmatrix} \tag{1.81}$$

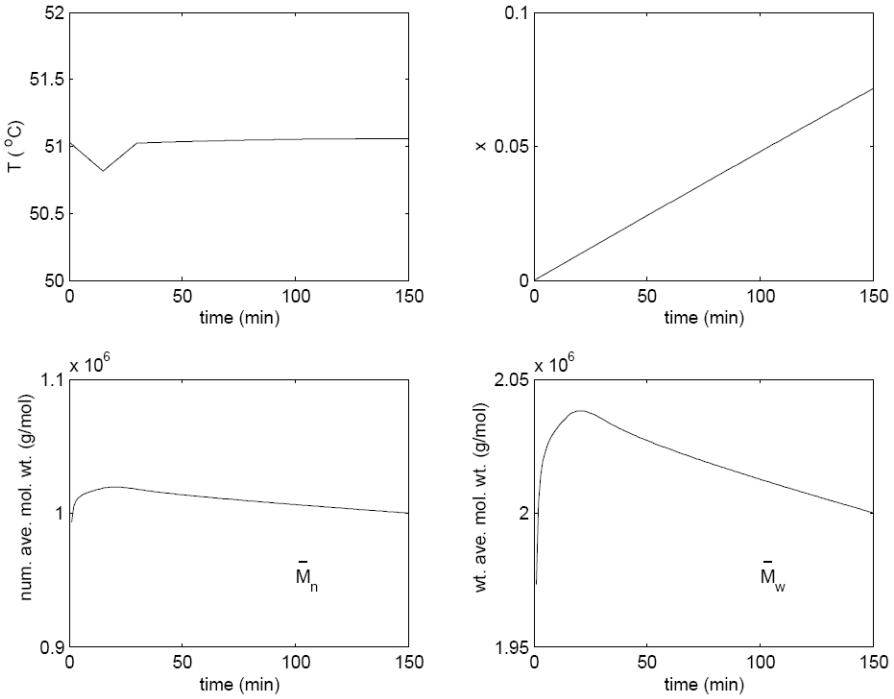


Fig. 1.13. Optimal temperature profile for Case B as by Equation (1.63). Simulation profiles of fractional conversion and average molecular-weight properties employing the optimal temperature profile. The initial initiator concentration and resulting average molecular-weight properties are reported in Table 1.8.

Table 1.9. Experimental conditions for free-radical polymerization of methyl methacrylate and benzoyl peroxide at 50 °C

Experiment	MMA (g)	BPO (g)	Initiator Concentration (wt%)
1	406.5	1.22	0.3
2	400.0	4.00	1.0
3	400.0	4.00	1.0
4	400.6	4.01	1.0

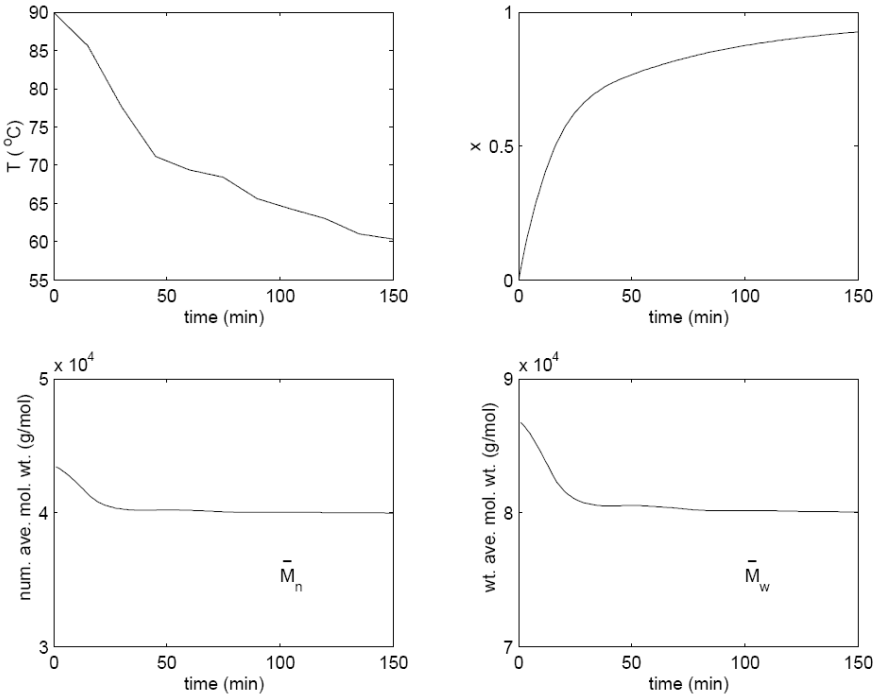


Fig. 1.14. Optimal temperature profile for Case C as determined by Equation (1.63). Simulation profiles of fractional conversion and average molecular-weight properties employing the optimal temperature profile. The initial initiator concentration and resulting average molecular-weight properties are reported in Table 1.8.

Table 1.10. Nominal parameters and confidence intervals obtained by least-squares parameter estimation applied to experimental data. Two cases are considered: Case 1 without monomer transfer ($k_{trm} = 0.0$), and Case 2 with monomer transfer ($k_{trm} \neq 0.0$). Confidence intervals are determined by Equation (1.61).

Parameters	Case 1	Case 2	Units
f	0.11	0.066	—
	$\pm$0.01	$\pm$0.007	
k_{trm}	0	1.2 $\pm$0.3	[L mol^{-1} min^{-1}]
$k_{td}^o \times 10^{-9}$	2.0	2.0	[L mol^{-1} min^{-1}]
	$\pm$0.3	$\pm$0.2	
k_{tc}^o	0	0	[L mol^{-1} min^{-1}]
$k_p^o \times 10^{-4}$	2.6 $\pm$0.2	3.4 $\pm$0.03	[L mol^{-1} min^{-1}]

and the correlation matrix

$$\mathbf{R} = \begin{bmatrix} 1.00 & -0.01 & -0.61 \\ -0.01 & 1.00 & 0.79 \\ -0.61 & 0.79 & 1.00 \end{bmatrix} \tag{1.82}$$

indicate some correlation between the estimated rate constant for propagation and the other two kinetic parameters.

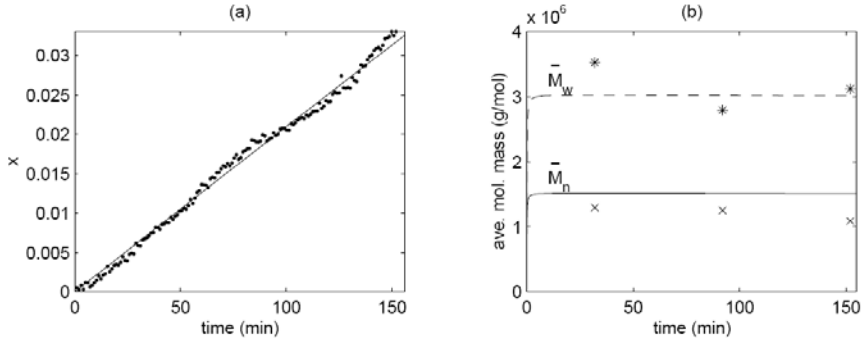


Fig. 1.15. (a) Experimental $(\cdots)$ and model prediction (—) conversion results for Case 2, Experiment 1. (b) Experimental $(\times)$, $(*)$ and model prediction (—), $(--)$ results for number- and weight-average molecular weight, respectively, for Experiment 1.

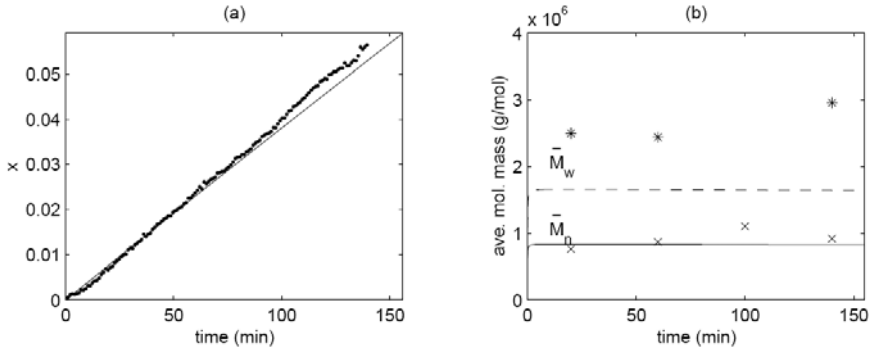


Fig. 1.16. (a) Experimental $(\cdots)$ and model prediction (—) conversion results for Case 2, Experiment 2. (b) Experimental $(\times)$, $(*)$ and model prediction (—), $(--)$ results for number- and weight-average molecular weight, respectively, for Experiment 2.

Figures 1.19–1.22 report the MWDs from experiments and simulated using the least-squares parameter estimates for Case 1. The largest polymer chain length measured by using

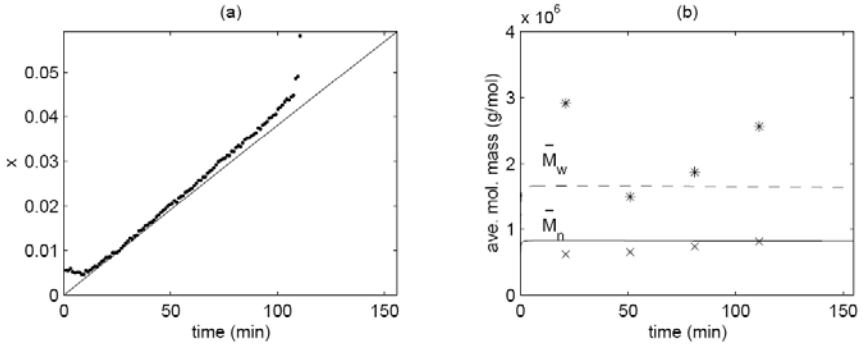


Fig. 1.17. (a) Experimental $(\cdots)$ and model prediction (—) conversion results for Case 2, Experiment 3. (b) Experimental $(\times)$, $(*)$ and model prediction (—), $(--)$ results for number- and weight-average molecular weight, respectively, for Experiment 3.

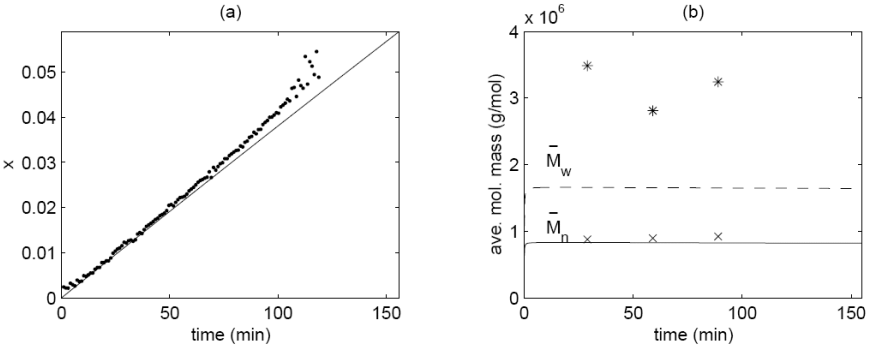


Fig. 1.18. (a) Experimental $(\cdots)$ and model prediction (—) conversion results for Case 2, Experiment 4. (b) Experimental $(\times)$, $(*)$ and model prediction (—), $(--)$ results for number- and weight-average molecular weight, respectively, for Experiment 4.

gel-permeation chromatography was $\sim 2 \times 10^7$ g mol^{-1}. The number of simulated polymer chain equations was 200,000 ($M_w \approx 20{,}000{,}000$ g mol^{-1}), which were distributed among 41 processors (1 master thread, 40 slave threads). The simulation times for each experiment are reported in Table 1.11. Table 1.12 reports the experimental and simulated polydispersity and average molecular-weights.

Although monomer transfer (Case 2) does not significantly affect conversion, it does affect the average molecular-weights, $\bar{M}_n$ and $\bar{M}_w$. The parameter estimates are similar to that of Case 1, with the exception of k_{trm} (*i.e.* a nonzero value). Case 2 has smaller confidence intervals for f, k_{td}^o, and k_p^o. Figures 1.23–1.26 compare the experimental and theoretical results obtained using the objective function (1.53). Parameter estimates obtained for Case 2 provide the best model fit to the experimental data, although the weight-average molecular weight, $\bar{M}_w$, is not as good a fit as conversion and number-average molecular weight. The covariance matrix is

Table 1.11. Simulation times reported for free-radical bulk polymerization of methyl methacrylate using parallel CVODE, Case 1

Experiment	Time (s)
1	521
2, 3, 4	327

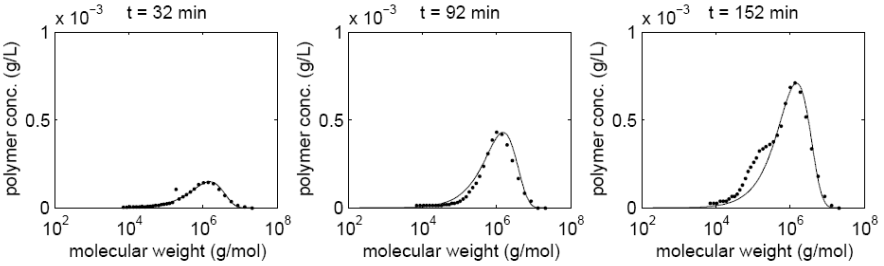


Fig. 1.19. Comparison of experimental $(\cdots)$ and simulated (—) MWD from Case 1, Experiment 1, for several time instants

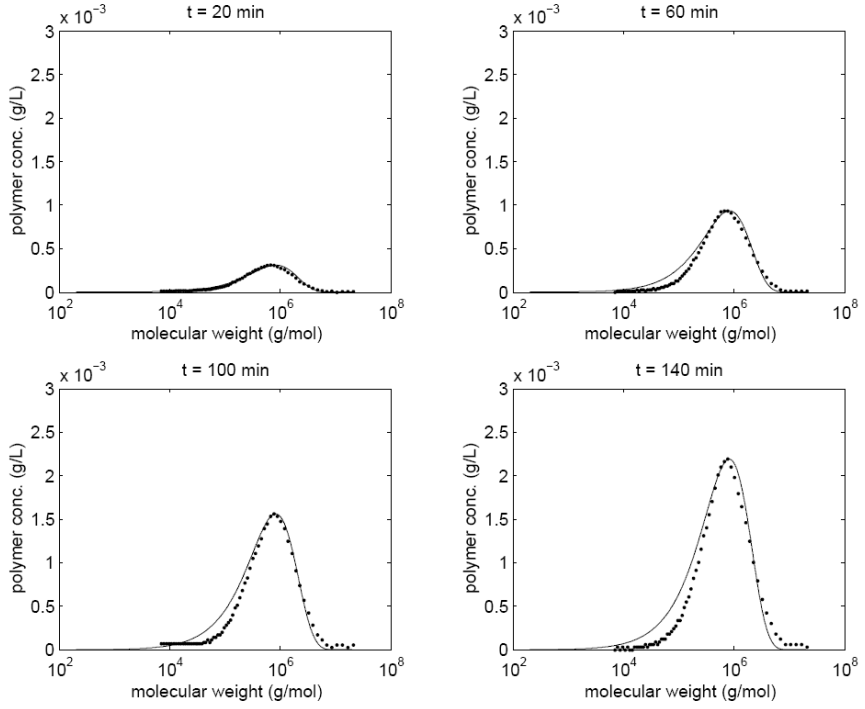


Fig. 1.20. Comparison of experimental $(\cdots)$ and simulated (—) MWD from Case 1, Experiment 2, for several time instants

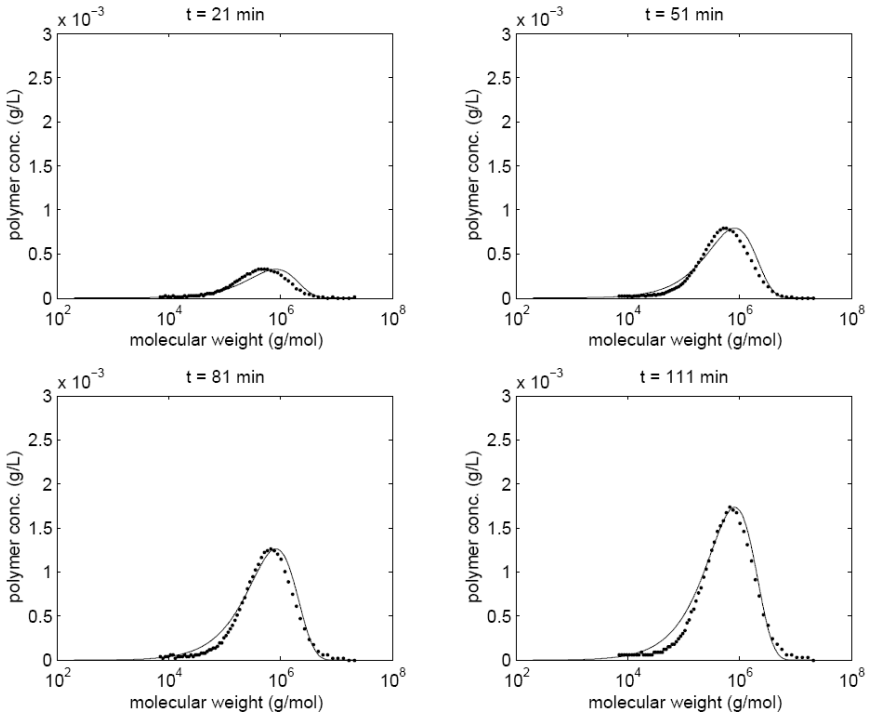


Fig. 1.21. Comparison of experimental $(\cdots)$ and simulated (—) MWD from Case 1, Experiment 3, for several time instants

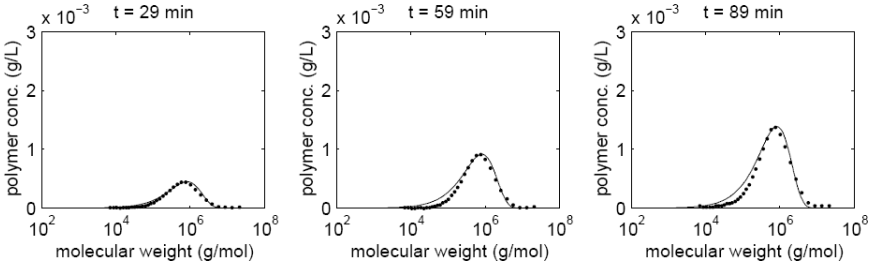


Fig. 1.22. Comparison of experimental $(\cdots)$ and simulated (—) MWD from Case 1, Experiment 4, for several time instants

Table 1.12. Experimental and simulation results (Case 1) for number-average molecular-weight, weight-average molecular-weight, and polydispersity

Experiment 1 time (min)	$\bar{M}_n^{exp}$ (g mol^{-1})	$\bar{M}_w^{exp}$ (g mol^{-1})	PD_{exp}	$\bar{M}_n^{sim}$ (g mol^{-1})	$\bar{M}_w^{sim}$ (g mol^{-1})	PD_{sim}
32	1.3×10^6	3.5×10^6	2.7	3.1×10^6	4.8×10^6	1.5
92	1.3×10^6	2.8×10^6	2.2	3.0×10^6	4.6×10^6	1.5
152	1.1×10^6	3.1×10^6	2.9	3.0×10^6	4.6×10^6	1.5

Experiment 2 time (min)	$\bar{M}_n^{exp}$ (g mol^{-1})	$\bar{M}_w^{exp}$ (g mol^{-1})	PD_{exp}	$\bar{M}_n^{sim}$ (g mol^{-1})	$\bar{M}_w^{sim}$ (g mol^{-1})	PD_{sim}
20	7.6×10^5	2.5×10^6	3.3	1.7×10^6	2.7×10^6	1.6
60	8.6×10^5	2.4×10^6	2.8	1.7×10^6	2.5×10^6	1.5
100	1.1×10^6	4.2×10^6	3.8	1.7×10^6	2.5×10^6	1.5
140	9.2×10^5	3.0×10^6	3.2	1.6×10^6	2.5×10^6	1.5

Experiment 3 time (min)	$\bar{M}_n^{exp}$ (g mol^{-1})	$\bar{M}_w^{exp}$ (g mol^{-1})	PD_{exp}	$\bar{M}_n^{sim}$ (g mol^{-1})	$\bar{M}_w^{sim}$ (g mol^{-1})	PD_{sim}
21	6.2×10^5	2.9×10^6	4.7	1.7×10^6	2.7×10^6	1.6
51	6.6×10^5	1.5×10^6	2.3	1.7×10^6	2.6×10^6	1.5
81	7.4×10^5	1.9×10^6	2.5	1.7×10^6	2.5×10^6	1.5
111	8.1×10^5	2.6×10^6	3.1	1.7×10^6	2.5×10^6	1.5

Experiment 4 time (min)	$\bar{M}_n^{exp}$ (g mol^{-1})	$\bar{M}_w^{exp}$ (g mol^{-1})	PD_{exp}	$\bar{M}_n^{sim}$ (g mol^{-1})	$\bar{M}_w^{sim}$ (g mol^{-1})	PD_{sim}
29	8.8×10^5	3.5×10^6	4.0	1.7×10^6	2.6×10^6	1.6
59	8.9×10^5	2.8×10^6	3.1	1.7×10^6	2.5×10^6	1.5
89	9.2×10^5	3.2×10^6	3.5	1.7×10^6	2.5×10^6	1.5

$$\mathbf{V}_\theta = \begin{bmatrix} f & k_{trm} & k_{td}^o & k_p^o \\ 4.83\times10^{-6} & -1.96\times10^{-4} & 1.34\times10^5 & -7.09\times10^{-2} \\ -1.96\times10^{-4} & 1.12\times10^{-2} & -5.50\times10^6 & 2.38 \\ 1.34\times10^5 & -5.50\times10^6 & 3.78\times10^{15} & -1.71\times10^9 \\ -7.09\times10^{-2} & 2.38 & -1.71\times10^9 & 1.10\times10^4 \end{bmatrix} \tag{1.83}$$

and the correlation matrix is

$$\mathbf{R} = \begin{bmatrix} 1.00 & -0.84 & 0.99 & -0.31 \\ -0.84 & 1.00 & -0.84 & 0.21 \\ 0.99 & -0.84 & 1.00 & -0.27 \\ -0.31 & 0.21 & -0.27 & 1.00 \end{bmatrix} \tag{1.84}$$

The correlation matrix indicates that the kinetic parameters f, k_{trm}, and k_{td}^o are highly correlated. In order to better address this correlation of kinetic parameters, optimal experimental design [4] should be investigated.

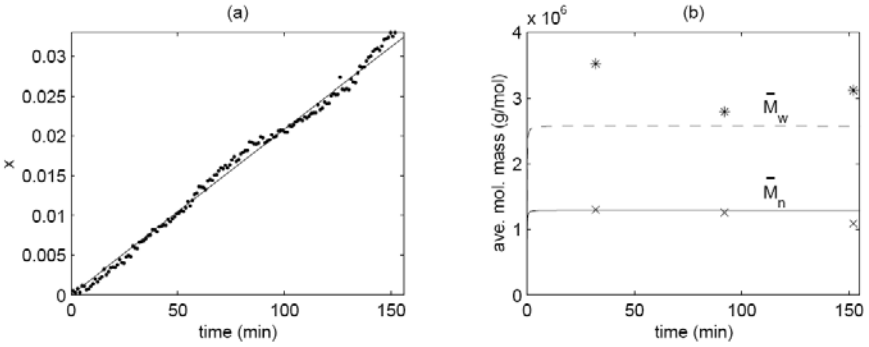


Fig. 1.23. (a) Experimental $(\cdots)$ and model prediction (—) conversion results for Case 3, Experiment 1. (b) Experimental $(\times)$, $(*)$ and model prediction (—), $(--)$ results for number- and weight-average molecular weight, respectively, for Experiment 1.

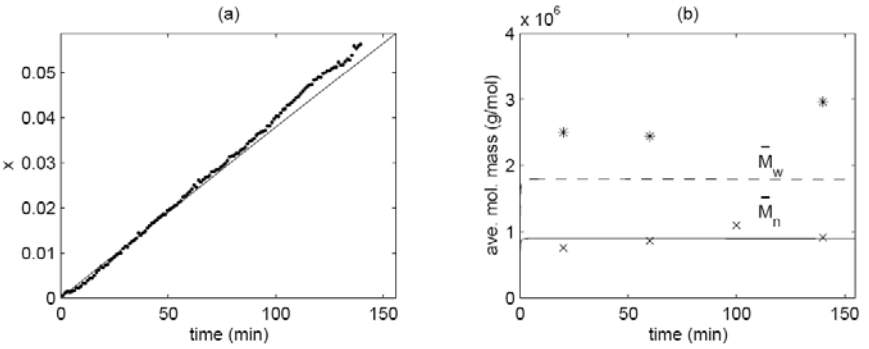


Fig. 1.24. (a) Experimental $(\cdots)$ and model prediction (—) conversion results for Case 3, Experiment 2. (b) Experimental $(\times)$, $(*)$ and model prediction (—), $(--)$ results for number- and weight-average molecular weight, respectively, for Experiment 2.

Figures 1.27–1.30 report the MWDs from experiments and simulated using the least-squares parameter estimates for Case 2, where monomer transfer is incorporated into the polymerization reaction mechanism. The simulation times for each experiment are reported in Table 1.13. The average polymer properties, $\bar{M}_n$, $\bar{M}_w$, and PD, are reported in Table 1.14.

1.4 Discrepancies Between Theory and Experimental Results

Let us explore in more detail the differences between the monomer conversion and average molecular-weights from experiments and the Case 1 model (Figures 1.15–1.18). The experimental monomer conversion for Experiment 1 appears to oscillate about the model values. We

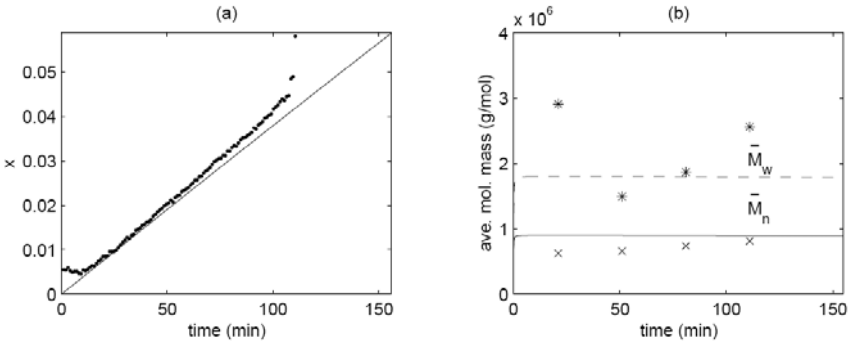


Fig. 1.25. (a) Experimental $(\cdots)$ and model prediction (—) conversion results for Case 3, Experiment 3. (b) Experimental $(\times)$, $(*)$ and model prediction (—), $(--)$ results for number- and weight-average molecular weight, respectively, for Experiment 3.

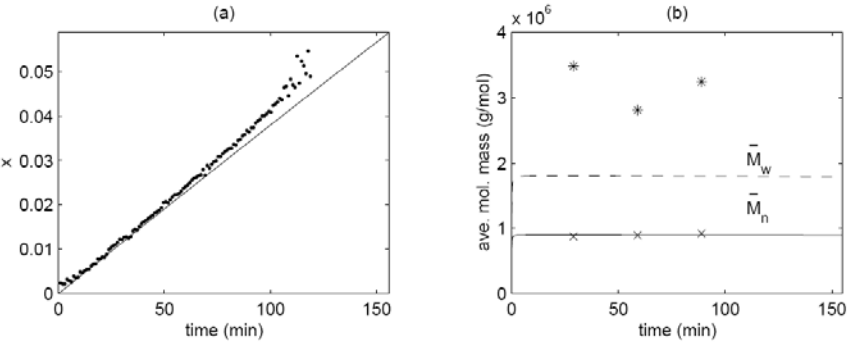


Fig. 1.26. (a) Experimental $(\cdots)$ and model prediction (—) conversion results for Case 3, Experiment 4. (b) Experimental $(\times)$, $(*)$ and model prediction (—), $(--)$ results for number- and weight-average molecular weight, respectively, for Experiment 4.

Table 1.13. Simulation times reported for free-radical bulk polymerization of methyl methacrylate using parallel CVODE, Case 2

Experiment	Time (s)
1	503
2, 3, 4	525

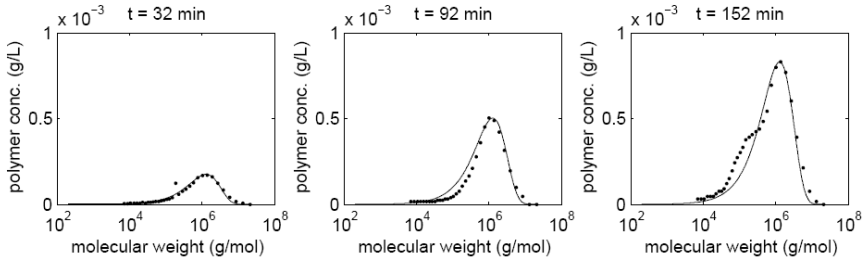


Fig. 1.27. Comparison of experimental $(\cdots)$ and simulated (—) MWD from Case 2, Experiment 1, for several time instants

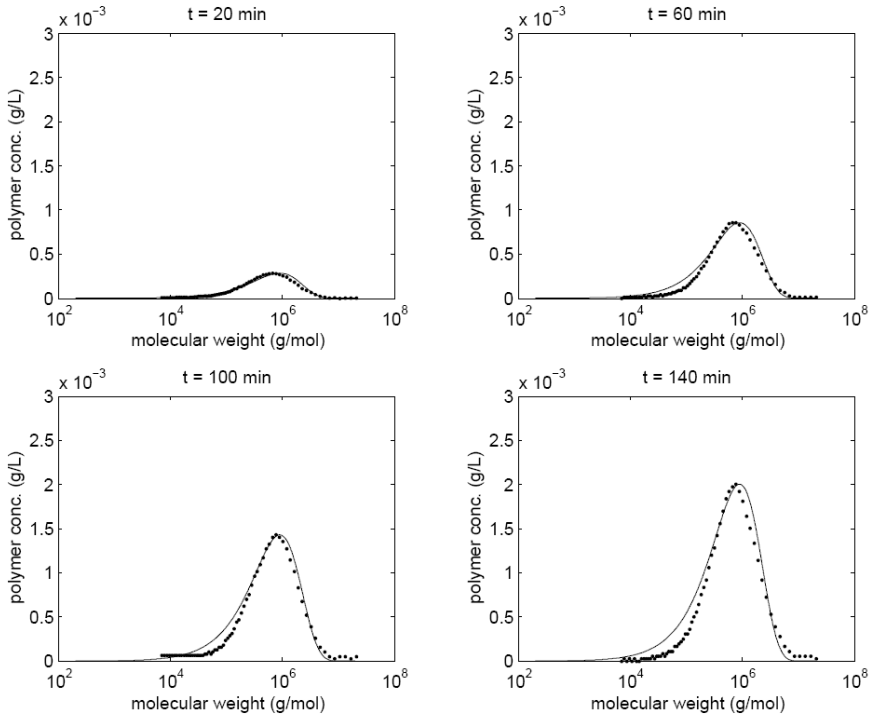


Fig. 1.28. Comparison of experimental $(\cdots)$ and simulated (—) MWD from Case 2, Experiment 2, for several time instants

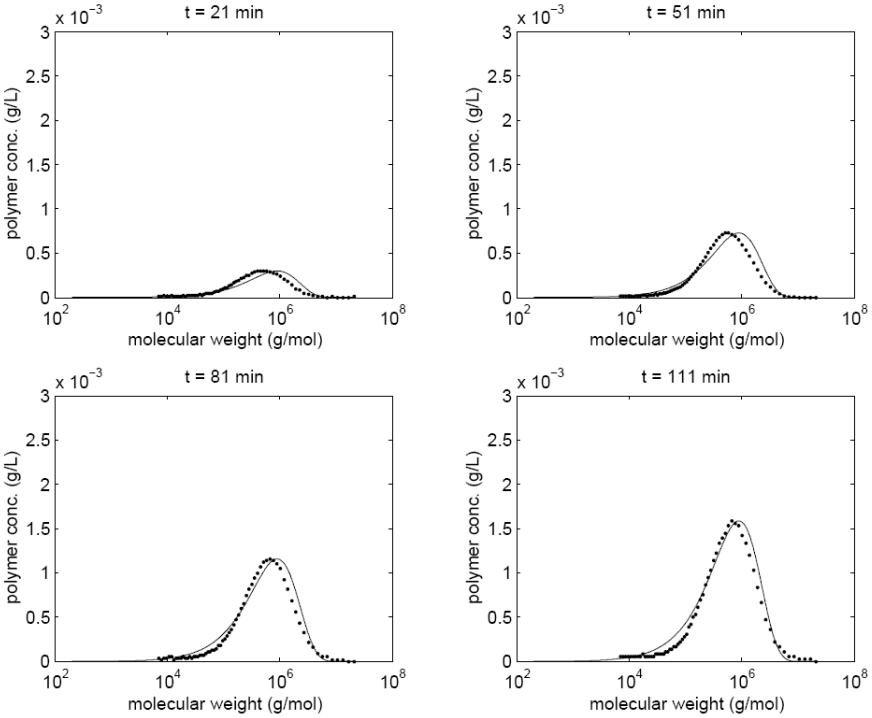


Fig. 1.29. Comparison of experimental $(\cdots)$ and simulated (—) MWD from Case 2, Experiment 3, for several time instants

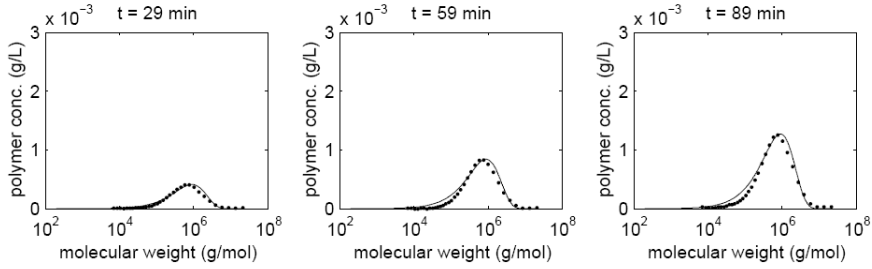


Fig. 1.30. Comparison of experimental $(\cdots)$ and simulated (—) MWD from Case 2, Experiment 4, for several time instants

Table 1.14. Experimental and simulation results (Case 2) for number-average molecular-weight, weight-average molecular-weight, and polydispersity

Experiment 1 time (min)	$\bar{M}_n^{exp}$ (g mol^{-1})	$\bar{M}_w^{exp}$ (g mol^{-1})	PD_{exp}	$\bar{M}_n^{sim}$ (g mol^{-1})	$\bar{M}_w^{sim}$ (g mol^{-1})	PD_{sim}
32	1.3×10^6	3.5×10^6	2.7	2.6×10^6	3.9×10^6	1.5
92	1.3×10^6	2.8×10^6	2.2	2.6×10^6	3.9×10^6	1.5
152	1.1×10^6	3.1×10^6	2.9	2.6×10^6	3.9×10^6	1.5

Experiment 2 time (min)	$\bar{M}_n^{exp}$ (g mol^{-1})	$\bar{M}_w^{exp}$ (g mol^{-1})	PD_{exp}	$\bar{M}_n^{sim}$ (g mol^{-1})	$\bar{M}_w^{sim}$ (g mol^{-1})	PD_{sim}
20	7.6×10^5	2.5×10^6	3.3	1.8×10^6	2.7×10^6	1.5
60	8.6×10^5	2.4×10^6	2.8	1.8×10^6	2.7×10^6	1.5
100	1.1×10^6	4.2×10^6	3.8	1.8×10^6	2.7×10^6	1.5
140	9.2×10^5	3.0×10^6	3.2	1.8×10^6	2.7×10^6	1.5

Experiment 3 time (min)	$\bar{M}_n^{exp}$ (g mol^{-1})	$\bar{M}_w^{exp}$ (g mol^{-1})	PD_{exp}	$\bar{M}_n^{sim}$ (g mol^{-1})	$\bar{M}_w^{sim}$ (g mol^{-1})	PD_{sim}
21	6.2×10^5	2.9×10^6	4.7	1.8×10^6	2.7×10^6	1.5
51	6.6×10^5	1.5×10^6	2.3	1.8×10^6	2.7×10^6	1.5
81	7.4×10^5	1.9×10^6	2.5	1.8×10^6	2.7×10^6	1.5
111	8.1×10^5	2.6×10^6	3.1	1.8×10^6	2.7×10^6	1.5

Experiment 4 time (min)	$\bar{M}_n^{exp}$ (g mol^{-1})	$\bar{M}_w^{exp}$ (g mol^{-1})	PD_{exp}	$\bar{M}_n^{sim}$ (g mol^{-1})	$\bar{M}_w^{sim}$ (g mol^{-1})	PD_{sim}
29	8.8×10^5	3.5×10^6	4.0	1.8×10^6	2.7×10^6	1.5
59	8.9×10^5	2.8×10^6	3.1	1.8×10^6	2.7×10^6	1.5
89	9.2×10^5	3.2×10^6	3.5	1.8×10^6	2.7×10^6	1.5

could find no dynamics in the mechanistic model that would explain such an oscillation, and it does not appear in the data for the other experiments. Three potential causes were considered:

- a slow drift in either temperature or monomer concentration that increased and decreased in such a manner as to *appear* to be an oscillation,
- an oscillatory bias in the monomer concentration, and
- an oscillatory disturbance in the temperature.

The oscillation in Figure 1.15 certainly looks real enough, so potential causes #2 and #3 were investigated. With many years of experience we had never seen an oscillatory bias from an ATR-FTIR measurement, which makes potential cause #2 unlikely. Regarding #3, although temperature was controlled within $\pm 0.2\,^{\circ}$C, a low-magnitude oscillation in temperature within those limits could lead to the observed oscillation in monomer concentration. The inlet hot water flow to the jacket was heated by a resistive heater (not so commonly seen in the USA, but quite common in hotel showers in some European countries) that used on-off control to regulate its output temperature. Further investigation determined that its outlet temperature, which

was the inlet to the jacket of the reactor, did oscillate under certain conditions. This motivated the replacement of the resistive heater with a steam-heated heat exchanger, which removed temperature oscillations in future experiments (unfortunately, too late for the experiments reported here).

Experiments 2–4 show up upward drift in monomer conversion that does not appear in the model (Figures 1.16–1.18). We could not explain the drift in terms of the reaction mechanism, and the drift is fairly reproducible for Experiments 2–4, suggesting that the drift is probably associated with the sensor. Drift in the ATR-FTIR measurement of monomer conversion cannot be ruled out since we have observed sensor drift in other ATR-FTIR studies, but such drifts were almost always observed only after 2 h, not within the 2 h reported in Figures 1.16–1.18. Further, ATR-FTIR drifts seen in other experiments have not been so reproducible. Our best guess on the cause is associated with the large increase in viscosity of the polymer solution that we observed between 50 and 100 min, which was measured by the increased torque on the stirring bar. The ATR-FTIR probe measures the infrared spectrum of the thin liquid film at its tip, and the liquid in this film will become less representative of the liquid in the rest of the reactor as liquid becomes more viscous and sticks to the probe tip. Then Figures 1.16–1.18 would report the monomer concentration next to the probe tip, with a small bias from the monomer concentration in the rest of the polymerization reactor (which is over 99.999% of the reactor volume).

There are no statistically significant temporal patterns in the deviations between the average molecular-weights between experiments and model outputs in Figures 1.15–1.18. The weight-average molecular-weight measurement appears to be quite noisy, and tends to be larger than that computed from the Case 1 model. The experimental number-average molecular-weight is much less noisy and is much better tracked by the model.

Although the simulated MWDs closely approximate the experimental MWDs in Case 1, there are some differences. The experimental MWD has an extra bump off the main peak for $t = 152$ min, whereas the simulated MWD is unimodal (see Figure 1.19c). As seen from Table 1.12, the simulated polydispersities are smaller than the experimental values for all experiments and the simulated number-average molecular-weights are larger for nearly all the experiments. For experiments 2–4, the simulated MWD is more asymmetric than the experimental MWD (Figures 1.20–1.22).

The model for Case 2 that includes monomer transfer results has average polymer properties similar to those of Case 1 that ignores monomer transfer, except for Experiment 1 in which all of the average molecular-weights for Case 2 are much closer to the experimental values. On average, the number-average molecular-weight for Case 2 are approximately 40% more accurate than for Case 1, and the weight-average molecular-weight for Case 2 is a factor of 2 more accurate than for Case 2 (compare Tables 1.12 and 1.14). Recall from Table 1.9 that Experiment 1 was operated under much lower BPO and initiator concentration than the other experiments, and that the MWD distribution from Experiment 1 (Figure 1.19) was qualitatively different in shape than for the other experiments.

Based on the above comparisons, it might be tempting to reach the conclusion that monomer transfer is important for the operating conditions for Experiment 1 and not important for the other experiments. However, from this data it is not clear whether this is really true. Recall that the model has many more experimental "outputs" to compare to, in the form of the molecular-weight distribution. The difference in quality of the model predictions between Cases 1 and 2 is not apparent when visually comparing the MWDs. Regardless of whether monomer transfer is ignored (Case 1) or considered (Case 2) in the model, both models produce MWDs that are reasonably coincident with experimental MWDs for a wide range of operating conditions and times. Subtle shifts in the MWD can have a dramatic effect on the

moments of the distribution and the molecular-weights, as seen by comparing the MWDs in Figures 1.19–1.30 with the values in Tables 1.12 and 1.14. The largest differences between the simulated and experimental MWDs were for the MWD with the extra bump at $t = 152$ min in Figures 1.19 and 1.27. Without repeating these experiments, it is not certain whether the extra bump is an artifact due to a mistake made by the experimentalist, or a real effect of the polymerization mechanism under those operating conditions. Often in industry, due to project deadlines the engineer does not have the luxury of carrying out a more comprehensive study, and the engineer must live with this ambiguity while moving on to the design or control stage.

The parameter sensitivity and estimation results indicate that Case 2 is the more accurate approach to determine the kinetic parameters. The smaller confidence intervals indicate a better estimate of the kinetic parameters. The next step of such a study would be to determine the kinetics under nonisothermal conditions and at higher conversions. This would help determine the functionality of the initiator efficiency, as well as the gel effect. An accurate estimate of these parameters would lead to more accurate models of free-radical polymerization.

In summary, a systematic approach to parameter estimation has been presented for determining kinetic parameters for the free-radical bulk polymerization of methyl methacrylate. The monomer concentration was monitored using ATR-FTIR spectroscopy and chemometrics. The average molecular-weights were determined by offline gel-permeation chromatography. Parameter sensitivity analysis demonstrated that it is inaccurate to exclude monomer transfer from free-radical polymerization models of MMA. Monomer transfer does not significantly affect monomer conversion, but it does have a significant effect on the average molecular-weights. The kinetic parameters were accurately determined using an objective function that equally accounted for monomer conversion and average molecular-weight properties.

References

[1] (accessed Feb. 1, 2007). `http://acts.nersc.gov/sundials`.

[2] (accessed Feb. 1, 2007). `http://turing.cse.uiuc.edu`.

[3] (accessed Feb. 1, 2007). `http://www.ncsa.uiuc.edu/UserInfo/Resources/Hardware/IA32LinuxCluster`.

[4] Beck, J. V. and Arnold, K. J. (1977). *Parameter Estimation in Engineering and Science*. Wiley, New York.

[5] Brown, P. N., Byrne, G. D., and Hindmarsh, A. C. (1989). VODE: A variable-coefficient ODE solver. *SIAM Journal on Scientific Computing* **10**, 1038–1051.

[6] Byrne, G. D. and Hindmarsh, A. C. (1998). User documentation for PVODE, an ODE solver for parallel computers. *LLNL Report UCRL-ID-130884* .

[7] Byrne, G. D. and Hindmarsh, A. C. (1999). PVODE, An ODE solver for parallel computers. *International Journal of High Performance Computing Applications* **13**, 354–365.

[8] Chang, J.-S. and Liao, P.-H. (1999). Molecular weight control of a batch polymerization reactor: Experimental study. *Industrial & Engineering Chemistry Research* **38**, 144–153.

[9] Chiu, W. Y., Carratt, G. M., and Soong, D. S. (1983). A computer model for the gel effect in free-radical polymerization. *Macromolecules* **16**, 348–357.

[10] Cohen, S. D. and Hindmarsh, A. C. (1994). CVODE user guide. *LLNL Report UCRL-MA-118618* .

[11] Crowley, T. J. and Choi, K. Y. (1997). Calculation of molecular weight distribution from molecular weight moments in free radical polymerization. *Industrial & Engineering Chemistry Research* **36**, 1419–1423.

[12] Crowley, T. J. and Choi, K. Y. (1997). Discrete optimal control of molecular weight distribution in a batch free radical polymerization process. *Industrial & Engineering Chemistry Research* **36**, 3676–3684.

[13] Crowley, T. J. and Choi, K. Y. (1998). Experimental studies on optimal molecular weight distribution control in a batch-free radical polymerization process. *Chemical Enginnering Science* **53**, 2769–2790.

[14] Curteanu, S. and Bulacovschi, V. (1999). Free radical polymerization methyl methacrylate. Modelling and simulation under semibatch and nonisothermal reactor conditions. *Journal of Applied Polymer Science* **74**, 2561–2570.

[15] Curteanu, S., Bulacovschi, V., and Lisã, C. (1998). Free radical polymerization of methyl methacrylate: Modeling and simulation by moment generating function. *Iranian Polymer Journal* **7**, 225–233.

[16] Curteanu, S., Bulacovschi, V., and Lisã, C. (1999). Algorithms for using some models of gel and glass effects in free-radical polymerization of methyl methacrylate. *Polymer-Plastics Technology & Engineering* **38**, 1121–1136.

[17] Fenouillot, F., Terrisse, J., and Rimlinger, T. (2001). Experimental study and simulation of the polymerization of methyl methacrylate at high temperature in a continuous reactor. *Journal of Applied Polymer Science* **79**, 2038–2051.

[18] Gropp, W., Lusk, E., and Skjellum, A. (1999). *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. MIT Press, 2nd edn.

[19] Hukkanen, E. J. (2004). *A Systems Approach to the Modeling and Control of Molecular, Microparticle, and Biological Distributions*. Ph.D. thesis, Univ. of Illinois at Urbana-Champaign.

[20] Kalfas, G., Yuan, H., and Ray, W. H. (1993). Modeling and experimental studies of aqueous suspension polymerization processes. 2. Experiments in batch reactors. *Industrial & Engineering Chemistry Research* **32**, 1831–1838.

[21] Kammona, O., Chatzi, E. G., and Kiparissides, C. (1999). Recent developments in hardware sensors for the on-line monitoring of polymerization reactions. *Polymer Reviews* **39**, 57–134.

[22] Kumar, A. and Gupta, R. K. (1998). *Fundamentals of Polymers*. The McGraw-Hill Companies, Inc.

[23] Maschio, G., Bello, T., and Scali, C. (1992). Optimization of batch polymerization reactors: Modelling and experimental results for suspension polymerization of methyl methacrylate. *Chemical Enginnering Science* **47**, 2609–2614.

[24] Petzold, L. R. (1983). A description of DASSL: A differential/algebraic system solver,. R. S. Stepleman *et al*, editor, *Scientific Computing*, pp. 65–68. North-Holland, Amsterdam.

[25] Ross, R. T. and Laurence, R. L. (1976). Gel effect and free volume in the bulk polymerization of methyl methacrylate. *AIChE Symp. Ser.* **72**, 74–79.

[26] Schmidt, A. D. and Ray, W. H. (1981). The dynamic behavior of continuous polymerization reactors – I. Isothermal solution polymerization in a CSTR. *Chemical Enginnering Science* **36**, 1401–1410.

[27] Tefera, N., Weickert, G., and Westerterp, K. R. (1997). Modeling of free readical polymerization up to high conversion. I. A method for the selection of models by simultaneous parameter estimation. *Journal of Applied Polymer Science* **63**, 1649–1661.

[28] Tefera, N., Weickert, G., and Westerterp, K. R. (1997). Modeling of free readical polymerization up to high conversion. II. Development of a mathematical model. *Journal of Applied Polymer Science* **63**, 1663–1680.

[29] Togkalidou, T., Fujiwara, M., Patel, S., and Braatz, R. D. (2000). A robust chemometrics approach to inferential modeling of particulate processes. *Proc. of the American Control Conference*, pp. 1732–1736.

[30] Togkalidou, T., Fujiwara, M., Patel, S., and Braatz, R. D. (2001). Solute concentration prediction using chemometrics and ATR-FTIR spectroscopy. *Journal of Crystal Growth* **231**, 534–543.

[31] Wittman, M. R. (1996). Testing of PVODE, a parallel ODE solver. *LLNL Report UCRL-ID-1225562* .

[32] Yuan, H. G., Kalfas, G., and Ray, W. H. (1991). Suspension polymerization. *Rev. Macromol. Chem. Phys.* **31**, 215–299.

[33] Zhou, J. L., Tits, A. L., and Lawrence, C. T. (1997). *User's Guide for FFSQP Version 3.7: A FORTRAN Code for Solving Constrained Nonlinear (Minmax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints*. Electrical Engineering Department and Institute for Systems Research, University of Maryland, College Park, MD 20742.

2

Open-cut Mine Planning *via* Closed-loop Receding-horizon Optimal Control

Cristian R. Rojas, Graham C. Goodwin, María M. Seron, and Meimei Zhang

The University of Newcastle, Centre for Complex Dynamic Systems and Control, School of Electrical Engineering and Computer Science, Callaghan NSW 2308, Australia
cristian.rojas@studentmail.newcastle.edu.au
{graham.goodwin,maria.seron}@newcastle.edu.au
MeiMei.Zhang@BHPBilliton.com

Summary. To obtain maximal return from a mining operation it is important that the sequence of mining steps be carefully planned. In this chapter we show how this problem can be converted into a closed-loop receding-horizon optimal control problem. Of particular interest is the formulation of the associated optimization problem in the face of uncertainty, *e.g.*, future ore prices. We show how one can formulate "open-loop", "reactive" and "closed-loop" policies to deal with price uncertainty. A "toy" example is presented to give insight into the problem. Also, a realistic mine-planning exercise is briefly described to highlight discrepancies between theory and practice.

Key words: Closed-loop control, mine planning, reactive control, receding horizon, scenario generation, stochastic programming

2.1 Background

In this chapter we will be concerned with optimal open-cut mine planning. This is a quintessential example of a complex optimization problem in the face of uncertainty.

Considerable benefits can be obtained by careful planning of mining operations. Hence, there has been a substantial research effort devoted to the topic of optimal mine planning, see for example [1–3, 6–13, 32].

The essence of this problem is as follows: One has preliminary available data on the location of an ore-body in a particular geological volume. Given the data, one would like to know "where" and "when" to dig so as to optimize the "net present value". A typical mining operation can span 15 to 20 years or more and hence there is a temporal aspect to questions. Also, the optimization needs to respect a host of constraints, *e.g.*, mining capacity in each year and slope constraints on the mine walls to avoid collapses. Finally, there are a number of uncertainties, *e.g.*, the location and extent of the ore body and, of course, one does not know the future price that the ore will have.

When formulating solutions to optimization problems of the type described above, it is helpful to distinguish between "open-loop", "reactive" and "closed-loop" solutions. An open-loop solution is the easiest to understand and solve. Here, one simply finds a single "control"

sequence over the optimization horizon. Here, and in the following, we use the term "control" to describe those variables that can be selected (at each time step) by the optimizer. Thus, in the case of mining, we could define $u_{jk}(t)$ as the action of mining (or not) at location (x_j, y_k) at time t. Then, $\{u_{jk}(t)\}$ becomes the "control variable".

"Reactive" strategies are closely related to open-loop strategies. The key difference is that, whereas in an open-loop strategy one applies the entire open-loop control sequence blindly, in a reactive strategy one applies only the current control. Then, when new information (*i.e.* the current ore price) arrives at the next time step, one does another open-loop plan beginning from the current "state". Again one implements only the first control step and so on.

"Closed-loop" strategies are quite different in the sense that, in these strategies, one takes account of the fact that *in the future* additional information will become available. Thus, closed-loop policies represent a mapping between the "information state" and the control. It is well known that for a very special class of problems (*e.g.*, linear-quadratic optimal control problems) separation holds, *i.e.* the current control is not affected by the fact that, in the future, additional information will become available. This is a remarkable result. Unfortunately, separation does not hold in general. Thus, for general temporal closed-loop optimization problems, such as the mine-planning case, one needs, in principle, to carry out a function optimization, where the function maps the "information state" to the control.

The obvious problem with closed-loop optimization, as described above, is the need for function optimization. Thus, in principle, one needs a different "control" for every possible outcome of the uncertain variables. This is a formidable task. Indeed, even in the open-loop case, typical optimization problems can be overwhelmingly complex. For example, in the mine-planning problem one typically has thousands of variables per time step. Also, if one has, say, 15 steps, then one can easily have 15^{1000} variables to consider. This number is way beyond comprehension and should be compared with 10^{80} that is the number of atoms in the known universe. This is the well-known "curse of dimensionality" in temporal optimization.

Clearly one needs to make various simplifications to the problem if one is to have any hope of solving it in a realistic time. Fortunately, by various "aggregation strategies" one can often reduce the problem to an acceptable size. For example, in typical mining problems, various strategies based on spatial quantization are used to simplify the problem. These strategies have resulted in open-loop optimal mine planning becoming a feasible idea and, indeed, it is frequently employed in practice.

One relatively recent idea, proposed by the current authors [17], is to also utilize *nonuniform* time quantization to simplify the calculations needed in open-loop planning. The basic idea here is that, although the future cannot be ignored, it often has a diminishing impact on the best control to use *now*. Based on this observation, one might conjecture that one will not compromise performance too much if one quantizes time more coarsely in the distant future. One then solves for the nonuniformly quantized "control" sequence and applies (or stores) the first step. One then advances time by one step and repeats the procedure using a different set of (nonuniformly) quantized times. This "rolling-horizon" approach allows one to build a strategy with a small uniform time step by combining a sequence of moving nonuniformly quantized problems. This idea has been adopted by industrial colleagues and used in realistic mine-planning problems. Indeed, this has led to computational savings of the order of $10:1$. Consequently, one might believe that there is now scope to try more daring strategies, *e.g.*, those based on some form of closed-loop planning.

The latter strategies have the potential to make the planning more "flexible" so that the mine is better able to respond to future price variations.

An immediate stumbling block is that, as discussed above, one needs to essentially map every possible realization of the uncertain variables into a different control sequence. Conse-

quently, one is obviously limited to a small number of possible realizations. Indeed, for the mining problem, it seems that only a handful (say 10 to 100) realizations could be contemplated.

Actually, the above problem has been long recognized in the planning literature, and there exists a substantial volume of work on how one can choose a *small*, *representative* set of realizations of uncertain variables. This is often called the problem of "scenario generation" and has been extensively discussed in the applied optimization literature including finance, management and statistics [5, 24, 25, 31, 34]. The problem is actually equivalent to approximating a given continuous probability distribution by a discrete distribution having finite support of given cardinality. In turn, this is a special type of quantization problem and it thus has links to contemporary literature in signal processing.

2.2 Experiment Description

2.2.1 Formulation of Mine-planning Problem

To set the work in the current chapter in context, we will next give an outline of the mine-planning problem. (Of course, real mine-planning problems involve a host of other practical issues, *e.g.*, processing capacity, stockpiling, *etc.*, not covered here.) The basic idea of open-cut mining can be visualized in Figure 2.1, which shows the ultimate pit of a typical mine, that is, the opening left in the ground after mining operations have been completed. (Actually this particular mine will be used as the basis of the discussion of discrepancies between theory and experiments in Section 2.4.) For simplicity of exposition, we represent the potential mine by the "box" shown in Figure 2.2, where the "surface" is divided into $(J + 1) \times (K + 1)$ rectangles.

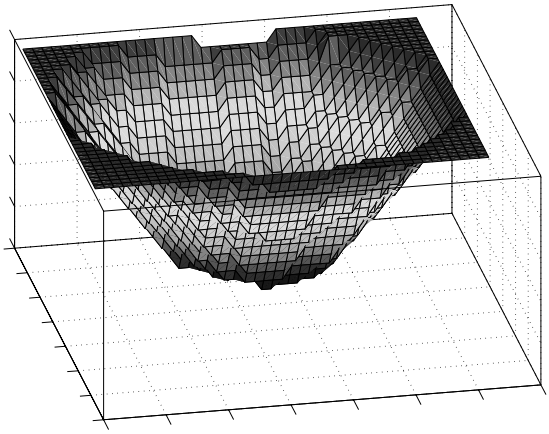


Fig. 2.1. Illustration of a typical mine ultimate pit

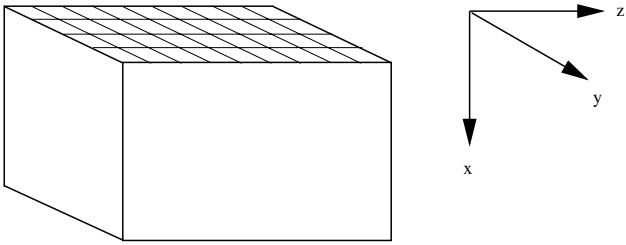


Fig. 2.2. Simplified representation of the mine

2.2.2 State-space Model of the Mine

So that we can utilize control-theory insights, we next cast the mine-planning problem in the form of a control problem. Accordingly, we define the mine state as the set of pit depths at the locations of the surface. We represent the evolution of this state via a linear, discrete-time dynamic model where mining action is the control input. Specifically, we denote by $x_{jk}(t)$ the mine depth at location jk at time t. Similarly, we denote by $u_{jk}(t)$ the action to mine (or not) at time t in the location jk, $j \in \{0, \ldots, J\}$, $k \in \{0, \ldots, K\}$. We thus think of $u_{jk}(t)$ as an $(J+1) \times (K+1)$ input vector. A state model for the system can then be written as

$$\begin{aligned} x_{jk}(t+1) &= x_{jk}(t) + b_1 u_{jk}(t); \quad t \in \mathbb{N}_0 \\ x_{jk}(0) &= 0; \quad j \in \{0, \ldots, J\}, \quad k \in \{0, \ldots, K\} \end{aligned} \tag{2.1}$$

where b_1 is a constant that reflects the effect of one unit of mining action.

2.2.3 Constraints

Mining operations are subject to a number of constraints including constraints on the order that material can be mined, slope constraints, mining capacity, *etc.*, see for example [1–3, 6–13, 30, 32]. In the state-space formulation presented in Section 2.2.2, these constraints can be incorporated in a natural way.

For example, note that $u_{jk}(t)$ can take either the value 1 or 0 indicating the action of mining or not at location jk at time t. Thus, $u_{jk}(t)$ is non-negative and the model (2.1) readily ensures that the mine depth cannot decrease at any location. Also, slope constraints on the mining depth can be directly incorporated by means of state constraints of the form

$$|x_{\ell n}(t) - x_{jk}(t)| \leq b_2; \qquad t \in \mathbb{N}_0, \quad |\ell - j| = 1, \quad |n - k| = 1 \tag{2.2}$$

The mining-capacity constraint can be easily handled by imposing an input constraint such that only a certain number of $u_{jk}(t)$ can be nonzero at any t. Other constraints, such as processing-plant constraints, can also be modeled by introducing functions to model ore content. Finally, the state-space formulation presented here can be extended to more complex situations, such as multiple processing plants with variable capacities, multiple material stockpiles, variable material price, *etc*.

2.2.4 The Cost Function Representing Net Present Value

The value of the body of ore at different locations is typically obtained by preliminary drilling work. Using this information one can construct a value function $V_{jk}(x_{jk})$ that represents the value assigned to the material in location jk at depth x_{jk}. We also introduce a time-discounting function d_t to yield the net present value and assume that the price of ore at time t is c_t. The cost function to use for mine planning, representing the net present value achieved by a given mining strategy over a planning horizon T, then takes the form

$$J := \sum_{t=1}^{T} \sum_{j=1}^{J} \sum_{k=1}^{K} d_t c_t \, V_{jk}(x_{jk}(t)) \, u_{jk}(t-1) \tag{2.3}$$

Note that we multiply by $u_{jk}(t-1)$ in Equation (2.3) since the value in the ore is only liberated when it is mined.

2.2.5 Introduction of Uncertainty

The formulation described so far assumes that all quantities are known into the future. To illustrate the impact of uncertainty on the mine-planning problem, we assume that the only uncertain variable is c_t, the price of the ore at time t.

The simplest possible model for the ore price is as a deterministic sequence $\{c_t\}$. For example, if we take copper as an illustration, then the price has been falling in real terms (i.e., when expressed in 2006 dollars). However, this falling trend has also been accompanied by random fluctuations and shocks. Thus, there is a strongly uncertain component in the planning description.

We will illustrate the nature of this uncertainty by examining copper price.

We will consider data normalized by the Consumer Price Index (CPI) conversion factor in the USA, that is, in terms of a fixed buying power for a unit of currency (USD). Typical data is shown in Figure 2.3 (USA copper data from 1967 to 2002) normalized by the CPI.

We see from Figure 2.3 that there has been a general trend downwards (exponential curve in the figure) with some added stochastic behavior including some key "shocks" at some points in time. It will be convenient for data-fitting purposes to first remove the deterministic trend by fitting an exponential and then subtracting it. The detrended data is shown in Figure 2.4.

It would be relatively easy to give a stochastic description for the data given in Figure 2.4. Indeed, there are a number of standard models used in mine-planning applications. These include mean-reverting models, *etc*. (see, for example, [22]). Note, however, that these models contain real-valued random variables. These models are useful for generating Monte Carlo simulations but are unsuitable for use in *closed-loop* mine planning due to the cardinality of the uncertainty description. Indeed, as we have already remarked, the mine-planning problem is barely computationally feasible even if all variables are considered deterministic and known. Hence, attempting function optimization with real-valued random variables for ore price would be absolutely out of the question. Thus, this represents an interesting theory/practice gap. Consequently, in practice we are forced to model the uncertain variables by a random variable having a discrete distribution. Indeed, the cardinality of the support of the distribution must be very small to make the problem computationally feasible. Thus, we must carefully choose a "representative" set of scenarios to describe the uncertain variables for the purpose of closed-loop planning. Further information on scenario generation is obtained in [15, 18–20, 23, 26–28]. Note that these methods have an additional complexity beyond that

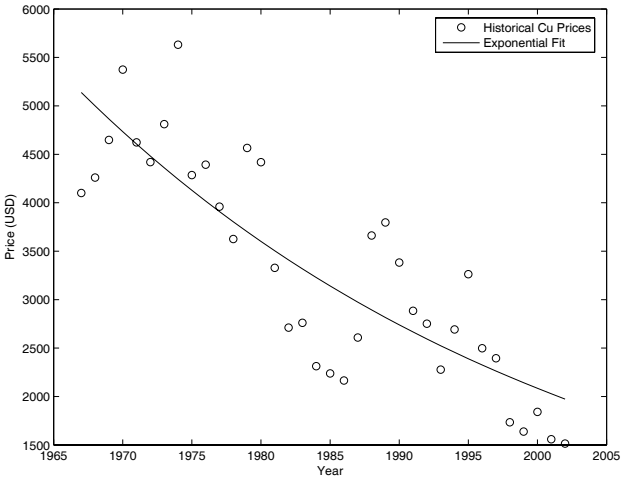


Fig. 2.3. USA copper price (normalized by the CPI) and its exponential trend

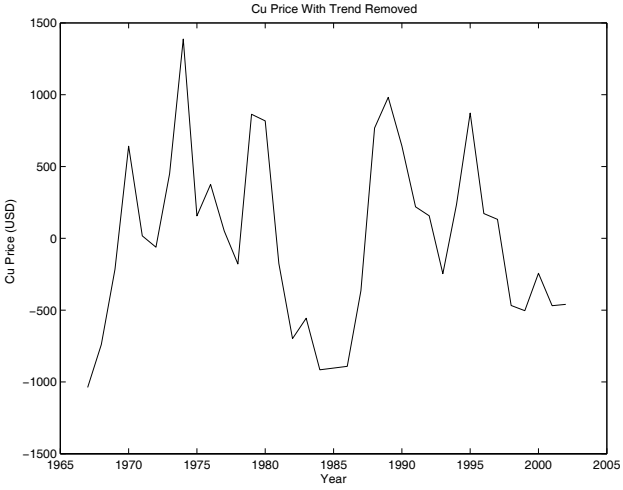


Fig. 2.4. Copper price after removing exponential trend

normally encountered in system identification [16], since the cardinality of the possible realizations must be restricted. Assuming that the scenarios are $c_t(s)$ for $s = 1, \ldots, S$, where t denotes the time (in years), then the cost function for the sth scenario becomes

$$J_s := \sum_{t=1}^{T} \sum_{j=1}^{J} \sum_{k=1}^{K} d_t c_t(s)\, V_{jk}(x_{jk}(t))\, u_{jk}(t-1) \tag{2.4}$$

In the following, we will describe methods for optimizing (the expected value of) cost functions of the general form of Equation (2.4). We first take a short digression to outline, in general terms, the key concepts underlying "open-loop", "reactive" and "closed-loop" planning.

2.2.6 Open-loop and Reactive Planning

In "open-loop" planning one solves an optimization problem at the outset based on the expected value of future ore price. Specifically, this policy minimizes (subject to the relevant constraints) the cost function

$$J^{OL} := \sum_{s=1}^{S} p_s J_s \tag{2.5}$$

where J_s, defined in Equation (2.4), is the cost associated with each scenario and p_s denotes the probability of the sth scenario. The resulting optimal mining strategy is a sequence of mining actions over the planning horizon T. Note that this policy does not use the information that future knowledge about price will be available, that is, that the current ore price will actually become known at each time step.

In "reactive" planning one applies only the first mining action resulting from the open-loop policy described above. At the next step one solves another open-loop problem over a planning horizon of length $T-1$ and "reacts" to the current value of ore price by using this value for the first time step. The procedure is repeated at each time step in a "rolling-horizon" fashion. Hence, although this policy uses the current value of ore price at each step, the fact that future information about ore price will be available is not taken into account.

2.2.7 Closed-loop Planning

"Closed-loop" planning (sometimes called "with recourse" in the stochastic programming literature) takes advantage of the fact that, in the future, the price will be known. Such policies involve a function optimization that links the information state to the control action.

A simple way of capturing the closed-loop planning idea is to allocate a separate control strategy to each scenario. This idea is illustrated by Figure 2.5, where four different price scenarios are considered. In this figure, price at stage 1 can only take the value $c_1 = v_1$; at stage 2, price can take the value $c_2 = v_2$, with probability α, or $c_2 = v_3$, with probability $1-\alpha$; at stage 3, if price at stage 2 was v_2, then price can take either the value $c_3 = v_4$, with *conditional* probability β, or $c_3 = v_5$, with *conditional* probability $1-\beta$; and similarly for the values v_6 and v_7. The price scenarios are then defined by each of the four branches of the scenario tree (for example, scenario 1 corresponds to $c_1 = v_1$, $c_2 = v_2$ and $c_3 = v_4$). In principle, we think of using 4 separate mines, each corresponding to a separate price scenario. Note that the scenarios have a tree like structure, *i.e.* $c_1(1) = c_1(2) = c_1(3) = c_1(4)$, $c_2(1) = c_2(2)$ and $c_2(3) = c_2(4)$. To bring all of the separated problems together, we note that we should not use future price information until we actually reach that time. This can be captured by ensuring that the mining strategies are nested (*i.e.* they share common components).

We now introduce four corresponding mining optimization problems with inputs $u_{jk}(t,s)$, for $s \in \{1,2,3,4\}$, where

- $u_{jk}(t,s) = 1$ means we mine at time t at the location jk under price scenario s.
- $u_{jk}(t,s) = 0$ means we do not mine at time t at the location jk under price scenario s.

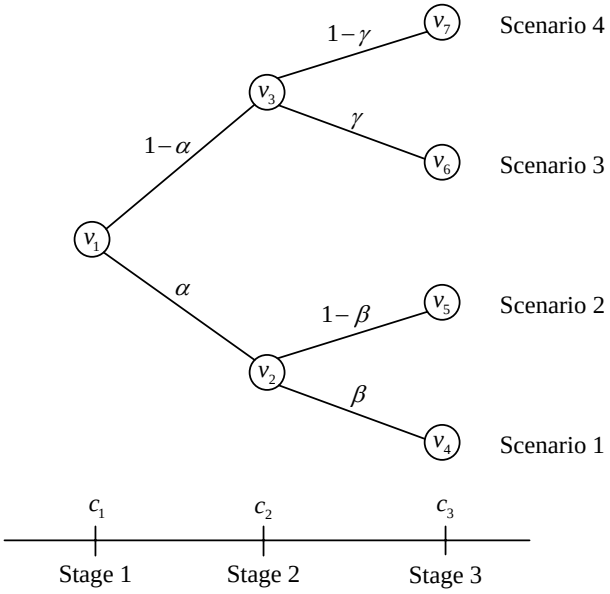


Fig. 2.5. An example of a scenario tree structure for closed-loop mine planning

Correspondingly, the state models satisfy, for $s = 1, 2, 3, 4$,

$$\begin{aligned} x_{jk}(t+1,s) &= x_{jk}(t,s) + b_1 u_{jk}(t,s); \qquad t \in \mathbb{N}_0 \\ x_{jk}(0,s) &= 0; \qquad j \in \{0,\dots,J\}, k \in \{0,\dots,K\} \end{aligned} \tag{2.6}$$

As outlined above, it is important that we do not use information about price until it becomes available. This is captured by adding constraints that ensure that the mining strategies are equal at each node of the price-scenario tree. For the example above, these constraints have the form:

$$\begin{aligned} &u_{jk}(0,1) = u_{jk}(0,2) = u_{jk}(0,3) = u_{jk}(0,4) \\ &u_{jk}(1,1) = u_{jk}(1,2) \\ &u_{jk}(1,3) = u_{jk}(1,4); \qquad j \in \{0,\dots,J\}, k \in \{0,\dots,K\} \end{aligned} \tag{2.7}$$

2.3 Simulation Results

Real mine-planning problems are exceedingly complex and can take weeks to solve on high-speed computers. Also, they are notoriously nonconvex. Hence, it is difficult to gain insights by examining real problems since one readily becomes lost in details. Accordingly, we will take

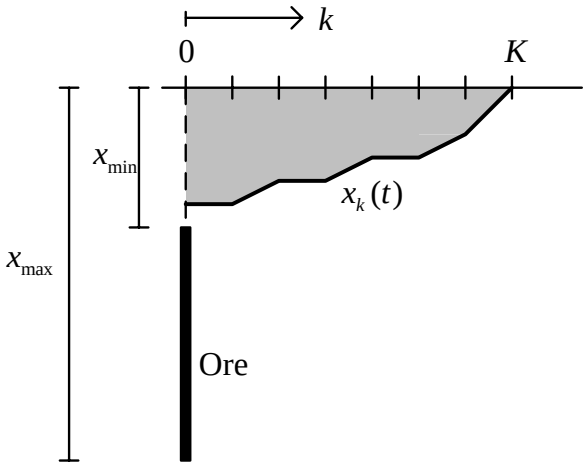


Fig. 2.6. A simplified mine-planning problem

a simplified "toy" example to gain insight into the structure of the various planning strategies. We thus choose a (very) simplified mining example as depicted in Figure 2.6.

A vertical ore bar of length $x_{\max} - x_{\min}$ is assumed to be located at a depth $x_{\min}$ in the soil. Our goal is to extract this ore in an optimal fashion subject to constraints. In particular, we assume that we can only extract a certain maximum amount of material per year, and that wall-slope constraints must be satisfied. We also assume that all extracted ore is immediately sold at the current ore price, which varies in a "random" way. Our goal is to maximize the net present value of the total return. Without loss of generality, we include the discount factor in the ore price.

We consider a two-dimensional version of the problem, where we are looking at a cross section of the soil. Using time and space quantization, let $x_k(t)$ be the depth of the soil at a distance $k \in \{0, \ldots, K\}$ to the right of the bar, at a time $t \in \{0, \ldots, T\}$, where $T \in \mathbb{N}$ is the time-horizon length and K is the maximum distance of the bar to be considered. Notice that, due to the symmetry of the problem, we need only consider positive values of k, since the solution for $k < 0$ is necessarily the mirror reflection of the solution for $k > 0$.

The cost function to be maximized is

$$J := E\left\{\sum_{t=1}^{T} c_t \mu(x_0(t-1), x_0(t))\right\} \tag{2.8}$$

where μ is a (positive Borel) measure of the ore distribution and $\{c_t\}_{t\in\{1,\ldots,T\}}$ is the ore price (considered as a stochastic process, to be specified later).

To simplify the problem further, we make the assumption that the bar has infinite length and that its top lies at the soil surface, *i.e.* that

$$x_{\min} = 0$$
$$x_{\max} = \infty \tag{2.9}$$

The cost function (2.8) can then be written as

$$J = E\left\{\sum_{t=1}^{T} c_t[x_0(t) - x_0(t-1)]\right\} \tag{2.10}$$

The constraints for this problem are:

1. *Initial condition*

$$x_k(0) = 0; \qquad \text{for all } k \in \{0, \ldots, K\} \tag{2.11}$$

2. *Maximum amount of material to be extracted per year*

$$\frac{x_0(t) - x_0(t-1) + x_K(t) - x_K(t-1)}{2} + \sum_{k=1}^{K-1}[x_k(t) - x_k(t-1)] \leq \frac{C}{2}; \qquad \text{for all } t \in \{1, \ldots, T\} \tag{2.12}$$

where $C \in \mathbb{R}^+$. The form of this constraint comes from a trapezoidal approximation to the integral of the extracted material.

3. *Non-negativity constraints*

$$x_k(t) \geq x_k(t-1); \qquad \text{for all } k \in \{0, \ldots, K\},\ t \in \{1, \ldots, T\} \tag{2.13}$$

4. *Slope constraints*

$$|x_{k-1}(t) - x_k(t)| \leq 1; \qquad \text{for all } k \in \{1, \ldots, K\},\ t \in \{1, \ldots, T\} \tag{2.14}$$

Notice that both the cost function and the constraints are linear in the decision variables. Thus, the problem can be described as a stochastic linear program and, as we will see, it can be solved by rewriting it as a deterministic linear program (thus avoiding having to state and solve Bellman's equation explicitly [4, 29]).

In the following, we will restrict our attention to a simple three-stage price model as shown in Figure 2.5.

According to this scenario tree, we have that

$$\begin{aligned} P\{c_2 = v_2\} &= \alpha \\ P\{c_2 = v_3\} &= 1 - \alpha \\ P\{c_3 = v_4 | c_2 = v_2\} &= \beta \\ P\{c_3 = v_5 | c_2 = v_2\} &= 1 - \beta \\ P\{c_3 = v_6 | c_2 = v_3\} &= \gamma \\ P\{c_3 = v_7 | c_2 = v_3\} &= 1 - \gamma \end{aligned} \tag{2.15}$$

We next transform the above stochastic problem into a deterministic one by introducing a new state sequence $x'_k(t, s)$, where k denotes spatial position, t denotes time (or stage) and s denotes the scenario number. Thus, we have

$$x_k(1) = x'_k(1,1) = x'_k(1,2) = x'_k(1,3) = x'_k(1,4); \qquad k \in \{0,\ldots,K\} \tag{2.16}$$

$$x_k(2) = \begin{cases} x'_k(2,1) = x'_k(2,2),\ c_2 = v_2 \\ x'_k(2,3) = x'_k(2,4),\ c_2 = v_3 \end{cases} \qquad k \in \{0,\ldots,K\} \tag{2.17}$$

$$x_k(3) = \begin{cases} x'_k(3,1),\ c_3 = v_4 \\ x'_k(3,2),\ c_3 = v_5 \\ x'_k(3,3),\ c_3 = v_6 \\ x'_k(3,4),\ c_3 = v_7 \end{cases} \qquad k \in \{0,\ldots,K\} \tag{2.18}$$

Then, the cost of the deterministic equivalent program can be written as

$$\begin{aligned} J &= E\{c_1[x_0(1) - x_0(0)] + c_2[x_0(2) - x_0(1)] + c_3[x_0(3) - x_0(2)]\} \\ &= v_1 x'_0(1,1) \\ &\quad + v_2\alpha[x'_0(2,1) - x'_0(1,1)] + v_3(1-\alpha)[x'_0(2,3) - x'_0(1,1)] \\ &\quad + \alpha\left(v_4\beta[x'_0(3,1) - x'_0(2,1)] + v_5(1-\beta)[x'_0(3,2) - x'_0(2,1)]\right) \\ &\quad + (1-\alpha)\left(v_6\gamma[x'_0(3,3) - x'_0(2,3)] + v_7(1-\gamma)[x'_0(3,4) - x'_0(2,3)]\right) \end{aligned} \tag{2.19}$$

The constraints for this program are

$$\frac{x'_0(1,1) + x'_K(1,1)}{2} + \sum_{k=1}^{K-1} x'_k(1,1) \le \frac{C}{2} \tag{2.20}$$

$$\frac{x'_0(2,s) + x'_K(2,s) - x'_0(1,1) - x'_K(1,1)}{2} + \sum_{k=1}^{K-1} [x'_k(2,s) - x'_k(1,1)] \le \frac{C}{2}; \qquad s \in \{1,3\}$$

$$\frac{x'_0(3,s) + x'_K(3,s) - x'_0(2,s) - x'_K(2,s)}{2} + \sum_{k=1}^{K-1} [x'_k(3,s) - x'_k(2,s)] \le \frac{C}{2}; \qquad s \in \{1,2,3,4\}$$

$$\begin{aligned} x'_k(1,1) &\ge 0; \quad k \in \{0,\ldots,K\} \\ x'_k(t,s) - x'_k(t-1,s) &\ge 0; \quad k \in \{0,\ldots,K\}, s \in \{1,2,3,4\}, t \in \{2,\ldots,T\} \end{aligned} \tag{2.21}$$

$$|x_{k-1}(t,s) - x_k(t,s)| \le 1; \qquad k \in \{1,\ldots,K\}, s \in \{1,2,3,4\}, t \in \{1,\ldots,T\} \tag{2.22}$$

The solution of this program gives the optimal (closed-loop) solution of the mine-planning problem.

Alternatively, if we are interested in the reactive solution to this problem, we can achieve this by adding extra constraints as follows:

$$\begin{aligned} x'_k(2,1) &= x'_k(2,3); && k \in \{0,\dots,K\} \\ x'_k(3,1) &= x'_k(3,2) = x'_k(3,3) = x'_k(3,4); && k \in \{0,\dots,K\} \end{aligned} \tag{2.23}$$

The inclusion of these constraints gives us a deterministic linear program whose solution provides $x'_k(1,1)$ for $k \in \{0,\dots,K\}$, since they force a open-loop solution for stage 1. Similarly, we can repeat for stage 2 by considering two problems depending on whether c_2 takes the value v_2 or v_3.

We will examine several sets of pricing data.

(i) Pricing data #1

Here we let $K = 40$, $C = 400$ and assume

$$\begin{aligned} v_1 &= 0.5 & \alpha &= 32/33 \\ v_2 &= 0.1 & \beta &= 17/24 \\ v_3 &= 10 & \gamma &= 17/24 \\ v_4 &= 0.01 \\ v_5 &= 0.25 \\ v_6 &= 1 \\ v_7 &= 25 \end{aligned} \tag{2.24}$$

Under these conditions, it turns out that the closed-loop and reactive policies coincide. The solution for the first two stages is shown in Figure 2.7, and the solution for stage 3 is shown in Figure 2.8.

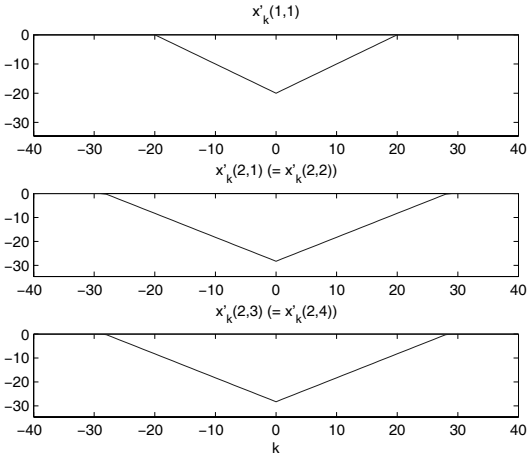


Fig. 2.7. First two stages of the solution to the three-stage mine planning problem, with parameters given by Equation (2.24)

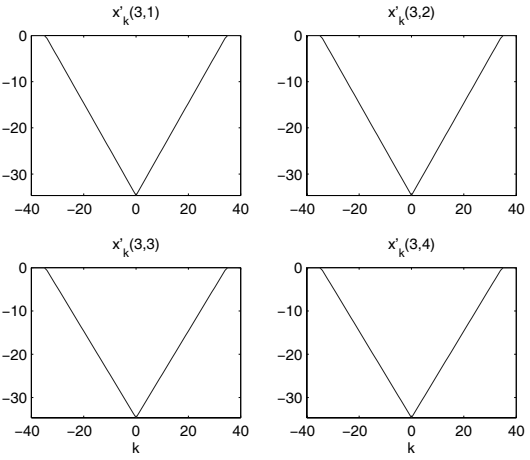


Fig. 2.8. Third stage of the solution to the three-stage mine planning problem, with parameters given by Equation (2.24)

It may seem strange that the reactive and closed-loop policies are identical for this example. However, this can be explained as follows:
We note that

$$\begin{aligned} c_1 = v_1 = 0.5 > 0.4 = E\{c_2\} \\ E\{c_2\} = 0.4 > 0.32 = E\{c_3\} \\ v_2 = 0.1 > 0.08 = E\{c_3 | c_2 = v_2\} \\ v_3 = 10 > 8 = E\{c_3 | c_2 = v_3\} \end{aligned} \tag{2.25}$$

Thus, at every stage the ore price "tends" to decrease (in a mean sense), so common sense tells us that the best strategy is to extract as much ore as possible as soon as one can in order to maximize the return. Since this tendency is exhibited in both a conditional and unconditional sense (with respect to the information available at the present time), the optimality of this strategy does not depend on how much information we actually have at a specific stage, and thus the closed-loop and reactive policies must coincide.
Notice that the coefficients of the cost function (2.19) are proportional to the differences in the *conditional* means. On the other hand, if we introduce the constraints (2.23) in the cost function (2.19), the coefficients of the resulting cost function are proportional to the differences in the *unconditional* means. This observation supports the conclusions of the previous paragraph.

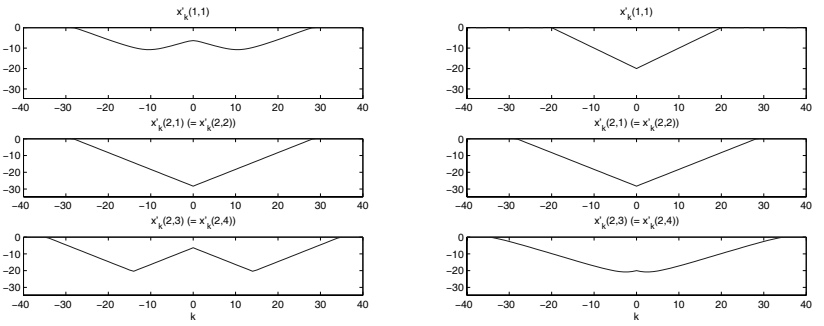
(ii) Pricing data #2
The above discussion suggests the form that alternative pricing data would need to take to ensure that reactive and closed-loop policies were different. For example, if we now let $K = 40$, $C = 100$ and choose

$$
\begin{aligned}
v_1 &= 0.41 & \alpha &= 32/33 \\
v_2 &= 0.1 & \beta &= 17/24 \\
v_3 &= 10.5 & \gamma &= 7/12 \\
v_4 &= 0.01 & & \\
v_5 &= 0.25 & & \\
v_6 &= 1 & & \\
v_7 &= 25 & &
\end{aligned}
\tag{2.26}
$$

we now have that

$$
\begin{aligned}
c_1 = v_1 = 0.41 > 0.40588 \approx E\{c_2\} \\
E\{c_2\} \approx 0.40588 > 0.40118 \approx E\{c_3\} \\
v_2 = 0.1 > 0.08 = E\{c_3 | c_2 = v_2\} \\
v_3 = 10.5 < 11 = E\{c_3 | c_2 = v_3\}
\end{aligned}
\tag{2.27}
$$

This means that the ore price exhibits an "unconditional tendency" to decrease, but if $c_2 = v_3 = 10.5$, c_3 "conditionally" tends to increase. Thus, the information available at stage 2 has the potential to be quite valuable when making a decision. The above line of reasoning suggests that, in this case, closed-loop and reactive policies might differ. This is indeed exactly what happens, as is shown in Figure 2.9.



(a) (b)

Fig. 2.9. Solution of the three-stage mine-planning problem for the parameters given in Equation (2.26). (a) Closed-loop solution. (b) Reactive solution. The third stage of the solution given by both policies is the same as the one shown in Figure 2.8.

2.4 Discrepancy Between Theory and Experiments

2.4.1 Discussion of the Simple Example of Section 2.3

Whilst it may seem, at first sight, that closed-loop planning would offer substantial advantages, the differences between closed-loop and reactive planning appear to be relatively modest for

the mine-planning problem. Specifically, the costs obtained for the example of Section 2.3 when using the closed-loop and reactive policies are 14.37744 and 14.23304, respectively. The difference may seem small (1%), but in the mine industry even a 1% improvement can equate to tens of millions of dollars! This means that there may indeed be cases where it might be worthwhile to obtain the closed-loop solution instead of just computing the reactive one.

Perhaps of more interest in this example, is the qualitative difference between the strategies. Indeed, we see from Figure 2.9b, that the reactive policy simply tries to extract as much ore as possible. On the other hand, the closed-loop policy (Figure 2.9a) makes preparations for a possible future price increase by removing some waste material so that one can be in a better position, at the next step, to exploit a potential price increase by then allowing more ore to be extracted.

2.4.2 Some General Observations

Although it is difficult in general to know in advance whether there will be differences between the three kinds of stochastic programming policies for a particular problem, we have seen that for the above mine-planning example it is relatively simple to arrive at some necessary conditions for the existence of discrepancies. In particular, there have to be discrepancies between the signs of the differences of the conditional means and the signs of the differences of the unconditional means.

Another interesting observation we can make from this example is that the number of possible optimal solutions is essentially finite, in the sense that to achieve the optimal cost, we only need consider a finite number of strategies. This follows from the fact that the mine-planning problem is equivalent to a deterministic *linear* program, which has a finite number of basic solutions [21, p. 20].

Moreover, from the sensitivity theory of linear programming [33, Chapter 7], we know that small changes in the coefficients of the cost function of a linear program do not change the optimal solution (except for a set of parameter values of Lebesgue measure zero), and the optimal cost is a continuous function of those coefficients.

Since the coefficients of the cost function are directly related to the scenario tree, the above line of reasoning implies that the solution of the mine-planning problem is "locally insensitive" to changes in the scenario tree (at least assuming that its structure remains unchanged), but it may suffer "global bifurcations", since the solution may "jump" to other entirely different solutions for other changes in the scenario tree. This behavior is evident in the difference between the solutions to the problem described above under Pricing Data #1 and Pricing Data #2.

The previous remark is quite interesting, since it implies that it is unhelpful to try to find an "exact" scenario tree to represent the uncertainty in the ore price. On the other hand, it can be very insightful to generate several scenario trees in order to study the robustness of the obtained solution to changes in the ore price structure. (This idea can be contrasted with the "contamination technique" developed in [14], which is aimed at studying the local sensitivity, or *stability*, of the stochastic program.)

2.4.3 Application to a Real Mine-planning Problem

The work described in this chapter has been carried out in collaboration with personnel at BHP Billiton Melbourne Technology Centre. They have tested the relative efficacy of the different policies on real problems. For simple ore geometries it turns out that reactive planning

achieves near-optimal performance. However, for more complex geometries, greater differences between open-loop, reactive and closed-loop planning become apparent.

To illustrate the potential discrepancies between the theory and practice, we consider a real mine-planning exercise (actually based on the mine illustrated in Figure 2.1). Three planning methods were tested, namely,

- open-loop,
- reactive,
- perfect knowledge.

The latter policy cannot be carried out in practice since future prices are unknown. However, it does provide an upper bound on the achievable closed-loop performance.

Table 2.1 tabulates the average NPVs obtained over 6 tests using the expected forward price conditional on the current spot price (*i.e.* "present knowledge" or "open-loop" approach), the reactive approach and perfect knowledge of future price. In each test, 25 equally likely price realizations were generated using Monte Carlo simulations based on a log-normal mean reverting price model.

Table 2.1. Average NPVs for different planning methods

Test No.	p_0	$\bar{p}$	$\mathrm{NPV}_{Conditional}$	$\mathrm{NPV}_{Reactive}$	$\mathrm{NPV}_{Perfect}$
1	0.68	0.89	1622	1662	1685
2	0.89	0.68	1262	1287	1306
3	0.68	0.68	1180	1207	1231
4	0.25	0.45	303	358	370
5	0.35	0.35	185	246	256
6	0.45	0.25	164	197	206

The notation used in Table 2.1 is as follows:

- p_0 [\$/lb]: the current spot price at time 0;
- $\bar{p}$ [\$/lb]: the long-term equilibrium commodity price;
- $\mathrm{NPV}_{Conditional}$ [million \$]: the average NPV obtained using the expected forward price conditional on the current spot price;
- $\mathrm{NPV}_{Reactive}$ [million \$]: the average NPV obtained using the reactive approach;
- $\mathrm{NPV}_{Perfect}$ [million \$]: the average NPV obtained assuming that the future price is perfectly known prior to producing a mine plan.

Figures 2.10 and 2.11 show some plots of the results of Test 4 (where $p_0 = 0.25$ and $\bar{p} = 0.45$). Figure 2.10 shows the NPVs obtained using the three planning methods for each price scenario. Figures 2.11a and 2.11b show the histograms of the NPV for each price scenario using the "present knowledge" approach and the "reactive" approach, respectively.

It can be seen from the above results that, for this particular mine, the reactive approach yields approximately 10 % improvement over the open-loop approach. This strongly suggests that reactive planning offers considerable benefits over open-loop planning. On the other hand, perfect knowledge yields only a further 2.7 % improvement. Thus, for this particular mine the gap between reactive planning and planning with perfect knowledge is quite small. Thus one might anticipate that the improvements resulting from closed-loop planning could be small. However, one should be careful to draw general conclusions from this simple example. Indeed, as discussed in Section 2.4.2, there can be situations where closed-loop planning can offer substantial qualitative and quantitative advantages over reactive planning.

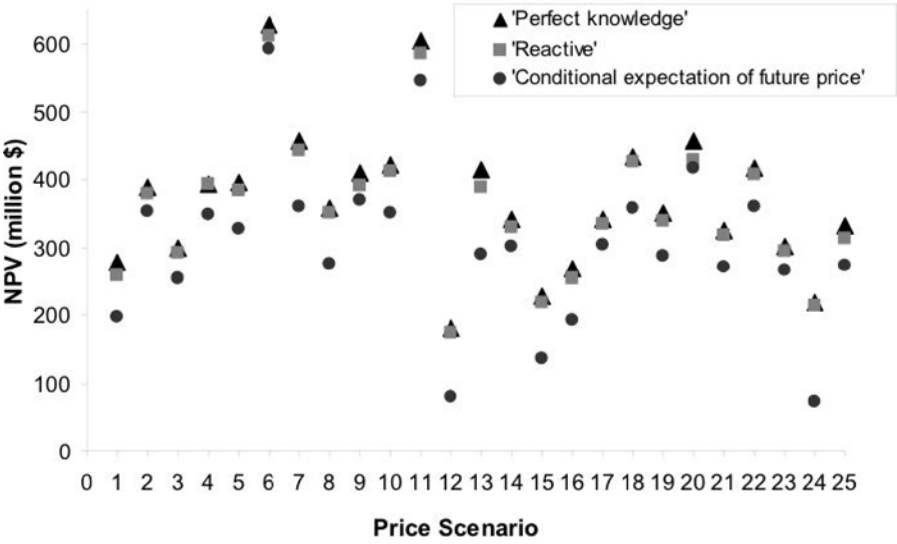


Fig. 2.10. NPV for each price scenario using three mining-project-evaluation methods

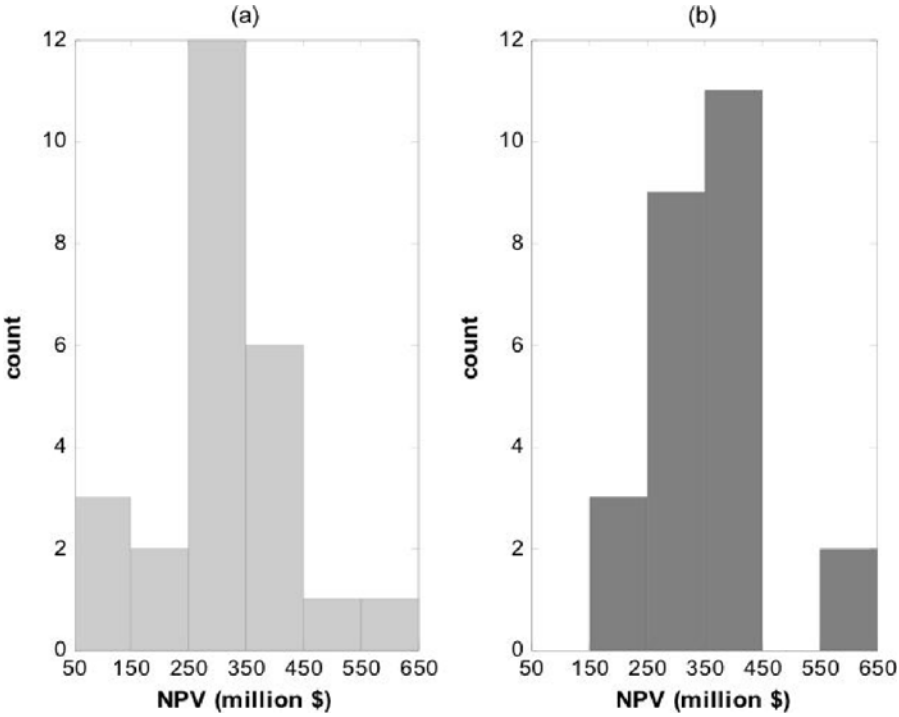


Fig. 2.11. Histograms of NPV for each price scenario using: (a) the method with expected price conditional on the current spot price and, (b) the reactive approach

2.4.4 Conclusions

This chapter has described the application of closed-loop stochastic optimal control to optimal mine planning. Key ideas that have emerged are:

1. This is a quintessential example of a complex optimization problem.
2. There is a substantial theory/practice gap associated with this problem and, indeed, to have the slightest hope of solving it, one needs to make major simplifications.
3. For certain ore geometries and problem parameters, there appears to be little practical gain to be achieved by using closed-loop planning rather than reactive planning.
4. For alternative ore geometries and/or different problem parameters, closed-loop planning can be beneficial.
5. Due to the enormous complexity of this problem, one needs to carefully choose a representative set of scenarios for the uncertain variables.
6. The cardinality of the scenarios must be quite small to be able to solve the closed-loop planning problem in practice.
7. Real mine-planning problems involve a host of complexities not touched on here.
8. The ideas presented here undoubtedly have relevance to other complex planning problems outside of the mining sphere.

Acknowledgements

The authors gratefully acknowledge the input of many others in writing this chapter. In particular, they wish to specifically mention Professor Richard H. Middleton and Mr. Brian F. Hennessy (University of Newcastle) and Dr. Peter M. Stone and Dr. Merab Menabde (BHP Billiton Melbourne Technology Centre).

References

[1] Appiah, P. B., Rosenman, M. A., and Sturgul, J. R. (1990). Introduction to Pareto optimal mine planning. *Journal of Mining and Geological Engineering* **8(4)**, 348–356.

[2] Appiah, P. B. and Sturgul, J. R. (1990). Pareto optimal stochastic mine planning. *International Journal of Surface Mining, Reclamation and Environment* **4(4)**, 181–195.

[3] Ataei, M. and Osanloo, M. (2003). Using a combination of genetic algorithm and the grid search method to determine optimum cutoff grades of multiple metal deposits. *International Journal of Surface Mining, Reclamation and Environment* **18(1)**, 60–78.

[4] Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, N. J.

[5] Beltratti, A., Consiglio, A., and Zenios, S. A. (1999). Scenario modeling for the management of international bond portfolios. *Annals of Operations Research* **85**(0), 227–247.

[6] Brazil, M., Lee, D. H., Van Leuven, M., Rubinstein, J. H., Thomas, D. A., and Wormald, N. C. (2003). Optimising declines in underground mines. *Mining Technology: Transactions of the Institution of Mining and Metallurgy, Section A* **112**, 164–170.

[7] Caccetta, L. and Hill, S. P. (2003). An application of branch and cut to open pit mine scheduling. *Journal of Global Optimization* **27**, 349–365.

[8] Clement, S. and Vagenas, N. (1994). Use of genetic algorithms in a mining problem. *International Journal of Surface Mining, Reclamation and Environment* **8**, 131–136.

[9] Darwen, P. J. (2001). Genetic algorithms and risk assessment to maximize NPV with robust open-pit scheduling. *Fourth Biennial Conference on Strategic Mine Planning*, pp. 29–34. Perth, Western Australia.

[10] Denby, B. and Schofield, D. (1994). Open-pit design and scheduling by use of genetic algorithms. *Transactions of the Inst. Min. Metall., Section A: Min. Industry* **103**, A21–A26.

[11] Denby, B. and Schofield, D. (1995). Inclusion of risk assessment in open pit design and scheduling. *Mining Technology: Transactions of the Institution of Mining and Metallurgy, Section A* **104**, A67–A71.

[12] Denby, B. and Schofield, D. (1995). The use of genetic algorithms in underground mine scheduling. *Proceedings of the 25th International Symposium Application of Computers Mineral Industries*, pp. 389–394. Brisbane, Australia.

[13] Denby, B., Schofield, D., and Surme, T. (1998). Genetic algorithms for flexible scheduling of open pit operations. *Proceedings of APCOM 27th International Symposium Mathematics in the Mineral Industries*. London, UK.

[14] Dupačová, J. (1996). Scenario-based stochastic programs: resistance with respect to sample. *Annals of Operations Research* **64**, 21–38.

[15] Dupačová, J., Consigli, G., and Wallace, S. W. (2000). Scenarios for multistage stochastic programs. *Annals of Operations Research* **100**, 25–53.

[16] Goodwin, G. C. and Payne, R. L. (1977). *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, New York.

[17] Goodwin, G. C., *et al.* (2006). Receding horizon control applied to optimal mine planning. *Automatica* **42**(8), 1337–1342.

[18] Høyland, K., Kaut, M., and Wallace, S. W. (2003). A heuristic for moment-matching scenario generation. *Computational Optimization and Applications* **24**(2–3), 169–185.

[19] Høyland, K. and Wallace, S. W. (2001). Generating scenario trees for multistage decision problems. *Management Science* **47**(11), 295–307.

[20] Keefer, D. L. (1994). Certainty equivalents for three-point discrete-distribution approximations. *Management Science* **40**(6), 760–773.

[21] Luenberger, D. G. (1984). *Linear and Nonlinear Programming*. Addison-Wesley, Reading, Massachussetts, 2nd edition.

[22] Metcalf, G. and Hassett, K. (1995). Investment under alternative return assumptions: comparing random walks and mean reversions. *Journal of Economic Dynamics and Control* **19**, 1471–1488.

[23] Miller, A. C. and Rice, T. R. (1983). Discrete approximations of probability distributions. *Management Science* **29**(3), 352–362.

[24] Mulvey, J. M. and Vladimirou, H. (1992). Stochastic network programming for financial planning problems. *Management Science* **38**(11), 1642–1664.

[25] Pflug, G. (2001). Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming, Ser. B* **89**, 251–257.

[26] Pflug, G. C. (1996). *Optimization of Stochastic Models: The Interface Between Simulation and Optimization*. Kluwer Academic Publishers, Boston.

[27] Pflug, G. C. (2001). Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programning, Serie B* **89**, 251–271.

[28] Rachev, S. T. (1991). *Probability Metrics and the Stability of Stochastic Models*. John Wiley and Sons, New York.

[29] Söderström, T. (1994). *Discrete-Time Stochastic Systems: Estimation and Control*. Prentice Hall, London.

[30] Stone, P., Froyland, G., Menabde, M., Law, B., Pasyar, R., and Monkhouse, P. (2004). Blasor–blended iron-ore mine planning optimisation at Yandi. *Orebody Modelling and Strategic Mine Planning", Proceedings of the International Symposium, AIMM*, pp. 285–288. Perth, Australia.

[31] Takriti, S., Birge, J., and Long, E. (1996). A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems* **11**(3), 1497–1508.

[32] Thomas, G. S. (1996). Optimization and scheduling of open pits via genetic algorithms and simulated annealing. *Proceedings of the 1st International Symposium on Balkema Publisher*, pp. 44–59. Rotterdam, The Netherlands.

[33] Vanderbei, R. J. (2004). *Linear Programming: Foundations and Extensions*. Springer, New York.

[34] Yu, L.-Y., Ji, X.-D., and Wang, S.-Y. (2003). Stochastic programming models in financial optimization: A survey. *AMO-Advanced Modeling and Optimization* **5**(1).

3

Energy Saving in a Copper Smelter by means of Model Predictive Control

Carlos Bordons[1], Manuel R. Arahal[1], Eduardo F. Camacho[1], and José M. Tejera[2]

[1] Departamento de Ingeniería de Sistemas y Automática. Universidad de Sevilla. Spain. {bordons, arahal, eduardo}@esi.us.es

[2] Atlantic Copper. Departamento de Servicios Generales, Electricidad e Instrumentación, Huelva, Spain. Jose-Maria-Tejera@fmi.com

Summary. This chapter presents an application of advanced control techniques on a copper smelter. The main objective of the control strategy is to keep the gas-circuit pressure at its desired value while achieving energy saving. Another objective of the control strategy is to reduce the risk of emissions. This chapter describes the design and implementation of the gas-circuit control.

The design phase includes an identification procedure. This is a multivariable process where a thorough analysis is needed for input-output matching. The identification phase included the determination of the best input-output pairing.

The control strategy has been devised taking into account not only system performance but also implementation issues. The designed controller runs on a distributed control system (DCS) using the available single-loop blocks and is able to perform a predictive control strategy with feedforward action using existing PID and lead-lag blocks.

3.1 Background

Advanced control techniques are normally used with two main objectives in mind: to reduce the manpower required to operate the system and to make it more efficient by allowing it to work closer to the optimal operating modes. model predictive control (MPC) is one of the most popular advanced control techniques. MPC formulation integrates optimal control, stochastic control, control of processes with dead time, multivariable control and future references when available. Another advantage of MPC is that because of the finite control horizon used, constraints and, in general nonlinear processes that are frequently found in industry, can be handled [3]. In many cases advanced control techniques are applied on processes already controlled by existing supervisory control and data acquisition (SCADA) systems. In these cases, the new control techniques have to be implemented on existing equipment with small requirements in terms of quality of data required for model tuning, computation capabilities and plant operators retraining. Furthermore, in these cases, the success of the control technique is not measured by its dynamic response but by economic gains obtained from an increase of the production and/or quality and savings in the use of resources such as energy, raw material or manpower.

This chapter describes an application where the ideas described above are illustrated. The main goal was to design a control scheme for the gas circuit of a copper smelter having in mind all the limitations described above. The process requires to move high masses of gas pumped by compressors with high energy consumptions. At the same time, because of environmental reasons the operating pressure has to be kept below the ambient pressure.

The current status and future trends in the automation of mineral and metal processing is reviewed in [5]. A survey of issues related to copper pyrometallurgical practice is presented in [7] and some maintenance aspects in copper smelters are mainly considered in [9]. Although there are not many reports about automatic control in copper smelters, some improvements related to control of copper converters such as the diagnosis system are presented in [1]. The relevance in terms of benefits obtained from the use of predictive controllers are discussed in this chapter and can be compared with those of [10].

3.1.1 Plant Description

Atlantic Copper Smelter facilities in Huelva (Spain) include a Flash Furnace and four Pierce–Smith converters, two of them blowing simultaneously. The three currents of gases generated in these processes are mixed in the mixing chamber (MC) and sent to three acid plants operating in parallel. Figure 3.1 shows an overview of the plant, whose annual production is around three hundred thousand tons of copper.

Fig. 3.1. General view of the copper smelter

The flash furnace and converters are controlled by means of a DCS (DCS); the acid plants are controlled by a SCADA-PLCs type system not connected to the DCS. Figure 3.2 shows the main components of the plant.

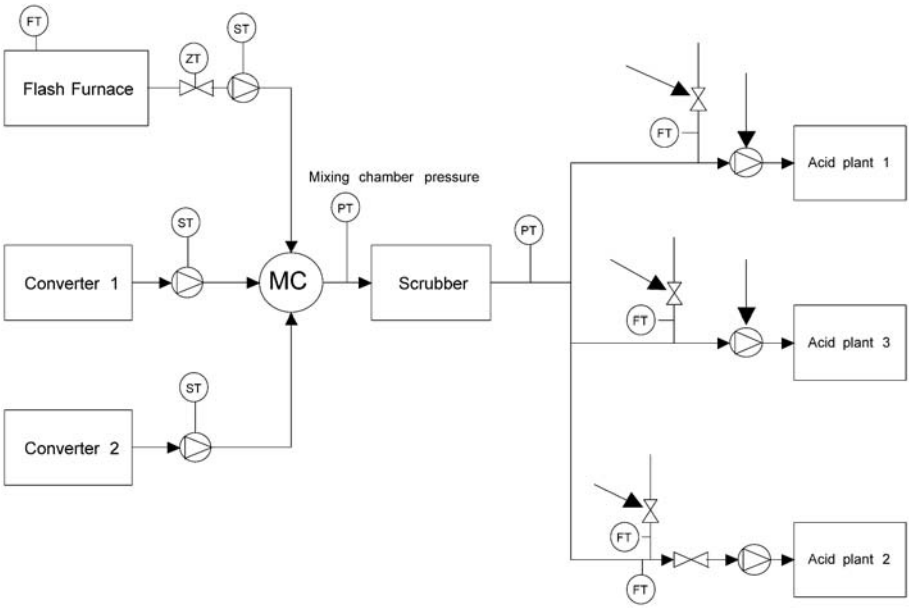


Fig. 3.2. Block diagram of the plant showing the main process variables and manipulated variables (thick arrows)

Taking into account that the converters operate on a batch mode, while those of the flash furnace and acid plants are continuous, extremely high disturbances both in flow and SO_2 concentration occur at the acid plants inlet due to the converters operating schedule. The existing control strategy, based on independent single-loop PID controllers, was not able to cope with those disturbances. Therefore, pressure-control performance was poor throughout the whole offgas system. Problems such as coupling between loops or limit cycles appear periodically, spreading these oscillations to the whole offgas system. This scenario brings about the following problems:

- The causes of the oscillations are extremely difficult to detect.
- No clock synchronization is possible, so there are considerable uncertainties when trying to measure cause–effect delays (uncertainties even as to deciding which is the cause and which is the effect).
- Extremely high operator supervisory effort, operating in manual about some 75 % of the time.
- No room for further process optimization.

The objective of the control strategy is to maintain the gas pressure in the mixing chamber at a desired value, always below ambient pressure in order to avoid gas losses to the atmosphere. The main problem is that the controller cannot act on the incoming gas (upstream),

since it comes form the plant's furnaces, but only on the gas flows that leave the chamber and enter into the sulfuric acid processing plants (downstream). This means that the upstream gas condition cannot be used as the manipulated variable but must be considered as a disturbance. These conditions must be measured in order to provide crucial information to the controller, which must anticipate its action to the change in upstream conditions. Since the influence of these measurable disturbances on the process output is high, the proposed control strategy is based on anticipative control.

The objective of this work was to increase pressure-control performance along the whole offgas system. Mixing-chamber pressure control error was chosen as the symptom for measuring purposes, due to its high sensitivity to oscillations in any process involved. The operator supervisory effort for keeping the pressure controlled should also be reduced. The variable selected for measuring this effort is the percentage of time the system is operating in automatic mode. The indicator that the control objective is fulfilled will be energy savings, measured as the total gas compressor power consumption.

3.2 Experiment Description

Models of the mixing-chamber pressure (y_{pcm}) have been obtained using data from tests carried out at the smelter. Two methods have been used: the reaction-curve fitting and least-squares identification to obtain first-order plus dead-time models. The models obtained from these approaches are compared later with data coming from new tests.

During the tests the plant is stabilized around an operation point and all input variables are kept as steady as possible in open loop except for the one that receives a step change in order to measure its influence on the pressure y_{pcm}. This is the only way to proceed since there are no sensors that would provide separately the contribution of each input on the output. This fact will not cause problems for the identification procedure as long as the experimentation is carried out with enough care to prevent the system receiving changes other than the step in the selected input. It will be shown later that this is easier said than done.

Proceeding in this way it is possible to obtain a submodel relating the change in the output variable y_{pcm} with changes in the input variable for each one of the possible inputs. Table 3.1 shows a list of the input variables and Figure 3.4 shows the global model obtained adding up the contributions of the different submodels.

The submodels are autoregressive but their contributions are not measurable. Therefore a difficulty arises when the global model is used. To work around this problem the submodels are used in simulation, using their own past outputs as inputs for future predictions that are added up to obtain $\hat{y}_{pcm}$, then the global error $e = y_{pcm} - \hat{y}_{pcm}$ is used to correct the future predictions.

3.2.1 Data

A good part of the modeling effort was devoted to perform the test at the plant that would supply data for identification. It has been pointed out previously that the test was carried out in open-loop mode in order to facilitate the identification. Also, the test had to be performed during normal operation of the plant in order to provide significative data. With great care some data was finally gathered during periods in which open-loop operation was possible with just one variable changing (significatively) at a time. As an example, Figure 3.3 shows

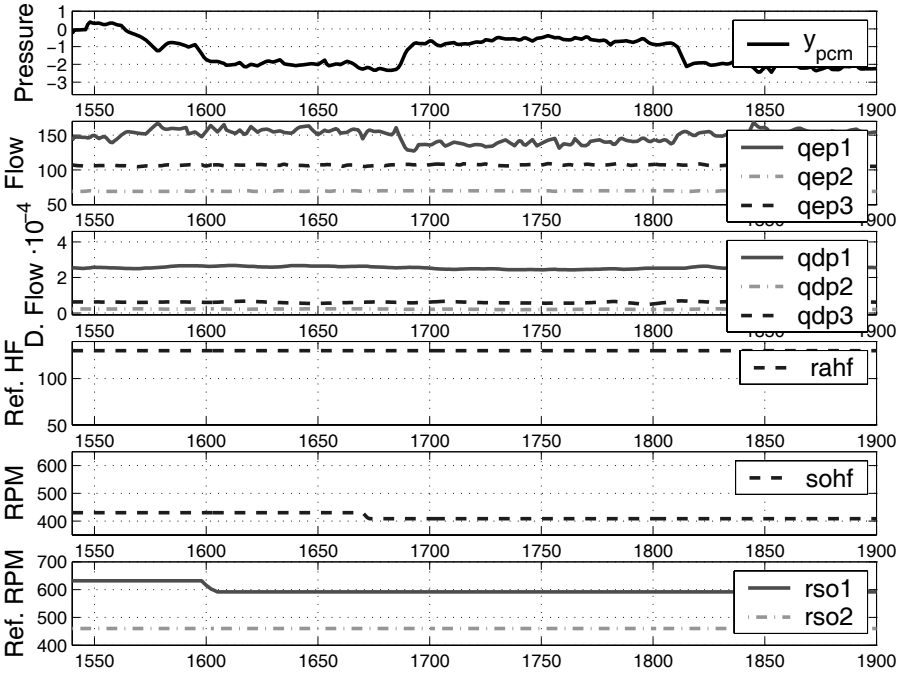


Fig. 3.3. Evolution of the pressure in the mixing chamber due to changes in the input variables during a typical test

the evolution of the pressure in the mixing chamber due to changes in the input variables during a typical test.

Filtering and removal of outliers was also necessary and did take more effort than previously thought.

3.2.2 Variables

The variable of interest is the pressure in the mixing chamber y_{pcm}, and its evolution is influenced by others that are divided in two groups: control signals and disturbances. In Table 3.1 a brief description of the considered variables is given.

3.2.3 Identification

According to the block diagram in Figure 3.4, the output is viewed as the sum of contributions v^i due to input signals u^i with $i \in \{1, 2, 3, 4, 5, 6\}$ and of contributions w^j due to measurable disturbances p^j, with $j \in \{1, 2, 4, 5\}$. For each input signal u^i and disturbance p^j a submodel will be calculated relating the input variable with its contribution to the output y_{pcm}.

The models considered are discrete-time transfer functions in the $\mathcal{Z}$-transform. The transfer function that relates $V^i(z)$ with $U^i(z)$ will be denoted as $G^i(z) = P^i(z)/Q^i(z)$. Similarly, $H^j(z) = R^j(z)/S^j(z)$ will stand for the transfer function relating $W^j(z)$ with $P^j(z)$.

Table 3.1. List of variables considered for modeling

Name	Code	Description	Units	Type
y_{pcm}	012 - y	Pressure in mixing chamber	mbar	Output
qep_1	009 - u^1	Flow to plant 1	kN m^3 h^{-1}	Control
qep_2	015 - u^2	Flow to plant 2	kN m^3 h^{-1}	Control
qep_3	013 - u^3	Flow to plant 3	kN m^3 h^{-1}	Control
qdp_1	059 - u^4	Dilution flow to plant 1	N m^3 h^{-1}	Control
qdp_2	064 - u^5	Dilution flow to plant 2	N m^3 h^{-1}	Control
qdp_3	065 - u^6	Dilution flow to plant 3	N m^3 h^{-1}	Control
ra_{hf}	032 - p^1	Reference for HF feeding	Tons h^{-1}	Measurable disturbance
vs_{hf}	004 - p^2	Flow control valve for HF	%	Measurable disturbance
so_{hf}	030 - p^3	Fan speed in HF	rpm	Measurable disturbance
rso_1	018 - p^4	Reference for fan speed line 1	rpm	Measurable disturbance
rso_2	020 - p^5	Reference for fan speed line 2	rpm	Measurable disturbance

To keep things simple first-order plus dead-time models are considered. Then $G^i(z) = P^i(z)/Q^i(z) = b^i z^{-r^i}/(1 - a^i z^{-1})$ and $H^j(z) = R^j(z)/S^j(z) = d^j z^{-s^j}/(1 - c^j z^{-1})$, being a^i, b^i, c^j and d^j parameters to be estimated using data from the tests.

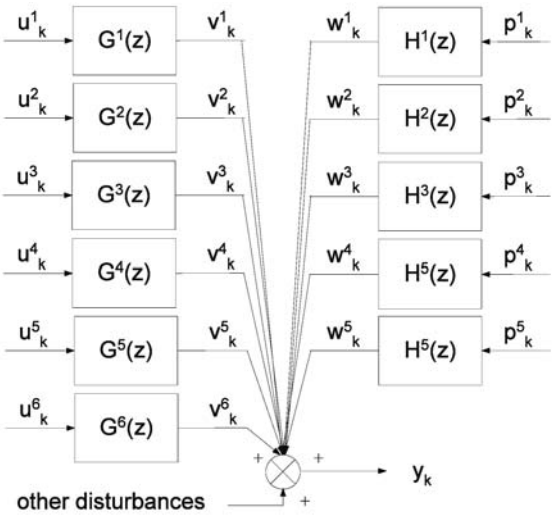


Fig. 3.4. Block diagram showing the different submodels

It is convenient to use an incremental model to calculate the predicted output $\hat{y}$ using past values of inputs and disturbances. To derive this model consider that

$$y(k) = y(k-1) + \triangle y(k) + e(k) \tag{3.1}$$

where $e(k)$ is the sum of not measurable disturbances and noise and $\triangle = (1 - z^{-1})$ is the

temporal increment operator. On the other hand, from the above definitions it is apparent that

$$\triangle y(k) = \sum_i \triangle v^i(k) + \sum_j \triangle w^j(k) \tag{3.2}$$

3.2.4 Parameter Estimation

During any of the tests, only an input variable is changed. For this reason the evolution observed in the output is due to this variable and to $e(k)$. This will allow the estimation of some parameters in the following way:

Suppose that in a experiment only u^* is changed, then

$$\triangle y(k) = \triangle v^*(k) + e(k) \tag{3.3}$$

If $e(k)$ has a random variation with zero mean, a least squares procedure can be used in

$$\triangle y(k) = a^* \triangle y(k-1) + b^* \triangle u^*(k - r^*) \tag{3.4}$$

to estimate the parameters a^*, b^* and r^* that produce a better adjustment of the model to the observed output. This procedure has been used with data from the tests producing the results presented in Table 3.2.

Table 3.2. Estimated parameters of the different submodels

Input Signal	Model	a	b	d
qep_1	01	0.5000	–0.0300000	0
qep_2	02	0.4100	–0.0300000	0
qep_3	03	0.6036	–0.0041430	0
qdp_1	04	0.5743	0.0000300	0
qdp_2	05	0.6519	0.0000448	0
qdp_3	06	0.4578	0.0000578	5
ra_{hf}	07	0.3520	0.0001680	4
vs_{hf}	08	0.6293	0.0317000	2
so_{hf}	09	0.3131	0.0129000	1
rso_1	10	0.4890	0.0044000	0
rso_2	11	0.3586	0.0114000	0

Figure 3.5 shows the output of the plant y_{pcm} and the one-step-ahead prediction calculated by the model y_m in a typical test.

In order to design the master controller the transfer function that relates the pressure in the mixing chamber and the flow to plants 1 and 2 has to be identified. This transfer function has been obtained as $M(z) = KG^1(z) + (1-K)G^2(z)$ since the flow ratio to plants 1 and 2 is fixed at the operational value of K_s, whose usual value is $K_s = 0.76$.

3.2.5 Validation of Submodels

As a first test the one-step-ahead prediction error will be calculated with data used in the identification of each submodel. The error of the prediction is calculated as

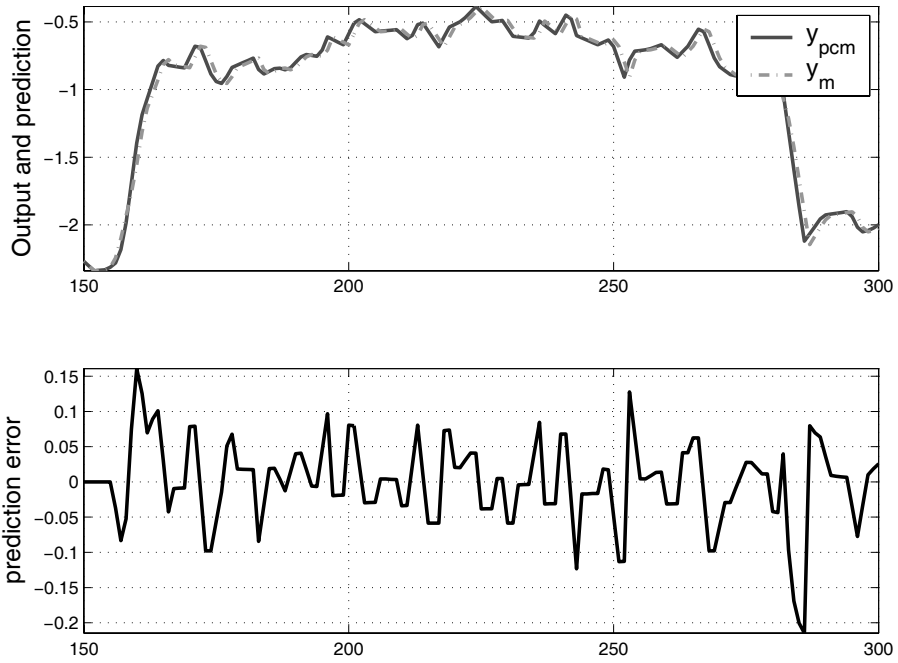


Fig. 3.5. The upper part shows the evolution of the output of the plant y_{pcm} and the one-step-ahead prediction calculated by the model y_m during a typical test. The lower part is the prediction error.

$$\hat{y}(k+1 \mid k) = y(k) + \triangle\hat{y}(k+1 \mid k) = y(k) + \sum_i \triangle v^i(k+1) + \sum_j \triangle w^j(k+1) \quad (3.5)$$

where $y(k)$ is the output (y_{pcm}) measured at sampling time k and $\hat{y}(k+1 \mid k)$ is the prediction done at that same instant for the output at the next sampling time $k+1$.

Please note that the contributions of each submodel v^i and w^j are not measurable and this implies that in expression (3.5) the values generated by the submodels have to be used. In this situation the submodels are not predicting but simulating their outputs. In spite of this, the global model is a prediction model because the value of $y(k)$ is used to predict $y(k+1)$.

It has to be taken into account that for each test a global model has been used. By global model we mean that the model contains all submodels.

3.2.6 Control

The models obtained in the previous section were used to develop a model-based control strategy. The following considerations were taken into account in order to implement the selected strategy:

- It should be easy to operate and to maintain (avoiding long training)
- It should be implemented using the existing control hardware, not requiring major investment in additional dedicated multivariable control systems.

- It should be aimed at reducing the energy used.

Therefore, the designed control strategy runs on the existing DCS using standard programming blocks, mainly PID single-loop regulators, lead-lag blocks for feedforward actions, and blocks for simple mathematic calculations. The operating interface is very similar to the existing one and the training period has been minimal.

As was mentioned above, the objective of the control strategy is to maintain the gas pressure in the mixing chamber at a desired value, always under ambient pressure in order to avoid gas losses to the atmosphere. In order to do so, set points of the downstream flow controller are manipulated by the master controller. A good operation strategy can drastically reduce energy expenses.

Description of the Control Strategy

The devised strategy is a two-level hierarchical control based on MPC [3, 6] and implemented on the existing DCS using available single-loop routines and simple calculation modules. The main loop is the mixing-chamber pressure control, that is performed as a generalized predictive controller (GPC) [4] with feedforward action and the low-level control is composed of slave controllers that are PIDs.

Since the upstream flow cannot be modified, the manipulated variable of the main loop is the downstream flow, which is the gas flow sent to the acid plants. This flow must be shared among the three acid plants, in a predetermined ratio. The main control loop (master) computes this total flow to track the pressure set point, which is usually set into the band -4 to -2 mbar. Notice that this value must always be kept below atmosphere pressure since higher values could result in leakage of gas with a great concentration of SO_2.

The gas circuit is composed of several compressors that must pump a large volume of gas, demanding high energy consumption. If the system is not well controlled, fast and underdamped control actions can make the energy demand reach undesirable. This happens because in that case the pressure set point must be kept far from the limit in order to avoid undesirable values and this is achieved by augmenting the depression using the compressors. If the control actions are able to anticipate, the energy demand can be drastically reduced. This can be done by the use of a predictive strategy that takes the disturbances into account.

The control strategy is composed of a master controller (predictive controller whose set point is the desired pressure in the mixing chamber) with feedfoward action acting on the slave controllers that regulate the gas flows to the acid plants. Figure 3.6 shows a block diagram of the control strategy. The blocks labelled as FFi correspond to the feedforward (anticipative) actions, as described in Section 3.2.6.

Apart from the normal operation, there also exists a safety mechanism that tracks the pressure at the electrofilter located downstream of the scrubber. This pressure must be negative, but cannot be too high (in absolute value) since a high negative pressure can seriously damage the filter. In order to avoid this situation, an override action is added to the proposed strategy. Another PI is used to control this variable, which has priority over the pressure in the MC controller. This is a fast action that produces a quick response and it only acts in emergency situations.

The gas coming from the chamber is diluted with fresh air before it enters acid plants 1 and 2. This is needed to keep the SO_2 content of the inlet gas at each acid plant below the designed maximum value. This is done by local dilution controllers that manipulate the flow of fresh air into the incoming stream to the acid plants. The changes in the gas flow produced by these local controllers act as disturbances to the pressure in the mixing chamber, since they

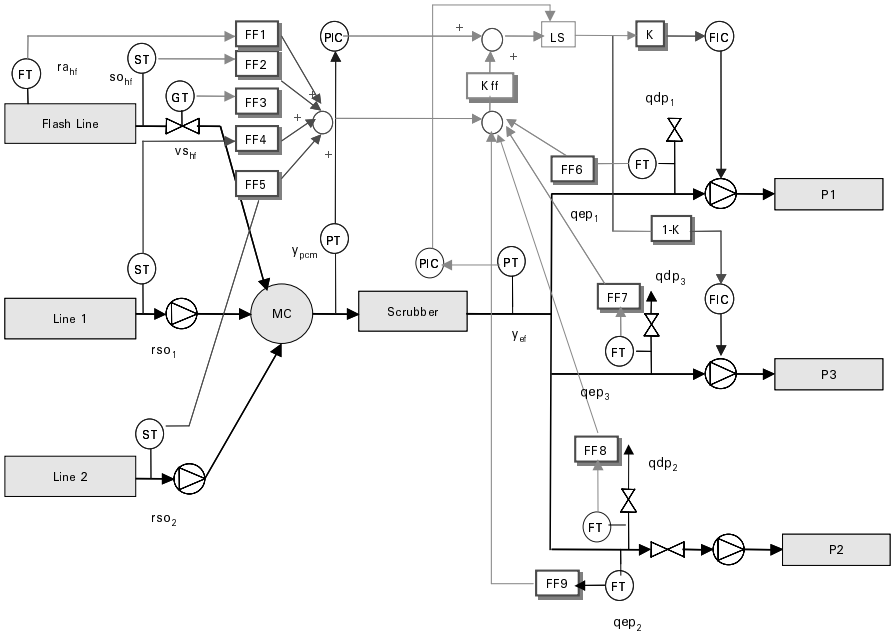


Fig. 3.6. Block diagram showing the proposed control strategy

are in fact manipulating the flow downstream of the chamber and, therefore, disturbing the pressure.

The following sections describe the two main parts of the control strategy: master controller and feedforward blocks.

Master-controller Development

MPC has developed considerably over the last decades, both within the research control community and in industry. This success can be attributed to the fact that MPC is, perhaps, the most general way of posing the process-control problem in the time domain. Although there are many applications of MPC in industry [8], this technique has not yet reached the popularity that its potential would suggest. One reason for this is that its implementation requires some mathematical complexities that are not a problem in general for the research control community, where mathematical packages are normally fully available, but that represent a drawback for the use of the technique by control engineers in practice. This work shows how the MPC concepts can be applied using the tools that are fully available in any commercial control system.

In order to develop a predictive controller with low computational requirements, a first-order plus dead-time model of the plant is used. This model is widely used in the process industry since it can easily describe the dynamics of most plants by means of a static gain, a time constant and a pure dead time:

$$G(s) = \frac{K}{1+\tau s} e^{-\tau_d s}$$

When the dead time τ_d is an integer multiple of the sampling time T ($\tau_d = dT$), the corresponding discrete transfer function has the form

$$G(z^{-1}) = \frac{bz^{-1}}{1 - az^{-1}} z^{-d} \tag{3.6}$$

where discrete parameters a, b and d can easily be derived from the continuous parameters by discretization of the continuous transfer function, resulting in the following expressions:

$$a = e^{-\frac{T}{\tau}} \qquad b = K(1 - a) \qquad d = \frac{\tau_d}{T}$$

Using this discrete model, a predictive controller can be derived. One of the most popular controllers of this family is GPC [4], which results in a linear control law that is easy to implement once the controller parameters are known. The derivation of the GPC parameters requires, however, some mathematical complexities such as recursively solving a Diophantine equation, forming some matrices and then solving a set of linear equations. Although this is not a problem for people in the research control community where mathematical packages are normally available, it may be discouraging for practitioners used to much simpler ways of implementing and tuning controllers. MPC can also be very difficult to implement using the tools that are normally available in a commercial DCS. In order to overcome this problem, a simple control law, similar to that of a simple PI, can be derived for first-order plus dead-time models. This has been done for the mixing-chamber pressure, following the method presented in [2] which is briefly described below.

The GPC algorithm consists of applying a control sequence that minimizes a multistage cost function of the form:

$$J(N) = \sum_{j=d+1}^{d+N} [\hat{y}(t + j \mid t) - w(t + j)]^2 + \sum_{j=1}^{N} \lambda [\triangle u(t + j - 1)]^2 \tag{3.7}$$

where $y(t + j \mid t)$ are the predictions of the plant output (controlled variable) along the horizon, $w(t + j)$ are the future values of the reference trajectory (future set-point values) and $\triangle u(t + j - 1)$ are the control action (manipulated variable) increments.

The computation of the control law involves an optimization performed online at every sampling instant along the future horizon. Notice that this is an ambitious objective and therefore this controller needs a computational effort higher than that of a classical PID. However, when constraints are not considered, a simple analytical solution can be found.

The cost function is minimized at every sampling time, giving rise to a control action that is given by (see [4]):

$$\mathbf{u} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f})$$

where $\mathbf{u}$ is the vector of future control actions (whose first component is $\triangle u(t)$), $\mathbf{G}$ is a matrix containing the step response coefficients, $\mathbf{w}$ is the vector of the future reference trajectory, $\mathbf{f}$ is the free response of the plant and $\mathbf{I}$ is the identity matrix. Since $\mathbf{G}$ and $\mathbf{f}$ depend solely on the plant parameters, the control law can be calculated in such a way that it can be computed once the plant is known and the control weighting factor λ is chosen.

As shown in [2], the control increment $\triangle u(t)$ can be rewritten as

$$\triangle u(t) = l_{y1} \hat{y}(t + d \mid t) + l_{y2} \hat{y}(t + d - 1 \mid t) + l_r r(t) \tag{3.8}$$

The coefficients l_{y1}, l_{y2} and l_{r1} do not depend on the dead time d and are functions of a, b, and λ. If the GPC is designed considering the plant to have a unit static gain, the coefficients

in Equation (3.8) will only depend on λ (which is supposed to be fixed) and on the pole of the plant that will change for the adaptive control case. Notice that by doing this, a normalized weighting factor λ is used and it should be corrected for systems with different static gains. The controller coefficients can be obtained by interpolating in a set of previously computed values. Notice that this can be accomplished in this case because the controller coefficients only depend on one parameter.

The predictor needed in the algorithm to calculate $\hat{y}(t+d \mid t)$, $\hat{y}(t+d-1 \mid t)$ is obtained by applying Equation (3.9) sequentially for $j = 1-d \cdots 0$. Notice that it basically consists of a model of the plant that is projected towards the future with the values of past inputs and outputs, and it only requires straightforward computation.

$$\begin{aligned}\hat{y}(t+d+j \mid t) = (1+a)\hat{y}(t+d+j-1 \mid t) - a\hat{y}(t+d+j-2 \mid t) \\ + b \triangle u(t+j-1)\end{aligned} \tag{3.9}$$

In the case of $\lambda = 0.8$ and for a control horizon of 15, the controller coefficients are given by:

$$\begin{aligned} l_{y1} &= -0.845 - 0.564\frac{a}{1.05-a} \\ l_{y2} &= 0.128 + 0.459\frac{a}{1.045-a} \\ l_r &= -l_{y1} - l_{y2} \end{aligned} \tag{3.10}$$

A general expression for these parameters when λ changes can be found in [3].

Notice that the control law (3.8) is similar to that provided by a simple proportional integral (PI) controller:

$$u(t) = u(t-1) + q_0 e(t) + q_1 e(t-1)$$

Taking into account that $l_r = -l_{y1} - l_{y2}$, Equation (3.8) can be rewritten as:

$$u(t) = u(t-1) + l_{y1}(\hat{y}(t+d \mid t) - r(t)) + l_{y2}(\hat{y}(t+d-1 \mid t) - r(t))$$

which corresponds to a PI with $q_0 = l_{y1}$ and $q_1 = l_{y2}$ that is fed with the "future" error $\hat{y}(t+d \mid t) - r(t)$ instead of the current one, $e(t)$. This means that a MPC can be implemented for this plant using the existing and roughly-tested PI routines available in existing control system. This also facilitates its use by plant operators.

Notice that this can be programmed in any DCS using the basic "loop" function and a simple calculation module that implements the prediction as given by Equation (3.9).

Then, the model obtained in the previous section is used to compute the GPC coefficients. The identification procedure provides the following values for the discrete model that is used by the GPC:

$$G(z) = \frac{-0.017052z^{-1}}{1 - 0.42z^{-1}}$$

using Equation (3.10), the controller coefficients are given by:

$$\begin{aligned} l_{y1} &= -1.2209 \\ l_{y2} &= 0.4364 \\ l_r &= 0.7845 \end{aligned}$$

Notice that, since the static gain of the plant is not unitary, these values must be divided by the static gain to get the real values to be used in the control law.

As mentioned in Section 3.2.6 the output of the master controller is split between two signals, which are used as set points to the flow controllers to the acid plants. The split ratio K_s is computed based on the pumping capacity of the feeding circuits to the plants.

Feedforward Action

There are many disturbances acting on the plant that can be measured and therefore they can be integrated in the control strategy, being added to the control action computed by the master controller, as shown in Figure 3.6. After the identification of all measurable disturbances in Section 3.2, the variables that are measured to compute the feedfoward actions are:

- Flash furnace: this is the main source of disturbance affecting the pressure. Three variables related to this furnace are considered: the flow of raw material, the gas-blower speed and the draft-valve position.
- Exhaust gas fans of lines 1 and 2: The converters batch operating schedule give rise to big changes in the gas flow that enters the mixing-chamber
- Dilution air flow of plants 1, 2 and 3: the local controllers that control the composition of the inlet gas of the acid plants influence the gas flow and consequently the pressure in the MC.
- Gas flow inlet plant 2: this flow is not controlled by the master (as happens with plant 1 and 3).

The computation of the feedforward action is independent of the master controller, and is calculated with the objective of canceling the disturbance effect. In this way, if the transfer function of the disturbance d is G_d and that of the plant is G, a block with transfer function G_{ff} can be included such that the effect of the new block on the plant output, $G_{ff}(s)G(s)D(s)$, cancels the effect of the disturbance $G_d(s)D(s)$

Therefore

$$G_{ff}(s) = -\frac{G_d(s)}{G(s)} \tag{3.11}$$

Under the assumption of linearity, the effect of each disturbance can be addressed independently, so nine feedforward blocks can be calculated and grouped into a compound signal that is added to the master's output (see Figure 3.6). The parameters of the nine feedforward blocks are computed using the models obtained in Section 3.2.

Implementation

The devised control strategy runs on a Foxboro I/A Distributed Control System using available routines. This makes production and maintenance personnel training almost unnecessary, since they are used to managing PIDs, lead-lags and algebraic blocks. They can easily accept the new control strategy since there are no abrupt changes in the way of operating the plant.

The coefficients of the master PI controller are obtained by translating the discrete values of q_0 and q_1 to the continuous ones (gain and integral time) with a sampling time of 2 s using the Tustin approach.

Since first-order plus delay models are used to describe the plant and the disturbances, and using Equation (3.11) the structure of each of the nine feedforward blocks is given by:

$$G_{ff}(s) = -\frac{K_d(1+\tau s)}{K(1+\tau_d s)} e^{-(dt_d - dt)s} = K\frac{s+c}{s+p} e^{-ds} \tag{3.12}$$

where K, τ and dt are the parameters of the plant and K_d, τ_d and dt_d those of the disturbance. It is seen that each feedforward loop gives rise to a transfer function with a static gain K, a zero c, a pole p and a pure delay d, which can be implemented using a lead-lag with a pure delay.

The parameters of the nine feedforward blocks were obtained based on the models developed in Section 3.2 and an onsite fine tuning. They had to be translated to the lead-lag block of the DCS, which has the form:

$$y(t) + \text{LTime}\frac{dy(t)}{dt} = x(t) + \text{LTime} \times \text{LGain}\frac{dx(t)}{dt} \tag{3.13}$$

with $x(t)$ the input signal and $y(t)$ the output signal. Parameter LTime is equivalent to the pole and LTime $\times$ LGain is equivalent to the zero of Equation (3.12). The parameters are shown in Table 3.3 (times are in minutes).

Table 3.3. Parameters of the lead-lag blocks

Block	Input	Lead-lag gain	Lead-lag time	Gain	Delay
1	Flash furnace raw material flow	3.2400	0.00319	–0.0035	0.1333
2	Gas-blower speed flash furnace	2.3380	0.0529	0.5703	0.53
3	Draft-valve flash furnace	1.8535	0.0720	–1.1402	0.0666
4	Fan-speed set-point Line 1	1.3470	0.1062	–0.1584	0
5	Fan-speed set-point Line 2	1.0350	0.1437	–0.2576	0
6	Dilution air-flow plant 1	2.1285	0.0601	–0.0009	0
7	Dilution air-flow plant 3	2.7110	0.0427	–0.0014	0.1666
8	Dilution air-flow plant 2	1.7405	0.0779	–0.0017	0
9	Gas flow inlet plant 2	2.9500	0.0374	0.6780	0

On the other hand, slave PIDs were experimentally tuned using the reaction curve method. Once the starting values were taken from the Ziegler–Nichols rules, fine tuning was performed with the help of operator experience.

The operator's display is very simple, see Figure 3.7. The main screen shows the most important disturbances acting on the system, the effect of the feedforward action, the master controller, the slave controllers and the dilution controllers. It also shows the override safety controller. The feedforward parameters of Table 3.3 can be updated by the maintenance operators through the graphical interface. Production operators are allowed to cancel or resume any feedforward block if necessary, for instance in case of instrument failure or misreading.

3.3 Results

This section shows the results of applying the proposed strategy to the plant. First, some operation results are shown and later the economic profit is quantified.

3.3.1 Operational Results

As has been depicted throughout the chapter, the main control objective is to keep the pressure in the mixing chamber below a threshold value in spite of the several disturbances that may

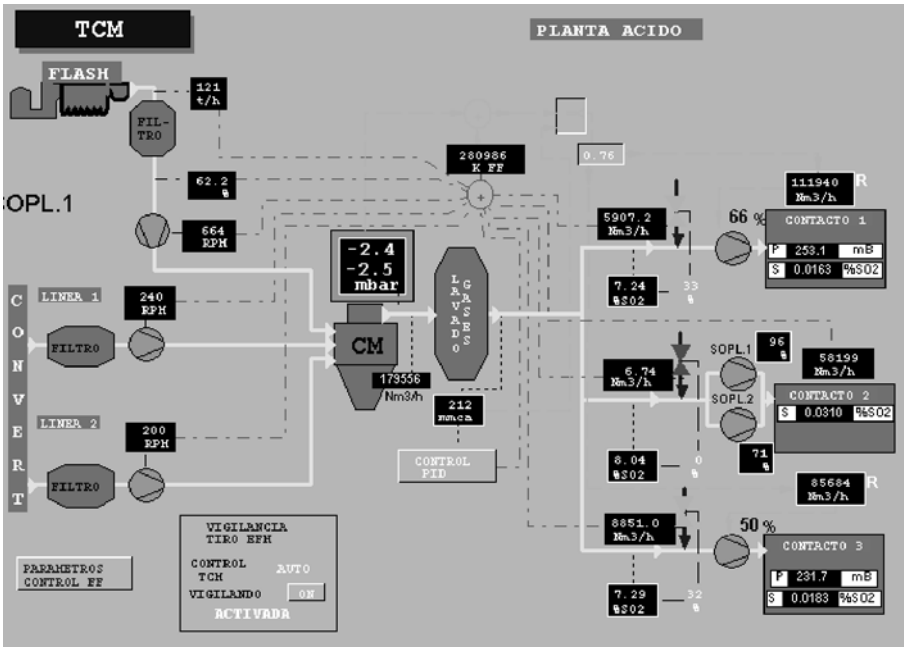


Fig. 3.7. Operator's display

appear in the process. The following figures describe the evolution of the controlled variable as well as the control action and main disturbances in several situations.

In order to show the disturbance rejection capabilities of the control strategy, Figure 3.8 shows how both the predictive controller and the feedforward actions make the pressure track the desired set point during normal operation, when several disturbances appear due to the changes in the operating regimes of the flash furnace and converters (lines 1 and 2) and the changes in the dilution of the acid plants. The computed control action is the set point to the slave controllers that track the desired value of the gas flow to plants 1 and 3. As described, the value computed by the pressure master controller is split between both flow slave controller using a specified ratio K_s that takes into account the capacity of each gas circuit. The figure shows the feedforward action (thin line) as well as the total control action.

One of the main sources of disturbance is the flash furnace, whose operating point can change very rapidly, with step changes in raw material feeding of 10 tons/h. This produces an abrupt change in the gases generated in the furnace, which clearly disturbs the pressure in the mixing chamber. This is reduced by the feedforward action as shown in Figure 3.9, which shows pressure (upper graph), total control action including feedforward (middle graph) and the set point of the feed controller, which is the disturbance (lower graph).

A similar source of disturbance is the load of the converters in the line upstream of the chamber. Changes in the state of the converters (blowing or not blowing) give rise to changes in the outlet gas flow of the converter, which can be measured and, consequently, an anticipative action can be taken (see Figure 3.10 for changes in line 1).

Another example of the feedforward action is given in Figure 3.11, when changes in two variables (gas flow in line 2 and air flow in the dilution loop of plant 1) appear during normal

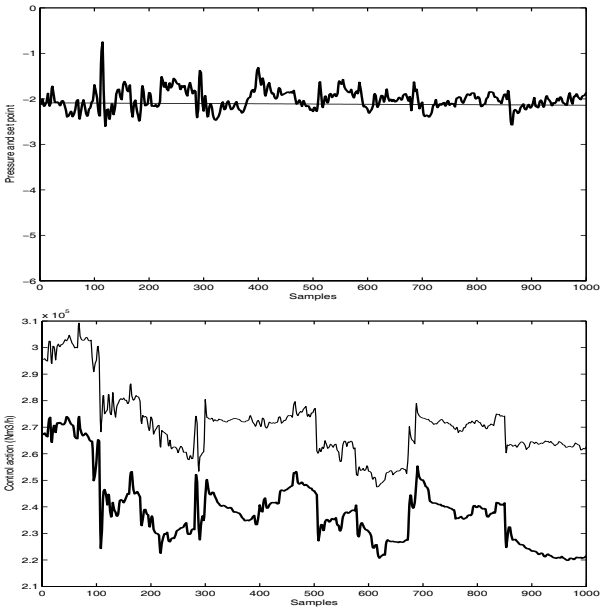


Fig. 3.8. Disturbance rejection. Pressure and control action (the thin line is the feedforward action).

operation. The flow of gas in line 2 changes during all the experiment, while the dilution air disturbance takes place between samples 900 and 1200.

The changes in the total gas flow produced by the dilution controllers are also disturbances to the pressure in the mixing chamber. Figure 3.12 shows how the feedforward action is able to reject the disturbance in the dilution flow of plant 3, which happens around sample 470.

One important element of the control strategy is the electrofilter safety control. As was mentioned before, the electrofilter pressure controller overrides the master controller in order to avoid low pressures. Figure 3.13 shows the behavior of this safety controller when pressure reaches the operator threshold (-47.5 mbar) at sample number 337 (lower graph). At this moment, the electrofilter controller overrides the master controller in order to relieve the pressure off the filter, until this pressure reaches a safe value (sample 927). The figure shows that this situation is repeated at sample 1347 until the end of the experiment. Notice that when the safety controller is ON, the master controller for the mixing chamber is OFF and therefore that pressure is out of control and might reach undesirable values (see upper graph). This only happens in particular situations.

3.3.2 Energy Savings

The economic profit resulting from the implementation of the control strategy comes from the fact that the system is now operating in automatic mode most of the time. This is achieved thanks to a good pressure control in the mixing chamber that takes all disturbances into account and avoids operators setting the loops to manual (mainly the dilution controllers). Table 3.4 depicts the situation prior to the implementation of the control strategy and shows that

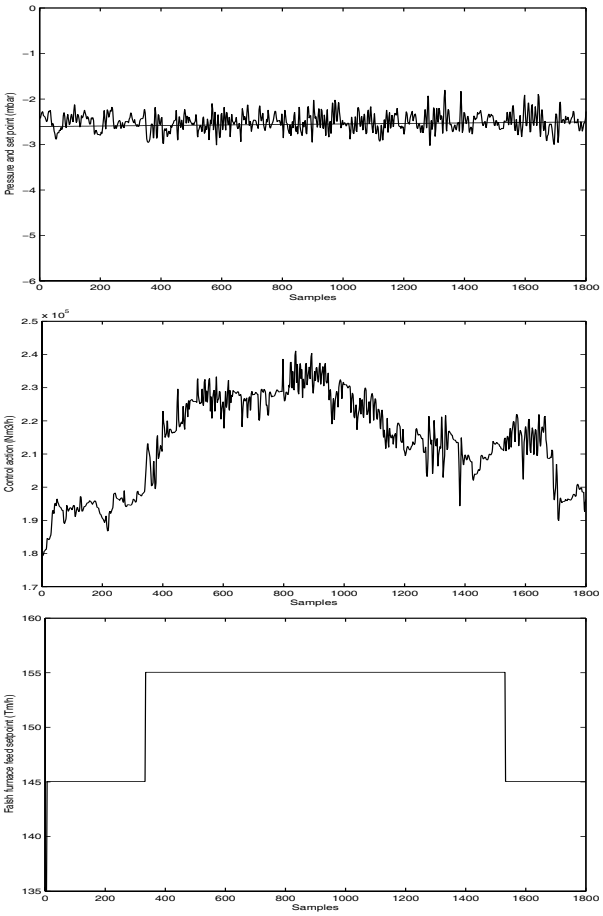


Fig. 3.9. Disturbance rejection. Changes in flash-furnace feeding.

most of the time the dilution controllers were switched to manual mode in order to avoid disturbances to the main pressure controller. When in manual mode the dilution valves are normally wide open and consequently consuming more energy than necessary. After a reliable control strategy is implemented, both pressure and dilution control can work in automatic mode, allowing the dilution blowers to consume only the necessary energy to fulfil the desired objective, as shown in Table 3.5

Another source of energy saving comes from the strategy that splits the signal generated by the master controller and is sent to the slave flow controllers of the acid plants. The three acid plants are different in terms of maximum gas flowrate and in maximum gas SO_2 concentration. Besides, they have different systems for flowrate regulation: Plant 2 by means of inlet guide vanes; and plants 1 and 3 by means of main compressor motor-speed regulation (VFD). For these reasons, it is not easy to decide on the best solution for splitting the total flowrate into the three acid plants. Due to its regulation system, plant 2 is left open and the rest is split between plants 1 and 3. A model was developed in order to evaluate the total energy

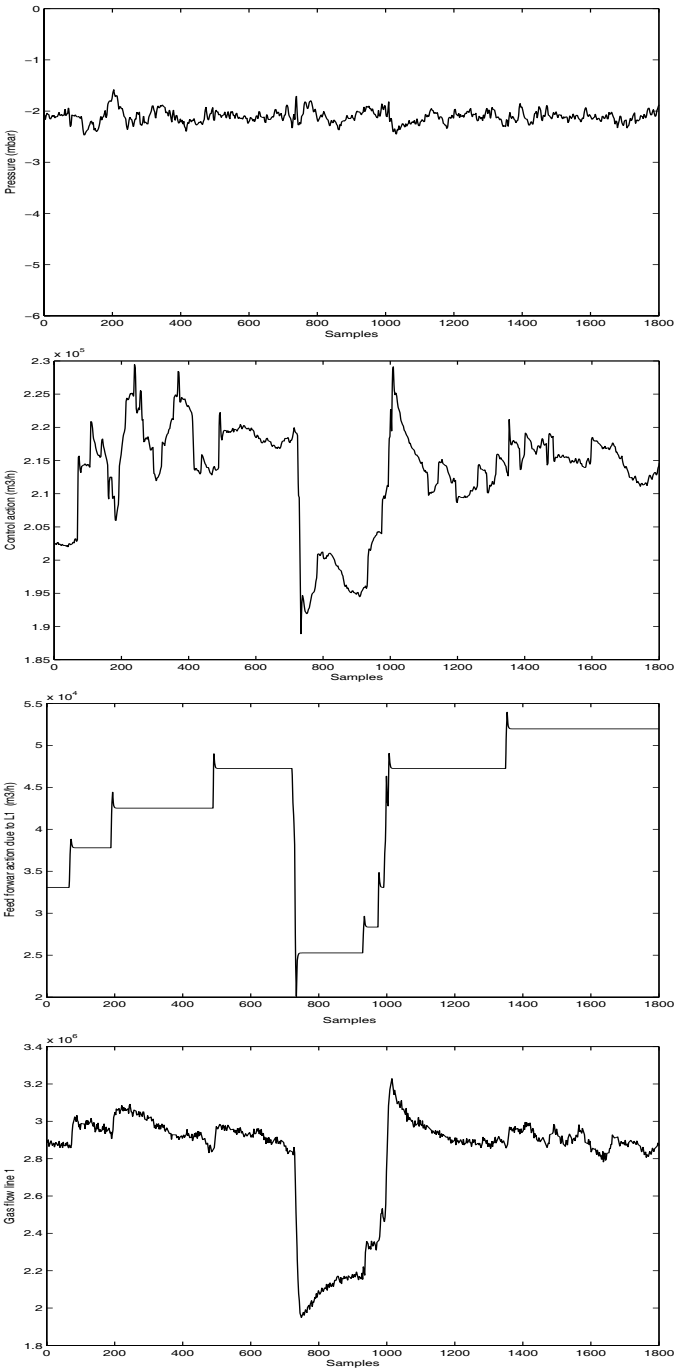


Fig. 3.10. Disturbance rejection. Changes in Line 1 feeding.

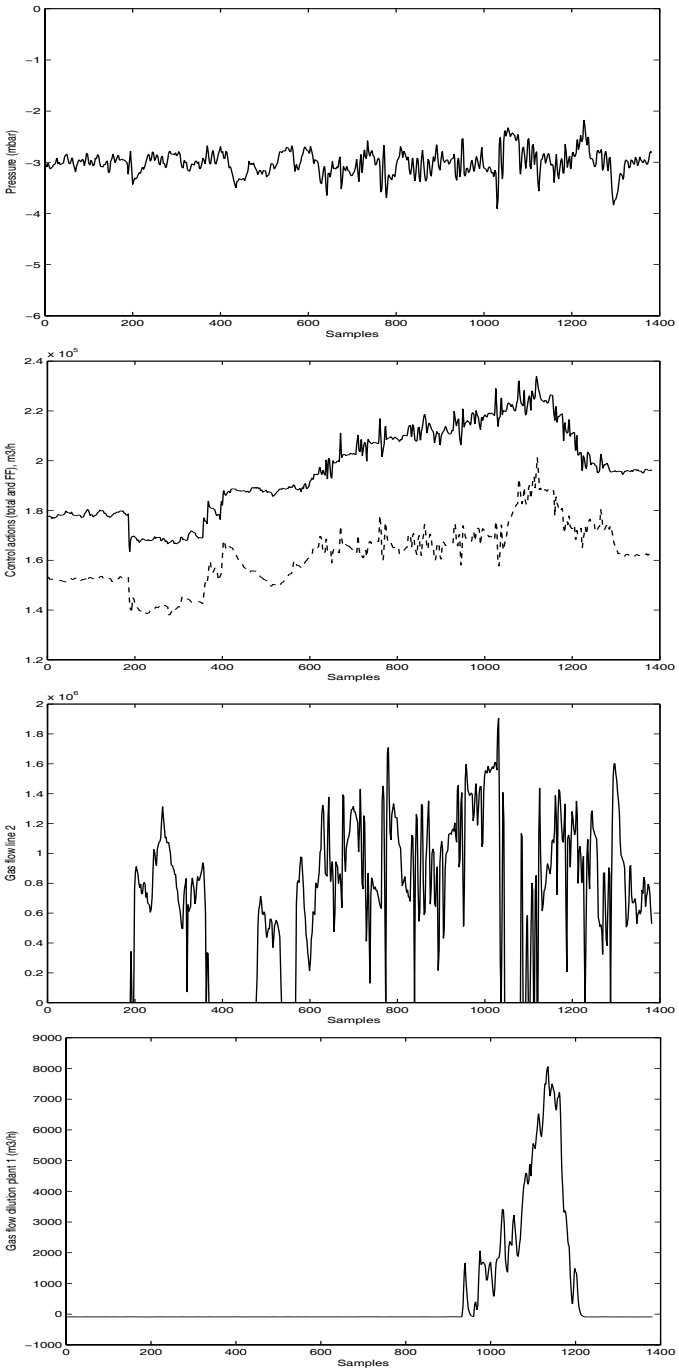


Fig. 3.11. Disturbance rejection. Changes in Line 2 feeding and air flow of dilution plant 1.

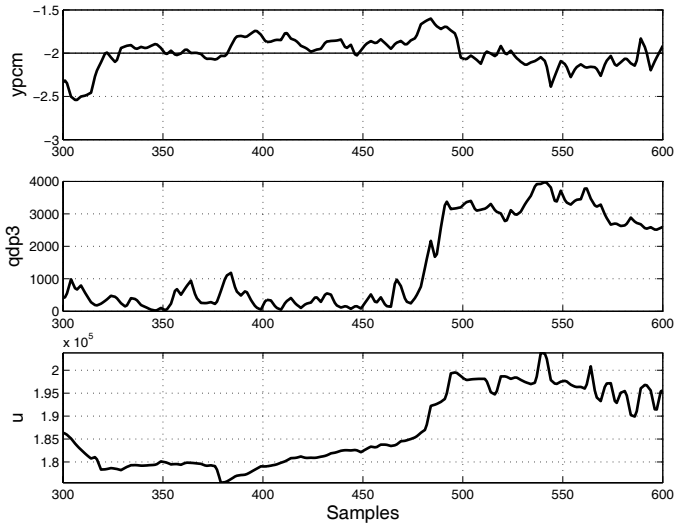


Fig. 3.12. Disturbance rejection. Dilution plant 3.

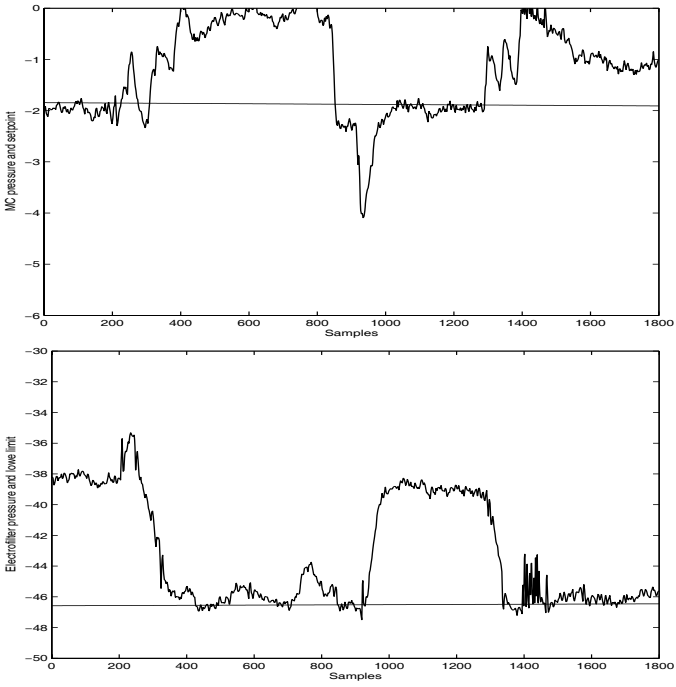


Fig. 3.13. Safety override controller

Table 3.4. Percentage of time at each operating mode before the implementation of the proposed strategy

Draft → / Dilution ↓	Automatic	Manual	Total
Automatic	27 %	8 %	35 %
Manual	55 %	10 %	65 %
Total	82 %	18 %	100 %

Table 3.5. Percentage of time at each operating mode after the implementation of the proposed strategy

Draft → / Dilution ↓	Automatic	Manual	Total
Automatic	84 %	0 %	84 %
Manual	15 %	1 %	16 %
Total	99 %	1 %	100 %

consumption in different scenarios under varied splitting strategies, in order to select the most economic splitting strategy, taking into account real plant conditions that can vary over time. The splitting constant K_s varies in the range $0.75 - 0.85$ and two tools have been developed to find the best value:

- Operator's tool: This is a display on the DCS that informs the operator in real time about which actions should be taken in order to operate at minimum energy cost.
- Plant manager tool: This is a MS Excel tool that captures real plant data, adjusts the economic calculations based on real plant conditions and gives advice for adjusting the flowrate splitting factor. The tool informs on the economic impact of the suggested adjustments.

The quantification of the benefits of the control strategy is shown in Figure 3.14. The bar graph shows the evolution of the energy consumption (mainly due to the compressor motors using to blow the gases). This consumption is specific, that is, corresponds to total energy divided by production, and therefore takes changes in production into account. Baseline corresponds to the average consumption of the six months previous to the implementation of the new control strategy.

It can be seen that the average is 4.22 %, reaching monthly values of even more than 5 % of reduction. This corresponds to a yearly saving of more that 1900 MWh, which is a considerable figure.

The following results have been achieved:

- Control performance: The standard deviation of the mixing-chamber pressure control has been reduced from 0.94 to 0.66 mm water column.
- Operator supervisory effort: The percentage of time operating in auto mode has been raised from 27 % to 84 %.
- Energy savings: The total electricity consumption of the three acid plants main gas compressors has been significantly reduced to 95.5 % of previous values. This implies slightly more than a 3 % reduction in the total acid plants electricity consumption.

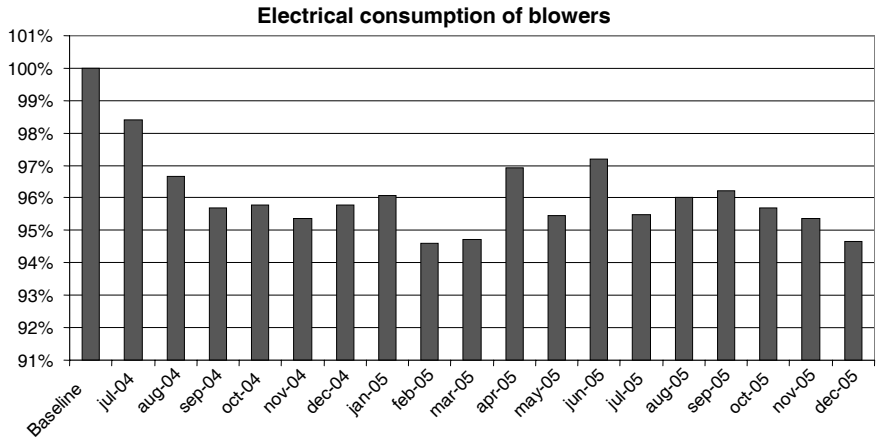


Fig. 3.14. Electrical consumption reduction

3.4 Discrepancies between Theory and Practice

The first discrepancies were found when developing control models. Although there was abundant data from every-day operations, it was very difficult to use this data for model identification. Data had to be analyzed and interpreted before it could be used for identification or model validation. There were signal excursions with no apparent cause that had to be investigated. The typical open-loop step-response tests that in theory seem so simple to perform, were extremely difficult to implement in practice. You have to keep the system stable in an open-loop mode and this is very difficult with a system with so many disturbances.

The second source of discrepancy between theory and practice was the complexity allowed in practice. The new control scheme had to be implemented over the existing equipment and the allowed complexity for the controller was limited. Classical control blocks had to be used. However, this approach had some obvious advantages such as lower cost, less commissioning time and less operator training requirements.

The third source of discrepancies came from the way in which the control objectives are considered. While in academia, the success of a controller is measured by its dynamic response, the main objective in practice is related to economic gains obtained from an increase in the production and/or quality savings and the use of resources such as energy, raw material or manpower. Although the two objectives are related, researchers rarely analyze the economic implications of advanced control applications.

Finally, consideration of the human factor. This is of paramount importance in practice. The need for training operators, or the level of knowledge required by them is seldom analyzed in academia. Furthermore, operators need to understand how the control equipment works in order to trust it. Otherwise, as soon as something strange occurs in the plant, the operators will switch to manual because they will not understand how the controller operates. The percentage of time a system works in automatic mode is a measure of the success of a control application that is seldom taken into consideration in control theory.

References

[1] Bergh, L., Chacana, P., Achurra, G., and Delgado, P. (2002). Improvements in the control of the injection system of copper concentrate in teniente converters. *Minerals Engineering* **15**, 369–372.

[2] Bordons, C. and Camacho, E. (1998). Generalized Predictive Controller for a Wide Class of Industrial Process. *IEEE Transaction on Control Systems Technology* **6**(3), 372–387.

[3] Camacho, E. and Bordons, C. (2004). *Model Predictive Control in the Process Industry. Second Edition*. Springer-Verlag, London.

[4] Clarke, D., Mohtadi, C., and Tuffs, P. (1987). Generalized Predictive Control. Part I. The Basic Algorithm. *Automatica* **23**(2), 137–148.

[5] Jämsä-Jounela, S. (2001). Current status and future trends in the automation of mineral and metal processing. *Control Engineering Practice* **9**(9), 1019–1024.

[6] Maciejowski, J. (2001). *Predictive Control with Constraints*. Prentice Hall, Harlow. ISBN 0-201-39823-0.

[7] Moskalyk, R. and Alfantazi, A. (2003). Review of copper pyrometallurgical practice: today and tomorrow. *Minerals Engineering* **16**, 893–919.

[8] Qin, S. and Badgwell, T. (2001). A survey of industrial model predictive control technoloy. *Control Engineering Practice* **11**, 733–764.

[9] Richet, D., Cotaina, N., Gabriel, M., and O'Reilly, K. (1995). Application of reliability centred maintenance in the foundry sector. *Control Engineering Practice Issue* **3**, 1029–1034.

[10] Tenno, R. and Jämsä-Jounela, S.-L. (1996). Copper flotation profit and control-system accuracy. *Control Engineering Practice* **4**, 1545–1551.

4

On Hybrid Model Predictive Control of Sewer Networks

Carlos Ocampo-Martinez[1], Alberto Bemporad[2], Ari Ingimundarson[1], and Vicenç Puig Cayuela[1]

[1] Sistemes Avançats de Control, ESAII, Universitat Politècnica de Catalunya (UPC), Rbla. Sant Nebridi 10, Terrassa, 08222 Barcelona, Spain.
{carlos.ocampo,ari.ingimundarson,vicenc.puig}@upc.edu

[2] Department of Information Engineering, University of Siena, Via Roma 56, I-53100 Siena, Italy. bemporad@dii.unisi.it

Summary. *Real-time control* (RTC) of sewer-network systems plays an important role in meeting increasingly restrictive environmental regulations to reduce release of untreated wastewater to the environment. This chapter presents the application of hybrid model predictive control (HMPC) on sewer systems. It is known from the literature that HMPC has a computational complexity growing exponentially with the size of the system to be controlled. However, the average solution time of modern *mixed integer program* (MIP) solvers is often much better than the predicted worst-case-solution time. The problem is to know when the worst-case computational complexity appears. In addition to presenting the application, a secondary aim of the chapter is to discuss the limits of applicability due to real-time constraints on computation time when HMPC is applied on large-scale systems such as sewer networks. By using a case study of a portion of the Barcelona sewer system, it is demonstrated how the computational complexity of HMPC appears for certain state and disturbance combinations.

4.1 Background

4.1.1 Introduction

Real-time control of sewer-network systems plays an important role in meeting increasingly restrictive environmental regulations to reduce release of untreated wastewater or *combined sewage overflow* (CSO) to the environment. Reduction of CSO often requires major investments in infrastructure within city limits and thus any improvement in efficient use of existing infrastructure, for example by improved control, is of interest. The advantage of sewer-network control has been demonstrated by a number of studies in the last decades, see [22, 34, 41, 42].

Extensive research has been carried out on real-time control of urban drainage systems. Comprehensive reviews that include a discussion of some existing implementations are given in [43] and references cited therein, while practical issues are discussed by [44], among others. The common idea is the use of optimization techniques to improve the system performance. Common control objectives are to try to avoid street flooding, prevent CSO discharges to the

environment, minimize the pollution, uniform utilization of the sewer-system storage capacity and, in most of cases, minimize the operating costs [21, 45, 46, 52]. The multivariable and large-scale nature of sewer networks has led to the use of some variants of *model predictive control* (MPC), as control strategies.

In order to apply MPC on a sewer system, a model able to predict its future states over a prediction horizon taking into account a rain forecast is needed. Sewer networks are systems with complex dynamics since water flows through sewers in open channel. Their dynamics are described by Saint-Vennant's partial differential equations that can be used to perform simulation studies but are highly complex to solve in real time.

Several control-oriented modeling techniques have been presented in the literature that deal with sewer systems, see [19, 34]. In [17, 39], a conceptual linear model based on assuming that a set of sewers in a catchment can be considered as a virtual tank or reservoir, has been used. The main reason to use a linear model is to preserve the convexity of the MPC optimization problem. A similar approach can be found in an early reference on MPC applied on sewer systems [22].

There exist, however, several inherent phenomena (overflows in sewers and tanks) and elements (weirs) in the system that result in distinct behavior depending on the state (flow/volume) of the network. These behaviors can not be neglected nor can they be modeled by a pure linear model. Instead, they require to be modeled using nonlinear functions depending on logical conditions [34]. This leads to the use of modeling methodologies that allow the inclusion of both continuous and discrete dynamics. The continuous dynamic part is typically associated with physical first principles while the discrete dynamic part comes from logic conditions that establish commutations of operational mode depending on internal system variables. This mixture of logical conditions and continuous dynamics gives rise to a *hybrid system*.

4.1.2 Hybrid Systems and Mixed Logical Dynamical Systems

The hybrid systems considered in this chapter are *mixed logical dynamical* (MLD) systems, introduced in [7]. MLD systems have recently been shown to be equivalent to other representations of hybrid systems such as *linear complementarity* (LC) systems, *min-max-plus scaling* (MMPS) systems and *piecewise affine* (PWA) systems, among others, under mild conditions, see [25]. By considering hybrid dynamical systems in discrete time a number of mathematical problems (like *Zeno behavior*, see [2], [26]) are avoided and allows models to be derived for which tractable analysis and optimal/predictive control techniques exist.

MLD systems are described by linear dynamic equations subject to linear mixed-integer inequalities, *i.e.* inequalities involving both continuous and binary (or logical, or 0–1) variables. These include physical/discrete states, continuous/integer inputs, and continuous/binary auxiliary variables. The general MLD form is [7]:

$$x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \tag{4.1a}$$

$$y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \tag{4.1b}$$

$$E_2\delta(k) + E_3z(k) \leq E_1u(k) + E_4x(k) + E_5 \tag{4.1c}$$

The meaning of the variables is the following:

- x are the continuous and binary states:

$$x = \begin{bmatrix} x_c \\ x_\ell \end{bmatrix}, \quad x_c \in \mathbb{R}^{n_c}, \quad x_\ell \in \{0,1\}^{n_\ell} \tag{4.2}$$

- y are the continuous and binary outputs:

$$y = \begin{bmatrix} y_c \\ y_\ell \end{bmatrix}, \quad y_c \in \mathbb{R}^{p_c}, \quad y_\ell \in \{0,1\}^{p_\ell} \tag{4.3}$$

- u are the continuous and binary inputs:

$$u = \begin{bmatrix} u_c \\ u_\ell \end{bmatrix}, \quad u_c \in \mathbb{R}^{m_c}, \quad u_\ell \in \{0,1\}^{m_\ell} \tag{4.4}$$

- Auxiliary binary variables: $\delta \in \{0,1\}^{r_\ell}$
- Auxiliary continuous variables: $z \in \mathbb{R}^{r_c}$.

Notice that by removing Equation (4.1c) and by setting δ and z to zero, Equations (4.1a) and (4.1b) reduce to an unconstrained linear discrete-time system. The variables δ and z are introduced when translating logic propositions into linear inequalities. All constraints are summarized in the inequality (4.1c).

The transformation of certain hybrid system descriptions into the MLD form requires the application of a set of given rules. To avoid the tedious procedure of deriving the MLD form by hand, a compiler was developed in [51] to automatically generate the matrices A, B_i, C, D_i and E_i in Equation (4.1) through the specification language HYSDEL (HYbrid System DEscription Language).

4.1.3 MPC and Hybrid Systems

Different methods for the analysis and design of hybrid control systems have been proposed in the literature during the last few years [7], [13], [31]. One of the most studied techniques involves optimal control and related variants such as *hybrid MPC* (HMPC). The formulation of the optimization problem in hybrid MPC follows the approach in standard linear MPC design; see [32]. The desired performance variables are expressed as affine functions of the control variables, initial states and known disturbances. However, due to the Boolean auxiliary variables present, the resulting optimization problem is a *mixed integer quadratic* or *linear program* (MIQP or MILP, respectively). The control law obtained in this way is also referred to as *mixed integer predictive control*.

In general, the HMPC structure is defined by the following optimal control problem. Assume that the hybrid system output should track a reference signal y_r and x_r, u_r, z_r are desired references for the states, inputs and auxiliary variables, respectively. For a fixed prediction horizon H_p, the sequences

$$\begin{aligned} \mathbf{x}_k &= \{x(1|k), x(2|k), \ldots, x(H_p|k)\} \\ \mathit{\Delta}_k &= \{\delta(0|k), \delta(1|k), \ldots, \delta(H_p-1|k)\} \\ \mathbf{z}_k &= \{z(0|k), z(1|k), \ldots, z(H_p-1|k)\} \end{aligned} \tag{4.5}$$

are generated applying the input sequence $\mathbf{u}_k = \{u(0), u(1), \ldots, u(H_p-1)\}$ to system (4.1) from initial state $x(0|k) \triangleq x(k)$, where $x(k)$ is the measurement of the current state.

Hence, using the previous concepts, the HMPC optimal control problem is defined as:

$$\min_{\mathbf{u}_k, \Delta_k, \mathbf{z}_k} J\left(\mathbf{u}_k, \Delta_k, \mathbf{z}_k, x(k)\right) \triangleq \left\|Q_{xf}\left(x(H_p|k) - x_f\right)\right\|_p + \sum_{i=1}^{H_p-1} \left\|Q_x\left(x(i|k) - x_r\right)\right\|_p + \sum_{i=0}^{H_p-1} \left\|Q_u\left(u(i) - u_r\right)\right\|_p + \sum_{i=0}^{H_p-1} \left\|Q_z\left(z(i|k) - z_r\right)\right\|_p + \sum_{i=0}^{H_p-1} \left\|Q_y\left(y(i|k) - y_r\right)\right\|_p \tag{4.6a}$$

$$\text{s. t.} \begin{cases} x_{(0|k)} = x(k), \\ x_f = x_r(H_p|k), \\ x(i+1|k) = Ax(i|k) + B_1 u(i) + B_2 \delta(i|k) + B_3 z(i|k), \\ y(i|k) = Cx(i|k) + D_1 u(i) + D_2 \delta(i|k) + D_3 z(i|k), \\ E_2\delta(i|k) + E_3 z(i|k) \leq E_1 u(i) + E_4 x(i|k) + E_5, \\ \text{for} \quad i = 0, 1, \cdots, H_p - 1 \end{cases} \tag{4.6b}$$

where $J(\cdot)$ is the cost function and x_f corresponds to the final desired value for the state variable over H_p. p is related to the selected cost norm (*i.e.* 1-norm, Euclidean or infinity).

Assuming that the HMPC problem (4.6) is *feasible* for $x(k)$, there exists an optimal solution given by the sequence

$$\mathbf{u}_k^* = \{u^*(0), u^*(1), \cdots, u^*(H_p - 1)\}$$

Then, the *receding horizon control* philosophy sets [15], [32]

$$u_{\text{MPC}}(x(k)) \triangleq u^*(0) \tag{4.7}$$

and disregards the computed inputs $u^*(1)$ to $u^*(H_p - 1)$. The whole process is repeated at the following time step. Equation (4.7) is known in the MPC literature as *the MPC control law*. The equality constraint related to the final state within the HMPC problem (4.6) can be relaxed so that $x(H_p|k)$ is only required to belong to a terminal constraint set $\mathbb{X}_T$, [28].

Thus, the HMPC problem (4.6) is defined by a cost function J in Equation (4.6a), which is given by the problem control objectives, and by a set of constraints in Equation (4.6b).

4.2 Hybrid Modeling and Control of Sewage Systems

4.2.1 Mathematical Modeling Principles

The water flow in sewer pipes is *open channel*. Open-channel flow is the flow of a certain fluid in a channel in which the fluid shares a free surface with an empty space above. The Saint-Venant equations[3], based on physical principles of mass conservation and energy, allow the accurate description of the open-channel flow in sewer pipes [36]. These equations are expressed as:

[3] Adhémar Jean Claude Barré de Saint-Venant (1797–1886) was a mechanician who developed the one-dimensional unsteady open-channel flow equations for shallow water or Saint-Venant equations that are a fundamental set of equations in modern hydraulic engineering.

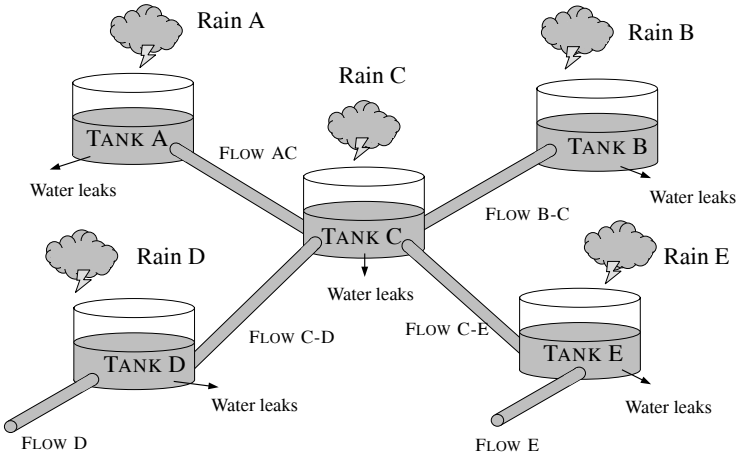


Fig. 4.1. Sewer network modeling by means of *virtual tanks*

$$\frac{\partial q(x,t)}{\partial x} + \frac{\partial A(x,t)}{\partial t} = 0 \tag{4.8}$$

$$\frac{\partial q(x,t)}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q(x,t)^2}{A(x,t)}\right) + gA(x,t)\frac{\partial L(x,t)}{\partial x} - gA(x,t)\ (I_0 - I_f) = 0 \tag{4.9}$$

where $q(x,t)$ is the flow ($\mathrm{m}^3\,\mathrm{s}^{-1}$), $A(x,t)$ is the cross-sectional area of the pipe (m^2), t is the time variable (s), x is the spatial variable measured in the direction of the sewage flow (m), g is the gravity ($\mathrm{m\,s}^{-2}$), I_0 is the sewer-pipe slope (dimensionless), I_f is the friction slope (dimensionless) and $L(x,t)$ is the water level inside the sewer pipe (m). This pair of partial-differential equations constitutes a nonlinear hyperbolic system. For an arbitrary geometry of the sewer pipe these equations lack an analytical solution. Notice that these equations describe the system behavior in great detail. However, such a level of detail is not useful for real-time control implementation due to the complexity of obtaining the solution of Equations (4.8) and (4.9) and the associated high computational cost.

Alternatively, several modeling techniques have been presented in the literature that deal with real-time sewer system control, see [19], [21], [33], [34], among many others. The modeling approach used in this chapter is close to the approach presented in [22]. Here, the sewer system is divided into catchments that are treated as *virtual tanks* (see Figure 4.1). At any given time, the stored volumes represent the amount of water inside the sewer pipes associated with the tank and are calculated on the basis of the rainfall of the catchment area of the tank and flow exchanges between other interconnected virtual tanks. The volume is calculated through the mass balance of the stored volume, the inflows and the outflow of the tank and the input rain intensity. The mentioned mass balance can be written as the difference equation

$$v_i(k+1) = v_i(k) + \Delta t \varphi_i S_i P_i(k) + \Delta t \left(q_i^{\text{in}}(k) - q_i^{\text{out}}(k)\right) \tag{4.10}$$

where φ_i is the *ground absorption coefficient* of the ith tank catchment, S is the surface area of the ith tank catchment, Δt is the time interval between measurements and P is the *rain intensity* corresponding to the ith tank catchment in Δt time units. $q_i^{\text{in}}(k)$ and $q_i^{\text{out}}(k)$ are the

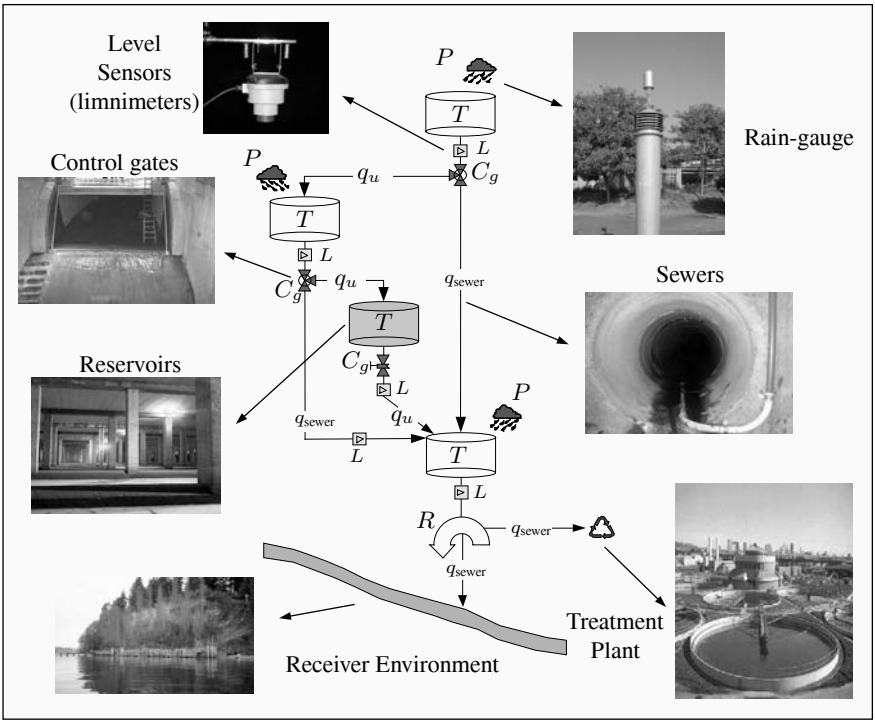


Fig. 4.2. Components for a basic scheme of a sewer network

sum of inflows and outflows, respectively. *Real retention tanks* are modeled in the same way but without the precipitation term.

The tanks (either, real or virtual) are connected with flow paths or links that corresponds to the main sewage pipes between the tanks. Figure 4.2 gives an idea of the interrelation between different components of a very simple sewer network using the proposed modeling methodology. The tank outflows are generally assumed to be a function of the tank volume. For the current application a linear relation has been assumed, that is,

$$q_i^{\text{out}}(k) = \beta_i v_i(k) \tag{4.11}$$

where β_i (given in s^{-1}) is defined as the *volume/flow conversion coefficient* as suggested in [48].

The manipulated variables of the system, denoted as q_{u_i}, are related to the outflows from the tanks. In the case of a real tank, a *retention gate* is present to control the outflow. Virtual tank outflows can not be closed but can be redirected with *redirection gates* (RG). The redirection gates divert the flow from a nominal flow path that the flow follows if the redirection gate is closed. This nominal flow is denoted as Q_i in the equation below, which expresses mass conservation at the redirection gate:

$$q_i^{\text{out}}(k) = Q_i(k) + \sum_j q_{u_i}^j(k) \tag{4.12}$$

where j is an index over all manipulated flows coming from the redirection gate. The flow path that Q_i represents is assumed to have a certain capacity and when this capacity reaches its limits, an overflow situation occurs. This flow limit will be denoted $\overline{Q}_i$. When Q_i reaches its capacity, two cases are considered: first, the water starts to flow on the streets, causing an overflow situation and secondly, it exits the sewer network and is considered lost to the environment. In the first case, the overflow water either follows the nominal flow path and ends up in the same tank as Q_i or it is diverted to another virtual tank. Flow to the environment physically represents the situation when the sewage water ends up in a river or, in the case of the Barcelona situation, in the Mediterranean sea.

When using this modeling approach where the inherent nonlinearities of the sewer network are simplified by assuming that only flowrates are manipulated, physical restrictions need to be included as constraints on system variables. For example, the manipulated variables $q^j_{u_i}$ that determine the redirection of an outflow from a tank should never be larger than the nominal outflow from the tank given by Equation (4.11). This is expressed with the following inequality

$$\sum_j q^j_{u_i}(k) \leq q^{out}_i(k) \tag{4.13}$$

Usually the range of actuation is also limited so that the manipulated variable has to fulfill $\underline{q}^j_{u_i} \leq q^j_{u_i}(k) \leq \overline{q}^j_{u_i}$, where $\underline{q}_{u_i}$ denotes the lower limit of manipulated flow and $\overline{q}_{u_i}$ denotes its upper limit. When $\underline{q}^j_{u_i}$ equals zero, this constraint is convex. But, if the lower bound is larger than zero, a constraint representing mass conservation has to be included in the range limitation leading to the following nonconvex inequality:

$$\min(\underline{q}^j_{u_i}, q^{out}_i(k) - \sum_{t \neq j} q^t_{u_i}(k)) \leq q^j_{u_i}(k) \leq \overline{q}^j_{u_i} \tag{4.14}$$

The sum in the expression is calculated for all outflows related to tank i except j.

A further complication is that if the control signal is an inflow to a real tank that has hard constraints on its capacity, then the situation can occur that this lower limit is also limited by this maximum capacity and the outflow from the real tank. The limit on the range of real tanks is expressed as

$$0 \leq v_i(k) \leq \overline{v}_i \tag{4.15}$$

where $\overline{v}_i$ denotes the maximum volume capacity given in m^3. As this constraint is physical, it is impossible to send more water to a real tank than it can hold.

The virtual tanks do not have a physical limit on their capacity. When they rise above a decided level an overflow situation occurs. This represents the case when the level in the sewers has reached a limit so that an overflow situation can occur in the streets. Notice that in practice, the difference between tank and link overflows is often small.

Model Calibration from Real Data

Real data from sensor measurements were used to estimate the parameters of the virtual tanks. Level is measured in sewer pipes using ultrasonic limnimeters. Notice that the sewer level is measured instead of the flow through the sewer. As these level sensors have no contact with the water flow, problems due to incorrect measurements caused by sensor faults are avoided. From these level measurements, the flow entering and exiting each virtual tank can be estimated

assuming steady uniform flow using the *Manning formula*[4] [36]:

$$q = vS_w \tag{4.16}$$

where S_w is the *wetted surface* that depends on the cross-sectional sewer area A and water level L within the sewer. The dependence of A and L on x and t are omitted for compactness. Moreover, v is the water velocity computed according to the parameters relation

$$v = \frac{K_n}{n} R_h^{2/3} I_0^{1/2} \tag{4.17}$$

where K_n is a constant whose value depends of the measurement units used in the equation, n is the Manning coefficient of roughness that depends of flow resistance offered by the sewer pipe material, R_h is the hydraulic radius defined as the relation of the cross-sectional area of flow and the wetted perimeter P_w as $R_h = A/P_w$, and I_0 is the sewer slope. For a given geometry of the sewer cross section, wetted perimeter and hydraulic radius can be expressed in terms of the sewer level L. For instance, given a rectangular cross section of width b, the wetted surface S_w is bL, the wetted perimeter P_w is $b + 2L$ and the hydraulic radius is given by $R_h = \frac{bL}{b+2L}$.

Rain intensity is measured using a *tipping bucket rain gauge* (see Figure 4.3). This gauge technology uses two small *buckets* mounted on a fulcrum (balanced like a see-saw). The tiny buckets are manufactured with tight tolerances to ensure that they hold an exact amount of precipitation. The tipping bucket assembly is located above the rain sewer, which funnels the precipitation to the buckets. As rainfall fills the tiny bucket, it becomes overbalanced and tips down, emptying itself as the other bucket pivots into place for the next reading. The action of each tipping event triggers a small switch that activates the electronic circuitry to transmit the count to the indoor console, recording the event as 1.2 mm h^{-1} of rainfall. The number of tipping events in 5 min (sampling time) is accumulated and multiplied by 1.2 mm h^{-1} in order to obtain the rain intensity P in m s^{-1} at each sampling time, after the appropriate units conversion.

Given rain intensities as well as limnimeter data, data series with P_i and input/output flows can be determined. By combining Equations (4.10) and (4.11), the following input/ouput equation in function of the flow in sewers and rain intensity in catchments can be obtained:

$$q_i^{out}(k+1) = aq_i^{out}(k) + b_1 P_i(k) + b_2 q_i^{in}(k) \tag{4.18}$$

where $a = (1 - \beta_i \Delta t)$, $b_1 = \beta_i \Delta t \varphi_i S_i$ and $b_2 = \beta_i \Delta t$. Figure 4.4 represents this equation and the interaction of all described parameters and measurements.

Equation (4.18) is linear in the parameters. This allows the parameters to be estimated using classical parameter estimation methods based on *least-squares algorithms* [30]. Hence, the parameter associated with the ground absorption coefficient is estimated as

$$\varphi_i = \frac{b_1}{b_2 S_i} \tag{4.19}$$

and the volume/flow conversion coefficient is estimated as

$$\beta_i = \frac{b_2}{\Delta t} \tag{4.20}$$

[4] The Manning formula is an empirical formula for open-channel flow, or flow driven by gravity. It was developed by the French engineer *Robert Manning* and proposed on 1891 in the Transactions of the Institution of Civil Engineers (Ireland).

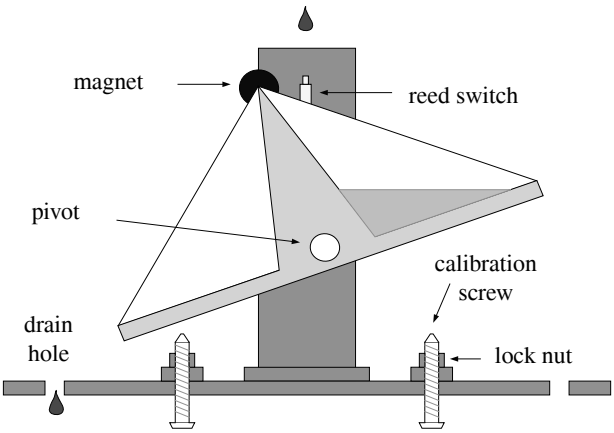


Fig. 4.3. Rain measurement principle using a tipping-bucket rain-gauge

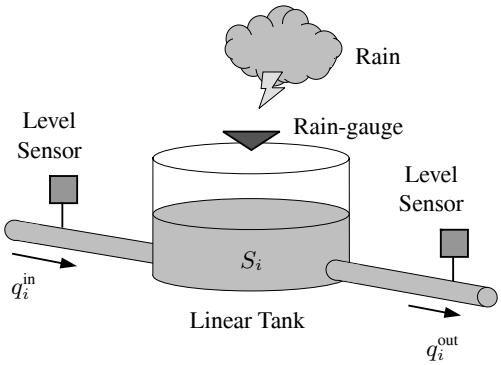


Fig. 4.4. Scheme of an individual virtual tank and its parameters and measurements

for the *i*th catchment.

Ground absorption and volume/flow conversion coefficients can be estimated online at each sampling time using Equation (4.18) and the recursive least-squares (RLS) algorithm [30]. Once estimated, these parameters are supplied to the MPC controller in order to take into account their time variation and neglected nonlinearities.

4.2.2 Hybrid Modeling of Sewer Systems

The presence of intense precipitation can cause some sewers to reach their limits of capacity. When this happens, excess sewage that normally would have been collected in the sewer can flow to other parts of the sewer system. In this way, as mentioned in Section 4.1, flow paths appear that are not always present and depend on the system state and inputs.

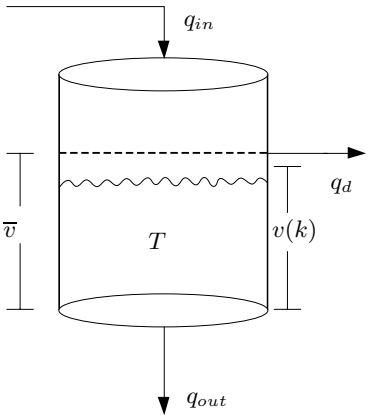


Fig. 4.5. Scheme of a virtual tank

This behavior and other particular hybrid phenomena of sewer systems that depend on system state can be conveniently modeled using the discrete-time hybrid dynamical models introduced in Section 4.1. In what follows it will be shown how some common elements of sewer systems can be represented with these models. Specifically, the elements that will be considered are virtual tanks, flow links and redirection gates. Other common sewer-system elements such as pumping stations can be easily modeled within the hybrid modeling methodology but will be omitted as they do not occur in the case study presented later. Hybrid dynamic behavior in sensors such as rain gauges and limnimeters is also omitted.

Virtual Tanks (VT)

When the maximum volume $\overline{v}$ in virtual tanks is reached, the excess volume above this maximum amount is redirected to another tank within the network. This creates a new outflow from the tank denoted with q_d in Figure 4.5. This phenomenon can be expressed mathematically as:

$$q_d(k) = \begin{cases} \frac{(v(k)-\overline{v})}{\Delta t} & \text{if} \quad v(k) \geq \overline{v} \\ 0 & \text{otherwise} \end{cases} \tag{4.21a}$$

$$q_{out}(k) = \begin{cases} \beta\overline{v} & \text{if} \quad v(k) \geq \overline{v} \\ \beta v(k) & \text{otherwise} \end{cases} \tag{4.21b}$$

where $v(k)$ corresponds to the tank volume (system state), $\overline{v}$ is its maximum volume capacity, Δt is the sampling time and β is a proportional factor between the volume and the outflow, see [38]. The flow q_d is referred to as *virtual tank overflow*. The difference equation for the virtual tank is

$$v(k+1) = v(k) + q_{in}(k) - q_{out}(k) - q_d(k) \tag{4.22}$$

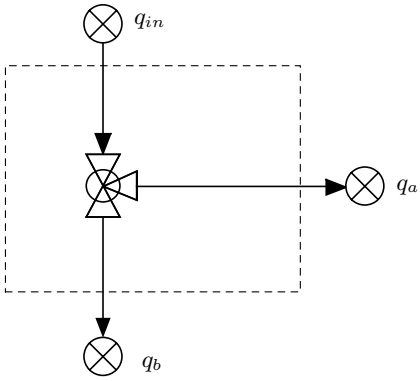


Fig. 4.6. Scheme of redirection gate element

Redirection Gates (RG)

These type of elements within a sewer network are used to redirect flow at a certain point in the network. An inflow q_{in} is redirected to outflows q_a and q_b as shown in Figure 4.6. Two types of redirection elements are considered. In the first type referred to as passive (also known as weirs), the flow follows the path q_a until a limit $\overline{q}_a$ is reached. If q_{in} is larger than $\overline{q}_a$ then the difference flows through q_b. In the other type, flow q_a can be manipulated within its physical limits. The expressions for the first type of redirection gates are:

$$q_a(k) = \begin{cases} q_{in}(k) & \text{if} \quad q_{in} \leq \overline{q}_a \\ \overline{q}_a & \text{otherwise} \end{cases} \tag{4.23a}$$

$$q_b(k) = q_{in}(k) - q_a(k) \tag{4.23b}$$

while in the second type, since the flow q_a is manipulated, the flow only has to fulfill the following restrictions:

$$0 \leq q_a(k) \leq q_{in} \tag{4.24a}$$

$$q_a(k) \leq \overline{q}_a \tag{4.24b}$$

$$q_a(k) = q_{in}(k) - q_b(k) \tag{4.24c}$$

Flow Links (FL)

The outflow from virtual tanks is assumed to be unlimited to guarantee a feasible solution. The same thing applies to the outflow q_b from the retention gate element. But, most often, flow links between elements in the sewer network have limited flow capacity. The flow link element serves to model this limited capacity. When the limit of flow capacity is exceeded, the resulting overflow is possibly redirected to another element in the system or is considered lost to the environment. The overflow is denoted q_c in Figure 4.7. The equations for the flow link are:

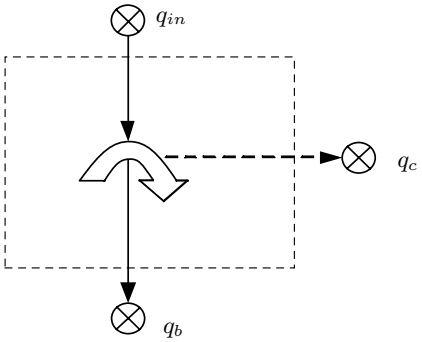


Fig. 4.7. Scheme for flow links

$$q_b(k) = \begin{cases} \overline{q}_b & \text{if} \quad q_{in} > \overline{q}_b \\ q_{in}(k) & \text{otherwise} \end{cases} \tag{4.25a}$$

$$q_c(k) = \begin{cases} q_{in}(k) - \overline{q}_b & \text{if} \quad q_{in} > \overline{q}_b \\ 0 & \text{otherwise} \end{cases} \tag{4.25b}$$

where $\overline{q}_b$ is the maximum flow through q_b and q_{in} is the inflow to the flow link.

The Hybrid Sewer System Model

The total sewer system model is constructed by connecting the system inflows (rain) and outflows (sewer treatment plants or outflows to the environment) with the inflows and outflows of the elements as well as connecting the elements themselves. The manipulated variables of the system, denoted as q_u, are the manipulated variables of each component as described before. The logical conditions presented to describe the dynamics of the sewer-system elements can be translated into linear integer inequalities as described in [7]. The whole sewer network expressed in MLD form can be written as:

$$v(k+1) = Av(k) + B_1 q_u(k) + B_2\delta(k) + B_3 z(k) + B_4 d(k) \tag{4.26a}$$

$$y(k) = Cv(k) + D_1 q_u(k) + D_2\delta(k) + D_3 z(k) + D_4 d(k) \tag{4.26b}$$

$$E_2\delta(k) + E_3 z(k) \leq E_1 q_u(k) + E_4 v(k) + E_5 + E_6 d(k) \tag{4.26c}$$

where $v \in \mathbb{R}_+^{n_c}$ corresponds to the vector of tank volumes (states), $q_u \in \mathbb{R}_+^{mi_c}$ is the vector of manipulated sewer flows (inputs), $d \in \mathbb{R}_+^{md}$ is the vector of rain measurements (disturbance), logic vector $\delta \in \{0,1\}^{r_\ell}$ collects the Boolean overflow conditions and vector $z \in \mathbb{R}_+^{r_c}$ is associated with variables that appear depending on system states and inputs. Variables δ and z are auxiliary variables associated with the MLD form. Equation (4.26c) collects the set of element constraints as well as translations from logic propositions. Notice that this model is a more general MLD than was presented in [7] due to the addition of the measured disturbances.

If it is assumed that the disturbances are described with a disturbance model $d(k+1) = A_d d(k)$, the MLD (4.26) form could be rewritten as:

$$v(k+1) = \begin{bmatrix} A & B_4 \\ 0 & A_d \end{bmatrix} \begin{bmatrix} v(k) \\ d(k) \end{bmatrix} + \begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix} \begin{bmatrix} q_u(k) \\ \delta(k) \\ z(k) \end{bmatrix} \tag{4.27a}$$

$$y(k) = \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} v(k) \\ d(k) \end{bmatrix} + \begin{bmatrix} D_1 & D_2 & D_3 \end{bmatrix} \begin{bmatrix} q_u(k) \\ \delta(k) \\ z(k) \end{bmatrix} \tag{4.27b}$$

$$\begin{bmatrix} E_2 & E_3 \end{bmatrix} \begin{bmatrix} \delta(k) \\ z(k) \end{bmatrix} \leq \begin{bmatrix} E_4 & E_6 \end{bmatrix} \begin{bmatrix} v(k) \\ d(k) \end{bmatrix} + \begin{bmatrix} E_1 & E_5 \end{bmatrix} \begin{bmatrix} q_u(k) \\ 1 \end{bmatrix} \tag{4.27c}$$

The type of model to be used for disturbance $d(k)$ in sewer-system control is, in general, an open research topic. Different types of rain-prediction models can be used (statistical, AR models, *etc.*) [49] or the rain precipitation can be measured and predicted more directly (radars, meteorological satellites, *etc.*) [53]. Combinations of the two approaches have also been presented. According to [16], when an optimal control law is used in the RTC of sewer networks with a short prediction horizon, assuming that the rain is constant over the prediction horizon results in a moderate performance loss compared to knowing exactly the rain over the horizon. This confirmed similar results reported in [22]. For constant rain over the prediction horizon, matrix A_d is set as the identity matrix of suitable dimensions.

4.2.3 The Hybrid Control Strategy

As discussed in the previous section, the proposed hybrid modeling methodology is very rich and allows the straightforward treatment of hybrid phenomena such as overflow and flooding. Morever, HMPC has been applied successfully to a variety of control problems in recent years using several approaches, see [5], [14], [29], [47], [50],

among others. This section presents a description of the HMPC formulation applied on sewer networks. The different aspects discussed here are presented considering the particular case study but can be easily extrapolated to other sewage-system topologies. The concepts and definitions of Section 4.1.3 are applied in this section in a straightforward manner but taking into account the particular notation used for sewer networks.

Control Objectives

The sewer-system control problem has multiple objectives with varying priority, see [34]. There exist many types of objectives according to the system design. In general, the most common objectives are related to the manipulation of the sewage in order to avoid undesired sewage overflow on the streets of the city. Other kinds of objectives are, for instance, related to the control energy, *i.e.* the energy cost of the regulation of the gate movements. According to the literature of sewer networks, the main objectives for the case study of this chapter are listed below in order of decreasing priority:

- *Objective 1:* minimize flooding in streets (virtual tank overflow).
- *Objective 2:* minimize flooding in links between virtual tanks.
- *Objective 3:* maximize sewage treatment.
- *Objective 4:* minimize control action.

A secondary purpose of the third objective is to reduce the volume in the tanks to anticipate future rainstorms. This objective also indirectly reduces pollution to the environment. This is because if the treatment plants are used optimally along with the storage capacity of the

network, pollution lost to the environment should be at a minimum. It should be noted that in practice the difference between the first two objectives is small. According to the hybrid model for sewer networks proposed in Section 4.2.2, all overflows and flows to treatment plants are defined by auxiliary variables z. However, as some of the performance objectives are sums of system variables, it is convenient to define system outputs where these sums are calculated.

The Cost Function

Each control objective corresponds to one term in the cost function. Hence, the expression of that function depends on its constitutive variables (auxiliary or output type). In general form, the structure for the cost function in Equation (4.6a) has the form

$$J[(\mathbf{u}_k, v(k)] \triangleq \sum_{i=0}^{H_p-1} \left\|Q_z(z_{(i|k)} - z_r)\right\|_p + \sum_{i=0}^{H_p-1} \left\|Q_y(y_{(i|k)} - y_r)\right\|_p \tag{4.28}$$

where Q_z and Q_y correspond to weight matrices of suitable dimensions and z_r, y_r are reference trajectories related to auxiliary and output variables, respectively. For the objectives 1 and 2, the references are zero flow. For the third objective, the references are the maximum capacity of the associated sewage-treatment plants. Priorities are set by selecting matrices Q_z and Q_y. The norm p can be selected as $p = 1, 2$ or $p = \infty$. Notice that due to the fact that all performance variables are positive, the case when $p = 1$ is actually a simple sum of the performance variables.

4.2.4 Case-study Description

The Barcelona Sewer Network

The city of Barcelona has a combined sewage system of approximately 1697 km length with a storage capacity of 3,038,622 m^3. It is a unitary system, that is, it combines waste and rainwater into the same sewers. It is worth noting that Barcelona has a population that is approximately around 1,593,000 inhabitants in an area of 98 Km^2, resulting in a very high density of population. Additionally, the yearly rainfall is not very high (600 mm $year^{-1}$), but it includes heavy storms (arriving to 90 mm h^{-1}) typical of the Mediterranean climate that can cause a lot of flooding problems and CSO to the receiving waters.

Clavegueram de Barcelona, S.A. (CLABSA) is the company in charge of the sewer system management in Barcelona. There has been a remote control system in operation since 1994 that includes, sensors, regulators, remote stations, communications and a control center in CLABSA. Nowadays, for control purposes, the urban drainage system contains 21 pumping stations, 36 gates, 10 valves and 8 detention tanks that are regulated in order to prevent flooding and CSO. The remote control system is equipped with 56 remote stations including 23 rain-gauges and 136 water-level sensors that provide real-time information about rainfall and water levels into the sewer system. All this information is centralized at the CLABSA Control Center through a supervisory control and data acquisition (SCADA) system. The regulated elements (pumps, gates and detention tanks) are currently controlled locally, *i.e.* they are handled from the remote control center according to the measurements of sensors connected only to the local station.

Barcelona Test Catchment

In this chapter, a representative portion of the Barcelona sewer system is studied. Figure 4.8 shows the catchment over a real map of Barcelona. A calibrated and validated model of the system following the methodology explained in Section 4.2.1 is available as well as rain-gauge data for an interval of several years.

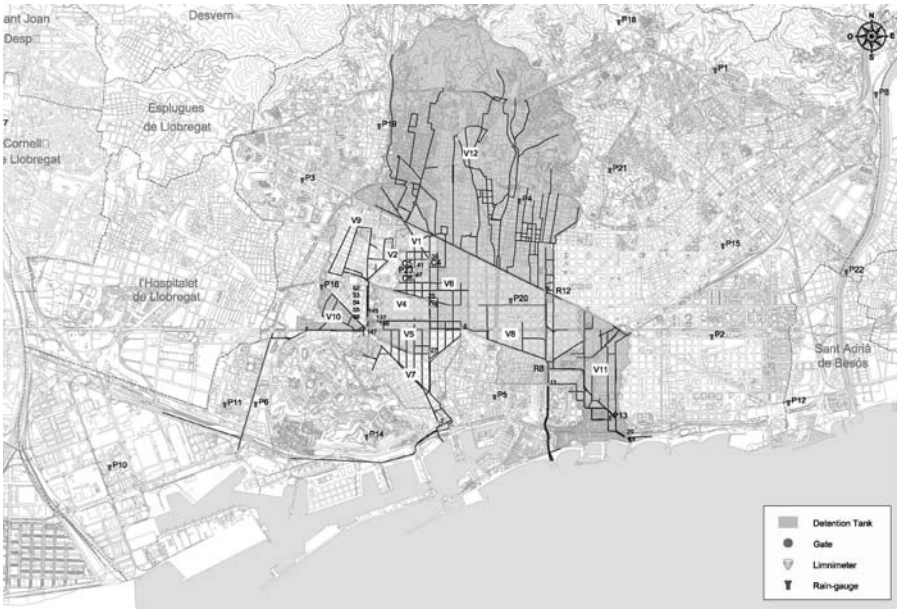


Fig. 4.8. Test Catchment located over the Barcelona map (Courtesy of CLABSA)

The catchment has a surface of 22.6 km^2 and includes typical elements of the larger network. Due to its size, there is a spatial difference in the rain intensity between rain gauges. The system is presented in Figure 4.10 using the virtual-reservoir methodology presented in Section 4.2.1. The catchment area considered has 1 retention gate associated with 1 real tank, 3 redirection gates, 11 subcatchments defining the same number of virtual tanks, several limnimeters and a pair of links connected to the same number of treatment plants. Also there are 5 rain-gauges in the catchment but some virtual tanks share the same rain sensor. The difference between the rain inflows for virtual tanks that share a sensor lies in the surface area A and the ground absorbtion coefficient φ. This results in a different amount of rain entering to the virtual tank even though the rain intensity is the same. The real tank corresponds to the *Escola Industrial* reservoir, which is located under a soccer field of the Industrial School of Barcelona (see Figure 4.9). It has a rectangle geometry of 94×54 m with a medium depth of 7 m and a maximum water capacity of 35000 m^3 [18].

The related system model has 12 state variables corresponding to the volumes in the 12 tanks (1 real, 11 virtual), 4 control inputs and 5 measured disturbances corresponding to the measurements of rain intensity over the virtual tanks. Two wastewater-treatment plants can be used to treat the sewer water before it is released to the environment. It is supposed that

Fig. 4.9. Retention tank located at Escola Industrial de Barcelona

all states (virtual-tank volumes) are estimated by using the limnimeters shown with a capital letter L in Figure 4.10. The flows to the environment as pollution, ($q_{10\mathrm{M}}$, $q_{7\mathrm{M}}$, $q_{8\mathrm{M}}$ and $q_{11\mathrm{M}}$ to the Mediterranean sea and q_{12s} to an other catchment) and the flows to the treatment plants ($Q_{7\mathrm{L}}$ and $Q_{11\mathrm{B}}$) are shown in the figure as well. Also appearing in the figure are the rain intensities P_{13}, P_{14}, P_{16}, P_{19} and P_{20}. The 4 controlled flows, denoted as q_{u_i}, $i = 1 \ldots 4$, have a maximum flow capacity of 9.1, 25.0, 7.0 and 29.3 $\mathrm{m}^3\ \mathrm{s}^{-1}$, respectively.

Figures 4.11a and 4.11b present the comparison between real level (from real data) and predicted level (using the model described in Section 4.2.1) corresponding to the output flows of virtual tank T_1 and T_2, respectively. It can be noticed that the fit obtained with the proposed modeling approach is satisfactory.

Tables 4.1 and 4.2 summarize the description of the case-study variables as well as the value of the parameters obtained by calibrating the system model following the procedures described in Section 4.2.1. In Table 4.1 (and also in Figure 4.10), T_i denotes the ith virtual tank and T_3 denotes the real tank. Moreover, $\overline{v}_i$ denotes the maximum capacity of the ith tank (virtual or real). Elements R_1 to R_5 denote passive redirection gates. The maximum capacities of the redirection gate flows as well as flow links are given in Table 4.2.

4.3 Simulation and Results

4.3.1 Rain Episodes

The rain episodes included in the case study are based on real rain-gauge data obtained within the city of Barcelona on the given dates (day-month-year) as seen in Table 4.4. These episodes were selected to represent the meteorological behavior of Barcelona. In Figure 4.12, the reading of the rain gauges for one of the rain episodes is shown. The rain storm presented in this figure caused severe flooding in the city area under study.

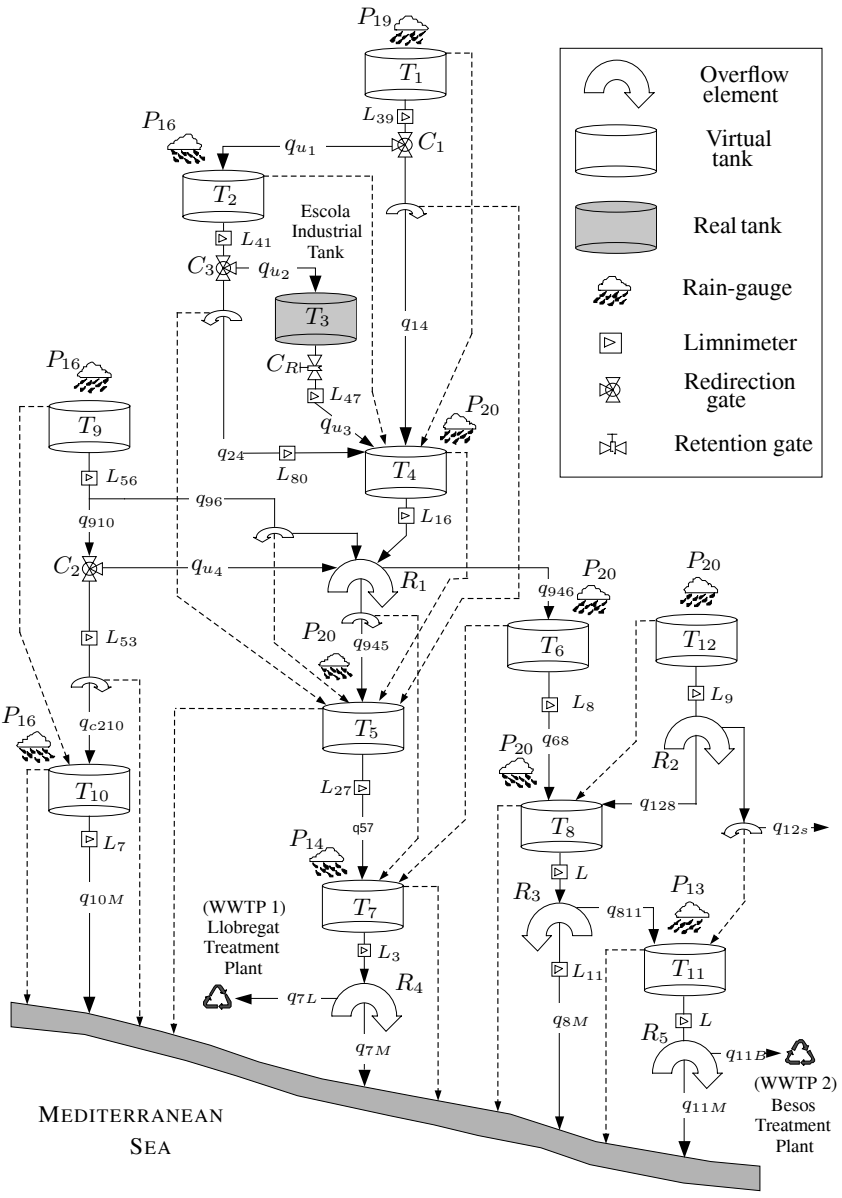
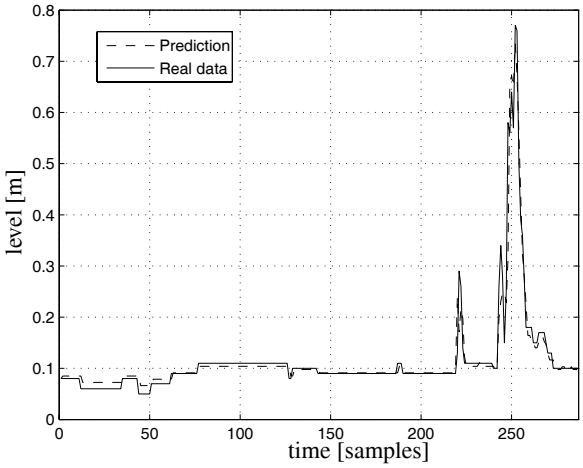


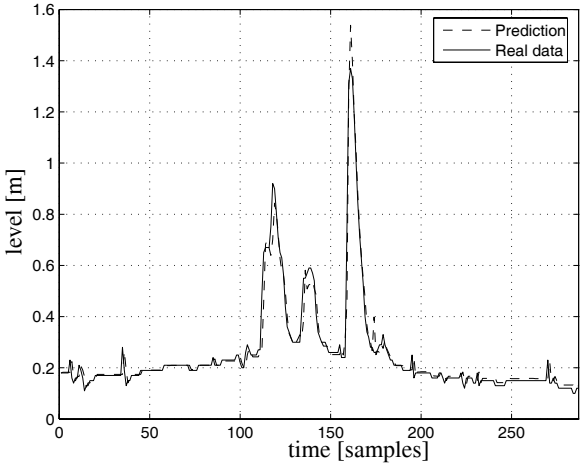
Fig. 4.10. Scheme of the Barcelona sewer-network catchment

4.3.2 Results

The purpose of this section is to show the performance of hybrid MPC for realistic episodes of rainstorms. The assumptions made for the implementation will be presented and their validity discussed before the results are given.



(a) Output level in T_1



(b) Output level in T_2

Fig. 4.11. Results of model calibration using the approach given in Section 4.2.1

The transformation of the hybrid system equations into the MLD form requires the application of the set of given rules [7]. The higher-level language and associated compiler HYSDEL (see [51]) is used here to avoid the tedious procedure of deriving the MLD form by hand. Given the MLD model, the controllers were designed and the scenarios simulated using the *Hybrid Toolbox* for MATLAB® (see [4]). Moreover, ILOG CPLEX 9.1 has been used for solving the MIP problems.

Table 4.1. Parameter values related to the subcatchments within the case study

Tank	S (m^2)	φ_i	β_i (s^{-1})	$\overline{v}_i$ (m^3)
T_1	323,576	1.03	7.1×10^{-4}	16,901
T_2	164,869	10.4	5.8×10^{-4}	43,000
T_3	5,076	–	2.0×10^{-4}	35,000
T_4	754,131	0.48	1.0×10^{-3}	26,659
T_5	489,892	1.93	1.2×10^{-4}	27,854
T_6	925,437	0.51	5.4×10^{-4}	26,659
T_7	1,570,753	1.30	3.5×10^{-4}	79,229
T_8	2,943,140	0.16	5.4×10^{-4}	87,407
T_9	1,823,194	0.49	1.3×10^{-4}	91,988
T_{10}	385,274	5.40	4.1×10^{-4}	175,220
T_{12}	1,913,067	1.00	5.0×10^{-4}	91,442
T_{12}	11,345,595	1.00	5.0×10^{-4}	293,248

Table 4.2. Description and maximum flow values of the main sewers in the case study

Sewer	$\overline{q}$ (m^3 s^{-1})	Sewer	$\overline{q}$ (m^3 s^{-1})
q_{14}	9.14	q_{128}	63.40
q_{24}	3.40	q_{57}	14.96
q_{96}	10.00	q_{68}	7.70
q_{c210}	32.80	q_{12s}	60.00
q_{945}	13.36	q_{811}	30.00
q_{910}	24.00	q_{7L}	7.30
q_{946}	24.60	q_{11B}	9.00

Simulation of Scenarios

The performance of the control scheme is compared with the simulation of the sewer system without control when the manipulated links have been used as passive elements, *i.e.* the amount of flows $q_{u1}(k)$, $q_{u2}(k)$ and $q_{u4}(k)$ only depend on the inflow to the corresponding gate and they are not manipulated (see Section 4.2.2), while $q_{u3}(k)$ is the natural outflow of the real tank.

Two different MLD models are used to simulate the scenarios, one for the hybrid MPC controller, MLD$_C$, which has been described in Section 4.2, and one tailored for the open-loop simulation, MLD$_{OL}$. The difference lies in the fact that when the system is simulated in open loop, logical conditions have to be added so that the physical constraints of the system are respected. The model MLD$_{OL}$ is therefore augmented so that the control signals from the controller are adjusted to respect the physical restrictions of the whole network. MLD$_{OL}$ contains more auxiliary variables for this reason.

The MLD$_C$ model implemented has 22 logical variables and 44 auxiliary variables. The prediction horizon H_p was set to 6, which is equivalent to 30 min (with the sampling time $\Delta t = 300\ s$, 5 min). The length of the simulation scenarios is 100 samples. The computation times presented in Section 4.4 were obtained on an INTEL® PENTIUM® M 1.73 GHz machine.

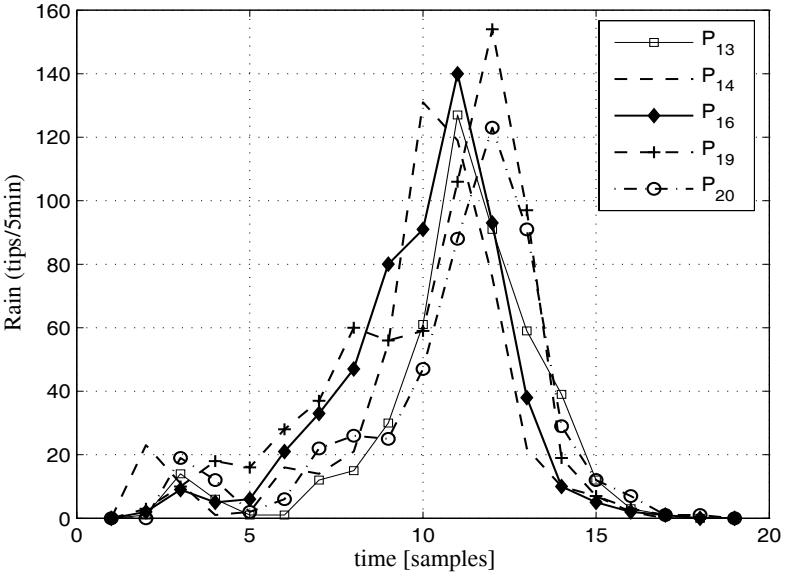


Fig. 4.12. Example of rain episode occurred in Barcelona on 14 September, 1999. Each curve represents a rain gauge P_i.

The control tuning is done taking into account the prioritization of the control objectives. In a preliminary study, different norms, cost-function structures and cost-function weights w_i have been used. In order to give a hierarchical priority to the control objectives, the relation of w_i between objectives is an order of magnitude.

Table 4.3 relates the auxiliary variables z with the control objectives discussed in Section 4.2.3. A suitable norm for the first objective is the ∞-norm. This minimizes the worst-case street flooding at each sample. On the other hand, for the second objective, the 1-norm is more suitable. As the variables are all positive, the following outputs are defined to implement the 1-norm indirectly with the ∞-norm-based cost function

$$y_1 = \sum_i z_{\text{sea}}(i)$$

$$y_2 = z_{41} \quad \text{and} \quad y_3 = z_{43}$$

Performance Improvement

Taking into account that the system performance in open loop for the considered rain has a flooding volume of 108×10^3, a pollution volume of 225.9×10^3 and a volume of treated water of 278.3×10^3 (all in m^3), the improvement obtained is between 4.5 % and 22.1 % for the first objective and the other objectives keep almost the same values for most of the cases, fulfilling the desired objective prioritization.

Table 4.4 summarizes the results for ten of the more representative rain episodes in Barcelona between 1998 and 2002. The results were obtained considering $p = 2$, a cost function containing only output variables and with the weight of the most important of its terms

Table 4.3. Relation between z variables and control objectives

Objective	z vector	z variable	Description
1	z_{str_v}	z_2	overflow in T_1
		z_6	overflow in T_2
		z_{10}	overflow in T_4
		z_{12}	overflow in T_9
		z_{20}	overflow in T_5
		z_{22}	overflow in T_6
		z_{24}	overflow in T_7
		z_{26}	overflow in T_{12}
		z_{32}	overflow in T_8
		z_{36}	overflow in T_{10}
		z_{40}	overflow in T_{11}
2	z_{str_q}	z_4	overflow in q_{14}
		z_8	overflow in q_{24}
		z_{14}	overflow in q_{96}
		z_{18}	overflow in q_{945}
		z_{30}	overflow in $q_{12\text{s}}$
		z_{34}	overflow in q_{c210}
3	z_{sea}	z_{29}	flow to environment ($q_{12\text{s}}$)
		z_{35}	flow to sea ($q_{10\text{M}}$)
		z_{38}	flow to sea ($q_{8\text{M}}$)
		z_{42}	flow to sea ($q_{11\text{M}}$)
		z_{44}	flow to sea ($q_{7\text{M}}$)
4	—	z_{43}	flow to Llobregat WWTP
	—	z_{41}	flow to Besòs WWTP

set as $w_{\text{str}_v} = 10^{-2}$. The system performance in general is improved when the hybrid control strategy is applied (see percentages for some values).

4.4 Discrepancies and Misfits

4.4.1 Computational Complexity

The results obtained in the simulation study of the previous section show that important performance improvements can be accomplished when HMPC is applied to sewer networks. Furthermore, the hybrid modeling methodology is very rich and allows the straightforward treatment of hybrid phenomena such as overflow and flooding.

However, the underlying optimization problem of HMPC is combinatorial and $\mathcal{NP}$-hard [40]. The worst-case computation time is exponential in the number of logical variables. Figure 4.13 shows how this problem appears in the case of the HMPC applied on the Barcelona sewer-network case study. In the top graph, rain intensity measured by the five rain gauges

Table 4.4. Obtained results of closed-loop performance using 10 representative rain episodes

Rain episodes	Open loop			Closed loop		
	Flooding $\times 10^3$ (m^3)	Pollution $\times 10^3$ (m^3)	Treated w. $\times 10^3$ (m^3)	Flooding $\times 10^3$ (m^3)	Pollution $\times 10^3$ (m^3)	Treated w. $\times 10^3$ (m^3)
14-09-1999	108	225.8	278.4	92.9 (14 %)	223.5	280.7
09-10-2002	116.1	409.8	533.8	97.1 (16 %)	398.8	544.9
03-09-1999	1	42.3	234.3	0 (100 %)	44.3	232.3
31-07-2002	160.3	378	324.4	139.7 (13 %)	374.6	327.8
17-10-1999	0	65.1	288.4	0	58.1 (11 %)	295.3
28-09-2000	1	104.5	285.3	1	98 (6 %)	291.9
25-09-1998	0	4.8	399.3	0	4.8	398.8
22-09-2001	0	25.5	192.3	0	25	192.4
01-08-2002	0	1.2	285.8	0	1.2	285.8
20-04-2001	0	35.4	239.5	0	32.3 (9 %)	242.5

available in this part of the network is presented for the critical portion (second rain peak) of the rain episode that occurred on October 17, 1999. This episode was relatively intensive with a return rate of 0.7 years within the city of Barcelona. In the second graph, the computation time to solve the MIP associated with the HMPC is shown as a function of time for the same scenario. Recalling that the desired sampling time for this system is 300 s, it can be seen that the MIP solver is unable to find the optimum within the desired sampling time. Furthermore, it is seen that computation time varies greatly. Before sample 16 the calculation time is very small.

If optimality is not achieved within a desired sampling time, feasibility is at least required. Often feasibility is sufficient for proof of stability of MPC scheme, see [35]. The ILOG CPLEX solver used in the current application can be configured to put special emphasis on finding a feasible solution before an optimal one [1]. It is also possible to limit the time the solver has to solve the problem at hand. In the second graph of Figure 4.13, the time required to find a feasible solution is shown. It was found by iteratively increasing the maximum solution time allowed for the solver until a feasible solution was found. The feature of CPLEX to put emphasis on finding feasible solution was activated. Again, it can be seen that the time required to find a feasible solution varies considerably. Furthermore, it should be stated that the feasible solutions found were often of such poor quality that running the system in open loop often yielded better performance.

In the current application, the MIQP problem solved in each sample has the following generical form:

$$\min_{\rho} \quad \rho^T H \rho + f^T \rho \tag{4.29}$$

$$\text{s.t} \quad A\rho \leq b + Cx_0 \tag{4.30}$$

where vector x_0 collects the system initial conditions and predicted disturbances (rain), which is the only thing that changes from sample to sample. The ability of the MIP solver to reduce computation time from the worst case depends on its ability to exclude from consideration as many nodes as possible when branching and bounding. This is done either by proving them to be infeasible or that their solution is suboptimal to other solutions. The increase in computation time is thus linked to an increase in the amount of feasible nodes. In the bottom

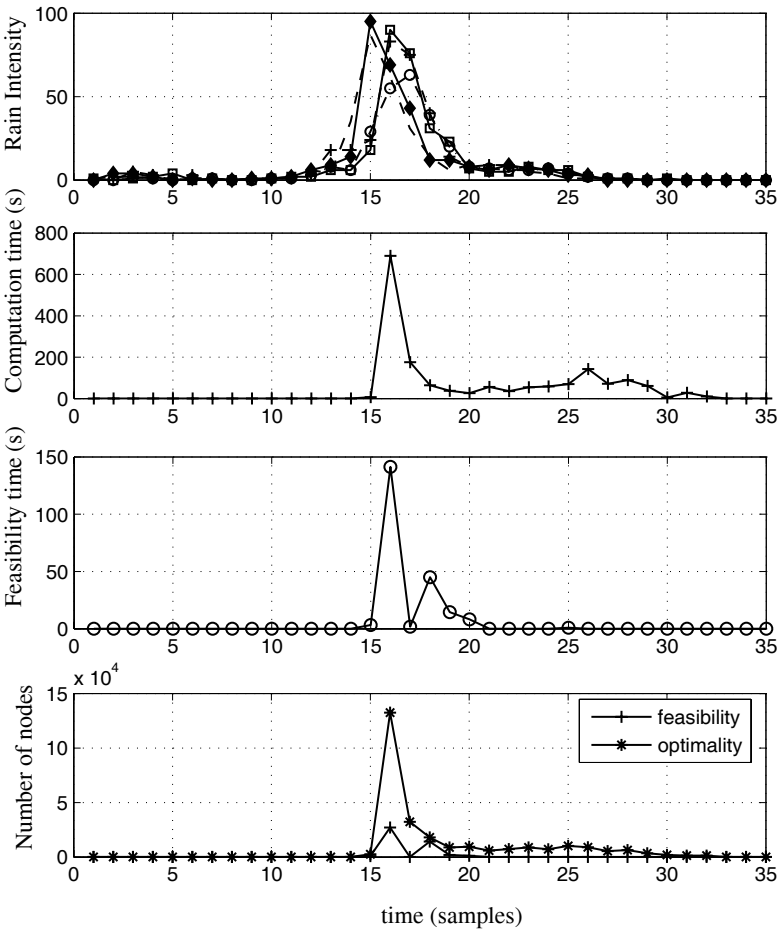


Fig. 4.13. MIP problem characteristics for the rain episode occurred on October 17, 1999

graph of Figure 4.13, the number of nodes the CPLEX solver explored during branching is shown. It is seen that there was a huge increase in the number of explored nodes between samples. There is thus a dramatic change in the complexity of the optimization problem for certain values of x_0.

Physical insight into the network dynamic behavior can explain the increase in complexity at time 11. At that time, due to the rain, many of the virtual tanks are very close to their overflow limit. This in turn means that more trajectories for distinct switching sequences Δ_k are feasible. Similar behavior was encountered in other rain episodes and when other cost functions were used.

The same problem regarding computation time occurs in nonlinear MPC, see [32]. When the optimization problem is no longer convex, a fundamental question is how long will the optimization take and whether the quality of the solution would be sufficient to justify the application of the MPC control approach.

It should be noted that it is common engineering practice to model the hybrid dynamics of sewer systems with *max* or *min* functions and apply nonlinear MPC schemes [34]. However, the problem of finding a feasible solution is not avoided and the nonlinear programming algorithms applied reach only a local optima which implies a suboptimal solution. Generally, heuristical methods are used to find an initial feasible solution.

4.4.2 Strategies to Deal with the Complexity in HMPC

Control strategies have been proposed where the HMPC problem is relaxed to make it computationally tractable. In [12] a *decentralized control* approach to HMPC was presented. The class of systems considered were those made up of dynamically uncoupled subsystems but where global control objectives were formulated with a global cost function.

A number of authors have also presented methods where the intent is to reduce complexity offline. In [11] an explicit solution to the constrained finite-time optimal control problem was presented for discrete-time linear hybrid systems. *Mode enumeration* (ME) [3, 23, 24] is an offline technique to compute and enumerate explicitly the feasible modes of piecewise affine PWA models. By pruning redundant modes where the dynamics are the same as in other modes, the complexity of the hybrid model can be reduced. Such methods are usually limited to systems of relatively small size.

The main source of complexity in the MIP problem is its combinatorial nature. In [27], a HMPC strategy was presented where additional restrictions were added to the MIP problem to reduce the number of possible combinations of the Boolean variables. Infeasibility is avoided by restricting the number of combinations around a nominal feasible trajectory.

4.4.3 Phase Transitions in MIP problems

The performance of MIP solvers has improved greatly in recent years [9]. The size limit of problems considered to be practically solvable has increased steadily. Part of the reason lies in the many orders of magnitude improvement of desktop computing power over the years. But there has also been tremendous improvement in solution algorithms for LPs and QPs, which are a cornerstone of MIP solvers [8]. Furthermore, modern solvers have incorporated many performance-improving features that have existed in the literature such as cutting-plane capabilities. Generally, the solvers apply a barrage of techniques on each problem. A recent improvement in solving the optimal control problem of HMPC by using symbolic techniques to solve constraint satisfactions problems (CSP), was presented in [6].

The MIP problem is still NP-complete and equivalent to the archetypal NP-complete K-satisfiability problem [37] (or the ZERO ONE INTEGER PROGRAMMING problem (ZIOP), see [10]). It has recently been shown that K-satisfiability problems exhibit phase-transitions in terms of computational difficulty and solution character when these aspects are considered as a function of parameters such as the ratio of number of constraints to number of variables. Away from the phase-transition region the problems become easier to solve, see [37] and references therein. This is of interest as it can give information about how the problem can be modified to distance it from the phase-transition region where it is most difficult to solve. *Phase-transition* behavior has been reported in for example multivehicle task-assignment problems, see [20].

4.4.4 Conclusions

Results presented in this chapter have shown that HMPC is a suitable control strategy for sewer networks. The hybrid modeling framework presented can take into account several inherent phenomena and elements that result in distinct behavior depending on the state of the network.

However, the exponential increase in computational complexity due to the NP-hard nature of the underlying HMPC optimization problems is an issue that should be addressed in order to face time constraints imposed by real-time implementation. For the case study presented, it has been shown that the computational complexity of the MIP problem related to the HMPC scheme can vary considerably depending on the measured state and disturbances.

Some techniques to address the computational complexity have been mentioned. Moreover, the existence of phase transitions in similar combinatorial problems should be explored to gain information on how to modify the optimization problem to move it to regions of less computational complexity.

Acknowledgements

The support received by the Research Commission of the *Generalitat de Catalunya* (ref. 2005SGR00537) and the Spanish CICYT (ref. DPI2006-11944) are gratefully acknowledged. We also appreciate the support provided by CLABSA.

References

[1] (2003). *ILOG CPLEX 9.1 User´s Manual*.

[2] Ames, A. and Sastry, S. (2005). Characterization of Zeno behavior in hybrid systems using homological methods. *Proceedings of the IEEE American Control Conference*, pp. 1160–1165. Portland, OR, USA.

[3] Bemporad, A. (2004). Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE Transactions on Automatic Control* **49**(5), 832–838.

[4] Bemporad, A. (2006). *Hybrid Toolbox - User's Guide*.

[5] Bemporad, A., Borrelli, F., and Morari, M. (2002). *Hybrid Systems: Computation and Control*, volume 228 of *Lecture Notes of Computer Science*, chapter On the optimal control law for linear discrete time hybrid systems, pp. 105–119. Springer, Berlin.

[6] Bemporad, A. and Giorgetti, N. (2006). Logic-based solution methods for optimal control of hybrid systems. *IEEE Transactions on Automatic Control* **51**(6), 963–976.

[7] Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica* **35**(3), 407–427.

[8] Bixby, R. (2002). Solving real-world linear programs: a decade and more of progress.

[9] Bixby, R., Fenelon, M., Gu, Z., Rothberg, E., and Wunderling, R. (2000). MIP: Theory and practice – closing the gap .

[10] Blondel, V. and Tsitsiklis, J. (2000). A survey of computational complexity results in systems and control. *Automatica* **36**, 1249–1274.

[11] Borrelli, F., Baotić, M., Bemporad, A., and Morari, M. (2005). Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica* **41**, 1709–1721.

[12] Borrelli, F., Keviczky, T., Balas, G., Stewart, G., Fregene, K., and Godbole, D. (2005). *Hybrid Systems: Computation and Control*, volume 3414, chapter Hybrid Decentralized Control of Large Scale Systems, pp. 168–183. Springer Berlin / Heidelberg.

[13] Branicky, M. and Zhang, G. (2000). Solving hybrid control problems: Level sets and behavioral programming. *Proceedings of the IEEE American Control Conference*.

[14] Branicky, M. S., Borkar, V. S., and Mitter, S. K. (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control* **43**(1), 31–45.

[15] Camacho, E. and Bordons, C. (2004). *Model Predictive Control*. Springer-Verlag, London, second edition.

[16] Cembrano, G. and Quevedo, J. (1999). *Optimization in Water Networks*. Research Studies Press.

[17] Cembrano, G., Quevedo, J., Salamero, M., Puig, V., Figueras, J., and Martí, J. (2004). Optimal control of urban drainage systems: a case study. *Control Engineering Practice* **12**(1), 1–9.

[18] CLABSA, C. d. B. S. (2005). Homepage. http://www.clabsa.es/.

[19] Duchesne, S., Mailhot, A., Dequidt, E., and Villeneuve, J. (2001). Mathematical modeling of sewers under surcharge for real time control of combined sewer overflows. *Urban Water* **3**, 241–252.

[20] Earl, M. and D'Andrea, R. (2005). Phase transitions in the multi-vehicle task assignment problem. *Proceedings of IMECE2005 2005 ASME International Mechanical Engineering Congress and Exposition*.

[21] Ermolin, Y. (1999). Mathematical modelling for optimized control of Moscow's sewer network. *Applied Mathematical Modelling* **23**, 543–556.

[22] Gelormino, M. and Ricker, N. (1994). Model-predictive control of a combined sewer system. *International Journal of Control* **59**, 793–816.

[23] Geyer, T. (2005). *Low Complexity Model Predictive Control in Power Electronics and Power Systems*. Ph.D. thesis.

[24] Geyer, T., Torrisi, F., and Morari, M. (2003). Efficient Mode Enumeration of Compositional Hybrid Models. *Hybrid Systems: Computation and Control* **2623**, 216–232.

[25] Heemels, W., De Schutter, B., and Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica* **37**, 1085–1091.

[26] Heymann, M., Lin, F., Meyer, G., and Resmerita, S. (2002). *Analysis of Zeno Behaviors in Hybrid Systems*. Technical report, Technion - Computer Science Department.

[27] Ingimundarson, A., Ocampo-Martinez, C., and Bemporad, A. (2007). Suboptimal model predictive control of hybrid systems using mode sequence constraints. Submitted to CDC 2007.

[28] Lazar, M., Heemels, W., Weiland, S., and Bemporad, A. (2006). Stability of hybrid model predictive control. *IEEE Transactions on Automatic Control* **51**(11), 1813 – 1818.

[29] Lincoln, B. and Rantzer, A. (2001). Optimizing linear system switching. *Proceedings of the 40th IEEE Conference on Decision and Control*, pp. 2063–2068.

[30] Ljung, L. (1999). *System Identification—Theory for the User, Second Edition*. Prentice Hall.

[31] Lygeros, J., Tomlin, C., and Sastry, S. (1999). Controllers for reachability specifications for hybrid systems. *Automatica* **35**(3), 349–370.

[32] Maciejowski, J. (2002). *Predictive Control with Constraints*. Prentice Hall, Great Britain.

[33] Marinaki, M. and Papageorgiou, M. (1998). Nonlinear optimal flow control for sewer networks. *Proceedings of the IEEE American Control Conference*, volume 2, pp. 1289–1293.

[34] Marinaki, M. and Papageorgiou, M. (2005). *Optimal Real-time Control of Sewer Networks*. Springer.

[35] Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica* **36**, 789–814.

[36] Mays, L. (2004). *Urban Stormwater Management Tools*. McGrawHill.

[37] Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B., and Troyansky, L. (1999). Determining computational complexity from characteristic "phase transitions". *Nature* **400**, 133–137.

[38] Ocampo-Martinez, C., Puig, V., Quevedo, J., and Ingimundarson, A. (2005). Fault tolerant model predictive control applied on the Barcelona sewer network. *Proceedings of IEEE Conference on Decision and Control (CDC) and European Control Conference (ECC)*.

[39] Ocampo-Martínez, C., Ingimundarson, A., Puig, V., and Quevedo, J. (2007). Objective prioritization using lexicographic minimizers for MPC of sewer networks. *IEEE Transactions on Control Systems Technology* (Accepted for publication).

[40] Papadimitriou, C. (1994). *Computational Complexity*. Addison-Wesley.

[41] Pleau, M., Colas, H., Lavallée, P., Pelletier, G., and Bonin, R. (2005). Global optimal real-time control of the Quebec urban drainage system. *Environmental Modelling & Software* **20**, 401–413.

[42] Pleau, M., Methot, F., Lebrun, A., and Colas, A. (1996). Minimizing combined sewer overflow in real-time control applications. *Water Quality Research Journal of Canada* **31**(4), 775 – 786.

[43] Schilling, W., Anderson, B., Nyberg, U., Aspegren, H., Rauch, W., and Harremöes, P. (1996). Real-time control of wastewater systems. *Journal of Hydraulic Resources* **34**(6), 785–797.

[44] Schütze, M., Butler, D., and Beck, B. (2002). *Modelling, Simulation and Control of Urban Wastewater Systems*. Springer.

[45] Schütze, M., Campisanob, A., Colas, H., W.Schillingd, and Vanrolleghem, P. (2004). Real time control of urban wastewater systems: Where do we stand today? *Journal of Hydrology* **299**, 335–348.

[46] Schütze, M., To, T., Jaumar, U., and Butler, D. (2002). Multi-objective control of urban wastewater systems. *Proceedings of 15th IFAC World Congress*.

[47] Schutter, B. D. (1999). Optimal control of a class of linear hybrid systems with saturation. *Proceedings of 38th IEEE Conference on Decision and Control*, pp. 3978–3983.

[48] Singh, V. (1988). *Hydrologic systems: Rainfall-runoff modeling*, volume I. Prentice-Hall, N.J.

[49] Smith, K. and Austin, G. (2000). Nowcasting precipitation - A proposal for a way forward. *Journal of Hydrology* **239**, 34–45.

[50] Tomlin, C., Lygeros, J., and Sastry, S. (2000). A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, volume 88, pp. 949–970.

[51] Torrisi, F. and Bemporad, A. (2004). Hysdel - A tool for generating computational hybrid models for analysis and synthesis problems. *IEEE Transactions on Control Systems Technology* **12**(2), 235–249.

[52] Weyand, M. (2002). Real-time control in combined sewer systems in Germany: Some case studies. *Urban Water* **4**, 347 – 354.

[53] Yuan, J., Tilford, K., Jiang, H., and Cluckie, I. (1999). Real-time urban drainage system modelling using weather radar rainfall data. *Physics and Chemistry on the Earth (B)* **24**, 915–919.

Part II

Aerospace

5

Nonlinear System Identification of Aeroelastic Systems: A Structure-detection Approach

Sunil L. Kukreja and Martin J. Brenner*

NASA Dryden Flight Research Center, Aerostructures Branch, Edwards, California, USA
{Sunil.Kukreja,Martin.J.Brenner}@nasa.gov

Summary. Identification methods for NARMAX models are applied to aeroelastic dynamics and its properties demonstrated via continuous-time simulations and experimental conditions. Identification of parametric nonlinear models involves estimating unknown parameters and detecting its underlying structure. Structure computation is concerned with selecting a subset of parameters to give a parsimonious description of the system that may afford greater insight into the functionality of the system or a simpler controller design. Structure-detection methods applicable to NARMAX modeling are applied to aeroelastic dynamics. Among other methods, a least absolute shrinkage and selection operator (LASSO) technique is investigated for computing efficient model descriptions of nonlinear systems. Simulation results from a nonlinear dynamic aircraft model demonstrate that methods developed for NARMAX structure computation provide good accuracy for selection of the exact model structure from an over-parameterized model description. Applicability of the method to more complex systems such as those encountered in aerospace applications is shown by identifying parsimonious system descriptions of the F/A-18 active aeroelastic wing (AAW) using flight-test data.

Key words: Nonlinear system identification, structure detection, limit cycle oscillation (LCO), flutter, aeroelasticity, NARMAX, parametric models, mathematical modeling

5.1 Background

5.1.1 Introduction

System identification or mathematical modeling is the process of developing or improving a mathematical representation of a physical system based on observed data. System identification is a critical step in aircraft development, analysis and validation for flight worthiness. One such application of system identification in the aerospace community is for the analysis of

aeroelasticity. Aeroelasticity is concerned with the interaction of inertia, structural and aerodynamic forces [38]. Previous approaches have modeled aeroelasticity with linear time-invariant (LTI) models [49]. These linear models have been successful in providing approximate estimates of an aircraft's response to gust, turbulence and external excitations [48]. However, at high angle-of-attack or transonic flight linear models no longer provide accurate predictions of the aircraft's behavior. Some of the behavior that cannot be modeled linearly includes air-flow separation and transonic effects that can induce nonlinear phenomena such as limit cycle oscillations (LCO) [11, 12]. The onset of LCOs has been observed on several aircraft such as the F-16C or F/A-18 and cannot be modeled properly as a LTI system [17]. This has necessitated the application of nonlinear identification techniques to accurately model LCO dynamics.

Significant achievements have been made in several areas of nonparametric nonlinear system identification ([27, 33, 45]). Recent work in the aerospace community has attempted to address nonlinear aeroelastic phenomena using Volterra kernel methods [42]. These methods provide a convenient means of characterizing LCOs but suffer from a highly overparameterized model description and do not lend themselves to efficient control synthesis.

Parametric representations of nonlinear systems typically contain a small number of coefficients that can be varied to alter the behavior of the equation and may be linked to the underlying system. Leontaritis and Billings [39, 40] have proposed the NARMAX (nonlinear autoregressive, moving average exogenous) structure as a general parametric form for modeling nonlinear systems. NARMAX models describe nonlinear systems in terms of linear-in-the-parameters difference equations relating the current output to (possibly nonlinear) combinations of inputs and past outputs. They are suitable for modeling both the stochastic and deterministic components of a system and capable of describing a wide variety of nonlinear systems [5, 13]. NARMAX models have been successfully demonstrated for modeling the input–output behavior of many complex systems such as ones found in engineering and biology [14, 35], and offer a promising framework for describing nonlinear behavior such as aeroelastic aircraft dynamics.

Often, this formulation yields compact model descriptions that may be readily identified and afford greater interpretability. This system representation, however, can yield a large number of possible terms required to represent the dynamic process [6]. In practice many of these candidate terms are insignificant and can be removed. Consequently, the structure-detection problem is that of selecting a subset of candidate terms that best predicts the output while maintaining an efficient system description.

There are two fundamental approaches to the structure-detection problem: (1) exhaustive search, where every possible subset of the full model is considered ([24, 58]), or (2) parameter variance, where the covariance matrix, P_θ, based on input–output data and estimated residuals is used to assess parameter relevance ([51]). Both have problems. Exhaustive search requires a large number of computations, whereas parameter-variance estimates are often inaccurate when the number of candidate terms is large.

A bootstrap method has been proposed to solve the structure-detection problem for overparameterized models [36]. Although it has been demonstrated that the bootstrap is a useful tool for structure detection of NARMAX models, there is a limitation of the model complexity that can be studied with this technique as a result of the large number of candidate terms, for a given model order, and the data length required to guarantee convergence. It was demonstrated that a necessary condition for bootstrap structure detection to yield accurate results is the number of data points needed for identification be *at least* 10 times the square of the initial number of candidate terms.

For many practical systems, collecting large data records may be financially and/or technically infeasible. Estimation techniques used for NARMAX identification all require an over-

determined system of equations to solve for the unknown system parameters. Due to the large number of possible candidate terms and limited data records available for any practical identification problem, it may not be feasible to analyze highly complex systems with the bootstrap technique. Nonlinear aeroelastic dynamics of aircraft are highly complex processes likely involving a large number of candidate terms that may not be accurately characterized by current approaches.

System identification is a critical step in aircraft development, analysis and validation for flight worthiness. The development and testing of aircraft typically takes many years and requires considerable expenditure of limited resources. One reason for lengthy development time/costs is inadequate knowledge of an appropriate model type or structure to use for parameter estimation. In aerospace systems analysis one of the main objectives is not only to estimate system parameters but to gain insight into the structure of the underlying system. Selection of an insufficient model structure may lead to difficulties in parameter estimation, giving estimates with significant biases and/or large variances [43]. Structure computation may indicate deficiencies in an analytical model and could lead to improved modeling strategies and also provide a parsimonious, blackbox, system description of complex aeroelastic behavior over a large operating range for control synthesis [30]. Therefore, structure computation is of significant relevance and importance to modeling, analysis, and design of aircraft and aerospace vehicles.

Although the NARMAX structure is well suited to modeling the input–output behavior of aeroelastic systems, it has not been investigated by the aerospace community. Therefore, a nonlinear pitch-plunge model of aircraft aeroelastic dynamics will be analyzed to derive its NARMAX representation and assess the applicability of NARMAX structure detection methods to a simulated model of aircraft freeplay. F/A-18 active aeroelastic wing (AAW) [56] flight-test data will then be investigated. Finally the application of a novel method for NARMAX model identification via a least absolute shrinkage and selection operator (LASSO) [64] will be demonstrated. This approach permits identification of NARMAX models in situations where current methods cannot be applied.

The NARMAX model structure is described in Section 5.1.2. Section 5.1.3 summarizes an estimation technique appropriate for NARMAX model identification, while Section 5.1.4 reviews commonly used techniques for assessing parameter significance as well as LASSO and its application as a tool for structure computation.

Section 5.2 presents a continuous-time representation of a nonlinear pitch-plunge model describing aircraft dynamics and its NARMAX representation is derived. Simulation results of several NARMAX structure detection algorithms are presented in Section 5.2.2, Section 5.2.3 illustrates application to experimental F/A-18 AAW data, and Section 5.2.4 provides a discussion and summary using the more common methods for structure detection. In Section 5.3, simulation results of LASSO's performance as a structure detection instrument are presented in Section 5.3.1, while in Section 5.3.2 application results to AAW flight-test data is presented and Section 5.3.3 discusses and summarizes LASSO results.

5.1.2 NARMAX Model Form

Kolmogorov–Gabor polynomials have been well known in control engineering for many years [22, 65]. However, these equations have recently been popularized by Billings and coworkers [39, 40] for use in identification, modeling, and control. The NARMAX (nonlinear autoregressive, moving average exogenous) structure is a general parametric form for modeling nonlinear systems [39]. This so-called NARMAX structure describes both the stochastic and deterministic components of nonlinear systems. Many nonlinear systems are a special case of the general

NARMAX structure [13]. The NARMAX structure models the input-output relationship as a nonlinear difference equation of the form

$$z(n) = f[z(n-1), \cdots, z(n-n_z), u(n), \cdots, u(n-n_u), e(n-1), \cdots, e(n-n_e)] + e(n) \tag{5.1}$$

where f denotes a nonlinear mapping, u is the controlled or exogenous input, z is the measured output, and e is the uncontrolled input or innovation. For the special case where f is a nonlinear mapping of polynomial form it may include a variety of nonlinear terms, such as terms raised to an integer power ($u^3(n-3)$), products of present and past inputs ($u(n)u(n-2)$), past outputs ($z^2(n-3)z(n-5)$), or crossterms ($u(n-1)z^2(n-3)$). In general, the nonlinear mapping, f, can be described by a wide variety of nonlinear functions such as sigmoids or splines [5, 13].

For simplicity of the present discussion, nonlinearities are assumed to be described by a polynomial expansion. Consider the following NARMAX model that is described by a polynomial expansion as

$$z(n) = \theta_1 z^2(n-2) + \theta_2 u^3(n-1) + \theta_3 u^2(n-1)u(n-2) + \theta_4 z(n-2)e(n-2) + \theta_5 e^2(n-2) + e(n) \tag{5.2}$$

This model contains a variety of nonlinear terms (regressors) but is *linear-in-the-parameters* and, therefore, pseudolinear regression techniques can be used for parameter estimation [4, 8, 26, 58]. The order of a polynomial class NARMAX model may be defined as

$$O = [n_u \; n_z \; n_e \; l] \tag{5.3}$$

where n_u is the maximum input lag, n_z the maximum output lag, n_e the maximum error lag and l the maximum nonlinearity order. For a nonpolynomial-class NARMAX model, l is simply replaced by an appropriate basis function. The maximum number of terms in a NARMAX model with n_u, n_z and n_e dynamic terms and lth-order nonlinearity is [8]

$$p = \sum_{i=1}^{l} p_i + 1; \tag{5.4}$$

$$p_i = \frac{p_{i-1}(n_u + n_z + n_e + i)}{i}, \quad p_0 = 1$$

For nonlinear systems, output additive noise can produce multiplicative terms between input, output and itself. To compute unbiased parameter estimates a noise model needs to be estimated. As a result, the number of candidate terms can become large for even moderately complex models making structure detection difficult. For example, consider model (5.2), which is of order $O = [2\ 2\ 2\ 3]$. A model of this order has $p = 120$ candidate terms to be initially considered for identification. Due to excessive parameterization (the curse of dimensionality), structure detection often leads to computationally intractable combinatorial optimization problems.

To perform structure detection, estimates of the unknown system parameters and their statistics are needed. An estimate of unknown system parameters can be obtained using standard prediction error (PE) identification techniques such as extended least-squares. Methods to obtain estimates of the parameter statistics for NARMAX models are discussed in Section 5.1.4.

5.1.3 Parameter Estimation

The structure-detection procedure begins by computing an unbiased estimate of the unknown system parameters. Although NARMAX models provide a concise system representation, any noise on the output enters the model as product terms with the system input and output [8]. Consequently, most parameter-estimation algorithms for linear systems cannot be applied directly because they assume that the noise terms in the model are independent [8, 43].

Extended least-squares (ELS) is one method appropriate for NARMAX models that easily enables unbiased estimates to be computed. ELS is a technique that addresses the bias problem by modeling the lagged errors to obtain unbiased parameter estimates. ELS for linear systems has been widely studied and is also referred to as Panuska's method, the extended matrix method, or approximate maximum likelihood [53, 54, 66].

In general, since the noise sequence is a realization of a stochastic process, it is not possible to solve for the noise source e, and it will not be equal to the prediction errors [8]. The prediction errors, $\hat{\boldsymbol{\epsilon}} \in \mathbb{R}^{N\times 1}$, are defined as

$$\hat{\boldsymbol{\epsilon}} = \mathbf{Z} - \hat{\mathbf{Z}} \tag{5.5}$$

where $\mathbf{Z} \in \mathbb{R}^{N\times 1}$ is the measured output and $\hat{\mathbf{Z}} = \boldsymbol{\Psi}\hat{\boldsymbol{\theta}} \in \mathbb{R}^{N\times 1}$ is the predicted output. In ELS, the NARMAX formulation of Equation (5.1) is redefined into a prediction-error model by replacing e with $\hat{\epsilon}$; making it a deterministic least-squares problem.

The ELS formulation is defined as

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\Psi}^T\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^T\mathbf{Z}, \qquad \boldsymbol{\Psi} = [\boldsymbol{\Psi}_{zu}\boldsymbol{\Psi}_{zu\hat{\epsilon}}\boldsymbol{\Psi}_{\hat{\epsilon}}] \tag{5.6}$$

where $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{p\times 1}$ is an ELS estimate of the system parameters, and $\boldsymbol{\Psi} \in \mathbb{R}^{N\times p}$ is a partitioned regressor matrix in which $\boldsymbol{\Psi}_{zu}$ is a function of z and u only, $\boldsymbol{\Psi}_{zu\hat{\epsilon}}$ represents all the crossproducts involving $\hat{\epsilon}$, and $\boldsymbol{\Psi}_{\hat{\epsilon}}$ is a polynomial function of the prediction errors only [8].

Often ELS is considered a pseudolinear approach to parameter estimation [26, 53, 66]. Strictly speaking, the introduction of prediction errors into the model formulation no longer makes the model linear-in-the-parameters because the prediction errors depend on the model output that is a function of all model parameters. The ELS technique solves a nonlinear optimization problem by ignoring the nonlinear character of the model and employing a least-squares approach. Essentially, ELS uses an approximate gradient of the model output with respect to the model parameters as a regression vector.

The prediction-error method (PEM) is another approach to parameter estimation for nonlinear systems. PEM implements the true gradient to solve the nonlinear optimization problem. ELS is an approximation of PEM. Here, ELS is chosen over PEM because it does not require the computation of the true gradient. Computing the true gradient may be difficult in the general case where the objective is to learn about the true form of the underlying system. Once an estimate of the system parameters is computed, an estimate of their statistics is needed to determine structure.

5.1.4 Structure Detection

In system identification it is necessary to form estimates of the unknown parameters governing a random process using a finite set of sample values. These can be computed using the ELS estimator. Parameter statistics are also needed to make a probability statement with respect to the unknown true parameter values. One such probability statement is to assign two limits to a

parameter and assert that, with some specified probability, the true value of the parameter will be situated between those limits, which constitute the confidence interval.

Structure detection is concerned with how to best make a probability statement of the parameter estimates while maintaining an efficient (parsimonious) system description and yet complex enough to predict the system output. Many methods for structure detection have been proposed over the years. In this section, techniques appropriate for NARMAX modeling are discussed that are (i) easy to implement and (ii) have previously shown promising results for polynomial NARMAX models. For a survey of the structure-detection literature the reader is referred to [29].

t-test

The t-test in combination with regression analysis is sometimes referred to as a form of hypothesis testing by computing the differences between means [19]. In regression, the significance of each parameter estimate is checked using a statistical measure. If the model that was postulated is more general than needed, tests of hypothesis are necessary to give a minimal model description.

Suppose the following model was fit

$$E(\mathbf{Z}) = \hat{\theta}_1 + \hat{\theta}_2\psi_2 + \hat{\theta}_3\psi_3 + \cdots + \hat{\theta}_p\psi_p \tag{5.7}$$

The $\hat{\boldsymbol{\theta}}$s are tested against the null hypothesis, H_0, that $\hat{\theta}_i = 0, i = 1, 2, \ldots, p$. This allows the experimenter to assess which parameters are significant and which are not, and consequently which ones to retain.

Often in engineering applications, it is assumed that the errors are Gaussian independent, identically distributed (iid) variables. This assumption implies that $\mathbf{Z}$ will be Gaussian iid with mean $\boldsymbol{\Psi}\boldsymbol{\theta}_0$ and variance $\sigma^2\mathbf{I}$,

$$\mathbf{Z} \in \mathcal{N}(\boldsymbol{\theta}_0, \sigma^2\mathbf{I}) \tag{5.8}$$

where $\mathcal{N}(\cdot)$ denotes the Gaussian (normal) distribution, $\boldsymbol{\theta}_0$ the true unknown parameter vector, σ^2 the variance and $\mathbf{I}$ is an identity matrix of appropriate dimension [43]. The estimate $\hat{\boldsymbol{\theta}}$ is also Gaussian iid because it is a linear combination of $\mathbf{Z}$,

$$\hat{\boldsymbol{\theta}} \in \mathcal{N}(\boldsymbol{\theta}_0, \mathbf{P}) \tag{5.9}$$

where $\mathbf{P} = \sigma^2(\boldsymbol{\Psi}^T\boldsymbol{\Psi})^{-1}$. While the matrix $(\boldsymbol{\Psi}^T\boldsymbol{\Psi})^{-1}$ is known to the user, σ^2 is typically not known, and must be estimated as

$$\hat{\sigma}^2 = \frac{1}{N-p}\left(\mathbf{Z} - \boldsymbol{\Psi}\hat{\boldsymbol{\theta}}\right)^T\left(\mathbf{Z} - \boldsymbol{\Psi}\hat{\boldsymbol{\theta}}\right) \in \chi^2(N-p) \tag{5.10}$$

where $\chi^2(N-p)$ denotes the Chi-squared distribution with $N-p$ degrees of freedom, and thus $\hat{\mathbf{P}} = \hat{\sigma}^2(\boldsymbol{\Psi}^T\boldsymbol{\Psi})^{-1}$.

These estimates are easily calculated as a byproduct to the regression procedure and lend themselves to forming a confidence interval of the parameters via the Student's t-distribution, defined as the ratio of a normal random variable divided by the square root of a Chi-squared random variable

$$t = \frac{\mathcal{N}(\hat{\boldsymbol{\theta}}_i, \hat{\mathbf{P}})}{\sqrt{\hat{P}_{ii}}} \tag{5.11}$$

where $\hat{P}_{ii}$ denotes the ith diagonal element of $\hat{\mathbf{P}}$.

Specifically, the confidence interval for an estimate, $\hat{\theta}_i$, is

$$\hat{\theta}_i \pm t(\gamma/2, N-p)\hat{P}_{ii} \tag{5.12}$$

where t is the tabulated t ratio at the $\gamma/2$ level of significance $(0 \leq \gamma \leq 1)$ with $N-p$ degrees of freedom. Therefore, the user is able to assess with $(1-\gamma)$ % confidence that the parameter lies within this range. If the interval includes zero, $\hat{\theta}_i$ is not significantly different from zero at the γ level and can be removed from the model.

This procedure assumes that an accurate estimate of parameter variances, *i.e.* residuals, is available [58]. Although this assumption is violated for overparameterized models, structure detection of moderately overparameterized polynomial NARMAX models have shown good results [36].

Stepwise Regression

The stepwise regression algorithm is a widely used tool for computing model structure [21]. Stepwise regression relies on the incremental change in the residual sums-of-squares (RSS) resulting from adding or removing a parameter. Two Fisher-Snedecor distribution levels (F-levels), F_{out} and F_{in}, are formed to determine whether a parameter should be removed from the model (F_{out}) or included in the model (F_{in}) [21, 58, 60, 61]. These F-levels are based on an F-distribution with $N-p$ degrees of freedom for a predetermined γth level of significance. The statistics F_{in} and F_{out} are estimated from the residual sums-of-squares for a model with p parameters as

$$F_{\text{in}} = \frac{\text{RSS}_p - \text{RSS}_{p+1}}{\text{RSS}_{p+1}/(N-p-1)} \quad \text{and} \quad F_{\text{out}} = \frac{\text{RSS}_{p-1} - \text{RSS}_p}{\text{RSS}_p/(N-p)} \tag{5.13}$$

For good model parameterizations, F_{out} must not be greater than F_{in} [21, 47, 60, 61]. This stepwise regression procedure can be summarized as follows [47].

1. Enter into the model any parameter(s) that are to be "forced" in.
2. Find the parameter from those not in the model but, available for inclusion, with the largest F_{in} value. If it is at least as great as a prespecified value of F_{in}, then add the parameter to the model. Stop if no parameter can be added.
3. Find the parameter among those in the model, other than those forced in, that has the smallest F_{out} value. If it is less than a prespecified value of F_{out}, then remove the parameter from the model. Repeat this step until no further parameter can be removed. Go to step 2.

It is well known this method is sensitive to the order in which the regressors are introduced [16, 31, 44, 60]; however, previous results for structure computation of polynomial NARMAX models have shown good results for simple models [36].

Bootstrap

Recently, the bootstrap has been shown to be a useful tool for structure detection of nonlinear models [36]. The bootstrap is a numerical method for estimating parameter statistics that requires few assumptions [20]. The conditions needed to apply bootstrap to least-squares estimation are quite mild; namely, that the errors be independent, identically distributed, and have zero mean.

The bootstrap is a technique to randomly reassign observations that enables re-estimates of parameters to be computed. This randomization and computation of parameters is done numerous times and treated as repeated experiments. In essence, the bootstrap simulates a Monte-Carlo analysis. For structure computation, the bootstrap method is used to detect spurious parameters, those parameters whose estimated values cannot be distinguished from zero.

Application of ELS to measured data gives the predicted model response $\hat{\mathbf{Z}}$, prediction errors $\hat{\boldsymbol{\epsilon}}$, and parameter estimate $\hat{\boldsymbol{\theta}}$. A bootstrap version of the residuals, $\hat{\boldsymbol{\epsilon}}^* = [\hat{\epsilon}_1^*, \hat{\epsilon}_2^*, \ldots, \hat{\epsilon}_N^*]$, is formed by random resampling, with replacement, of the estimated residuals $\hat{\boldsymbol{\epsilon}} = [\hat{\epsilon}_1, \hat{\epsilon}_2, \ldots, \hat{\epsilon}_N]$. This resampling is accomplished by randomly selection from $\hat{\boldsymbol{\epsilon}}$ with an equal probability associated with each of the N elements. As an example of the bootstrap resampling process, consider a residual sequence with $N = 4$ data points. A possible bootstrap version of these residuals might be, $\hat{\boldsymbol{\epsilon}}^* = [\hat{\epsilon}_3, \hat{\epsilon}_1, \hat{\epsilon}_3, \hat{\epsilon}_2]$. These resampled errors are added to the model response to generate a bootstrap replication of the original data,

$$\mathbf{Z}^* = \boldsymbol{\Psi}\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\epsilon}}^* \tag{5.14}$$

This bootstrap sample, $\mathbf{Z}^*$, enables a new parameter estimate, $\hat{\boldsymbol{\theta}}^*$, to be computed This procedure is repeated B times to provide a set of parameter estimates from the B bootstrap replications,

$$\hat{\boldsymbol{\Theta}}^* = \left[\hat{\boldsymbol{\theta}}_1^*, \ldots, \hat{\boldsymbol{\theta}}_B^*\right] \tag{5.15}$$

Parameter statistics are then easily computed from $\hat{\boldsymbol{\Theta}}^*$ by forming percentile intervals at a chosen γ level of significance [36].

LASSO

As stated earlier, using least-squares it is difficult to estimate accurate parameter variance when the number of candidate terms is large. A novel procedure that may enable structure selection of highly overparameterized models is now considered.

The least absolute shrinkage and selection operator (LASSO) [64] is a least-squares-like problem with the addition of an ℓ_1 penalty on the parameter vector as

$$\min_{\theta} \frac{1}{2} \left\|(\mathbf{Z} - \boldsymbol{\Psi}\theta)\right\|_2^2 + \lambda \left\|\theta\right\|_1 \tag{5.16}$$

where $\|\cdot\|_2$ denotes the ℓ_2-norm and $\|\cdot\|_1$ denotes the ℓ_1-norm. The regularization parameter $\mathbb{R} \ni \boldsymbol{\lambda} = [\lambda_{min}, \ldots, \lambda_{max}]$ controls the tradeoff between approximation error and sparseness. The LASSO shrinks the least-squares estimator

$$\min_{\theta} \frac{1}{2} \left\|(\mathbf{Z} - \boldsymbol{\Psi}\boldsymbol{\theta})\right\|_2^2 \tag{5.17}$$

towards 0 and potentially sets $\theta_j = 0$ for some j. Consequently, LASSO behaves as a structure-selection instrument.

Solution of LASSO

A solution to LASSO can be constructed in a quadratic programming framework [15]. With the introduction of slack variables the solution to this optimization problem can be written as a simple bound-constrained quadratic program (QP)

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T\mathbf{M}\mathbf{x} + \mathbf{c}^T\mathbf{x} \tag{5.18}$$

such that $x_k \geq 0$ and where

$$\mathbf{M} = \begin{bmatrix} \boldsymbol{\Psi}^T\boldsymbol{\Psi} & -\boldsymbol{\Psi}^T\boldsymbol{\Psi} \\ -\boldsymbol{\Psi}^T\boldsymbol{\Psi} & \boldsymbol{\Psi}^T\boldsymbol{\Psi} \end{bmatrix}, \ \mathbf{c} = \lambda\mathbf{1} - \begin{bmatrix} \boldsymbol{\Psi}^T\mathbf{Z} \\ -\boldsymbol{\Psi}^T\mathbf{Z} \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} \theta^+ \\ \theta^- \end{bmatrix}$$

The model parameters are given by

$$\boldsymbol{\theta} = \boldsymbol{\theta}^+ - \boldsymbol{\theta}^-$$

The quadratic program can be solved readily using standard optimizers [46]. Thus, given a suitable regularization parameter, the general structure computation problem can be solved. A method that determines an appropriate regularization parameter, λ, for the penalty term in (5.16) is the method of crossvalidation [59]. This approach allows the prediction error

$$PE = E\left[\mathbf{Z} - \boldsymbol{\Psi}\theta\right]^2 \tag{5.19}$$

to be estimated. The regularization parameter, λ, is chosen to minimize this estimate.

Unique Optimum and Convergence of LASSO

For identification it is often assumed that the excitation signal is persistently exciting, which implies that $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ is positive-definite. As a result, the first term of Equation (5.16) is a strictly convex function. Since the second term is convex, it follows that the sum is strictly convex and a unique optimizer is guaranteed [10, 50]. Assume the optimal regularization parameter, λ^*, is known. Since Equation (5.16) is strictly a convex optimization problem the solution will converge to a unique global minimum [50]. From parametric optimization theory it is known that $PE(\lambda)$ is not necessarily a convex function (it is a piecewise quadratic function) [28]. Hence, for several choices of λ giving the same PE, different model structures can result. Section 5.3.1 uses a simulated nonlinear model to investigate the validity of LASSO to select the correct model structure.

5.2 Application I

5.2.1 Nonlinear Aeroelastic Model

The aeroelastic equations governing pitch-plunge dynamics can be described as

$$\begin{bmatrix} m & mx_\alpha b \\ mx_\alpha b & I_\alpha \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} c_h & 0 \\ 0 & c_\alpha \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} k_h & 0 \\ 0 & k_\alpha \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} = \begin{bmatrix} -L \\ M \end{bmatrix} \tag{5.20}$$

where h denotes plunge motion, α pitch angle, x_α the nondimensional distance between the elastic axis and the center of mass, m wing mass, I_α mass moment of inertia of the wing about the elastic axis, b the semichord of the wing, $\{c_h,\ c_\alpha\}$ plunge and pitch structural damping coefficients, $\{k_h,\ k_\alpha\}$ plunge and pitch structural spring constants, and $L,\ M$ aerodynamic lift and moment.

Typically, quasisteady aerodynamic forces and moments are assumed that can be modeled as

$$L = \rho U^2 b c_{l_\alpha} \left[\alpha + \frac{\dot{h}}{U} + \left(\frac{1}{2} - a\right) b\frac{\dot{\alpha}}{U}\right] + \rho U^2 b c_{l_\beta}\beta \tag{5.21}$$
$$M = \rho U^2 b^2 c_{m_\alpha} \left[\alpha + \frac{\dot{h}}{U} + \left(\frac{1}{2} - a\right) b\frac{\dot{\alpha}}{U}\right] + \rho U^2 b^2 c_{m_\beta}\beta$$

where ρ denotes density of air, U free-stream velocity, c_{m_α}, c_{l_α} moment and lift coefficients per angle-of-attack, c_{m_β}, c_{l_β} moment and lift coefficient per control surface deflection β, and a the nondimensional distance from midchord to elastic axis. Combining Equations (5.20) and (5.21) gives the equations of motion as

$$\begin{bmatrix} m & mx_\alpha b \\ mx_\alpha b & I_\alpha \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} c_h + \rho U b c_{l_\alpha} & \rho U b^2 c_{l_\alpha}(\frac{1}{2} - a) \\ \rho U b^2 c_{m_\alpha} & c_\alpha - \rho U b^3 c_{m_\alpha}(\frac{1}{2} - a) \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \end{bmatrix} \tag{5.22}$$
$$+ \begin{bmatrix} k_h & \rho U^2 b c_{l_\alpha} \\ 0 & -\rho U^2 b^2 c_{m_\alpha} + k_\alpha \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} = \begin{bmatrix} -\rho b c_{l_\beta} \\ \rho b^2 c_{m_\beta} \end{bmatrix} U^2 \beta$$

This model can be easily transformed to a continuous-time state-space form to simulate a system response [41]. To transform these equations to the state-space form, define the following

$$\mathbf{M} = \begin{bmatrix} m & mx_\alpha b \\ mx_\alpha b & I_\alpha \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} c_h + \rho U b c_{l_\alpha} & \rho U b^2 c_{l_\alpha}(\frac{1}{2} - a) \\ \rho U b^2 c_{m_\alpha} & c_\alpha - \rho U b^3 c_{m_\alpha}(\frac{1}{2} - a) \end{bmatrix} \tag{5.23}$$
$$\mathbf{K} = \begin{bmatrix} k_h & \rho U^2 b c_{l_\alpha} \\ 0 & -\rho U^2 b^2 c_{m_\alpha} + k_\alpha \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -U^2 \rho b c_{l_\beta} \\ U^2 \rho b^2 c_{m_\beta} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} h \\ \alpha \end{bmatrix}, \quad \mathbf{u} = \beta$$

Substituting definitions (5.23) into (5.22) and combining terms gives a state-space formulation as

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^{-1}\mathbf{F} \end{bmatrix} \mathbf{u} \tag{5.24}$$

This system representation provides both pitch and plunge accelerations. Only plunge accelerations are considered here. Given a persistently exciting input, the results for the pitch component will be similar. In the subsequent development, denote plunge acceleration ($\ddot{h}$) as z to conform to the notation used in Equations (5.1)–(5.14).

It has been observed that aeroelastic structures present a well-defined low-frequency elastic-mode composition [25, 62]. Therefore, the linear modal components of this case example are represented using a fourth-order linear time-invariant (LTI) system [2].

Typically, an assumption is made that a structural nonlinearity exists in an aeroelastic system that affects not only flutter speed, but also the characteristics of the motion [18, 38]. An example of an aeroelastic nonlinearity is a preloaded spring tab system. It has been proposed that this type of nonlinearity be modeled as a saturator [38]. This type of nonlinearity is chosen as a hyperbolic tangent, $\tanh(\cdot)$, because a wing section response typically saturates smoothly.

Consider the model class given in Figure 5.1. This blocked structured N-L model (static nonlinearity followed by a causal, linear, time-invariant, dynamic system) is commonly known as a Hammerstein model. Baldelli *et al.* [3] suggest that block-oriented model structures, including Hammerstein models, are capable of representing the nonlinear behavior of most aeroelastic systems [52]. For this reason, the simulated aeroelastic example is a Hammerstein model – a static map represented by a hyperbolic tangent followed by a fourth-order LTI system.

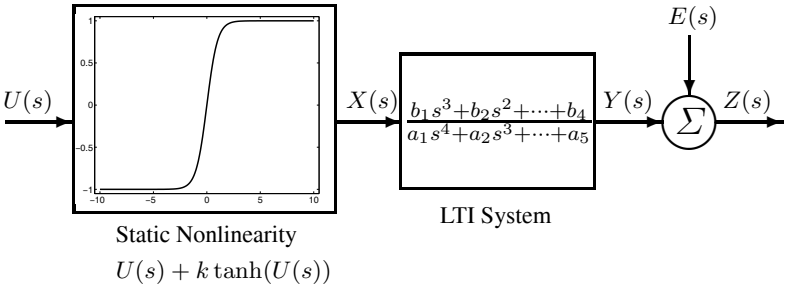


Fig. 5.1. Hammerstein model structure

Continuous to Discrete-domain Approximations

A system's response is sampled via an analog-to-discrete converter [43]. As a result, many system representations are given in discrete time for identification and digital control synthesis.

Several methods exist for discrimination of continuous-time infinite impulse response (IIR) systems. Most commonly used are Tustin's (also known as the bilinear transform) and zero-order hold (ZOH) methods ([1, 23, 43]). Tustin's and ZOH methods map the entire left-half s-plane into the unit circle but do not offer the same model approximation. Tustin's method maps a proper continuous-time transfer function to a proper discrete-time transfer function but maps a strictly proper continuous-time transfer function also to a proper discrete-time transfer function. This often results in a large error in signal emulation due to loss of the inherent time delay (lag). The ZOH explicitly accounts for the delay inherent in many continuous-time systems and provides for better signal emulation. Although Tustin's method can also provide good signal estimation for high sampling rates, it provides a model that is not intuitive. Models based on this approximation will include the current input as one of the model terms, which may not be physically practical. For the aeroelastic system under investigation, the ZOH provides a system description that is both stable and intuitive. For these reasons ZOH is chosen to model the system dynamics.

Discrete-time Representation of Nonlinear Aeroelastic Model

The aeroelastic system in Figure 5.1 is given in continuous time. This section shows how the model can be converted to discrete time and rewritten as a NARMAX model. To do so, note that each block can be analyzed separately.

The first block represents the aeroelastic nonlinearity and as a hyperbolic tangent, is unchanged from continuous time to discrete time since it is a static map [43]. The second block characterizes the linear modal component of an aeroelastic system as a fourth-order LTI system and is converted to discrete time using the ZOH approximation [1]. In addition, assume the model is corrupted by output additive (measurement) noise as

$$z(n) = y(n) + e(n) \tag{5.25}$$

where $z(n)$ is the noise-corrupted output, $y(n)$ the unmeasured noise-free output and $e(n)$ the measurement noise.

After collecting terms and combining, the overall nonlinear model was represented as a nonlinear difference equation with 16 terms as

$$\begin{aligned} z(n) = {} & \theta_1 z(n-1) + \theta_2 z(n-2) + \theta_3 z(n-3) + \theta_4 z(n-4) \\ & + \theta_5 u(n-1) + \theta_6 u(n-2) + \theta_7 u(n-3) + \theta_8 u(n-4) \\ & + \theta_9 \tanh(u(n-1)) + \theta_{10} \tanh(u(n-2)) + \theta_{11} \tanh(u(n-3)) \\ & + \theta_{12} \tanh(u(n-4)) + \theta_{13} e(n-1) + \theta_{14} e(n-2) + \theta_{15} e(n-3) + \theta_{16} e(n-4) \\ & + e(n) \end{aligned}$$

This is a NARMAX model since (i) it includes input–output terms that are combinations of linear and nonlinear terms and (ii) is linear-in-the-parameters.

Structure detection can provide useful insights that can be used in subsequent development or refinement of physical models. Therefore, the sequel investigates (i) the performance of the algorithms outlined in Section 5.1.4 on a simulated NARMAX model of aeroelastic structural stiffness and, (ii) applicability to experimental aircraft data.

5.2.2 Simulation Studies

The efficacy of the bootstrap, stepwise regression and t-test structure-detection algorithms was assessed using Monte-Carlo simulations of a continuous-time model representing aeroelastic structural stiffness dynamics. The inputs were uniformly distributed, white, zero-mean, random sequences with a variances of 8 rad^2. One hundred Monte-Carlo simulations were generated in which each input–output realization was unique, and had a unique Gaussian white, zero-mean, noise sequence added to the output. The output additive noise amplitude was increased in increments of 5 dB, from 20 to 5 dB signal-to-noise ratio (SNR). For identification, the data length was $N_e = [10,000 \;\; 80,000]$ points and increased in increments of $10,000$. For the bootstrap method, an initial estimate of the system parameters was computed and $B = 100$ bootstrap replications were generated to assess the distribution of each parameter. The number $B = 100$ was used because it was previously shown that it is computationally inefficient to select large B, $B \gtrsim 100$ [34, 36]. For all three techniques, each parameter was tested for significance at the 95 % confidence level and a backwards elimination approach was implemented to reduce the fully parameterized model [19, 36]. Specifically, using backwards elimination, parameters of an overparameterized model were estimated and their significance tested to determine if they could be distinguished from zero. If the structure-detection method indicated that the parameter(s) cannot be distinguished from zero, it/they were removed from the model. Next, the reduced model parameters and residuals were updated and tested again until convergence.

The model posed for structure computation was an additive nonlinear model of the form

$$z(n) = \sum_{v=1}^{q} \theta_v \psi(n) + \sum_{w=1}^{r} \theta_w f(\psi(n)) + e(n), \;\; q + r = p \tag{5.26}$$

The regressors posed for this identification problem were of the form given in Equation (5.1), up to lag order 4, and the nonlinear mapping $f(\cdot)$ was chosen to be a hyperbolic tangent

function ($\tanh(\cdot)$), $O = [4\ 4\ 4\ \tanh]$. As previously stated, a $\tanh$ was selected as a basis because a wing-section response is limited due to structural stiffness and appears to saturate smoothly. This gave a full model description with 27 candidate terms.

For each input–output realization, the structure-detection result was classified into one of three categories:

1. Exact model: A model that contains only true system terms,
2. Overmodeled: A model with all its true system terms plus spurious parameters and
3. Undermodeled: A model without all its true system terms. An undermodeled system may contain spurious terms as well.

The continuous-time values used in this study correspond to those found in experiments. The continuous-time parameters used in this study are shown in Table 5.1. Figure 5.2 presents a 50 s slice of a typical input–output data set used for this study. The top panel shows a uniform white input used as control surface deflection and the bottom panel shows the "measured" plunge acceleration response (5 dB SNR) superimposed on top of the noise-free plunge acceleration.

Table 5.1. Continuous-time system coefficients

CT Coefficient	Value
U	18.0 m s^{-1}
a	-0.600 m
b	0.135 m
I_α	0.065 m^2 kg
k_h	2844 N m^{-1}
k_α	2.82 N m rad^{-1}
x_a	0.247 m
c_h	27.4 kg s^{-1}
c_α	0.180 m^2 kg s^{-1}
ρ	1.23 kg m^{-3}
c_{l_α}	3.28
c_{l_β}	3.36
c_{m_α}	–0.628
c_{m_β}	–0.635
k	2

Notice the input flap deflection (Figure 5.2, top) is large (5 rads or $\sim$286 deg), and would not be possible for a real actuator. For this simulated example, an input of this amplitude was selected to provide good coverage for the $\tanh$ nonlinearity and to avoid rescaling. With experimental data a scaled nonlinearity is more appropriate.

Figure 5.3 shows the results of this study. The plots show the per cent rate at which each structure-detection algorithm selected the exact model, overmodeled and undermodeled for increasing data length and increasing SNR (decreasing noise amplitude).

The bootstrap technique had an approximately 0 % rate of exact modeling for low SNR and small data length (5 dB SNR and $N_e = 10,000 - 20,000$), where undermodeling dominated with a rate of 100 %. However, the rate of exact modeling increased linearly from

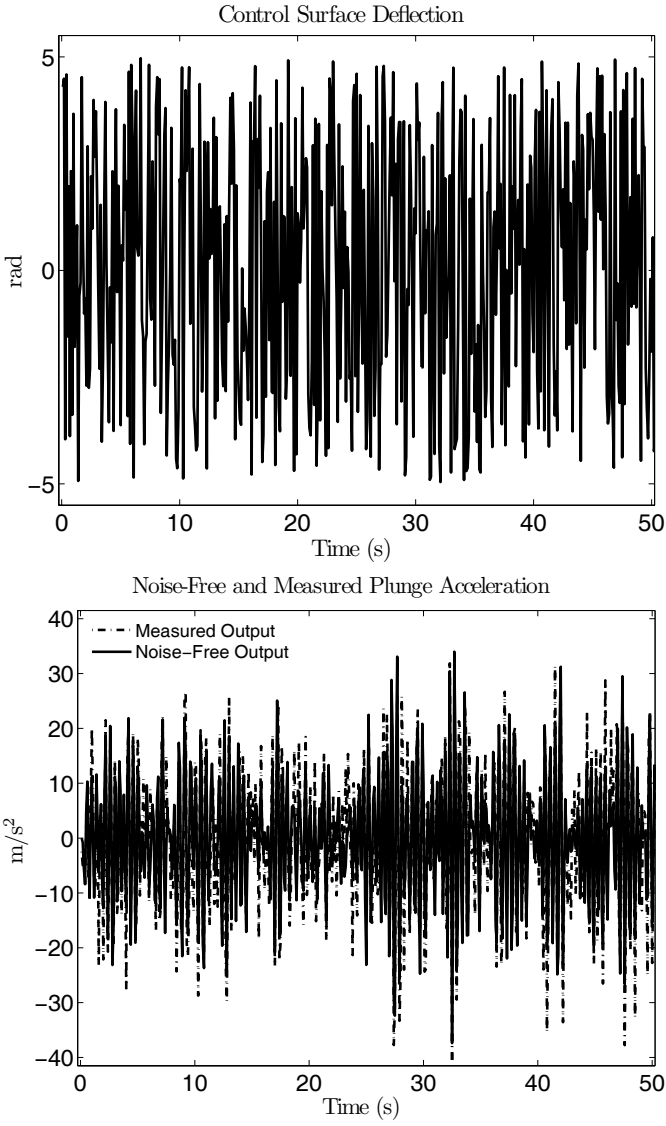


Fig. 5.2. Fifty-second slice of a typical input–output data set used for structure computation. Top: Control-surface deflection. Bottom: Measured (5 dB SNR) plunge-acceleration output superimposed on top of noise-free plunge-acceleration output.

6 % to 95 % for $N_e = 10,000 - 80,000$ and 10–20 dB SNR. The rate of overmodeling was 0 % for $N_e = 10,000 - 20,000$ at 5 dB SNR. For higher SNR (10–20 dB) and $N_e = 20,000 - 80,000$ the overmodeling rate decreased linearly from 42 % to 5 %. Al-

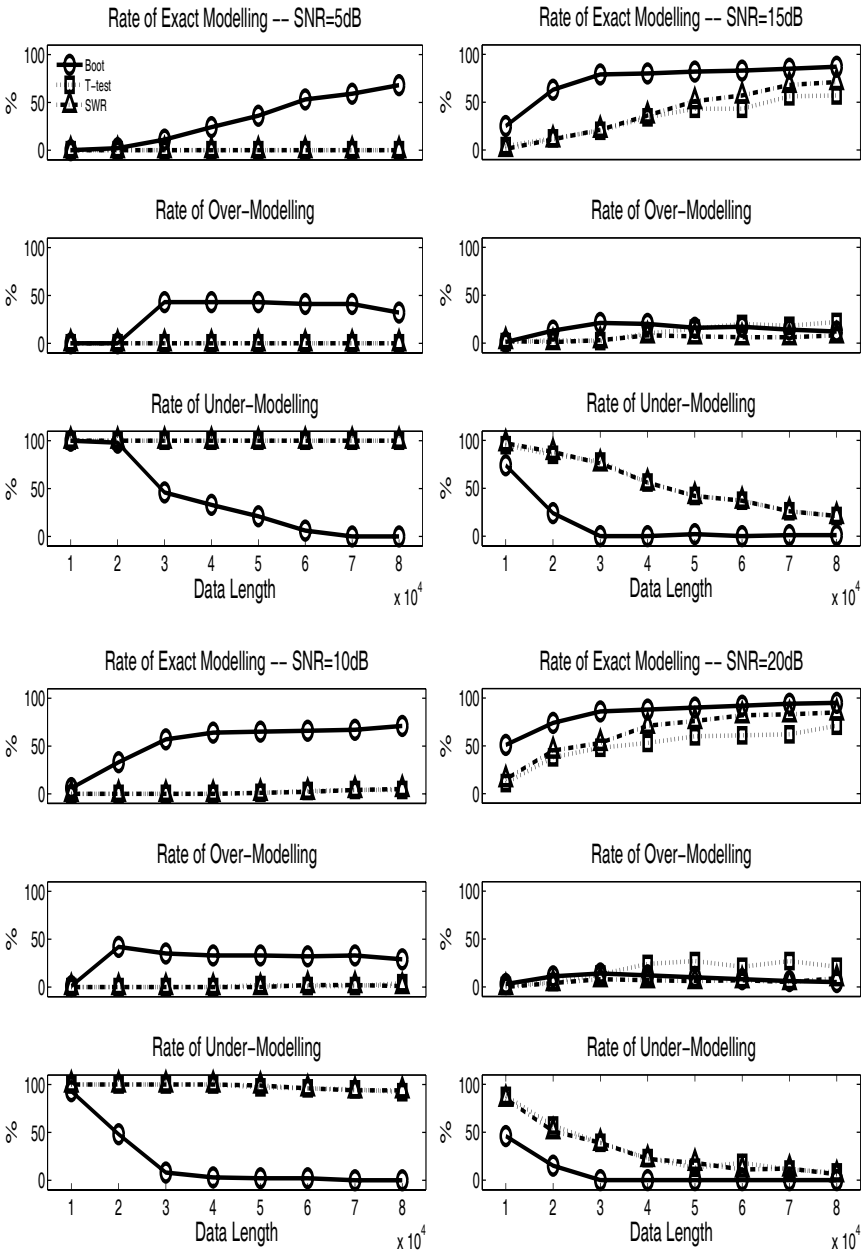


Fig. 5.3. Exact, over- and undermodeling rate for a Hammerstein model describing aeroelastic dynamics. Top panels: rate of exact modeling; Center panels: rate of overmodeling; Bottom panel: rate of undermodeling. Ordinate: Per cent selection, Abscissa: Data length.

though undermodeling dominated for low SNR and short data length, it decreased linearly from 48 % to 0 % for $N_e = 20,000 - 80,000$ and 10–20 dB SNR.

Stepwise regression undermodeling dominated the structure computation effort for all data lengths for 5–10 dB SNR, with a rate of 100 %. However, for high SNR levels, 15–20 dB, and $N_e = 10,000 - 80,000$ the undermodeling rate decreased from 97 % to 6 %. The rate of overmodeling was low for all SNRs and data lengths, ranging from a minimum of 0 % to a maximum of 9 %. The rate of selecting the exact model was constant at ~ 0 % for all data lengths and 5–10 dB SNR but increased from 1 % to 85 % for high SNR, 15–20 dB, and $N_e = 10,000 - 80,000$.

The t-test approach undermodeling also dominated the structure-computation effort with a rate of 100 % for all data lengths at 5–10 dB SNR. For higher SNR levels (15–20 dB) and all data lengths, the undermodeling rate decreased linearly from 95 % to 8 %. The rate of overmodeling was ~ 0 % for all data lengths and 5–10 dB SNR. For higher SNRs of 15–20 dB, the rate of over-modeling remained approximately constant for all data lengths with a rate of ~ 20 %. The rate of selecting the exact model was constant at ~ 0 % for all data lengths and 5–10 dB SNR but increased from 4 % to 70 % for all data lengths and high SNR, 15–20 dB.

For this overparameterized model describing aeroelastic dynamics, the bootstrap method clearly outperformed the t-test and stepwise regression. As a result, the subsequent section implements only the bootstrap structure method for the analysis of flight-test data.

5.2.3 Structure Detection of Experimental AAW Data

The applicability of using structure detection via a bootstrap approach for blackbox modeling was assessed on experimental flight-test data from the F/A-18 active aeroelastic wing (AAW) project at NASA Dryden Flight Research Center. The data analyzed for this study used collective aileron-command input and structural accelerometer-response output.

Procedures

Flight data was gathered during subsonic flutter clearance of the AAW. At each flight condition, the aircraft was subjected to multisine inputs corresponding to collective and differential aileron, collective and differential outboard leading edge flap, rudder, and collective stabilator excitations in the range of 3–35 Hz for 26 s. Accelerometer data measured during the collective aileron sweeps at Mach 0.85, 4572 m (15,000 ft) was considered. The input collective aileron command was the exact signal sent by the onboard computer. The output was taken as the response of an accelerometer mounted near the left wing, outerfold aft. Data was sampled at 400 Hz. For analysis, the recorded flight-test data were decimated by a factor of 2, resulting in a final sampling rate of 200 Hz.

The system was identified by applying the bootstrap approach, as outlined in Section 5.2.2, except here scaled hyperbolic tangent functions were used because the input amplitude is less than ± 1 (Figure 5.4). The scale factors used for the input, output and error signals were in the range of $\nu = [0.1\ 1.0]$ and increased in increments of 0.1. A scaled hyperbolic tangent is denoted as $\tanh(\cdot, \nu)$. Models with every possible combination of scale factors were considered (*i.e.* structure computation was performed on 1000 models). The model that yielded the highest crossvalidation per cent fit was deemed the best-fit model.

For computation of a model structure, $N_e = 5200$ points were used from accelerometer-response measurements on the left wing. To validate the computed model structure a cross-validation data set containing $N_v = 5200$ points was used from data collected at a similar location on the right wing.

Results

The results of identifying the AAW data are presented. Figure 5.4 shows the input–output trial used for this analysis. The data represents collective aileron-command sequence and structural accelerometer response (left wing, outer wing-fold aft) used to compute the system structure.

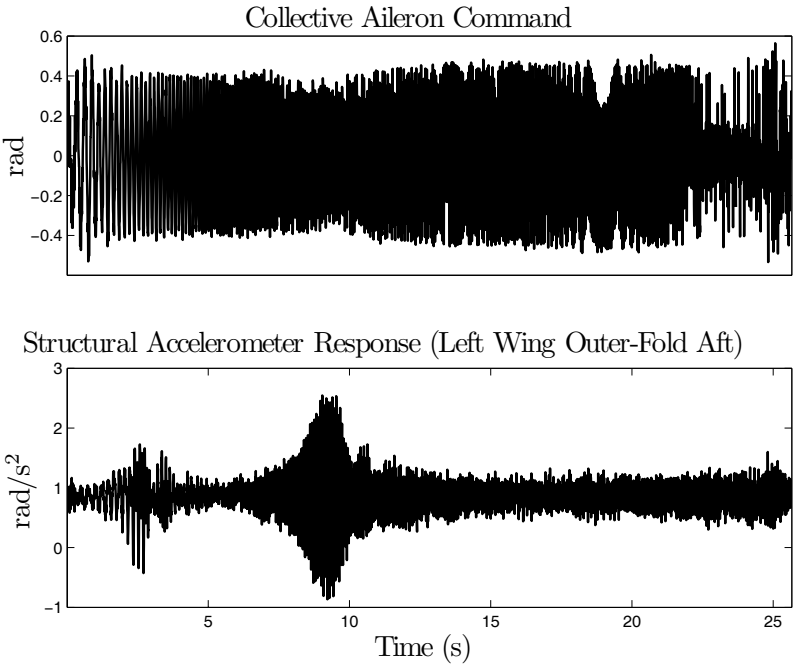


Fig. 5.4. Experimental F/A-18 AAW data. Top: Recorded collective aileron position. Bottom: Recorded structural accelerometer response (left wing outer wing-fold aft).

Equation (5.27) depicts the model structure computed by the bootstrap method.

$$\begin{aligned} z(n) = {} & \hat{\theta}_0 + \theta_1 z(n-1) + \hat{\theta}_2 z(n-3) + \hat{\theta}_3 z(n-4) + \hat{\theta}_4 u(n) + \hat{\theta}_5 u(n-2) \\ & + \hat{\theta}_6 u(n-4) + \hat{\theta}_7 \tanh(u(n), 0.3) + \hat{\theta}_8 \tanh(u(n-1), 0.3) \\ & + \hat{\theta}_9 \tanh(u(n-3), 0.3) + \hat{\theta}_{10} \tanh(u(n-4), 0.3) + \hat{\theta}_{11} e(n-1) \\ & + \hat{\theta}_{12} e(n-3) + \hat{\theta}_{13} e(n-4) + e(n) \end{aligned} \tag{5.27}$$

The computed model structure is represented as a Hammerstein model with direct feed-through terms and contains a total of 14 terms.

Figure 5.5 shows the predicted output for a crossvalidation data set for the identified structure (Equation (5.27)). The upper panel displays the full 26 s time history of the accelerometer

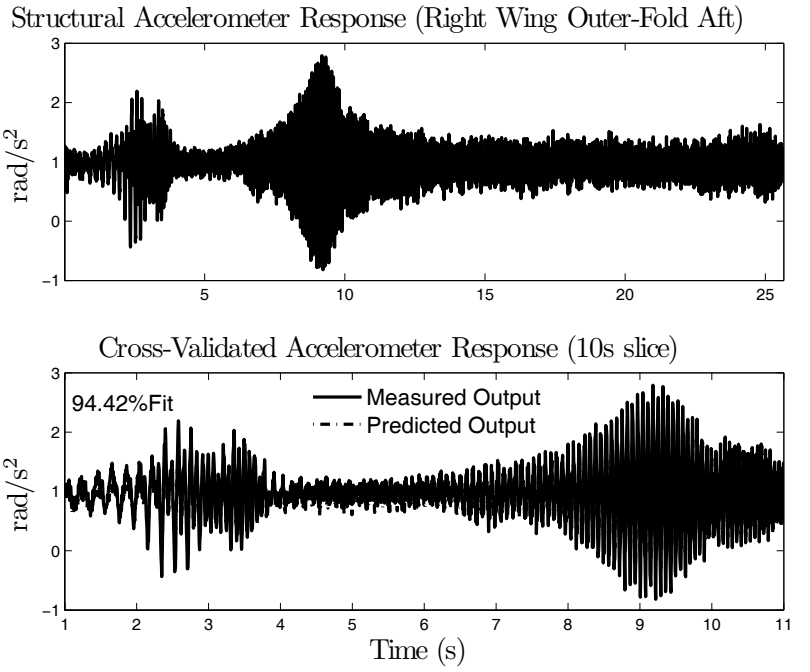


Fig. 5.5. Top: Recorded structural accelerometer response (right wing, outerfold aft). Bottom: Predicted accelerometer response of right wing outerfold aft superimposed on top of measured velocity output for identified NARMAX AAW model from flight-test data ($N_v = 5200$).

response recorded on the right wing outer wing-fold aft. The lower panel displays a 10 s slice of the predicted output superimposed on top of the measured output that accounts for over 94 % of the measured output variance. The result demonstrates that the computed model structure is capable of reproducing the measured output with high accuracy.

5.2.4 Theory *versus* Simulation/Experimental Results

Simulations

For the simulated model of aeroelastic structural stiffness dynamics, results demonstrated that for all SNR levels and sufficient data length ($N_e = 60,000 - 80,000$), the bootstrap method had a low rate of selecting an undermodeled model (2–0 %) and a high rate of selecting the exact model (60–95 %). The t-test and stepwise regression both had difficulty computing the

correct structure in this case study with a selection range of 3–70 % and 2–85 % respectively, for equivalent data lengths and SNR levels. Moreover, both had a significantly higher rate of undermodeling, 100–6 % and 100–8 %, respectively.

Both the t-test and stepwise regression had difficulty, but for different reasons. For the t-test, extraneous parameters model the noise. This results in a biased estimate of the variance, which may give models with incorrect structure. Stepwise regression also fails since it uses a F-ratio test to determine significance. This is sensitive to the order in which the terms are entered into the model and how much each term contributes to the output [36].

Experimental Aircraft Data

Experimental results demonstrate that structure computation as a tool for blackbox modeling may be a useful tool for the analysis of dynamic aircraft data. The bootstrap successfully reduced the number of regressors posed to aircraft aeroelastic data yielding a parsimonious model structure. Additionally, this parsimonious structure was capable of predicting a large portion of the crossvalidation data, collected on the adjacent wing and with a different sensor. This suggests that the identified structure and parameters explain the data well. Using per cent fit alone as an indicator of model goodness could lead to incorrect interpretations of model validity. However, for nonlinear models often this may be the only indicator that is readily available.

Hyperbolic tangent with fourth-order input, output and error lag was used as a basis function to explain the nonlinear behavior of the F/A-18 AAW data. Different basis functions and a higher dynamic order (lag-order) could be investigated to determine if another basis can produce accurate model predictions with reduced complexity. Moreover, further studies could be used to evaluate whether the model structure is invariant under different operating conditions, such as Mach number and altitude, and model parameterizations.

Significance

This study illustrates the usefulness of structure detection as an approach to compute a parsimonious model of a highly complex nonlinear process, as demonstrated with experimental data of aircraft aeroelastic dynamics. Moreover, analysis of input–output data can provide useful process insights that can be used in subsequent development or refinement of physical models. In particular, morphological models are based on assumptions (these effects are important and those are negligible), which may be incorrect [55, 57]. A structure computation approach to model identification may help uncover such surprises.

Summary

Simulation results demonstrate that a bootstrap approach for structure computation of aircraft structural stiffness provided a high rate of true model selection. The t-test and stepwise regression methods had difficulty providing accurate results for the system studied yet have previously been shown to provide reasonable results for polynomial class models. Using bootstrap, it is possible to compute good estimates of parameter statistics resulting in accurate estimates of model structure.

Moreover, this study contributes to the understanding of the use of structure detection for modeling and identification of aerospace systems. These results may have practical significance in the analysis of aircraft dynamics during envelope expansion and could lead to

more efficient control strategies. In addition, this technique could allow greater insight into the functionality of various systems dynamics by providing a quantitative model that is easily interpretable.

5.3 LASSO Applications

5.3.1 Simulation Example

The efficacy of LASSO as a tool for structure detection was assessed using Monte–Carlo simulations of a polynomial nonlinear system. In these simulations, a white input with uniform distribution was used. One thousand Monte–Carlo simulations were generated in which each input-output realization was unique and had a unique Gaussian distributed, white, zero-mean, noise sequence added to the output. The output additive noise amplitude was increased in increments of 5 dB, from 20 to 0 dB signal-to-noise ratio (SNR). Each input–output set consisted of 1000 data points.

The regularization parameter, λ, was determined by numerically minimizing the cross-validation error across a discrete set of 1000 logarithmically spaced λ values ($10^{\lambda_{min}} \leq \lambda \leq 10^{\lambda_{max}}$). The min-max regularization parameter levels were set to $\lambda_{min} = -10$ and $\lambda_{max} = 1.5$. For crossvalidation, the last 1/3 of each data set was used; 667 points for estimation and 333 for validation. Initially, with a single realization at each noise level, the min-max regularization parameter levels were set to $\lambda_{min} = -10$ and $\lambda_{max} = 1.5$ using a coarse discrimination of 100 logarithmically spaced λs. Then, the min-max levels were reset to be in the neighborhood of the λ that best minimized the PE in this coarse grid and the discrimination increased to 1000 logarithmically spaced λs before starting the Monte-Carlo analysis.

For each input–output realization, the structure detection result was again classified into one of three categories:

1. Exact model: A model that contains only true system terms,
2. Overmodeled: A model with all its true system terms plus spurious parameters and
3. Undermodeled: A model without all its true system terms. An undermodeled model may contain spurious terms as well.

Model Order Known

Consider the nonlinear polynomial system,

$$\begin{aligned} z(n) = {} & 0.4[u(n-1) + u(n-1)^2 + u(n-1)^3] \\ & + 0.8z(n-1) - 0.8e(n-1) \end{aligned}$$

It was assumed that the system order is fully known, $O = [1, 1, 1, 3]$, to assess the accuracy of LASSO to compute the correct structure when the model is mildly overparameterized. This system is described by 3 lagged inputs, 1 lagged output, 1 lagged error, and third-order nonlinearity. A model of this order has 35 candidate terms, but the true system has only 5 true terms. With 30 spurious parameters this NARMAX model is highly overparameterized.

The left panel in Figure 5.6 shows the LASSO method had a 0 % rate of undermodeling for 20–10 dB SNR, which then increased from 12.3–30.3 % for 5–0 dB SNR. The rate of over-modeling increased for 20–5 dB SNR, from 48.6–76.8 % then deceased at 0 dB SNR to 62.6 % where undermodeling started to increase rapidly. The rate of selecting the exact model

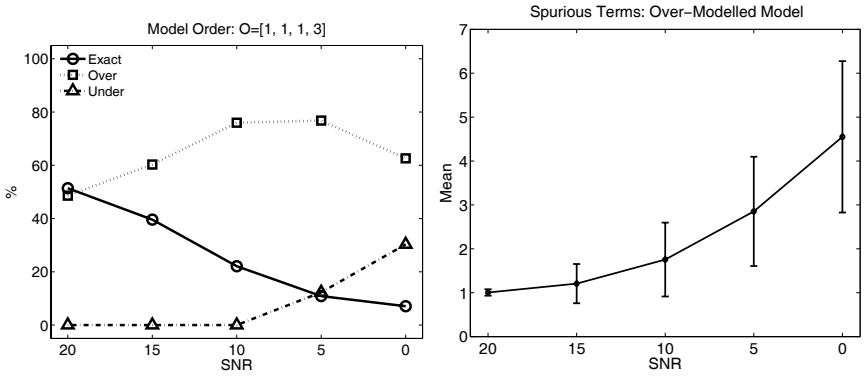


Fig. 5.6. Left: Selection rate of exact, over- and undermodeling. Right: Mean and STD of spurious term selection rate for overmodeling (Note that the abscissa is shown in decreasing SNR, which corresponds to increasing noise intensity).

decreased across all SNR levels with a maximum of 51.4 % at 20 dB and a minimum of 7.1 % at 0 dB.

The right panel illustrates the rate of selecting spurious terms when overmodeling occurred. This rate was low for all SNR levels with a minimum of 1.01 ± 0.0733 and a maximum of 4.55 ± 1.72. For this third-order model with known model order, LASSO performed well for high SNRs since it did not drop any true terms. The performance of LASSO deteriorated when the SNR decreased. When LASSO selection was overmodeled the rate of spurious term selection was low for all SNR levels.

Model Order Unknown but Bounded

Next is examined the performance of LASSO for the following system,

$$z(n) = 0.4[u(n-1) + u(n-1)^2] + 0.8z(n-1) - 0.8e(n-1) + e(n)$$

Here, it was assumed the system order was unknown but could be bounded as $O_{max} = [3, 3, 3, 3]$. A model of this order has 286 candidate terms. However, only 4 true parameters describe this system: two lagged inputs, one lagged output and one lagged error term. With 282 spurious parameters this NARMAX model is highly overparameterized.

The left panel in Figure 5.7 shows the LASSO method had a 0 % rate of undermodeling for all SNR levels. The rate of overmodeling increased with decreasing SNR, from 63.09–100 %. The rate of selecting the exact model decreased as the SNR decreased from 36.91–0 %. The right panel illustrates the rate of selecting spurious terms when overmodeling occurred. This rate was low for most SNR levels from 20 to 5 dB, with an average overmodeling rate of 1.87 ± 0.754 at SNR=20 dB to 10.97 ± 2.87 at SNR=5 dB. However, at SNR=0 dB the average overmodeling rate increased significantly to 53.53 ± 10.2 For this second-order model with unknown but bounded model order, LASSO had a lower rate of selecting the exact model for all noise levels. It was deemed that the performance of this approach is acceptable

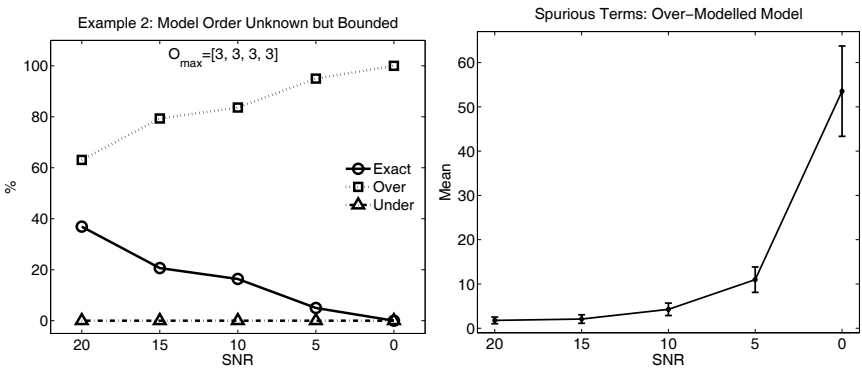


Fig. 5.7. Example 2 selection rate: unknown but bounded model order. Left: Rate of exact, over- and undermodeling. Right: Mean and STD of spurious term selection rate for overmodeling (note that the abscissa is shown in decreasing SNR that corresponds to increasing noise intensity).

because, although the overparameterization was excessive, no true system terms were dropped and the rate of spurious term selection was reasonable for all noise levels.

5.3.2 Experimental Aircraft Data

Finally, the LASSO technique was assessed on experimental flight-test data from the F/A-18 AAW. The data analyzed for this study used collective aileron-position input and structural accelerometer-response output.

Procedures

Flight data was gathered during subsonic flutter clearance of the AAW. At each flight condition, the aircraft was subjected to multisine inputs corresponding to collective and differential aileron, collective and differential outboard leading edge flap, rudder, and collective stabilator excitations in the range of 3–35 Hz for 26 s. This study considers accelerometer data measured during the collective aileron sweeps at Mach 0.85 at 4572 m (15,000 ft). The input collective aileron position was obtained as the average of four position transducer measurements from the right and left ailerons. The output was taken as the response of an accelerometer mounted near the wing leading edge just inside the wing fold. Data was sampled at 400 Hz. For analysis, the recorded flight test data was decimated by a factor of 2, resulting in a final sampling rate of 200 Hz.

For identification a model with fourth-order input–output and error dynamics and third-order nonlinearity, $O = [3, 4, 4, 4]$, was used. A model with fourth-order dynamics was selected because it has been observed that aeroelastic structures are well defined by a fourth-order linear time-invariant (LTI) system [63]. The nonlinearity order was chosen as third order because models of higher nonlinear order can often be decomposed to second or third order. This gave a full model description with 560 candidate terms.

The system was identified by applying the LASSO, as outlined in Section 5.3.1. For estimation, $N_e = 5200$ points were used from accelerometer-response measurements on the right wing. For crossvalidation, $N_v = 5200$ points were used from data collected at a similar location on the left wing. In both the estimation and crossvalidation sets the input was the collective aileron position. The min-max regularization parameter levels were set to $\lambda_{min} = -10$ and $\lambda_{max} = 1.0$ ($10^{\lambda_{min}} \leq \lambda \leq 10^{\lambda_{max}}$) with a discrimination grid of 1000 logarithmically spaced λs. Each model class was identified separately and the one yielding the best output prediction was deemed the best model.

Results

The results of identifying the AAW data are presented. Figure (5.8) shows the input-output trial used for this analysis. The data represents collective aileron position and structural accelerometer response (right wing) used to compute the system structure.

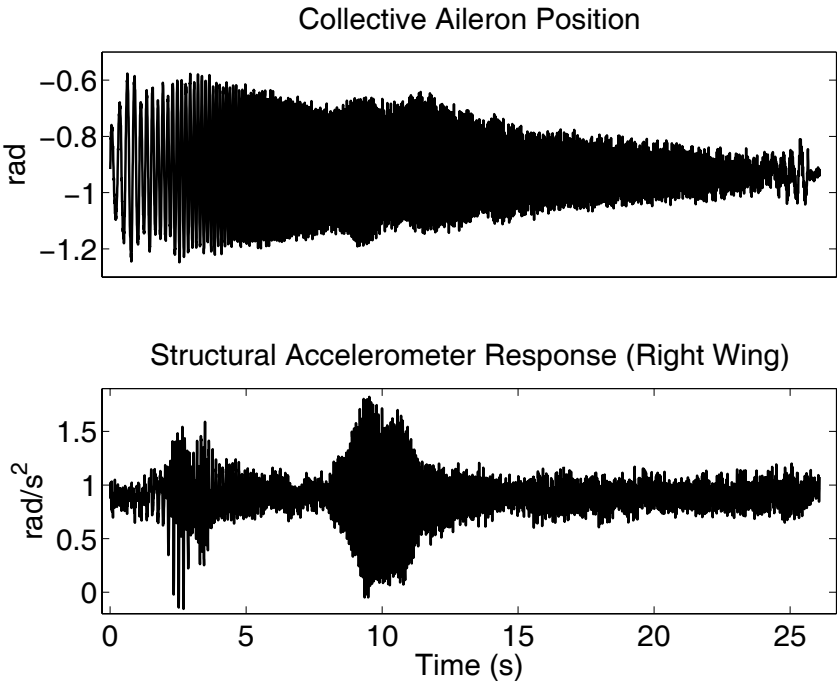


Fig. 5.8. Top: Collective aileron position. Bottom: Structural accelerometer response.

Equation (5.28) depicts the model structure computed by the LASSO method

$$\begin{aligned} z(n) &= \hat{\theta}_0 + \hat{\theta}_1 u(n-1) + \hat{\theta}_2 u(n-2) + \hat{\theta}_3 u(n-4) \\ &+ \hat{\theta}_4 u^2(n-1) + \hat{\theta}_5 u^2(n-2) + \hat{\theta}_6 u^2(n-4) \\ &+ \hat{\theta}_7 z(n-1) + \hat{\theta}_8 z(n-4) + \hat{\theta}_9 u^2(n-1) z(n-4) \\ &+ \hat{\theta}_{10} u^2(n-2) z(n-1) + \hat{\theta}_{11} u^2(n-4) z(n-4) \\ &+ \hat{\theta}_{12} z^3(n-1) + \hat{\theta}_{13} z^3(n-4) + \hat{\theta}_{14} \hat{\epsilon}(n-1) \\ &+ \hat{\theta}_{15} \hat{\epsilon}(n-4) + \hat{\theta}_{16} u^2(n-1) \hat{\epsilon}(n-4) \\ &+ \hat{\theta}_{17} u^2(n-2) \hat{\epsilon}(n-1) + \hat{\theta}_{18} u^2(n-4) \hat{\epsilon}(n-4) \\ &+ \hat{\theta}_{19} z^2(n-1) \hat{\epsilon}(n-1) + \hat{\theta}_{20} z(n-1) \hat{\epsilon}^2(n-1) \\ &+ \hat{\theta}_{21} \hat{\epsilon}^3(n-1) + \hat{\theta}_{22} z^2(n-4) \hat{\epsilon}(n-4) \\ &+ \hat{\theta}_{23} z(n-4) \hat{\epsilon}^2(n-4) + \hat{\theta}_{24} \hat{\epsilon}^3(n-4) \end{aligned} \tag{5.28}$$

The computed model structure is represented by a combination of linear and nonlinear, lagged input-output terms and contains 25 terms. Hence, the LASSO technique successfully produced a parsimonious model description from the full set of 560 candidate terms.

5.3.3 Theory *versus* Simulation/Experimental Results

Simulations

The LASSO approach to structure detection yields good results for the nonlinear polynomial model considered. It had a high rate of selecting the exact structure for high SNR levels. For lower SNRs, overmodeling dominated the structure computation procedure except at 0 dB at which the rate of undermodeling started to increase quickly. However, the rate of selecting spurious terms was low, giving a model that is easily interpretable for all SNR levels when the computed model was overmodeled.

For the highly overparameterized model, with unknown but bounded model order, overmodeling dominated for all SNR levels. but gave an acceptable rate of selecting the correct model structure for low noise amplitude (20 dB). Moreover, the rate of overparameterization was low for all noise levels except 0 dB where it dominated completely. The results for this case study were obtained using only 667 data points for estimation and 333 for validation. Often there is more data available in many system-identification applications and, therefore, the exact selection rate should improve given more data for identification.

Although the utility of LASSO to underdetermined identification cases was not demonstrated, it has been shown in some applications that it provides an accurate and unique solution [15]. This ill-posed problem is solved by the introduction of a regularizer or penalty function to the standard least-squares metric. Moreover, the results demonstrate that LASSO, at worst, provides a model that is slightly overparameterized. Consider an overparameterized model better than an undermodeled model since it is not possible (with our backward selection approach) to re-enter a parameter into the model once it has been removed. An overparameterized model that still contains its true parameters is clearly more useful than one that has dropped a true parameter.

LASSO tends to provide a model that is slightly overparameterized. This feature may be very useful in many applications such as those in aerospace or biomedical research where there is little or no *a priori* information regarding the true underlying system structure. In these situations, LASSO may be useful to reduce a large number of possible regressors to a manageable level as demonstrated with the highly overparameterized model, with unknown

but bounded model order. Once the number of possible terms has been reduced, these remaining terms can be further assessed for significance using a variety of tools such as covariance matrix, ANOVA, state-dependent parameter estimation or bootstrap approaches [36, 51].

Experimental Aircraft Data

Experimental results demonstrate that LASSO may be a useful tool for structure computation of dynamic aircraft data. LASSO successfully reduced the large number of regressors posed to aircraft aeroelastic data yielding a parsimonious model structure. Additionally, this parsimonious structure was capable of predicting a large portion of the observed crossvalidation data, collected on an adjacent wing and with a different sensor. This suggests that the identified structure and parameters explain the data well. Using per cent fit alone as an indicator of model goodness may lead to incorrect interpretations of model validity. However, in many cases, for nonlinear models this may be the only indicator that is readily available.

For this study, only a polynomial map was used as a basis function to explain the nonlinear behavior of the F/A-18 AAW data. Clearly, different basis functions should be investigated to determine if another basis can produce accurate model predictions with reduced or comparable complexity. Moreover, further studies are necessary to evaluate whether the model structure is invariant under different operating conditions, such as Mach number and altitude, and for model parameterizations.

There may be more efficient ways to address the solution of Equation (5.16) for different λ. The LASSO optimization problem can be viewed as a special case of a parameterized QP. It is known that the solution $\theta(\lambda)$ is a piecewise function [28] and there are reasonably efficient ways to construct this function for low-order dimension on the parametric variable λ [37]. Since λ is scalar in our case, a parametric approach might be tractable. If the piecewise function $\theta(\lambda)$ has a compact description, it could be more computationally efficient than a brute-force gridding.

It should be acknowledged that the overall problem with the two stages, (i) computing a finite set of optimal θ for varying λ and, (ii) optimizing λ in a second phase to minimize the crossvalidation criteria, can be interpreted as an *ad hoc* approach to solve a bilevel optimization problem. These problems are notoriously difficult to solve exactly. Improved computation of suboptimal solutions may be possible by exploring more advanced approaches to address the bilevel optimization problem

The optimization criteria in the LASSO setup is motivated by the well-known fact that a $\|\cdot\|_1$ penalty appended to a quadratic objective tends to yield a sparse solution. However, it is only a heuristic for addressing the underlying problem: achieving few nonzero parameters. An alternative way to address this, in an optimization framework, is to use combinatorial optimization. Solving the regression problem (5.17) with a bound on the number of nonzero parameters may be achieved in a straightforward manner using mixed binary quadratic programming. Instead of performing the crossvalidation optimization problem over a bank of solutions, computed using different λ, one could compute solutions for a different number of nonzero parameters and use these solutions in the crossvalidation phase.

Summary

Nonlinear models have the advantage that they cover a wider range of system dynamics than linear models that could allow for faster envelope expansion. Using nonlinear models to characterize aeroelastic phenomena can provide significant time and cost savings for test and development of aerospace vehicles.

A model-validation technique for nonlinear systems, using higher-order correlations was developed by Billings and Voon [7, 9]. Korenberg and Hunter [32] showed that this model-validation technique fails for simple cases. Therefore, this approach was not implemented and was used as an indicator of model goodness.

LASSO enhances the capability for structure detection of an overparameterized polynomial NARMAX model in the presence of additive output noise. Application of structure computation to aeroelastic modeling is again presented using flight-test data and shown to yield a parsimonious model structure while maintaining a high per cent fit to crossvalidation data.

5.4 Concluding Remarks

NARMAX structure-detection techniques show a high degree of accuracy for selection of the exact model structure with simulated data, but the applied structure-detection technique is a key element to achieving satisfactory accuracy and may not apply in all cases. Analysis of experimental flight-test data generally provides a parsimonious system description with a high per cent fit for crossvalidated data. These analyses contribute to the understanding of the use of parametric identification techniques and NARMAX structure detection for blackbox modeling of nonlinear aerospace systems. Algorithms need to be developed for distributed processing and automated for fast and efficient algorithms over a sensor array.

LASSO is a novel approach for detecting the structure of overparameterized nonlinear models in situations where other methods may be inadequate. The main point is that the LASSO technique is clearly amenable to the study of a wide range of nonlinear systems. Application of structure computation to aeroelastic modeling is again presented using flight-test data and shown to yield a parsimonious model structure while maintaining a high per cent fit to crossvalidation data. These results may have practical significance in the analysis of aircraft dynamics during envelope expansion and could lead to more efficient control strategies. This technique could also allow greater insight into the functionality of various systems dynamics by providing a model that is easily interpretable.

References

[1] Åström, K. and Wittenmark, B. (1996). *Computer-Controlled Systems: Theory and Design*. Prentice Hall, Englewood Cliffs NJ, 3rd edition.

[2] Baldelli, D., Chen, P. C., Liu, D. D., Lind, R., and Brenner, M. (2004). Nonlinear aeroelastic modeling by block-oriented identification. *45th AIAA/ASME/ASCE/AHS/ Structures, Structural Dynamics & Materials Conference*, pp. AIAA–2004–1938. Palm Springs, CA.

[3] Baldelli, D., Mazzaro, M., and Sánchez Peña, R. (2001). Robust identification of lightly damped flexible structures by means of orthonormal bases. *IEEE Transactions on Control Systems Technology* **9**(5), 696–707.

[4] Beck, J. and Arnold, K. (1977). *Parameter Estimation in Engineering and Science*. Wiley series in probability and mathematical statistics. John Wiley & Sons, New York.

[5] Billings, S. and Chen, S. (1989). Extended model set, global data and threshold model identification of severely non-linear systems. *International Journal of Control* **50**(5), 1897–1923.

[6] Billings, S. and Jones, G. (1992). Orthogonal least-squares parameter estimation algorithms for non-linear stochastic systems. *International Journal of Systems Science* **23**(7), 1019–1032.

[7] Billings, S. and Voon, W. (1983). Structure detection and model validation tests in the identification of nonlinear systems. *IEE Proceedings, Part D.* **130**, 193–199.

[8] Billings, S. and Voon, W. (1984). Least squares parameter estimation algorithms for non-linear systems. *International Journal of Systems Science* **15**(6), 601–615.

[9] Billings, S. and Voon, W. (1986). Correlation based model validity tests for non-linear models. *International Journal of Control* **44**(1), 235–244.

[10] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge, U.K., 1st edition.

[11] Bunton, R. and Denegri, C. (2000). Limit cycle oscillation characteristics of fighter aircraft. *AIAA Journal of Aircraft* **37**(5), 916–918.

[12] Chen, P., Sarhaddi, D., and Liu, D. (1998). Limit cycle oscillation characteristics of fighter aircraft with external stores. *AIAA-98*, p. 1727.

[13] Chen, S. and Billings, S. (1989). Representations of non-linear systems: the NARMAX model. *International Journal of Control* **49**(3), 1013–1032.

[14] Chen, S., Billings, S., Cowan, C., and Grant, P. (1990). Practical identification of NARMAX models using radial basis functions. *International Journal of Control* **52**, 1327–1350.

[15] Chen, S., Donoho, D., and Saunders, M. (2001). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Compututing* **43**(1), 129–159.

[16] Copas, J. (1983). Regression, prediction and shrinkage. *Journal of the Royal Statistical Society B* **45**(2), 311–354.

[17] Denegri, C. (2000). Limit cycle oscillation flight test results of a fighter with external stores. *AIAA Journal of Aircraft* **37**(5), 761–769.

[18] Dowell, E., Edwards, J., and Strganac, T. (2003). Nonlinear aeroelasticity. *AIAA Journal of Aircraft* **40**(1), 857–874.

[19] Draper, N. and Smith, H. (1981). *Applied Regression Analysis*. John Wiley & Sons, New York, 2nd edition.

[20] Efron, B. (1979). Computer and the theory of statistics: Thinking the unthinkable. *SIAM Review* **21**(4), 460–480.

[21] Efroymson, M. (1960). *Multiple Regression and Correlation*. in Mathematical Methods for Digital Computers. John Wiley & Sons, New York, 1st edition.

[22] Eykhoff, P. (1974). *System Identification*. John Wiley & Sons, New York, 1st edition.

[23] Franklin, G., Powell, J., and Emami-Naeini, A. (2002). *Feedback Control of Dynamic Systems*. Addison-Wesley, New York, 4th edition.

[24] Freund, J. (1962). *Mathematical Statistics*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1st edition.

[25] Gawronski, W. (1996). *Balanced Control of Flexible Structures*. Springer-Verlag, London, 1st edition.

[26] Goodwin, G. and Payne, R. (1977). *Dynamic System Identification: Experiment Design and Data Analysis*, volume 136 of *Mathematics in Science and Engineering*. Academic Press, New York.

[27] Greblicki, W. and Pawlak, M. (1991). Nonparametric identification of a cascade nonlinear time series system. *Signal Processing* **22**, 61–75.

[28] Grigoriadis, M. and Ritter, K. (1969). A parametric method for semidefinite quadratic programs. *SIAM J. Control* **7**(4), 559–577.

[29] Haber, R. and Unbehauen, H. (1990). Structure identification of nonlinear dynamic systems-a survey on input/output approaches. *Automatica* **26**, 651–677.

[30] Harris, C. and Billings, S. (1985). *Self Tuning and Adaptive Control: Theory and Applications*. Peter Peregrinus, London, 2nd edition.

[31] Hurvich, C. and Tsai, C. L. (1990). The impact of model selection on inference in linear regression. *The American Statistician* **44**, 214–217.

[32] Korenberg, M. and Hunter, I. (1990). The identification of nonlinear biological systems: Wiener kernel approaches. *Annals of Biomedical Engineering* **18**, 629–654.

[33] Kosut, R., Lau, M., and Boyd, S. (1992). Set-membership identification of systems with parameteric and nonparametric uncertainty. *IEEE Transactions on Automatic Control* **37**(7), 929–942.

[34] Kukreja, S. (2003). A suboptimal bootstrap method for structure detection of nonlinear output-error models. *Proc. 13th IFAC Symp. System Identification*, volume 13, pp. 1566–1571. Rotterdam, The Netherlands.

[35] Kukreja, S., Galiana, H., and Kearney, R. (2003). NARMAX representation and identification of ankle dynamics. *IEEE-TBME* **50**(1), 70–81.

[36] Kukreja, S., Galiana, H., and Kearney, R. (2004). A bootstrap method for structure detection of NARMAX models. *International Journal of Control* **77**(2), 132–143.

[37] Kvasnica, M., Grieder, P., Baotić, M., and Morari, M. (2004). Multi Parametric Toolbox (MPT). *Hybrid Systems: Computation and Control*, volume 2993 of *Lecture Notes in Computer Science*, pp. 448–462. Springer Verlag, Philadelphia.

[38] Lee, B., Pricei, S., and Wong, Y. (1999). Nonlinear aeroelastic analysis of airfoils:bifurcation and chaos. *Progress in Aerospace Sciences* **35**, 205–334.

[39] Leontaritis, I. and Billings, S. (1985). Input-output parametric models for non-linear systems part I: deterministic non-linear systems. *International Journal of Control* **41**(2), 303–328.

[40] Leontaritis, I. and Billings, S. (1985). Input-output parametric models for non-linear systems part II: stochastic non-linear systems. *International Journal of Control* **41**(2), 329–344.

[41] Lind, R. and Brenner, M. (1999). *Robust Aeroservoelastic Stability Analysis: Flight Test Applications*. Springer-Verlag, New York, 1st edition.

[42] Lind, R., Prazenica, R., and Brenner, M. (2003). Estimating nonlinearity using volterra kernals in feedback with linear models. *AIAA-03*, volume 44, pp. 1406–1416. Norfolk, Virginia.

[43] Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 2nd edition.

[44] Mantel, M. (1970). Why stepdown procedures in variable selection. *Technometrics* **12**, 621–625.

[45] Masri, S. and Caughey, T. (1979). A nonparametric identification technique for nonlinear dynamic problems. *Journal of Applied Mechanics* **46**(2), 433–447.

[46] Mészáros, C. (1998). *The BPMPD interior point solver for convex quadratic problems*. Technical Report 98–8, Hungarian Academy of Science, Budapest.

[47] Miller, A. (1996). The convergence of Efroymson's stepwise regression algorithm. *The American Statistician* **50**(2), 180–181.

[48] Mukhopadhyay, V. (1995). Flutter suppression control low design and testing for the acting flexible wing. *AIAA Journal of Aircraft (Special Adaptive Flexible Wing Issue)* **32**(1), 45–51.

[49] Mukhopadhyay, V. (2003). Historical perspective on analysis and control of aeroelastic responses. *AIAA Journal of Guidance, Control, and Dynamics* **26**(5), 673–684.

[50] Osborne, M., Presnell, B., and Turlach, B. (2000). On the LASSO and its dual. *Journal of Computational and Graphical Statistics* **9**(2), 319–337.

[51] Ostrom, C. (1990). *Time Series Analysis: Regression Techniques*. Sage Publications, Newberry Park, Calif., 2nd edition.

[52] Palanthandalam-Madapusi, H., Hoagg, J., and Berstein, D. (2004). Basis-function optimization for subspace-based nonlinear identification of systems with measured-input nonlinearities. *Proc. IEEE American Control Conference*, volume 5, pp. 4788–4793. Boston, Massachusetts.

[53] Panuska, V. (1968). A stochastic approximation method for identification of linear systems using adaptive filtering. *Proceedings 9th Joint Automatic Control Conference*, pp. 1014–1021. IEEE, Ann Arbor, MI, USA.

[54] Panuska, V. (1969). An adaptive recursive least squares identification algorithm. *Proceedings 8th IEEE Symp. on Adaptive Processes*, p. paper 6e. University Park, PA, USA.

[55] Pearson, R. (1995). Nonlinear input/output modelling. *Journal of Process Control* **5**(4), 197–211.

[56] Pendleton, E., Bessette, D., Field, P., Miller, G., and Griffin, K. (2000). Active aeroelastic wing flight research program: Technical program and model analytical development. *AIAA J. Aircraft* **37**(4), 554–561.

[57] Pottmann, M. and Pearson, R. (1998). Block-oriented NARMAX models with output multiplicities. *AIChE Journal* **44**(1), 131–140.

[58] Seber, G. (1977). *Linear Regression Analysis*. John Wiley & Sons, New York, 1st edition.

[59] Shao, J. (1993). Linear model selection by cross-validation. *Journal of the American Statistical Association* **88**, 486–494.

[60] Sjöberg, J., *et al.* (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica* **31**(12), 1691–1724.

[61] Smillie, K. (1966). *An Introduction to Regression and Correlation*. Academic Press Inc., London, 1st edition.

[62] Smith, R. (1995). Eigenvalue perturbation models for robust control. *IEEE Automatic Control* **40**(6), 10631–1066.

[63] Smith, R. S. (1995). Model validation for robust control: An experimental process control application. *Automatica* **31**(11), 1637–1647.

[64] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B* **58**(1), 267–288.

[65] Unbehauen, H. (1996). Some new trends in identification and modeling of nonlinear dynamical systems. *Applied Mathematics and Computation* **78**, 279–297.

[66] Young, P. (1968). The use of linear regression and relaxed procedures for the identification of dynamic processes. *Proceedings 7th IEEE Symp. on Adaptive Processes*, pp. 501–505. Los Angeles, CA, USA.

6

Modeling and Control of Flexible Structures in Frequency Domain

Luis Alvergue[1], Jie Chen[2], and Guoxiang Gu[1]

[1] Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803-5901 USA {ggu,lalver1}@lsu.edu
[2] Department of Electrical Engineering, University of California, Riverside, CA 92521 USA jchen@ee.ucr.edu

6.1 Background

The problem of robust stability and performance for feedback control systems has been studied for many years. A number of design procedures are proposed to tackle the robustness issue for feedback control systems. These include design techniques in both the time and frequency domains. While time-domain methods are popular in academic research, frequency-domain techniques dominate the engineering practice. The primary reason is the existence of the model uncertainty in both magnitude and phase responses of the plant model that is difficult to tackle in the time domain that gives rise to $\mathcal{H}_\infty$ control. In retrospect, $\mathcal{H}_\infty$ control revived classical frequency design techniques, and renewed traditional interests in analyzing feedback control systems in the frequency domain. It provides the worst-case guarantee on robust stability and performance provided that the model uncertainty is quantified in $\mathcal{H}_\infty$ norm. In particular, $\mathcal{H}_\infty$ loopshaping as investigated in [6] extends the lead/lag compensator design to multivariable systems, and generalizes the synthesis method based on Bode plots to frequency shaping in $\mathcal{H}_\infty$. The rapid development in $\mathcal{H}_\infty$ control also motivated modeling of uncertain systems in the frequency domain. Identification in $\mathcal{H}_\infty$ proposed in [5] attracted significant interest in the control community that aims at modeling and quantification of $\mathcal{H}_\infty$ norm-bounded uncertainties compatible with $\mathcal{H}_\infty$ control. The monograph [2] provides a glimpse in this research area.

In spite of the aforementioned progress, modeling and control of uncertain systems are studied separately, though not independently in light of [2, 5]. A more challenging issue is the integration of modeling and control under the same framework of $\mathcal{H}_\infty$ that will be investigated in this chapter with JPL flexible structure [1] as an application platform. It is assumed that the design specifications for feedback control of the lightly damped system are given in the frequency domain, specified by the frequency shape of the open-loop system. Based on the samples of the experimental frequency response measurements, normalized coprime factors of the shaped plant model are identified with quantification of the modeling error in $\mathcal{H}_\infty$ norm. Several existing algorithms are employed to identify the plant model, and the modeling errors in terms of the normalized coprime factor uncertainty are compared in $\mathcal{H}_\infty$ norm. In addition feedback controllers are synthesized to stabilize the true but unknown plant based on the $\mathcal{H}_\infty$ loopshaping method. A complete design procedure will be presented that integrates modeling and control in the frequency domain. The performance of the feedback control system with the

true plant is analyzed based on the modeling error bound quantified in the identification stage. For consistency, the proposed modeling and control technique will be developed in discrete time. The notations are standard and will be made clear as we proceed.

6.2 Robust Stabilization

Robust stabilization refers to stabilization of a set of systems by a single feedback controller. In [6] stabilization of uncertain systems described by normalized coprime factors is investigated. It assumes that the nominal plant model is given by a transfer matrix of size $p \times m$ represented by

$$P_0(z) = \tilde{M}_0^{-1}(z)\tilde{N}_0(z) = N_0(z)M_0^{-1}(z) \tag{6.1}$$

where $\tilde{M}_0(z), \tilde{N}_0(z), M_0(z), N_0(z)$ are normalized coprime factors or stable and proper transfer matrices that satisfy[3]

$$\begin{array}{l} \tilde{M}_0(z)\tilde{M}_0(z)^{\sim} + \tilde{N}_0(z)\tilde{N}_0(z)^{\sim} = I \\ M_0(z)^{\sim}M_0(z) + N_0(z)^{\sim}N_0(z) = I \end{array} \tag{6.2}$$

If $P_0(z)$ has finitely many poles, then computation of its normalized coprime factors is straightforward. Specifically in this case $P_0(z)$ admits a state-space realization with realization matrices $\{A_0, B_0, C_0, D_0\}$. By convention

$$P_0(z) = \left[\begin{array}{c|c} A_0 & B_0 \\ \hline C_0 & D_0 \end{array}\right] := D_0 + C_0(zI - A_0)^{-1}B_0 \tag{6.3}$$

by an abuse of notation. Denote $R_0 = I + D_0'D_0$ and $\tilde{R}_0 = I + D_0D_0'$, and $X \geq 0$ and $Y \geq 0$ as the stabilizing solutions to the following algebraic Riccati equations (AREs):

$$X = A_0'XA_0 + C_0'C_0 - F_0'(R_0 + B_0'XB_0)F_0 \tag{6.4}$$

$$F_0 = -(R_0 + B_0'XB_0)^{-1}(B_0'XA_0 + D_0'C_0)$$

$$Y = A_0YA_0' + B_0B_0' - L_0(\tilde{R}_0 + C_0'YC_0)L_0' \tag{6.5}$$

$$L_0 = -(A_0YC_0' + B_0D_0')(\tilde{R}_0 + C_0YC_0')^{-1}$$

Then the normalized coprime factors of $P_0(z)$ are given by [9]

$$G_0(z) = \left[\begin{array}{c} M_0(z) \\ N_0(z) \end{array}\right] = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} A_0 + B_0F_0 & B_0\Omega_0^{-1} \\ \hline F_0 & \Omega_0^{-1} \\ C_0 + D_0F_0 & D_0\Omega_0^{-1} \end{array}\right] \tag{6.6}$$

$$\tilde{G}_0(z) = \left[\begin{array}{cc} \tilde{M}_0(z) & \tilde{N}_0(z) \end{array}\right] = \left[\begin{array}{c|cc} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array}\right] = \left[\begin{array}{c|cc} A_0 + L_0C_0 & L_0 & B_0 + L_0D_0 \\ \hline \tilde{\Omega}_0^{-1}C_0 & \tilde{\Omega}_0^{-1} & \tilde{\Omega}_0^{-1}D_0 \end{array}\right] \tag{6.7}$$

that are stable and satisfy $G_0(z)^{\sim}G_0(z) = I$ and $\tilde{G}_0(z)\tilde{G}_0(z)^{\sim} = I$ where

$$\Omega_0 = (R_0 + B_0'XB_0)^{1/2}, \qquad \tilde{\Omega}_0 = (\tilde{R}_0 + C_0YC_0')^{1/2} \tag{6.8}$$

Denote the set of all stable and proper transfer matrices by $\mathcal{H}_\infty$. For any $T(z) \in \mathcal{H}_\infty$, its $\mathcal{H}_\infty$ norm is given by

[3] Notation $T(z)^{\sim} = T'(z^{-1})$ is used.

$$\|T\|_\infty := \sup_{|z|>1} \sigma_{\max}[T(z)] \tag{6.9}$$

with $\sigma_{\max}(\cdot)$ the maximum singular value. The true system is in the form of

$$\begin{aligned} P(z) &= [\tilde{M}_0(z) + \Delta_{\tilde{M}}(z)]^{-1}[\tilde{N}_0(z) + \Delta_{\tilde{N}}(z)] \\ &= [N_0(z) + \Delta_N(z)]\,[M_0(z) + \Delta_M(z)]^{-1} \end{aligned} \tag{6.10}$$

It assumes that $\Delta_{\tilde{M}}(z), \Delta_{\tilde{N}}(z), \Delta_M(z), \Delta_N(z) \in \mathcal{H}_\infty$ are unknown but a norm bound $\|\Delta\|_\infty \le \delta$ or $\|\tilde{\Delta}\|_\infty \le \tilde{\delta}$ is available where

$$\Delta(z) = \begin{bmatrix} \Delta_M(z) \\ \Delta_N(z) \end{bmatrix}, \qquad \tilde{\Delta}(z) = \begin{bmatrix} \Delta_{\tilde{M}}(z) & \Delta_{\tilde{N}}(z) \end{bmatrix} \tag{6.11}$$

Hence, the true system $P(z)$ lies in a set of uncertain systems with size δ or $\tilde{\delta}$ and described by the normalized coprime factors of its approximant or $P_0(z)$. Given the true plant $P(z)$ and nominal model $P_0(z)$, the norm bounds on the modeling error can be computed according to [3]

$$\delta = \inf_{Q\in\mathcal{H}_\infty} \|G_0 - GQ\|_\infty, \qquad \tilde{\delta} = \inf_{\tilde{Q}\in\mathcal{H}_\infty} \|\tilde{G}_0 - \tilde{G}\tilde{Q}\|_\infty \tag{6.12}$$

in which $G(z)$ and $\tilde{G}(z)$ are defined in the same way as $G_0(z)$ and $\tilde{G}_0(z)$, respectively with $P_0(z)$ replaced by $P(z)$. Under some mild condition the set of uncertain systems described by normalized coprime factors is equivalent to the set of gap metric uncertain systems [3] that in turn implies that

$$\Delta(z) = G(z)Q(z) - G_0(z), \qquad \tilde{\Delta}(z) = \tilde{Q}(z)\tilde{G}(z) - \tilde{G}_0(z) \tag{6.13}$$

for some $Q(z), \tilde{Q}(z) \in \mathcal{H}_\infty$. It follows that $P(z)$ in Equation (6.10) is characterized by $P_0(z)$ and the norm bound δ or $\tilde{\delta}$. Robust stabilization requires design of a single feedback controller that stabilizes the set of plants in Equation (6.10) satisfying the given norm bound. The solution to such a robust stabilization problem is provided in [6] for continuous-time systems and will be provided next for discrete-time systems.

Denote the feedback controller as $K_0(z)$ that is synthesized based on $P_0(z)$. Let $K_0(z) = V_0^{-1}(z)U_0(z)$ be its left coprime factorization. Stabilization of $P_0(z)$ is equivalent to design of $\{V_0(z), U_0(z)\}$ such that

$$I = V_0(z)M_0(z) + U_0(z)N_0(z) = G_0^+(z)G_0(z) \tag{6.14}$$

where $G_0^+(z) \in \mathcal{H}_\infty$ is a left inverse of $G_0(z)$. Because

$$V_0(z)[M_0(z) + \Delta_M(z)] + U_0(z)[N_0(z) + \Delta_N(z)] = I + G_0^+(z)\Delta(z) \tag{6.15}$$

robust stabilization amounts to design of $G_0^+(z) \in \mathcal{H}_\infty$ such that $\delta\|G_0^+\|_\infty < 1$ which is the so-called corona problem [3]. We skip the dual case for uncertain systems described by left normalized coprime factors.

Remark 1. The equality (6.14) and $\delta\|G_0^+\|_\infty < 1$ imply

$$\gamma_0 = \|G_0^+\|_\infty = \left\| \begin{bmatrix} V_0 & U_0 \end{bmatrix} \right\|_\infty = \left\| \begin{bmatrix} P_0 \\ I \end{bmatrix} (I - K_0P_0)^{-1} \begin{bmatrix} K_0 & I \end{bmatrix} \right\|_\infty < \delta^{-1} \tag{6.16}$$

It turns out that the set of uncertain systems stabilized by the above $K_0(z)$ is larger than that described earlier. Indeed, denote $\mathcal{L}_\infty$ as the set of all proper transfer matrices with bounded

$\|\cdot\|_\infty$. Then the set of uncertain systems with $\Delta(z) \in \mathcal{L}_\infty$ and $\|\Delta\|_\infty \leq \delta$ is also stabilized [8] provided that

$$\text{wno}\left[\det(I + P^\sim P_0)\right] + \eta(P_0) - \eta(P) - \eta_0(P) = 0 \tag{6.17}$$

holds, where $\eta(\cdot)$ and $\eta_0(\cdot)$ denote the number of poles outside and on the unit circle, respectively, and wno$(\cdot)$ is the winding number of the Nyquist plot about the origin. This observation is important because modeling of the normalized coprime factors does not always guarantee stability of the uncertainty $\Delta(z)$ or $\tilde{\Delta}(z)$ but the condition (6.17) can be rather easily enforced. In addition, given $G_0(z)$ and $G(z)$, consisting of the normalized coprime factors of $P_0(z)$ and $P(z)$, the uncertainty bound can be computed according to [8]

$$\delta = \inf_{Q \in \mathcal{L}_\infty, \text{wno}(Q)=0} \|G_0 - GQ\|_\infty \tag{6.18}$$

Recall that (A, B, C, D) is a realization of $G_0(z)$ with $(p + m) \times m$ the size of D, assuming $n \times n$ is the size of A_0 or A. Then D has full column rank and $G_0(z)$ is strictly minimum phase implying the existence of proper and stable left inverses $G_0^+(z)$. Indeed we have

$$G_0(z) = \left[I + C(zI - A)^{-1}B_L\right] D, \qquad B_L = BD^+ - LD_\perp^+$$

for any L of size $n \times p$ where D^+ is a left inverse of D and $D_\perp^+$ of size $(p+m) \times m$ satisfying

$$\begin{bmatrix} D^+ \\ D_\perp^+ \end{bmatrix} \begin{bmatrix} D & D_\perp \end{bmatrix} = \begin{bmatrix} I_m & 0 \\ 0 & I_p \end{bmatrix} \tag{6.19}$$

where I_k denotes the identity matrix of dimension $k \times k$. Moreover, the following holds

$$\begin{aligned} \text{rank}\left\{\begin{bmatrix} A - zI & B \\ C & D \end{bmatrix}\right\} &= \text{rank}\left\{\begin{bmatrix} I & -BD^+ \\ 0 & D_\perp^+ \\ 0 & D^+ \end{bmatrix} \begin{bmatrix} A - zI & B \\ C & D \end{bmatrix}\right\} \\ &= m + \text{rank}\left\{\begin{bmatrix} zI - A + BD^+C \\ D_\perp^+ C \end{bmatrix}\right\} \end{aligned}$$

Thus, $G_0(z)$ being strictly minimum phase is equivalent to the detectability of $(D_\perp^+ C, A - BD^+C)$. It follows that there exists an L such that $A_L = A - BD^+C + LD_\perp^+ C$ is a stability matrix and

$$G_0^+(z) = D^+\left[I + C(zI - A)^{-1}B_L\right]^{-1} = D^+\left[I - C(zI - A_L)^{-1}B_L\right] \tag{6.20}$$

is stable and proper. Define an augmented square transfer matrix as

$$G_a(z) = \left[G_0(z)\ G_{0\perp}(z)\right] := \left[I + C(zI - A)^{-1}B_L\right]\left[D\ D_\perp\right] \tag{6.21}$$

Then $G_a(z)$ is strictly minimum phase provided that $G_0(z)$ is, and L is stabilizing. Hence

$$\begin{aligned} G_a^{-1}(z) &= \left[D\ D_\perp\right]^{-1}\left[I + C(zI - A)^{-1}B_L\right]^{-1} \\ &= \begin{bmatrix} D^+ \\ D_\perp^+ \end{bmatrix}\left[I - C(zI - A_L)^{-1}B_L\right] =: \begin{bmatrix} G_0^+(z) \\ G_{0\perp}^+(z) \end{bmatrix} \end{aligned} \tag{6.22}$$

is stable as well. The following characterizes all the stable and proper left inverses.

Lemma 1. *Suppose that $G_0(z) = D + C(zI - A)^{-1}B$ is strictly minimum phase and D has full column rank. Then there exists a stabilizing L such that the left inverse as in Equation* (6.20) *is stable and proper. Moreover, all stable and proper left inverses of $G_0(z)$ are parameterized by*

$$G_0^{\text{inv}}(z) = G_0^+(z) + J(z)G_{0\perp}^+(z) \tag{6.23}$$

where $J(z)$ is an arbitrary stable and proper transfer matrix.

Proof. Clearly for any stable and proper $J(z)$, $G_0^{\text{inv}}(z)$ is stable and proper as well. Moreover $G_0^{\text{inv}}(z)G_0(z) = I$ holds by the relation in Equation (6.21). Conversely if $G_0^{\text{inv}}(z)$ is a stable and proper left inverse of $G_0(z)$. Then

$$G_0^{\text{inv}}(z)G_a(z) = G_0^{\text{inv}}(z)\left[G_0(z)\ G_{0\perp}(z)\right] = \left[I\ J(z)\right] \tag{6.24}$$

by taking $J(z) = G_0^{\text{inv}}(z)G_{0\perp}(z)$, which is proper and stable. It follows that

$$G_0^{\text{inv}}(z) = \left[I\ J(z)\right]G_a^{-1}(z) = G_0^+(z) + J(z)G_{0\perp}^+(z) \tag{6.25}$$

that is in the same form as in Equation (6.23) and depends only on L and $J(z)$.

In the case of normalized coprime factors, the following holds [6]

$$\inf_{J(z)\in\mathcal{H}_\infty} \|G_0^+ + JG_{0\perp}^+\|_\infty = \frac{1}{\sqrt{1 - \lambda_{\max}[XY(I+XY)^{-1}]}} =: \gamma_{\text{opt}} \tag{6.26}$$

In fact $J(z)$ can be replaced by static gain J in computing a stable and proper left inverse $G_0^{\text{inv}}(z)$ such that $\|G_0^{\text{inv}}\|_\infty < \gamma$ for any $\gamma > \gamma_{\text{opt}}$.

Theorem 1. *Consider the stable and proper left inverse $G_0^{\text{inv}}(z)$ as in Equation* (6.25). *There exist stabilizing gains L and J such that $\|G_0^{\text{inv}}\|_\infty < \gamma$, if and only if the following* ARE

$$\Sigma = (A - B\Pi D'C)\Sigma[I + C'(I - D\Pi D')C\Sigma]^{-1}(A - B\Pi D'C)' + B\Pi B' \tag{6.27}$$

with $\Pi = \gamma^2(\gamma^2 D'D - I)^{-1}$ has a stabilizing solution $\Sigma \geq 0$ such that

$$\gamma^2(D'D)^2 - \left[D'D + D'C\Sigma(I + C'D_\perp D_\perp'C\Sigma)^{-1}C'D\right] > 0 \tag{6.28}$$

In this case, a left inverse satisfying $\|G_0^{\text{inv}}\|_{[\infty,T]} < \gamma$ is specified by

$$\begin{bmatrix} L \\ J \end{bmatrix} = -\begin{bmatrix} A - BD^+C \\ D^+C \end{bmatrix} \Sigma C'D_\perp(I + D_\perp'C\Sigma C'D_\perp)^{-1}$$

That is, $J(\cdot)$ can be chosen as a constant gain.

Proof. By the expression of $G_0^{\text{inv}}(z)$ in Equation (6.25), the required left inverse corresponds to the $\mathcal{H}_\infty$ output estimation problem described by

$$\begin{aligned} x(t+1) &= (A - BD^+C)x(t) - BD^+\eta(t) \\ \zeta(t) &= -D^+Cx(t) + D^+\eta(t) \\ y(t) &= D_\perp'Cx(t) + D_\perp'\eta(t) \end{aligned} \tag{6.29}$$

where $x(t)$ is the state vector of $G_0(z)$, $\zeta(t)$ is the output signal to be estimated, $y(t)$ is the output observation, and $\eta(t)$ is an arbitrary energy bounded noise sequence. Specifically, the output estimator has the structure

$$\begin{aligned}\hat{x}(t+1) &= (A - BD^+C)x(t) - L[y(t) - D'_\perp C\hat{x}(t)] \\ \hat{\zeta}(t) &= D^+Cx(t) - J[y(t) - D'_\perp C\hat{x}(t)]\end{aligned} \tag{6.30}$$

Denote the estimator errors by $x_e(t) = x(t) - \hat{x}(t)$ and $\zeta_e(t) = \zeta(t) - \hat{\zeta}(t)$. Then, with the expressions in Equations (6.29) and (6.30), we arrive at

$$\begin{bmatrix} x_e(t+1) \\ \zeta_e(t) \end{bmatrix} = \begin{bmatrix} A - BD^+C + LD_\perp C & -BD^+ + LD'_\perp \\ D^+C + JD'_\perp C & D^+ + JD'_\perp \end{bmatrix} \begin{bmatrix} x_e(t) \\ \eta(t) \end{bmatrix} \tag{6.31}$$

that is identical to that in Equation (6.25). It is now clear that the existence of L and $J(z)$ such that $G_0^{\text{inv}}(z)$ is stable and satisfies $\|G_0^{\text{inv}}\|_\infty < \gamma$ for some $\gamma > 0$ is an $\mathcal{H}_\infty$ output estimation problem and the formulas in [4] can be applied directly. In fact, the formulas in [4] can be further simplified by noting the relation in Equation (6.19). Hence, we conclude that there exists an output estimator such that $G_0^{\text{inv}}(z)$ is stable and $\|G_0^{\text{inv}}\|_\infty < \gamma$, if and only if the ARE (6.27) admits a stabilizing solution $\Sigma \geq 0$ such that the inequality (6.28) holds. The detailed calculations are skipped.

Note that the ARE (6.27) can be rewritten in a more familiar form

$$\Sigma = A_\Pi \Sigma A'_\Pi - A_\Pi \Sigma C'(R + C\Sigma C')^{-1} C\Sigma A'_\Pi + B\Pi B' \tag{6.32}$$

with $A_\Pi = A - B\Pi D'C$ although $R = I - \gamma^2 DD'$ is sign-indefinite.

Remark 2. Theorem 1 provides a procedure for computing the robust controller. Let $G_0^{\text{inv}}(z) = \begin{bmatrix} V_0(z) & U_0(z) \end{bmatrix}$ be the stable and proper left inverse of $G_0(z)$ whose $\mathcal{H}_\infty$-norm is bounded by $\gamma > 0$. Then $K_0(z) = V_0^{-1}(z)U_0(z)$ is the feedback controller that stabilizes the uncertain system described by normalized coprime factors [3, 6] as long as the $\mathcal{H}_\infty$-norm of the uncertainty is no more than $\delta < \gamma^{-1}$. In fact, a two degree-of-freedom controller can be employed as in Figure 6.1 to achieve not only robust stability but also good tracking performance [8].

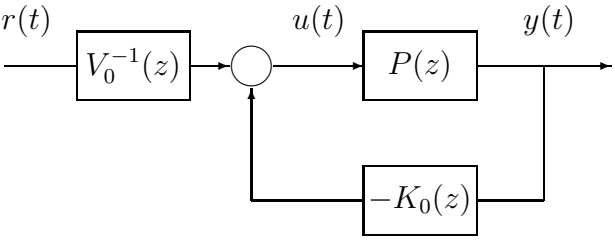


Fig. 6.1. Two degree-of-freedom control system where $K_0(z) = V_0^{-1}(z)U_0(z)$

An application of the robust stabilization result is the $\mathcal{H}_\infty$ loopshaping design procedure developed in [6] in which it is observed that robust stabilization and performance objective share the same cost function measured by $\mathcal{H}_\infty$ norm. To be specific, assume that $P_0(z)$ has the desired frequency shape dictated by the desired loop-gain performance. Then we have the following result [6].

Theorem 2. *Suppose that $P_0(z)$ admits normalized coprime factorizations as in Equation* (6.1) *and $G_0(z)$ is defined as in Equation* (6.6). *Let $G_0^{\text{inv}}(z)$ be as in Theorem 1 satisfying the condition $\gamma_0\delta < 1$ with $\gamma_0 = \|G_0^{\text{inv}}(z)\|_\infty < \gamma$. Then the following holds*

(i) *If* $\sigma_{\min}(P_0(e^{j\omega})) > \sqrt{\gamma^2-1}$, *then*

$$\sigma_{\min}(K_0(e^{j\omega})) \geq \frac{\sigma_{\min}(P_0(e^{j\omega})) - \sqrt{\gamma^2-1}}{1+\sigma_{\min}(P_0(e^{j\omega}))\sqrt{\gamma^2-1}}$$

(ii) *If* $\sigma_{\max}(P_0(e^{j\omega})) < 1/\sqrt{\gamma^2-1}$, *then*

$$\sigma_{\max}(K_0(e^{j\omega})) \leq \frac{\sqrt{\gamma^2-1} + \sigma_{\max}(P_0(e^{j\omega}))}{1-\sigma_{\max}(P_0(e^{j\omega}))\sqrt{\gamma^2-1}}$$

(iii) *For all* $\Delta(z) \in \mathcal{H}_\infty$ *such that* $\delta = \|\Delta\|_\infty < 1/\gamma$,

$$\left\| \begin{bmatrix} P \\ I \end{bmatrix} (I - K_0 P)^{-1} \begin{bmatrix} K_0 & I \end{bmatrix} \right\|_\infty < \frac{\gamma}{1-\delta\gamma}$$

where $P(z)$ *is in the form of Equation* (6.10).

The facts (i) and (ii) show that the loopshape of the feedback system consisting of the nominal model is similar to that of the nominal model that can be suitably shaped by some frequency weighting functions. This is the main reason why the robust stabilizing controller can achieve the $\mathcal{H}_\infty$ loopshaping performance. In light of Remark 1, fact (iii) implies that the true loopshape of the feedback system in Figure 6.1 has a similar performance to that of the nominal one provided that $\gamma_0\delta << 1$. That is, the robust stabilizing controller achieves the robust performance as well.

The robust stabilization result in this section poses a modeling problem: How to identify the normalized coprime factors with modeling error quantified in $\mathcal{H}_\infty$ norm? In fact by Remark 1, the uncertainty $\Delta(z)$ can be allowed to be unstable, provided that the condition (6.17) holds. This problem will be investigated in the next section.

6.3 Robust Modeling

Modeling of the normalized coprime factors is studied in this section. It is assumed that the experimental data consist of frequency-response measurements of the physical plant, corrupted by an additive-type noise:

$$E_k = P(e^{j\omega_k}) + \eta_k, \quad \omega_k \in [0,\ \pi], \quad \text{for } k = 1, 2, \cdots, N \tag{6.33}$$

It is also assumed that frequency shape of the plant P represents the performance of the loop gain. If this is not true, then weighting functions should be employed to shape the plant P that is discussed in detail in [6]. The synthesis of the weighting function can be related to filtering of the measurement noise $\{\eta_k\}_{k=1}^N$. It is often the case that the low-frequency range is the operation band, while the high-frequency range is not. Thus, integrators, or low-pass filters are the typical weighting functions to shape the plant that is beneficial to improve the signal-to-noise ratio in experimental data sequence $\{E_k\}_{k=1}^N$. For this reason, it is assumed that

$$\eta_k \in \mathcal{N}_\varepsilon(E) := \left\{ \mathcal{N} :\ \sigma_{\max}((I + E_k E_k^*)^{-1/2}\mathcal{N}) \leq \varepsilon \right\} \tag{6.34}$$

with superscript * denoting transpose and conjugate. Hence, in the low-gain frequency range the additive error in the plant gain measurements is small, and in the high-gain frequency

range it is large, in contradiction with the assumption in [5]. It is also noted that Equation (6.34) involves only experimental data, and thus the noise level ε can be estimated.

The first difficulty in modeling is the experimental frequency response data that is obtained for the physical plant P, rather than for its normalized coprime factors. Denote

$$\Phi[P] = \begin{bmatrix} P^{\sim} \\ -I \end{bmatrix} (I + PP^{\sim})^{-1} \begin{bmatrix} P & -I \end{bmatrix} = \begin{bmatrix} \tilde{N}^{\sim} \\ -\tilde{M}^{\sim} \end{bmatrix} \begin{bmatrix} \tilde{N} & -\tilde{M} \end{bmatrix} \tag{6.35}$$

where $P(z) = \tilde{M}^{-1}(z)\tilde{N}(z) = N(z)M^{-1}(z)$ with $\{\tilde{M}, \tilde{N})\}$ and $\{M, N\}$ left and right normalized coprime factors of $P(z)$. The next result is useful.

Proposition 1. *Let $P_0(z)$ in Equation* (6.1) *be the nominal model with $G_0(z)$ as in Equation* (6.6). *Let $P(z)$ in Equation* (6.10) *be the true plant and $\Delta(z)$ as in Equation* (6.11) *satisfying condition* (6.17) *and $\|\Delta\|_\infty \leq \delta$. Then*

$$\delta_0 := \|\Delta\|_\infty = \|\Phi[P]G_0\|_\infty \tag{6.36}$$

holds, where $\Phi[P]$ is the same as in Equation (6.35). *Furthermore, the following holds*

$$\delta_d = \|\Delta(j\omega_k)\|_{\ell\infty} := \max_k \sigma_{\max}(\Delta(e^{j\omega_k})) \leq \|\Phi[E_k]G_0(e^{j\omega_k})\|_{\ell\infty} + \varepsilon \tag{6.37}$$

Proof. In light of Remark 1,

$$\delta_0 = \|\Delta\|_\infty = \inf_{Q \in \mathcal{L}_\infty, \mathrm{wno}(Q)=0} \|G_0 - GQ\|_\infty \tag{6.38}$$

If condition (6.17) is true, then by [8],

$$\delta_0 = \left\| \begin{bmatrix} \tilde{N} & -\tilde{M} \end{bmatrix} \begin{bmatrix} M_0 \\ N_0 \end{bmatrix} \right\|_\infty = \|(I - GG^{\sim})G_0\|_\infty = \|\Phi[P]G_0\|_\infty \tag{6.39}$$

By a similar argument as in [8] and the assumption in Equation (6.34),

$$\begin{aligned} \|\Phi[E_k] - \Phi[P_0(e^{j\omega_k})]\|_{\ell\infty} &= \|(I + P_0(e^{j\omega_k})P_0(e^{j\omega_k})^*)^{-1/2}\eta_k(I + E_k^*E_k)^{-1/2}\|_{\ell\infty} \\ &\leq \varepsilon \end{aligned}$$

holds. Thus, the discretized error function satisfies

$$\begin{aligned} \delta_d &= \|\Delta(e^{j\omega_k})\|_{\ell\infty} = \|\Phi[P(j\omega_k)]G_0(e^{j\omega_k})\|_{\ell\infty} \\ &\leq \|\Phi[E_k]G_0(e^{j\omega_k})\|_{\ell\infty} + \| \left(\Phi[P(e^{j\omega_k})] - \Phi[E_k]\right) G_0(e^{j\omega_k})\|_{\ell\infty} \\ &\leq \|\Phi[E_k]G_0(e^{j\omega_k})\|_{\ell\infty} + \varepsilon \end{aligned}$$

by the fact that $G_0(e^{j\omega_k})^*G_0(e^{j\omega_k}) = I$ for $k = 1, 2, \cdots, N$.

Proposition 1 shows that the modeling error is hinged to minimization of

$$\begin{aligned} \delta_{\max} &= \|\Phi[E_k]G_0(e^{j\omega_k})\|_{\ell\infty} \\ &= \left\| \begin{bmatrix} E_k^* \\ -I \end{bmatrix} (I + E_kE_k^*)^{-1} \begin{bmatrix} E_k^* & -I \end{bmatrix} G_0(e^{j\omega_k}) \right\|_{\ell\infty} \end{aligned} \tag{6.40}$$

Indeed, as $M(z)$ and $N(z)$ are right normalized coprime factors of $P(z)$,

$$M(z) = [I + P(z)^{\sim}P(z)]^{-1/2}, \qquad N = P(z)[I + P(z)^{\sim}P(z)]^{-1/2}$$

Hence, large abrupt variations of the frequency response of plant $P(z)$ are smoothed away in $M(e^{j\omega_k})$ and $N(e^{j\omega_k})$. Moreover if $G_0(z)$ admits a smooth frequency response, the modeling error $\delta_0 = \|\Delta\|_\infty$ tends to be close to $\delta_{\max}$, provided that the noise bound ε is small, and the number of experimental data points is large. The rest of this section will thus focus on seeking the model $G_0(z)$ such that $\delta_{\max}$ is minimized.

Parameterize $G_0(z)$ as

$$G_0(z) = \begin{bmatrix} M_0(z) \\ N_0(z) \end{bmatrix} = \begin{bmatrix} R(z) \\ S(z) \end{bmatrix} T^{-1}(z) \tag{6.41}$$

$$X(z) = \begin{bmatrix} R(z) \\ S(z) \end{bmatrix} = \sum_{k=0}^{n} \begin{bmatrix} R_k \\ S_k \end{bmatrix} z^{-k} = \sum_{k=0}^{n} X_k z^{-k} \tag{6.42}$$

where $R(z)^{\sim}R(z) + S(z)^{\sim}S(z) = T(z)^{\sim}T(z)$ with $T(z)$ square of size $m \times m$ and X_k of size $(p+m) \times m$. Recall that m is the number of inputs and p the number of outputs of the plant. Stability of the left normalized coprime factors requires that $\det[T(z)] \neq 0, \forall |z| \geq 1$. Clearly $T(z)$ is a spectral factor of $R(z)^{\sim}R(z) + S(z)^{\sim}S(z)$ that is a polynomial of z and z^{-1}. Denote

$$\underline{X}' = \begin{bmatrix} X_0' & X_1' & \cdots & X_n' \end{bmatrix}, \qquad \Omega_k = \begin{bmatrix} I & e^{-j\omega_k} I & \cdots & e^{-jn\omega_k} I \end{bmatrix} \tag{6.43}$$

All the above identities have size $(p+m) \times (p+m)$. The following relations hold

$$X(e^{j\omega_k}) = \begin{bmatrix} R(e^{j\omega_k}) \\ S(e^{j\omega_k}) \end{bmatrix} = \Omega_k \underline{X}, \quad T(e^{j\omega_k})^* T(e^{j\omega_k}) = \underline{X}' \Omega_k^* \Omega_k \underline{X} \tag{6.44}$$

It follows from the proof of Proposition 1 and Equation (6.40) that for $k = 1, 2, \cdots, N$,

$$\begin{aligned} \delta_{\max}^2 I &\geq \begin{bmatrix} M_0(e^{j\omega_k})^* & N_0(e^{j\omega_k})^* \end{bmatrix} \Phi[E_k] \begin{bmatrix} M_0(e^{j\omega_k}) \\ N_0(e^{j\omega_k}) \end{bmatrix} \\ &= T^{-1}(e^{j\omega_k})^* \begin{bmatrix} R_0^*(e^{j\omega_k}) & S_0^*(e^{j\omega_k}) \end{bmatrix} \Phi[E_k] \begin{bmatrix} R(e^{j\omega_k}) \\ S(e^{j\omega_k}) \end{bmatrix} T^{-1}(e^{j\omega_k}) \end{aligned}$$

Multiplying $T(e^{j\omega_k})^*$ from left, and $T(e^{j\omega_k})$ from right, and using the relations in Equation (6.44) give the following equivalent expression:

$$\underline{X}' \Omega_k^* \left(\delta_{\max}^2 I - \Phi[E_k] \right) \Omega_k \underline{X} \geq 0 \tag{6.45}$$

for $1 \leq k \leq N$. Hence $\delta_{\max}$ can be alternatively defined as

$$\delta_{\max} := \inf \left\{ \delta : \ \underline{X}' \Omega_k^* \left(\delta^2 I - \Phi[E_k] \right) \Omega_k \underline{X} \geq 0 \ \text{ for } \ 1 \leq k \leq N \right\} \tag{6.46}$$

Modeling of the right normalized coprime factors $\{M_0(z), N_0(z)\}$ amounts to minimization of $\delta_{\max}$ over all full rank matrices X with size $(n+1)(p+m) \times m$ due to the fact that $G_0(e^{j\omega})$ is an orthogonal matrix and thus has full rank at each frequency.

For a fixed value of δ, it is difficult to find a feasibility solution $\underline{X}$ having full rank such that $\underline{X}' \Omega_k^* \left(\delta^2 I - \Phi[E_k] \right) \Omega_k \underline{X} \geq 0$ for $1 \leq k \leq N$. In what follows, an alternative is sought to synthesize the matrix $\underline{X}$ that parameterizes the identified plant P_0. Instead of searching for an optimal solution to $\delta_{\max}$, SK-iteration [7] will be employed to find a local minimum for the ℓ^∞ objective function. To be more specific, at the ith step SK-iteration seeks X of size $(n+1)p \times m$ having full rank such that it minimizes

$$\delta_q(\underline{X}) = \| \left(\underline{X}_i' \Omega_k^* \Omega_k \underline{X}_i\right)^{-1/2} \Phi[E_k] \Omega_k \underline{X} \|_{\ell^q}, \tag{6.47}$$

with $\|\cdot\|_{\ell^q}$ the ℓ^q-norm and assigns $\underline{X}_{i+1} = \underline{X}$ for $i = 0, 1, \cdots$. To avoid numerical difficulty, R_0, the first m rows of $\underline{X}$, can be normalized to identity. The initial solution $\underline{X}_0$ is arbitrary provided that $\Omega_k \underline{X}_0$ has full rank for each k. For instance, $\underline{X}_0$ can be taken to satisfy $\underline{X}_0 \Omega_k = I$ for $1 \le k \le N$. By norm properties, $\delta_p(\underline{X})$ in Equation (6.47) is a nondifferentiable and convex function with respect to Ω_k. Thus various algorithms for convex programming can be employed to search for the optimal solution to $\delta_p(\underline{X})$. Moreover, for the case of $p = 2$, an analytic solution is available for the minimum of $\delta_p(\underline{X})$ as in Equation (6.47) that is far more efficient to compute than convex programming. As indicated in [1], the SK-iteration for δ_p with $p = 2$ may also lead to a small error in $\delta_{\max}$. It should be clear that if SK-iteration converges, the limit $\{\underline{X}_i\}$ is a local minimum of δ_p. Therefore the SK-iteration provides an effective numerical tool for searching the identified model $P_0(z)$ that is parameterized by $P_0(s) = S(z)R^{-1}(z)$. However, minimization of δ_p alone is not adequate, even if $\delta_{\max}$ is small because condition (6.17) may not be satisfied. This will be addressed in the next result.

Theorem 3. *Suppose that P_0 has no fewer unstable poles than P, i.e. $\eta(P_0) + \eta_0(P_0) \ge \eta(P) + \eta_0(P_0)$. Then*

$$\kappa = \text{wno}\left[\det(I + P^\sim P_0)\right] + \eta(P_0) - \eta(P) - \eta_0(P) \ge 0 \tag{6.48}$$

provided that $\|N_0^\sim (P - P_0) M_0\|_\infty < 1$ or

$$\left\| (I + P_0^\sim P_0)^{-1/2} P_0^\sim (P - P_0)(I + P_0^\sim P_0)^{-1/2} \right\|_\infty < 1 \tag{6.49}$$

Proof. Denote $E_{id}(z) = P_0(z) - P(z)$. Then

$$\begin{aligned} I + P^\sim P_0 &= I + P_0^\sim P_0 - E_{id}^\sim P_0 = (M_0^\sim)^{-1} M_0^{-1} - E_{id}^\sim N_0 M_0^{-1} \\ &= (M_0^\sim)^{-1} \left(I - M_0^\sim E_{id}^\sim N_0\right) M_0^{-1} \end{aligned}$$

as $P_0 = \tilde{M}_0^{-1} \tilde{N}_0$ is a left normalized coprime factorization. Thus,

$$\det(I + P^\sim P_0) = \det(I - M_0^\sim E_{id}^\sim N_0) / [\det(M_0) \det(M_0^\sim)] \tag{6.50}$$

It follows from the above and condition (6.49) that

$$\begin{aligned} \text{wno}\left[\det(I + P^\sim P_0)\right] &= \text{wno}\left[\det(I - M_0^\sim E_{id}^\sim N_0)\right] - \text{wno}\left[\det(M_0)\right] - \text{wno}\left[\det(M_0^\sim)\right] \\ &= -\text{wno}\left[\det(M_0)\right] - \text{wno}\left[\det(M_0^\sim)\right] \end{aligned}$$

Using the property for the winding number and $\eta(M_0^{-1}) = \eta(P_0)$,

$$\text{wno}\left[\det(I + P^\sim P_0)\right] = \eta_0(P_0) \tag{6.51}$$

It is now straightforward to see that

$$\begin{aligned} \kappa &= \text{wno}\left[\det(I + P^\sim P_0)\right] + \eta(P_0) - \eta(P) - \eta_0(P) \\ &= \eta(P_0) + \eta_0(P_0) - \eta(P) - \eta_0(P) \ge 0 \end{aligned}$$

This concludes the proof.

The condition (6.49) is difficult to enforce unless $\eta_0(P_0) = \eta_0(P)$. The next result provides a simpler condition.

Theorem 4. *Suppose that P_0 has no fewer poles than P in $\mathbf{C}_+$, i.e., $\eta(P_0) \geq \eta(P)$. Then $\kappa = \text{wno}\left[\det(I + P^{\sim}P_0)\right] + \eta(P_0) - \eta(P) - \eta_0(P) \geq 0$, if*

$$\left\|(I + P^{\sim}P)^{-1/2}P^{\sim}(P - P_0)(I + P^{\sim}P)^{-1/2}\right\|_{\infty} < 1 \tag{6.52}$$

Proof. We follow the same procedure as in the proof of Theorem 3 with $E_{id}(z) = P(z) - P_0(z)$. Then

$$\begin{aligned} I + P^{\sim}P_0 &= I + P^{\sim}P - P^{\sim}E_{id} = (M^{\sim})^{-1}M^{-1}(z) - P^{\sim}E_{id} \\ &= (M^{\sim})^{-1}\left(I - N^{\sim}E_{id}M\right)M^{-1} \end{aligned}$$

as $P(z) = N(z)M^{-1}(z)$ is a right normalized coprime factorization. Thus,

$$\det(I + P^{\sim}P_0) = \det(I - N^{\sim}E_{id}M)/[\det(M)\det(M^{\sim})]$$

It follows from the condition (6.52) that

$$\begin{aligned} \text{wno}\left[\det(I + P^{\sim}P_0)\right] &= \text{wno}\left[\det(I - N^{\sim}E_{id}M)\right] - \text{wno}\left[\det(M)\right] - \text{wno}\left[\det(M^{\sim})\right] \\ &= \eta_0(P) \end{aligned}$$

Hence, the hypothesis $\eta(P_0) \geq \eta(P)$ implies that

$$\kappa = \text{wno}\left[\det(I + P^{\sim}P_0)\right] + \eta(P_0) - \eta(P) - \eta_0(P) = \eta(P_0) - \eta(P) \geq 0$$

This concludes the proof.

While Theorem 3 requires $\eta_0(P_0) = \eta_0(P)$, the condition of Theorem 4 is more mild, and easy to be incorporated into numerical computation. Indeed, $\|(I + P^{\sim}P)^{-1/2}P^{\sim}\|_{\infty} < 1$, if P has no pole on the imaginary axis and in general $\|(I + PP^{\sim})^{-1/2}P^{\sim})\|_{\infty} \leq 1$. Thus condition (6.49) is ensured if

$$\begin{aligned} \|(P_0 - P)(I + P^{\sim}P)^{-1/2}\|_{\infty} &= \|(P_0 - P)M\|_{\infty} = \left\|\begin{bmatrix} P_0 & -I \end{bmatrix} G\right\|_{\infty} \\ &= \left\|\begin{bmatrix} P_0 & -I \end{bmatrix} GG^{\sim}\right\|_{\infty} < 1 \end{aligned} \tag{6.53}$$

The above can be written alternatively by defining

$$\Psi[P] = \begin{bmatrix} -P \\ I \end{bmatrix}(I + P^{\sim}P)^{-1}\begin{bmatrix} -P^{\sim} & I \end{bmatrix} = \begin{bmatrix} -N \\ M \end{bmatrix}\begin{bmatrix} -N^{\sim} & M^{\sim} \end{bmatrix} \tag{6.54}$$

In addition, the following equality also holds

$$\|\Psi[P(e^{j\omega_k})] - \Psi[E_k]\|_{\ell^{\infty}} \leq \varepsilon \tag{6.55}$$

Note that we can rewrite the condition in Equation (6.52) as

$$\left\|\begin{bmatrix} I & P_0 \end{bmatrix}\Psi[P]\right\|_{\infty} < 1 \tag{6.56}$$

Due to the fact that $P(z)$ is unknown and only N experimental data are available, the above condition is replaced by

$$\left\|\begin{bmatrix} I & P_0(e^{j\omega_k}) \end{bmatrix}\Psi[E_k]\right\|_{\ell^{\infty}} < 1 - \varepsilon \tag{6.57}$$

Because the modeling error is, according to [8],

$$\begin{aligned}\delta = \|\Delta\|_\infty &= \|(I + P^\sim P)^{-1/2}(P_0 - P)(I + P_0 P_0^\sim)^{-1/2}\|_\infty \\ &\le \|(P_0 - P)(I + P^\sim P)^{-1/2}\|_\infty\end{aligned}$$

the condition (6.53) is consistent with the minimization of $\|\Delta\|_\infty$. Therefore, to ensure condition (6.53), the ℓ^q objective function in Equation (6.47) can be modified as

$$\delta_q(\alpha, \underline{X}) = \left[\delta_q^q(\underline{X}) + \alpha \left\| \left[I \;\; \Omega_k^p \underline{X} (\Omega_k^m \underline{X}_i)^{-1} \right] \Psi[E_k] \right\|_{\ell^q}^q \right]^{1/q} \tag{6.58}$$

with $\alpha > 0$ a scalar weighting, and $\Omega_k^m = \Omega_k S_m$ and $\Omega_k^m = \Omega_k S_p$ where

$$S_m = I_{n+1} \otimes \begin{bmatrix} I_m \\ 0 \end{bmatrix}, \qquad S_p = I_{n+1} \otimes \begin{bmatrix} 0 \\ I_p \end{bmatrix}, \qquad S_m + S_p = I$$

With a suitable choice of the weighting, it is possible to obtain an identified plant P_0 such that Equation (6.53) is true. The condition $\eta(P_0) \ge \eta(P)$ is relatively easy to satisfy since if it is violated, an unstable transfer function, with prescribed number of unstable poles, having arbitrarily small magnitude response can be added into P_0 that has little effect on the modeling error. Thus, the SK-iteration provides a simple, and effective method for the modeling problem studied in this chapter.

Remark 3. For robust stabilization, $\kappa = 0$ is required in light of the condition (6.17). In the case $\eta(P_0) > \eta(P)$, we may employ optimal Hankel norm approximation to reduce the number of unstable poles of $P_0(z)$. If the modeling error is δ_0 with the approximant $P_0(z)$. Then with the optimal Hankel-norm approximant, the new $P_0(z)$ has the same number of unstable poles as $P(z)$ and the total modeling error is no more than $2\delta_0$. The argument is similar to the two-stage nonlinear algorithm in [2] that is omitted.

Once the approximate model $P_0(z)$ is obtained with modeling error δ, then the robust feedback controller can be synthesized as in the previous section. In the next section our integrated modeling and control method will be demonstrated with the JPL flexible structure as the application example.

6.4 Simulation Results for the JPL Flexible Structure

In this section our proposed integrated approach to modeling and control of flexible structures will be tested with the JPL flexible structure [1], for which a schematic diagram is given in the graphic image of Figure 6.2.

Because the structure has a single input/output and is stable, modeling is simplified by parameterizing it as a fraction of two polynomials:

$$P_0(z) = \frac{a(z)}{b(z)} = \frac{a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}}{1 + b_1 z^{-1} + \cdots + b_n z^{-n}} \tag{6.59}$$

The condition in Theorem 4 is automatically satisfied. In addition, the error function in Proposition 1 can be simplified into

$$\delta_0 = \sup_\omega \left| \frac{P_0(e^{j\omega}) - P(e^{j\omega})}{\sqrt{1 + |P_0(e^{j\omega})|^2}\sqrt{1 + |P(e^{j\omega})|^2}} \right| \tag{6.60}$$

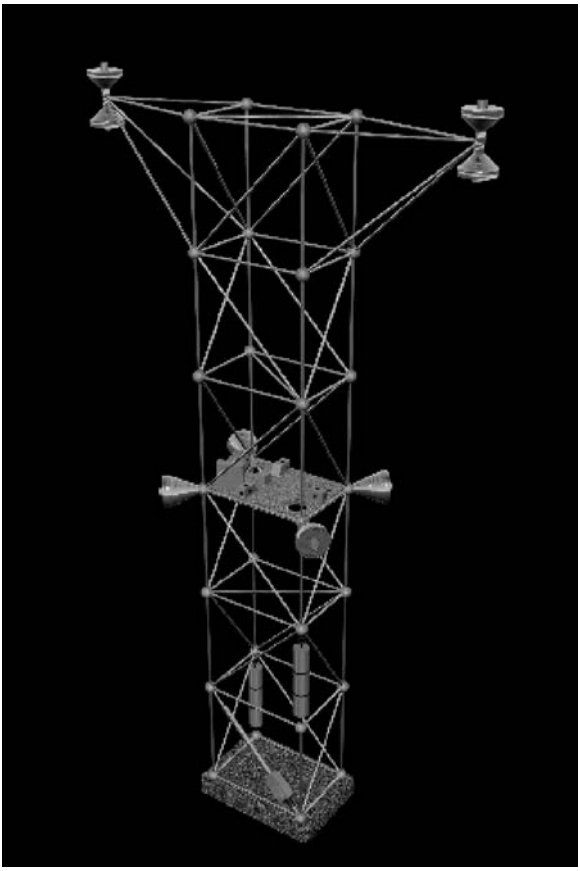


Fig. 6.2. JPL structure

that is, the ν-metric defined in [8] for $\kappa = 0$, with κ defined in Equation (6.48). Because $P(z)$ is unavailable except its experimental measurement data E_k, the discretized error uncertainty

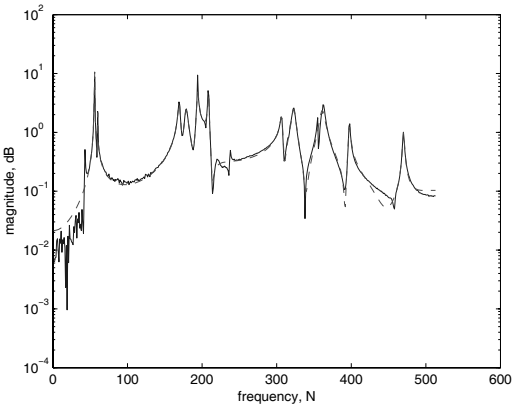
$$|\Delta(e^{j\omega_k})| = \left| \frac{P_0(e^{j\omega_k}) - E_k}{\sqrt{1+|P_0(e^{j\omega_k})|^2}\sqrt{1+|E_k|^2}} \right| \tag{6.61}$$

will be used. Let Ω_k be as in Equation (6.43) and parameter vector be

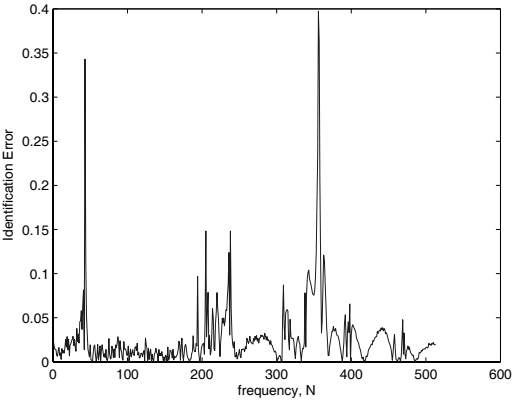
$$\underline{X}' = \begin{bmatrix} 0 & a_0 & b_1 & a_1 & \cdots & b_n & a_n \end{bmatrix} \tag{6.62}$$

Then, the following identity holds

$$\begin{bmatrix} b(e^{j\omega_k}) \\ a(e^{j\omega_k}) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \Omega_k \underline{X} \tag{6.63}$$



(a)



(b)

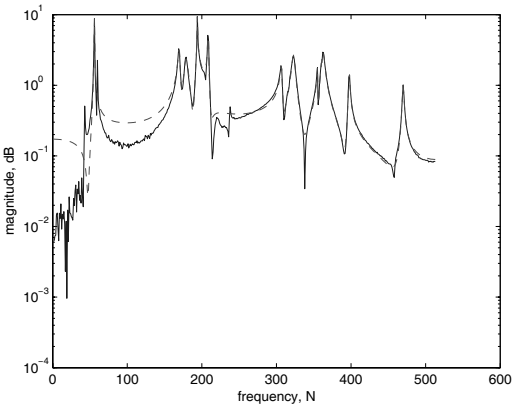
Fig. 6.3. (a) Frequency response of the true plant (solid) and estimated plant (dashed); (b) Identification error plot for the curve-fitting method.

Hence the error uncertainty has the form

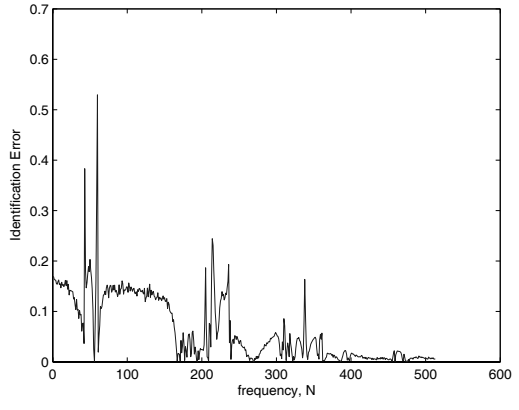
$$|\Delta(e^{j\omega_k})| = \left| \frac{E_k + \begin{bmatrix} E_k & -1 \end{bmatrix} \Omega_k \underline{X}}{\sqrt{h_k(\underline{X})}\sqrt{1+|E_k|^2}} \right| \tag{6.64}$$

$$h_k(\underline{X}) = \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \Omega_k \underline{X} \right)^* \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \Omega_k \underline{X} \right)$$

The experimental data are given by $N = 1024$ frequency-response measurements uniformly distributed over $[0,\ 2\pi)$. In fact, only 513 of them are independent measurements. Its magnitude response in dB is plotted next with solid line. We employed the SK-iteration [7] that is a popular curve-fitting algorithm to compute the parameter vector $\underline{X}$ with ℓ_2 norm that



(a)



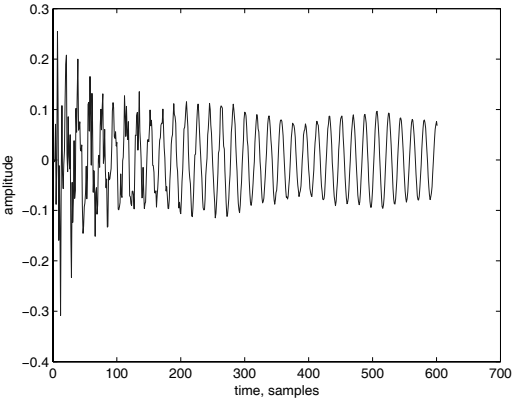
(b)

Fig. 6.4. (a) Frequency response of the true plant (solid) and estimated plant (dashed); (b) Identification error plot for the Kung's algorithm.

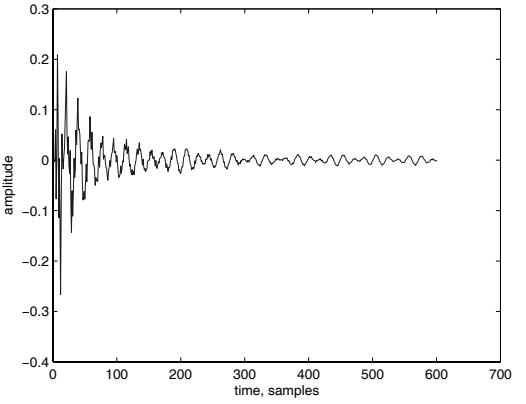
uses the previously computed parameter vector in $h(\cdot)$ to obtain the new estimated parameter vector. We also employed Kung's algorithm to identify the open-loop plant as a comparison. The nominal model of the identified plant is represented by a 24-order model.

In Figure 6.3, the magnitude responses of the true and estimated plants are plotted for the identification method based on the curve-fitting algorithm as discussed earlier. Because the error criterion is ℓ_2, the peak error is not minimized. The interesting part is that the estimated model is stable. Figure 6.4 shows the corresponding results based on Kung's algorithm.

Kung's algorithm tends to achieve a small error in the form of additive uncertainty. The identification error plot in Figure 6.4b corresponds to the discretized error uncertainty in Equation (6.61). The maximum error using the curve-fitting method is $|\Delta|_{max} = 0.3972$, while



(a)



(b)

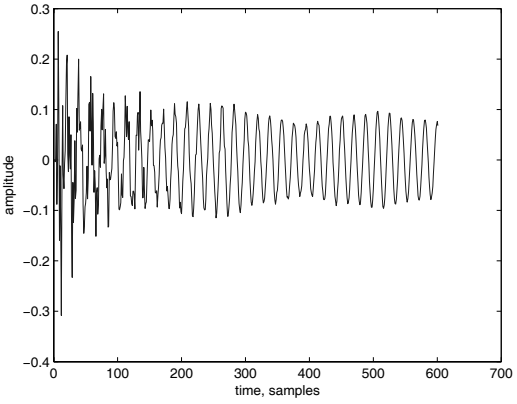
Fig. 6.5. (a) Open-loop response of the true plant. (b) Closed-loop response with the $\mathcal{H}_\infty$ loopshaping controller based on the curve-fitting model.

the maximum error using Kung's algorithm is $|\Delta|_{max} = 0.5296$. It can be seen that the curve-fitting method performs better at reducing the error uncertainty.

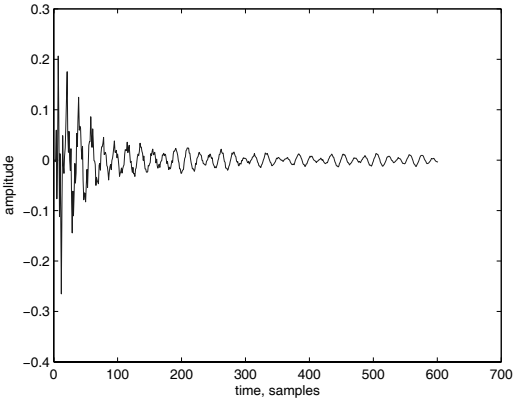
After the estimated models are available, the loopshaping algorithm in Section 6.2 is applied to design the feedback controller. Figure 6.5 shows the disturbance rejection performance based on the model obtained from the curve-fitting method compared with the performance without feedback control.

The disturbance rejection performance for the case based on Kung's algorithm is shown in Figure 6.6.

It needs to be pointed out that with the curve-fitting model, the corresponding optimal value as in Equation (26) is given by $\gamma_{\rm opt} = 1.5819$, with Kung's algorithm model, $\gamma_{\rm opt} =$



(a)



(b)

Fig. 6.6. (a) Open-loop response of true plant. (b) Closed-loop response with the $\mathcal{H}_\infty$ loopshaping controller based on Kung's algorithm model.

1.6223, which are very close to each other. In addition $\gamma = 3$ is used in the design of the $\mathcal{H}_\infty$ loopshaping controller for both cases. The disturbance is a unit impulse. We comment that it is hard to see the differences in terms of the output responses even though the modeling errors are quite different.

6.5 Conclusion

An integrated approach to modeling and control of uncertainty systems is investigated in discrete time. It is assumed that the design specifications for a feedback control system are given

in the frequency domain, specified by the frequency shape of the open-loop system. Based on the samples of the experimental frequency-response measurements, normalized coprime factors of the shaped plant model are identified with quantification of the modeling error in $\mathcal{H}_\infty$ norm. In addition, feedback controllers are synthesized to stabilize the true but unknown plant based on the $\mathcal{H}_\infty$ loopshaping method. A complete design procedure has been presented that integrates modeling and control in the frequency domain. The performance of the feedback control system with the true plant is guaranteed provided that the modeling error bound quantified for the ν-metric uncertainty is suitably small. The results are applied to modeling and control of lightly damped systems with a JPL flexible structure as the application platform. Different existing algorithms are employed to identify the plant model, and the modeling errors in terms of the normalized coprime factor uncertainty are compared in the $\mathcal{H}_\infty$ norm. The simulation results are presented to illustrate the proposed integrated approach based on the $\mathcal{H}_\infty$ loopshaping design and frequency-domain identification method. It is interesting to observe that the disturbance rejection performances are very similar even though different identified models are used with quite different modeling errors in the frequency domain, which demonstrates the advantage of our proposed integrated approach in this chapter.

Acknowledgments

The authors would like to thank Professor Roy Smith for providing the graphic image of the JPL flexible structure shown in Figure 6.2.

References

[1] Bayard, D. (1993). Multivariable state-space identification: algorithms and experimental results. *Proceedings of the American Control Conference*, pp. 3038–3042.

[2] Chen, J. and Gu, G. (2000). *Control-Oriented System Identification - An $\mathcal{H}_\infty$ Approach*. John Wiley Publisher.

[3] Georgiou, T. and Smith, M. C. (1990). Optimal robustness in the gap metric. *IEEE Transactions on Automatic Control* **35**, 673–686.

[4] Green, M. and Limebeer, D. (1995). *Linear Robust Control*. Prentice Hall.

[5] Helmicki, J., Jacobson, C., and Nett, C. (1991). Control Oriented System Identification: A worst case/deterministic approach in $\mathcal{H}_\infty$. *IEEE Transactions on Automatic Control* **36**(10), 1163–1176.

[6] McFarlane, D. and Glover, K. (1990). *Robust Controller Design Using Normalized Coprime Factor Plant Descriptions*. Springer-Verlag, Lecture Notes in Control and Information Sciences edition.

[7] Sanathanan, C. and Koerner, J. (1963). Transfer function synthesis as a ratio of two complex polynomials. *IEEE Transactions on Automatic Control* **8**, 56–58.

[8] Vinnicombe, G. (1993). Frequency domain uncertainty and the graph topology. *IEEE Transactions on Automatic Control* **38**, 1371–1383.

[9] Zhou, K., Doyle, J. C., and Glover, K. (1996). *Robust and Optimal Control*. Prentice–Hall.

Part III

Vision and Sound

7

Robust Identification and Model (In)Validation of Active-vision Systems

Tamer Inanc[1], Mario Sznaier[2], and Octavia Camps[2]

[1] ECE Department, University of Louisville, Louisville, KY 40292, USA
t.inanc@louisville.edu
[2] ECE Department, Northeastern University, Boston, MA 02115, USA
{msznaier,camps}@ece.neu.edu

Summary. Recent hardware advances have rendered active-vision a viable option for a diverse spectrum of applications ranging from MEMS manufacture to assisting individuals with disabilities. However, there are relatively few instances where these techniques have been successfully applied in uncontrolled environments. This can be traced to the *fragility* of active-vision systems designed using classical methods. In this chapter we show how a combination of robust identification and model (in)validation tools can be used to obtain models of these systems that combine the dynamics of the physical hardware and the image-processing algorithms, along with an associated worst-case uncertainty description, suitable to be used in the context of modern robust control tools. As shown here, this combination leads to systems that achieve good tracking performance under a wide range of conditions. These results are experimentally validated using a BiSight/UniSight robotic head–eye platform.

Key words: Active vision, visual servoing, robust control, robust identification, model (in)validation.

7.1 Background

7.1.1 Introduction and Motivation

In the past few years, active-vision systems – *i.e.* systems incorporating vision as an integral part of the loop – have emerged as a viable option for a large number of applications, ranging from vision-based assembly [22, 23, 55, 68, 80] to vision-assisted surgery [35, 39, 46, 78, 79], assisting individuals with disabilities [42, 67, 69, 77], and intelligent vehicle highway systems [6, 25, 66, 75]. In practice, using active systems in dynamic scenes requires both real-time visual processing and real-time closed-loop control. Recent hardware developments have made this now possible, leading to a number of systems [16, 17, 20, 27, 56, 63].

Active-vision systems appeared as far back as the late 1970s [33] and have been the subject of much research since. An excellent survey of the state-of-the art as of 1996 and as of 2006 and a comprehensive literature reviews up to then can be found in [34] and [9, 10], respectively. Earlier active-vision systems dealt with stability issues by detuning the controller,

at the expense of performance, until stability was accomplished [34]. Later approaches combined PID controllers with some prediction to explicitly address time delays [7, 24]. However, these predictors can tolerate only small amounts of uncertainty [60]. Furthermore, the combination PID controller/predictor had to be tuned empirically, a process that entailed considerable experimentation [18].

Set-point tracking can be improved by using a two-degrees of freedom (2-DOF) controller [18]. However, this approach can improve neither robustness nor disturbance rejection [45]. Optimal controllers have the potential to improve performance [30, 31, 52], but can lead to fragile systems [52]. Empirical results [51, 65] show that adaptive controllers can accommodate calibration errors. However, this approach does not allow for achieving an *a priori* established robustness level or to balance robustness *versus* performance.

A switching approach to visual servo control was introduced in [26]. In this approach two controllers were designed: one that relied on the homography between initial and goal images, and a second one based on an affine transformation approximating the motion between the initial and goal camera configurations. However, stability of the switched system was not formally addressed and it is difficult to prove.

Recent work has recognized the fact that robustness issues are central to the success of active-vision systems. Robustness to calibration errors and estimation noise has been addressed in [28, 74] and [62], respectively. 2.5D or homography-based visual servo regulation and tracking was addressed in [21] and [11], respectively, where a Lyapunov-based adaptive control strategy was used to compensate for unknown depth measurements. However, a fixed camera position was assumed (for regulation) and uncertainties in the camera-calibration parameters as well as mechanical system dynamics were not taken into account. Furthermore, the intrinsic[3] camera-calibration matrix was needed for tracking.

Nonlinear tracking controllers were introduced in [81], where two position-tracking controllers were developed: One addressed nonlinear robot dynamics and the second compensated for parametric uncertainties in the dynamics as well as in the camera-calibration parameters. However, this approach assumed a fixed-camera configuration and a limited camera orientation, and required calculating the inverse Jacobian. Moreover, higher overshoots occurred during the transient response due to the high number of parameter estimates needed by the adaptive controller. More recently, the adaptive controller proposed in [1] avoided overparameterization and eliminated the need for restricting the camera orientation, but it still required the calculation of the inverse Jacobian.

An alternative approach relies on the use of self-calibration methods to measure the camera intrinsic parameters [15, 43, 54]. However, these techniques fail for camera-in-hand (camera translation) applications. Indeed, [70] showed that in these cases, it is impossible to measure the camera internal parameters. Visual servoing invariant to changes in camera-calibration parameters was considered in [41].

An additional drawback of the approaches mentioned above, is the implicit use of a *"separation principle"*, where the computer vision and control aspects of the problem are treated, to a very large extent, independently. The interaction between these components was explicitly taken into account in [57], but neither model uncertainty nor measurement errors were considered.

In [72] we used a simplified scenario, restricted motion of a single target in the absence of clutter, as a vehicle to illustrate how to address the issues indicated above using robust control

[3] The camera-calibration parameters consist of intrinsic and extrinsic parameters. The intrinsic parameters are the image center, the camera scale factor and the camera magnification factor. The extrinsic parameters are the camera position and orientation.

tools. Motivated by these results and those in [71], in this chapter we show *how recently developed robust identification, model (in)validation and robust control synthesis techniques can be brought to bear on the problem to synthesize robust active-vision systems capable of delivering good performance for a range of conditions.*

This chapter is organized as follows: in Section 7.1 we introduce the notation that we use and give a summary of the relevant theoretical and algorithmic tools. In Section 7.2 we describe the identification and control issues that arise in the context of designing active-vision systems and propose a solution based on a combination of robust identification/robust control tools. In Section 7.3 we provide experimental results validating this approach In Section 7.4 we comment on the shortcomings of existing tools, indicate possible solutions, and benchmark the resulting systems against those obtained using classical design methods. Finally, in Section 7.5 we summarize our results and indicate directions for future research.

7.1.2 Notation

By $\mathcal{L}_\infty$ we will denote the Lebesgue space of complex-valued matrix functions essentially bounded on the unit circle, equipped with the norm:

$$\|G(z)\|_\infty \doteq ess \sup_{|z|=1} \overline{\sigma}(G(z))$$

where $\overline{\sigma}$ denotes the largest singular value. By $\mathcal{H}_\infty$ we denote the subspace of functions in $\mathcal{L}_\infty$ with a bounded analytic continuation outside the unit disk, equipped with the norm $\|G(z)\|_\infty \doteq ess\sup_{|z|>1} \overline{\sigma}(G(z))$. Also of interest is the Banach space $\mathcal{H}_{\infty,\rho}$ of transfer matrices in $\mathcal{H}_\infty$ that have analytic continuation outside the disk of radius $\rho < 1$, *i.e.* the space of exponentially stable systems with a stability margin of $(1-\rho)$, equipped with the norm $\|G(z)\|_{\infty,\rho} \doteq \sup_{|z|>\rho} \overline{\sigma}(G(z))$. $\mathcal{H}_{\infty,\rho}(K)$ denotes the open K–ball in $\mathcal{H}_{\infty,\rho}$, *i.e.* $\{H \in \mathcal{H}_{\infty,\rho} \colon \|H\|_{\infty,\rho} < K\}$. In the following, we will use simply $\mathcal{BH}_\infty$ for the case where $K = \rho = 1$.

Finally, ℓ^2 denotes the space of bounded real sequences $h = \{h_i\}$ equipped with the norm $\|h\|_{\ell^2}^2 \stackrel{\text{def}}{=} \sum_{i=0}^{\infty} h_i^2$. Similarly, ℓ^∞ denotes the space of bounded real sequences equipped with the norm $\|h\|_{\ell^\infty} \stackrel{\text{def}}{=} \sup_k |h_k|$.

7.1.3 Theoretical background

In this section, we summarize, for ease of reference, the theoretical and algorithmic results used to identify and validate a *combined* model of the active-vision system, and to synthesize a robust controller based on this model. A more detailed description can be found in several textbooks such as [12] and [60].

Robust Identification

In principle, a detailed model of an active-vision system can be obtained by using first principles to model the dynamics of both, the platform where the cameras are mounted and of the imaging-processing algorithms. However, the resulting model will depend on several parameters, including the optical parameters of the imaging systems and the mass of a camera-mounting structure such as a pan/tilt unit in our case. In addition, such an approach does not

fully capture the interaction between the imaging and mechanical portions of the system, leading to poor performance or even instability. Finally, classical identification procedures [40] are not well suited to be used in combination with robust control techniques (see [60], Chapter 10).

To avoid these difficulties in this chapter we will use recently developed nonparametric robust identification techniques that, starting from experimental data and some mild *a priori* assumptions on the plant, generate a nominal model as well as bounds on the worst-case identification error suitable to be used by robust control synthesis methods. In particular, since the experimental data used in this chapter originates from time-domain experiments and the controller-synthesis algorithm requires $\mathcal{H}_\infty$ bounds on the uncertainty description, we will use $\mathcal{H}_\infty/\ell^1$ identification [13]. In the following we briefly cover the fundamentals of this technique. A tutorial to the robust identification field and key to the relevant references can be found for instance in [60], Chapter 10.

The experimental data used by the $\mathcal{H}_\infty/\ell^1$ method consists of the first N_t samples, corrupted by additive noise, of the time response of the system
$y(k) = h(k) + \eta_t(k)$, $k = 0, \ldots, N_t - 1$ corresponding to a known input $u(k)$. The *a priori* assumptions are:

1. The system to be identified belongs to the class $\mathcal{H}_{\infty,\rho}(K)$ *i.e.* the set of exponentially stable systems with a stability margin of $(1-\rho)$, and a peak response to complex exponential inputs of K.
2. A bound ϵ_t of the measurement noise $\eta_t(k)$, *i.e.* $\|\eta_t\|_{\ell^\infty} \leq \epsilon_t$ is known.

The goal of the method is twofold:

1. Firstly, to establish whether or not the experimental data and the *a priori* assumptions are consistent, *i.e.* whether or not there exist models in $\mathcal{H}_{\infty,\rho}(K)$ that interpolate the data points within the experimental errors (the *"consistency"* problem, see Definition 8.1.1 in Chapter 8) and,
2. If so, to obtain one such model as well as a bound on the worst-case identification error.

It can be shown [53] that the consistency problem reduces to a (convex) linear matrix inequality (LMI) feasibility problem that can be efficiently solved using commercially available code. Specifically, the *a posteriori* experimental data and the *a priori* information are consistent if and only if there exists a vector $\mathbf{h} = [h_o, h_1, \ldots, h_{N_t-1}]$ such that the following LMIs hold:

$$M_R(\mathbf{h}) = \begin{bmatrix} R^{-2} & \frac{\mathcal{F}^T}{K} \\ \frac{\mathcal{F}}{K} & R^2 \end{bmatrix} > 0 \tag{7.1}$$

$$-\epsilon_t < \mathcal{F}^T \mathbf{u} - \mathbf{y}^t < \epsilon_t$$

where $R = \text{diag}[\,1 \;\; \rho \;\; \rho^2 \;\; \ldots \;\; \rho^{N_t-1}\,]$ and

$$\mathcal{F} = \begin{bmatrix} h_0 & h_1 & \ldots & h_{N_t-1} \\ 0 & h_0 & \ldots & h_{N_t-2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & h_0 \end{bmatrix} \tag{7.2}$$

$$\mathbf{y} = [y_0, y_1, \ldots, y_{N_t-1}]^T$$

$$\mathbf{u} = [u_0, u_1, \ldots, u_{N_t-1}]^T$$

Once the generalized Pick matrix M_R (7.1) is found, the set of all models consistent with both the *a priori* assumptions and the *a posteriori* experimental data can be parameterized as a linear fractional transformation (LFT) of a free parameter $q(z) \in \mathcal{BH}_\infty$, $G(z) = F_\ell[L(z), q(z)]$, where $L(z)$ depends only on M_R. In particular, the choice $q(z) = 0$ leads to the central model, of order less than or equal to the number of experimental data points N_t. In this case an explicit state-space realization of $G(z)$ is given by [3, 36]:

$$
\begin{aligned}
G(z) &= \left[\begin{array}{c|c} A_G & B_G \\ \hline C_G & D_G \end{array}\right] \\
A_G &= A - [C_-^T C_- + (A^T - I)M_R]^{-1} C_-^T C_- (A - I) \\
B_G &= -[(A^T - I)M_R + C_-^T C_-]^{-1} C_- \\
C_G &= C_+ [(A^T - I)M_R + C_-^T C_-]^{-1} C_-^T C_- (A - I) - C_+ (A - I) \\
D_G &= C_+ [(A^T - I)M_R + C_-^T C_-]^{-1} C_-^T
\end{aligned}
\tag{7.3}
$$

where

$$
A = \begin{bmatrix} 0 & I_{N_t \times N_t} \\ 0 & 0 \end{bmatrix},\ C_- = [\overbrace{1 \quad 0 \quad \ldots \quad 0}^{N_t}],\ C_+ = \frac{\mathbf{h}^T R}{K}
$$

Finally, the worst-case identification error is bounded by [12]:

$$
\|e\|_\infty \leq \sum_{k=0}^{N-1} \min \left\{ \epsilon \sum_{i=0}^{k} |w_i| . \frac{K}{\rho^k} \right\} + \frac{K \rho^{N-1}}{1 - \rho} \tag{7.4}
$$

where w_i depends only on the known input signal[4].

Model (In)Validation

In the previous section we have outlined a robust identification algorithm that generates nominal models and uncertainty descriptions suitable to be used by robust control synthesis methods. However, before this description can be used by the control engineer, it must be *validated*, based on additional experimental data. This leads to the following model (in)validation problem:

- Given experimental data, corrupted by additive noise, find whether or not this data could have been produced by the combination of the nominal model and some uncertainty in the uncertainty set.
- If the answer is negative, then the assumed model does not provide a correct description of the physical system.

In addition, these techniques allow for refining the uncertainty description, by establishing the lowest uncertainty level that does not invalidate the model. These bounds are usually much tighter than the one provided by Equation (7.4), leading to less conservative controller design.

Model (in)validation of linear time invariant (LTI) systems has been extensively studied in the past decade (see for instance [12] and references therein). For ease of reference, we

[4] In the simplest case where $u = \delta$ then $w_0 = 1, w_i = 0, i \geq 1$

quote the main result showing that in the case of unstructured uncertainty entering the plant as an LFT, model (in)validation reduces to a convex optimization problem that can be efficiently solved.

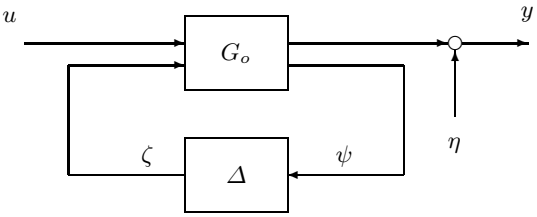


Fig. 7.1. Setup for model (in)validation

Consider the interconnection shown in Figure 7.1, where $G_o = \left[\begin{array}{c|c} P & Q \\ \hline R & S \end{array}\right]$ is a known given system, Δ represents unstructured bounded dynamic uncertainty, and the signals u and y represent a known test input and the corresponding output corrupted by measurement noise η. The *a priori* information consists on bounds on the "size" of the uncertainty, $\|\Delta\|_\infty \leq \gamma$ and the magnitude of the measurement noise $\|\eta\|_\infty \leq \epsilon_N$. In this context it can be shown [14] that, given measurements of the input $\mathbf{u}$ and output $\mathbf{y}$, the model is not invalidated by this experimental information if and only if there exist a vector $\zeta = \left(\zeta_1 \ldots, \zeta_n\right)$ such that the following set of LMIs are feasible:

$$M(\zeta) \doteq \left[\begin{array}{c} (T_R T_u)^* T_R T_u + (T_R T_u)^* T_S T_\zeta + T_\zeta^* T_S^* T_R T_u \\ \left(\frac{1}{\gamma^2} I - T_S^* T_S\right)^{\frac{1}{2}} T_\zeta \end{array} \right. \quad \left. \begin{array}{c} T_\zeta^* \left(\frac{1}{\gamma^2} I - T_S^* T_S\right)^{\frac{1}{2}} \\ I \end{array} \right] > 0 \tag{7.5}$$

$$\|\mathbf{y} - T_P \mathbf{u} + T_Q \zeta\|_{\ell^\infty} \leq \epsilon_N$$

where, for a given sequence $h \in \ell^2$, T_h represents its associated lower Toeplitz matrix:

$$T_h = \left[\begin{array}{cccc} h_0 & 0 & \ldots & 0 \\ h_1 & h_0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-1} & h_{n-2} & \ldots & h_0 \end{array}\right] \tag{7.6}$$

and, for a given linear operator $\mathcal{L}\colon \ell^2 \to \ell^2$, $T_\mathcal{L}$ denotes the lower Toeplitz matrix associated with its convolution kernel. Note in passing that since Equation (7.5) is linear in $\frac{1}{\gamma^2}$, the problem of finding the tightest uncertainty bound such that the model is not invalidated by the experimental data leads to a convex LMI optimization problem.

Overview of Robust Controller Design *via* μ-synthesis

Once a nominal model of the plant has been identified and validated using the techniques outlined above, standard control techniques can be used to synthesize suitable controllers that achieve good performance for the *nominal* system. However, as noted in [32], the resulting closed-loop system can be *fragile* to uncertainty and changes in the parameters of the camera, such as the focal length f. To avoid this difficulty, in this chapter we will use robust control tools (μ-synthesis) to design a controller, based on both the nominal model *and* the uncertainty description provided by the identification step, that achieves robust performance.

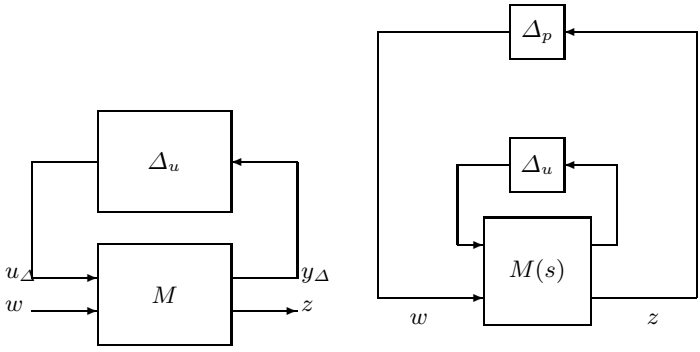


Fig. 7.2. (a) LFT uncertainty description. (b) Equivalence between robust performance and stability.

Specifically, consider the generalized system interconnection shown in Figure 7.2, consisting of a stable transfer function matrix M (in our case the combination of the pan–tilt unit, the vision sensor and the controller) and a "feedback" term Δ_u, representing model uncertainty with a block-diagonal structure of the form:

$$\Delta_u(s) \in \mathbf{\Delta} \equiv \Big\{ \Delta(s) = \text{block} - \text{diag}\big\{\Delta_1(s), \Delta_2(s), \ldots, \Delta_n(s)\big\},\ \Delta_i(s) \text{ stable} \Big\} \tag{7.7}$$

It can be shown that this interconnection is stable for all $\|\Delta_u(s)\|_\infty \leq 1$ if and only if $\mu_{\Delta_u}(M) < 1$, where the structured singular value μ is defined as[5] [19]:

$$\mu_{\Delta_u}(M) = \begin{cases} \dfrac{1}{\min\limits_{\Delta_u \in \mathbf{\Delta}} \{\|\Delta_u\|_\infty \,:\, det(I + M(j\omega)\Delta_u(j\omega)) = 0 \text{ for some } \omega\}} \\ 0 \text{ if no } \Delta_u \in \mathbf{\Delta} \text{ destabilizes } M \end{cases} \tag{7.8}$$

Robust performance (*i.e.* guaranteed performance for all possible model uncertainties in the set) can be addressed by recasting the problem into an augmented robust stability problem by introducing an additional fictitious perturbation block Δ_p, as shown in Figure 7.2b,

[5] Roughly speaking, μ is the inverse of the "size" of the smallest destabilizing perturbation.

where w and z represent exogenous inputs and outputs subject to performance specifications, respectively. It can be shown (see the main loop theorem in [50]) that *robust performance* is achieved if and only if:

$$\mu_{RP} = \sup_{\omega} \mu_{\Delta}(M) < 1$$

where $\Delta = \text{ diag}\{\Delta_p, \Delta_u\}$ contains now both the uncertainty and the performance blocks.

μ provides a useful tool for robustness analysis that combines unstructured and structured uncertainty, robust stability and robust performance in a unified, nonconservative, framework. It can even be extended to cover parametric uncertainty (real μ). Unfortunately, at the present time there are no efficient algorithms for computing the exact value of μ for general perturbation structures. Instead, the following upper bound is used [19, 58]:

$$\mu_{\Delta}(M) \leq \inf_{D \in \mathcal{D}} \|DMD^{-1}\|_{\infty} \tag{7.9}$$

where D represents a set of positive-definite Hermitian matrices with a diagonal block structure that commutes with that of Δ. It can be shown that problem (7.9) can be recast as a convex optimization problem, leading to efficient computational algorithms. Moreover, this upper bound coincides with the exact value of μ for perturbation structures having up to 3 blocks. For more than 3 blocks, the bound is no longer tight and recent results show that the gap can increase linearly in the number of blocks [44]. However, in most cases arising in practice the gap is small (less than $15\ \%$ [50]).

From the discussion above it follows that controllers guaranteeing robust stability or robust performance can be synthesized by solving the following optimization problem:

$$\min_{K \text{ stabilizing}} \mu_{\Delta}\Big\{M(K)\Big\}$$

where the notation $M(K)$ is used to indicate explicitly that the closed-loop transfer matrix M is a function of the controller K. Due to the difficulties in computing the exact value of μ, the upper bound (7.9) is used instead, yielding the following optimization problem (in K and D):

$$J = \min_{K \text{ stabilizing}} \left\{ \inf_{D \in \mathcal{D}} \|DM(K)D^{-1}\|_{\infty} \right\} \tag{7.10}$$

Robust stability or robust performance is achieved if $J < 1$.

While the optimization problem (7.10) is convex either in the scales D or in the controller K, *it is not jointly* convex in D and K. Thus, there potentially exist local minima where an optimization algorithm may get trapped. The solution method currently used alternates between finding the tightest possible upper bound by optimizing the scales D while holding the controller constant (an infinite-dimensional convex optimization problem); and finding an internally stabilizing controller that minimizes this upper bound (a standard $\mathcal{H}_{\infty}$ control problem). This algorithm, known as the "D–K" iteration, is implemented both in the Robust control toolbox [59] and μ-analysis and synthesis toolbox [2], and can be summarized as follows:

1. $\mathcal{H}_{\infty}$ Synthesis. Holding D fixed, use $\mathcal{H}_{\infty}$ synthesis to solve:

$$\min_{K \text{ stabilizing}} \|DM(K)D^{-1}\|_{\infty} \tag{7.11}$$

In the first iteration D is often set to I(identity matrix). After the first iteration the D scale obtained in step 3 is used.

2. μ Analysis. Calculate the upper bound of μ for the closed-loop system obtained using the controller K from step 1. This entails solving the following infinite-dimensional optimization problem:
$$\inf_{D\in\mathcal{D}} \|DM(K)D^{-1}\|_\infty \tag{7.12}$$
This problem is approximately solved by finding the value of D over a finite grid of frequency points ω_i.
3. D Fitting. The approximate solution to the optimal scaling problem of step 2 is found by fitting the values $D(j\omega_i)$ with a real-rational, proper, stable, minimum-phase transfer matrix $D(s)$. Note that the order of the controller is that of the augmented plant (plant + weights) + 2$\times$ order of D. Thus, in order to obtain controllers with reasonable complexity, the order of D should be kept low (usually first or second order).
4. Go to step 1 until the stop criterion is met, which means that the condition $J < 1$ is satisfied.

As mentioned before, while due to the lack of joint convexity in K and D, this algorithm is not theoretically guaranteed to converge to the global minimum, it works well in practice, and has allowed for solving many difficult engineering problems (see, for instance, [64]).

7.2 Description of Application Example

The control-related issues involved in active-vision can be illustrated by considering the problem of smooth tracking of a noncooperative target, illustrated in the block diagram shown in Figure 7.3. Here, the goal is to internally stabilize the plant and to track the target motion, y_{target}, using as measurements images possibly corrupted by noise.

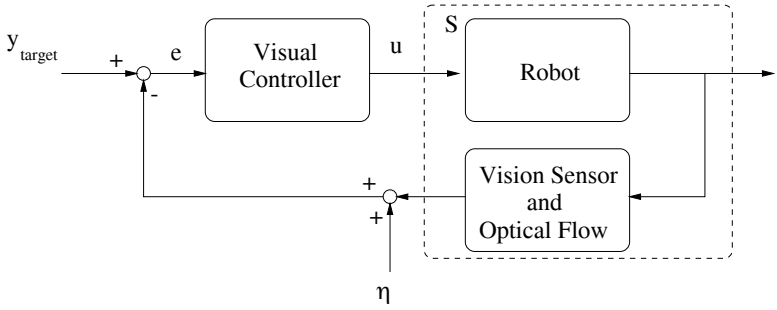


Fig. 7.3. Block diagram of a visual tracking system

It can be shown (see [52] for details) that a simplified model of the plant is given by the following state-space model:

$$\begin{aligned} \underline{x}(k+1) &= A_f\underline{x}(k) + B_f\underline{u}(k) + E_f\underline{v}(k) \\ \underline{y}(k) &= \underline{x}(k) + \underline{\eta}(k) \end{aligned} \tag{7.13}$$

where the state $\underline{x} \doteq (x)^T \in \mathbb{R}^2$ represents the position of the feature in the image plane, $\underline{v} \doteq (v_x, v_y)^T \in \mathbb{R}^2$ models the unknown (but bounded) target velocity, $\underline{u} \doteq (\theta_x, \theta_y)^T \in \mathbb{R}^2$

represents the control input (pan and tilt motion of the camera), $\underline{y}$ represents the measured position of the feature corrupted by the noise $\underline{\eta}$, and where the matrices are given by:

$$A_f = I_2,\ E_f = \frac{\tau f}{Z_s(k)} I_2$$

$$B_f = \tau \begin{bmatrix} \frac{x(k)y(k)}{f} & -(f + \frac{x^2(k)}{f}) \\ (f + \frac{y^2(k)}{f}) & -\frac{x(k)y(k)}{f} \end{bmatrix} \tag{7.14}$$

where f is the focal length of the camera, Z_s its distance to the object, and τ the sampling interval. In this context, the objective is to design a controller that stabilizes the model (7.13) and keeps $\|e\|$, the norm of the error signal, small.

If the change in the coordinates of the target feature is small, Equation (7.13) can be linearized around the present position, yielding an (approximately) equivalent LTI problem, that can be solved using a number of techniques either classical (PID [18, 52], pole-placement [18, 52]) or modern (linear quadratic Gaussian (LQG) optimal [31, 51, 52]). However, while the resulting closed-loop systems appear to have good performance, experimental results show that actual performance can be far worse than expected based on simulations using the model (7.13). As discussed in [71], this is largely due to the fact that this approach does not take into account either the dynamical effects of the image-processing algorithms used to locate the object in each frame, or potentially destabilizing modeling errors that include variations in the optical parameters and unmodeled robot dynamics. The goal of this chapter is to illustrate how these effects can be taken into account by exploiting a combination of recently developed robust identification and model (in)validation techniques.

7.2.1 Hardware Description

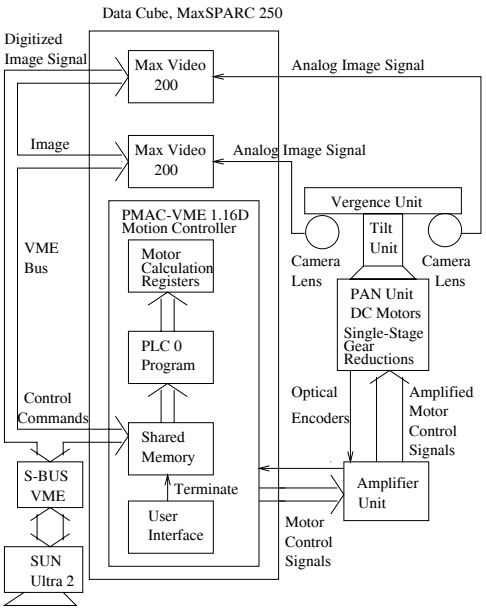
The hardware setup used in this chapter, shown in Figure 7.4a, consists of a UniSight pan–tilt platform equipped with a BiSight stereo head containing two Hitachi KP-M1 cameras and Fujinon H10X11EMPX-31 motorized lenses. The input commands to the head and the lenses are given using a 10-channel PMAC $\delta - \tau$ controller, and the image processing required to capture the images and locate the target are performed using a Datacube MaxSPARC S250 hosted by a dual processor Sun Ultra 2 workstation. A block diagram of the complete system showing the interconnection of the various components is shown in Figure 7.4b.

This robotic head has four mechanical degrees of freedom:

- Pan is the rotation of the intercamera baseline about a vertical axis.
- Tilt is the rotation of the intercamera baseline about a horizontal axis.
- Right vergence is the rotation of the right camera about the vertical axis.
- Left vergence is the rotation of the left camera about the vertical axis.

There is a mechanical coupling between pan and tilt axes: if the pan motor is turned while the tilt motor is held stationary, the tilt-axis will also rotate by $1/4$ the amount of the pan-axis rotation. However, moving the tilt motor by itself has no effect on the pan axis. To correct this coupling, the tilt motor must be turned by a ratio of the pan-motor rotation in the opposite direction of the pan-motor motion. The system also has six optical degrees of freedom (zoom, focus and aperture of left and right cameras). The mechanical properties of the "head" can be found in [36].

(a)



(b)

Fig. 7.4. (a) The experimental setup. (b) Corresponding block diagram.

7.2.2 Tracking Algorithms

Next, we briefly discuss the choice of the image-processing algorithms used in this chapter. The so-called *"motion correspondence"* problem – *i.e.* to determine the image position of the object being tracked in the frames of the sequence has been extensively studied in the computer-vision literature, and a large number of techniques have been proposed, both for known and unknown objects (see, for instance, [4, 5, 8, 17, 29, 34, 37, 47, 56, 63, 76] and ref-

erences therein). Correspondences between individual frames are usually integrated over time to exploit the dynamical properties of the target, using for instance condensation trackers [47]. These trackers generalize Kalman-filter-based ones by allowing more general (multimodal) observation-noise models, although in some cases can result in impractical computational requirements [38].

Selection of the image-processing algorithm entails a compromise between complexity and robustness, since time delays stemming from more sophisticated image-processing algorithms have negative impact on the stability and overall performance of the closed-loop system. Thus, as a compromise between complexity and robustness, we selected a *normalized crosscorrelation with template update* [37] algorithm to track the target through a sequence of frames. As shown in the following, this algorithm briefly described below achieves good performance tracking targets at video rate, even in the presence of clutter.

Let T be a template image representing a memorized pattern of the object of interest and $I_{i,j}$ be the input image where the object must be found. The *normalized crosscorrelation* between the template image T and a region R of the input image $I_{i,j}$ is given by:

$$N(i,j) = \frac{\sum_{x,y \in \mathbb{R}} T(x,y) \cdot I_{i,j}(x,y)}{||T(x,y)|| \cdot ||I_{i,j}(x,y)||} \tag{7.15}$$

where $I_{i,j}(x,y) = I(x-i, y-j)$, and the denominator is a normalization factor to reduce the effect of illumination variations. Regions R containing a pattern similar to the memorized template result in high values of $N(i,j)$ ($N(i,j) = 1$ corresponds to a perfect match). Thus, searching for the object of interest can be reduced to the problem of finding a region R in the input image $I_{i,j}$ that has a high normalized crosscorrelation value with the stored template T. Search time can be reduced by looking in a small window around the expected location of the target.

The approach outlined above allows for tracking targets without prior models, it is invariant to translation of the target and it is robust to illumination changes. Additional robustness to orientation and scale variations, to a certain degree, is obtained by using *template updating*. That is, the template is replaced by the best matched region from the current frame. However, in order to prevent *correlator walkoff* due to occlusion and image clutter, the template is updated only when the crosscorrelation factor of the current frame drops from the previous value by more than a threshold. In our experiments we have used 128×128 images, 18×18 template size, 9×9 search window with increments of 2 pixels, and template update threshold of 0.04.

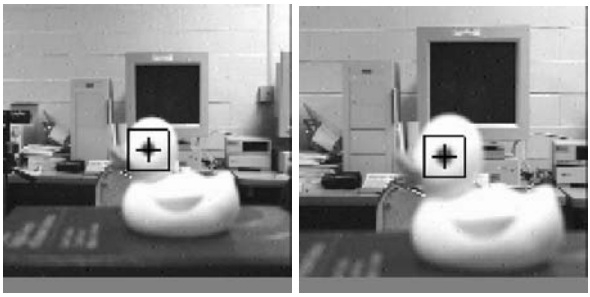
7.3 Experimental Results

Synthesizing a robust active-vision controller requires first obtaining a model of the plant, as well as an estimate of the associated uncertainty. In the following we describe the results of applying the robust identification and model (in)validation algorithms outlined in Section 7.1.3 to find a *combined model of the system that includes both the dynamics of the BiSight pan and tilt unit and the computer vision module (image-processing algorithms)*[6]. For the sake of briefness an abbreviated description is given of the tilt-axis results, since the procedure is similar to the one used for the pan axis.

[6] Identifying a single model combining the dynamics of the pan–tilt unit and the computer-vision module captures better the interaction among them, while avoiding artificially inflating the order of the resulting model.

7.3.1 Identification Results

In order to identify the transfer function from the command input u to the pan and tilt units to the displacements y_p and y_t of the target in the image, *measured in pixels*, the system was sequentially excited in each axis with a step input of amplitude 67 *encoder units*[7] (roughly corresponding to an angular displacement of 1.5°) and the position of a target (originally located at the center of the image) was measured in pixels[8]. This process was repeated for eight different zoom settings ranging from 0 to 70 % in increments of 10 %, with the goal of obtaining an accurate description of the system for this range of zoom values. The corresponding images of the target for some representative zoom values are shown in Figure 7.5.



(a) 10 % Zoom (b) 30 % Zoom

(c) 50 % Zoom (d) 70 % Zoom

Fig. 7.5. Sample target images for different zoom values

Finally, the *a priori* information was determined by repeatedly measuring the location of the target in the absence of input, to obtain an experimental noise level of approximately

[7] Encoder units are the units used in the servo level PMAC controller commands to drive the servo motors of the pan and tilt unit of BiSight Head-Eye platform.

[8] Note that the input to the system is in the encoder units while the output of the system is in pixels (output of the normalized crosscorrelation algorithm).

$\epsilon_t = 2$ pixels[9], and by measuring the time-constants of the pan and tilt, leading to an estimated value $\rho \equiv 0.5$. Using this data, the robust identification algorithm outlined in the previous section was run in each case of $0\ \%$ through $70\ \%$ zoom levels with $10\ \%$ increments[10]. Higher level zoom values (above $70\ \%$) are not considered since very high level zoom values caused too much blurring in our laboratory environment due to distance, the resulting images were useless. For each case, $N_t = 26$ data samples were used that were followed by a model-reduction step.

Figure 7.6 compares the step responses of the resulting identified models against the experimental data points (normalized by the input) for different zoom values. Here "o" denotes an experimental data point used in the identification, while "$*$" denotes additional experimental data, plotted for validation purposes[11]. As shown in the plots, these responses interpolate the experimental data points within the normalized error bounds $\left(\frac{2 \text{ pixels}}{67 \text{ encoder units}}\right)$. Similar results, omitted for space reasons, were obtained for the remaining values of the zoom. It is also worth noticing the existence of a time delay between the time when the input was applied and the system started to react. This delay, due to the image processing, fluctuates between 67 and 96 ms.

In the following we will use $f = 50\ \%$ as the nominal value of the zoom[12] for robust controller design.

7.3.2 Uncertainty Description and Model Validation

Figures 7.7a and 7.7b show the frequency response corresponding to models identified for different values of f. From these plots it follows that *changes in f result in significant changes in the dynamics*, particularly in the low-frequency range. In turn, these changes can lead to loss of stability or significant performance degradation[13]. In order to synthesize a robust controller capable of accommodating these changes in the plant dynamics, in this chapter we will model them as multiplicative dynamic uncertainty. Specifically, the transfer functions $G_{f,pan}(z)$ and $G_{f,tilt}(z)$ corresponding to a chosen nominal value of the f will be modeled as:

$$\begin{aligned} G_{f,pan}(z) &= G_{nom,pan}(z)\left[1 + \Delta_{pan}(z) W_{u,pan}(z)\right] \\ G_{f,tilt}(z) &= G_{nom,tilt}(z)\left[1 + \Delta_{tilt}(z) W_{u,tilt}(z)\right] \end{aligned} \tag{7.16}$$

where $G_{nom,pan}$ and $G_{nom,tilt}$, the nominal transfer functions are given in Equation (7.18), $W_{u,pan}(z), W_{u,tilt}(z)$ are fixed weighting functions containing all the information available about the frequency distribution of the uncertainty and $\Delta_{pan}(z),\ \Delta_{tilt}(z) \in \mathcal{BH}_\infty$ represent dynamic model uncertainties.

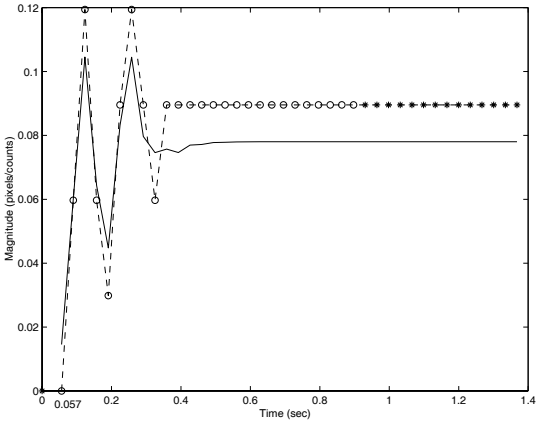
[9] This experimental error is mainly due to fluctuating conditions such as ambient light.

[10] It is important to note that our proposed method does not require calibration of cameras. Zoom changes were simply obtained through PMAC controller unit. Camera focal length is not measured.

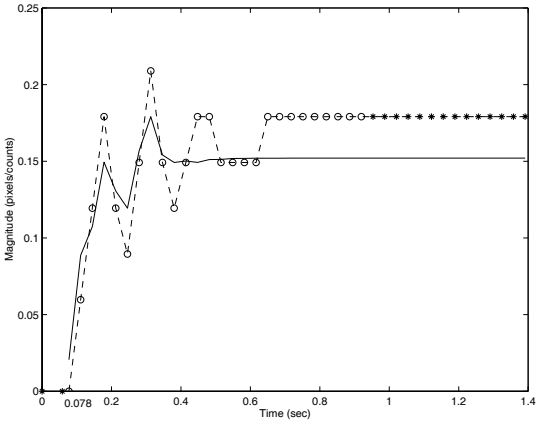
[11] Due to space constraints, only the pan axis is shown, results for the tilt axis are similar.

[12] As shown in the following, the frequency response of the corresponding transfer function is roughly in the center of the family of models.

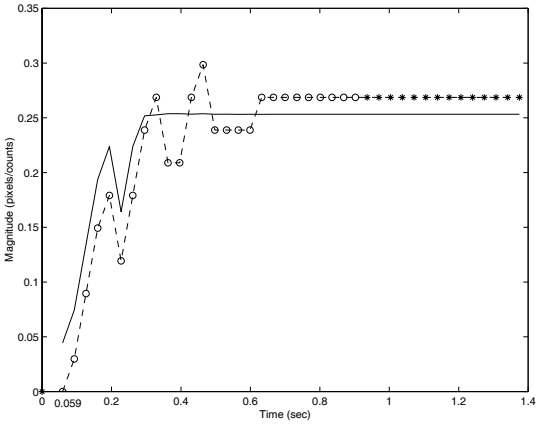
[13] As we will show in the following, a classical PID controller designed for f=50 %, goes unstable when f increases to 70 %. A PID controller designed for $f = 70\ \%$ maintains stability for the entire zoom range, but leads to very slow closed-loop systems for the lower values of f due to the small value of the resulting loop gain.



(a) 10 % Zoom



(b) 50 % Zoom



(c) 70 % Zoom

Fig. 7.6. Model of the pan axis (solid line) *versus* experimental data (dashed line) for different zoom values

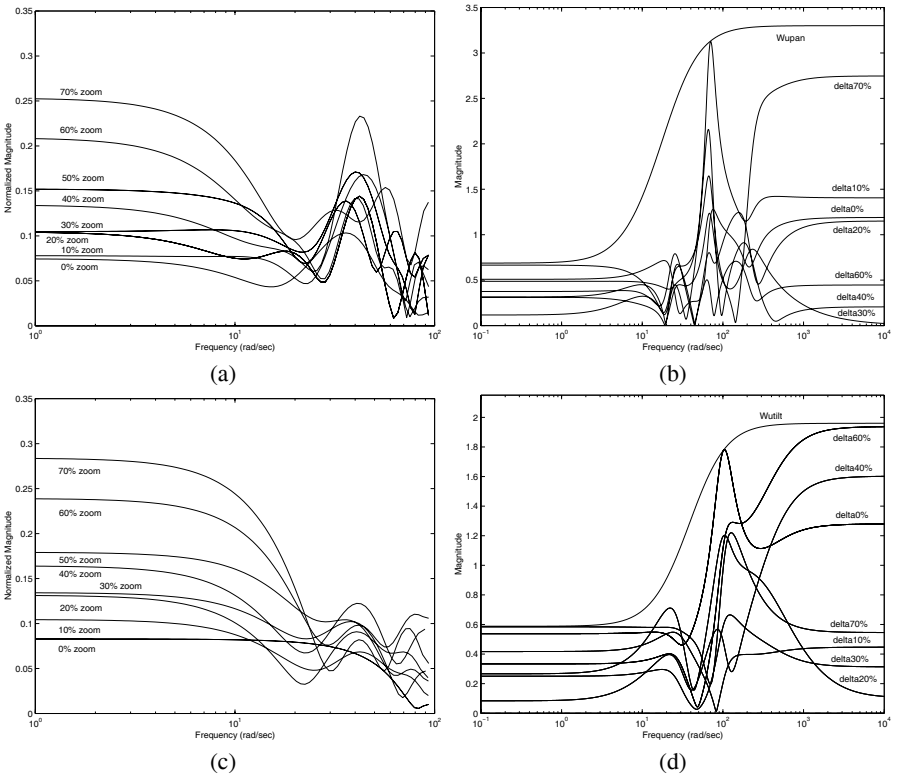


Fig. 7.7. Frequency responses for different fs, the corresponding model uncertainty and the overall uncertainty weights. (a) and (b): pan axis; (c) and (d): tilt axis.

Uncertainty Weight Function Selection

The transfer functions $W_{u,i}(z)$ should be selected so that the family (7.16) covers all possible plants [49]. Equivalently, its magnitude Bode plot should cover the plots of $\frac{G_{f,i}(z)-G_{nom,i}(z)}{G_{nom,i}(z)}$ for all possible plants, shown in Figures 7.7b and 7.7d, where i =pan, tilt.

Based on our experimental data, we selected as uncertainty weights:

$$W_u = \begin{bmatrix} W_{u,pan}(z) \\ \\ W_{u,tilt}(z) \end{bmatrix} = \begin{bmatrix} \frac{2.5479z-2.152}{z-0.4242} \\ \\ \frac{1.3324z-0.7944}{z-0.0851} \end{bmatrix} \tag{7.17}$$

As shown in Figures 7.7b and 7.7d, these weights cover the uncertainty due to the combination of changes in f and high-order unmodeled dynamics.

(a) 10 % Zoom (b) 30 % Zoom

(c) 50 % Zoom (d) 70 % Zoom

Fig. 7.8. A different target used to validate the models

Model and Uncertainty (In)Validation

In order to validate the models and uncertainty descriptions, the experiments were repeated using a different target, shown in Figure 7.8. The new data collected was used to construct the model (in)validation LMIs given in Equation (7.5). Table 7.1 shows the results of minimizing γ subject to feasibility of these LMIs. Since in all cases $\gamma < 1$ and $\epsilon_N \leq \epsilon_t$ this new data does not invalidate the model. Similar results were obtained with other targets.

7.4 Discrepancies/Misfits between Theory and Experimental Results

In this section, we briefly comment on several issues where the existing theory failed to give a complete solution to the identification problem and indicate the tools used to overcome these difficulties.

7.4.1 Reduced-order Models

As indicated in Section 7.1, all models consistent with the *a priori* information and *a posteriori* experimental data can be parametrized as an LFT of a fixed transfer matrix, generated by

Table 7.1. Model validation results

	Pan	axis	Tilt	axis
Zoom	γ	ϵ_N	γ	ϵ_N
0 %	0.4448	0.02985	0.5594	0.02985
10 %	0.4383	0.02984	0.4884	0.02985
20 %	0.2777	0.02985	0.3331	0.02985
30 %	0.2199	0.02984	0.2283	0.02985
40 %	0.2626	0.02985	0.1420	0.02985
50 %	0.0468	0.02984	0.0000	0.02984
60 %	0.2327	0.02985	0.5368	0.02982
70 %	0.5921	0.02985	0.5900	0.02971

the identification algorithm, and a free parameter: a transfer matrix $q(z) \in \mathcal{BH}_\infty$. In particular, the central interpolant (corresponding to the choice $q = 0$) leads to a model of order N_t, the number of experimental data points, with a state-space realization given by Equation (7.3). Thus, in this case the central interpolant has order 26. Since popular robust control methods lead to controllers having at least the same order as the plant, using these models would lead to complex controllers. In principle, this difficulty can be avoided by using the additional degrees of freedom available in q to search for lower-order models inside the consistency set. However, the resulting problem is difficult to solve, since the order of the model (or equivalently the rank of the associated Hankel operator) is not a convex function of q. Thus, rather than pursuing this approach, we simply model reduced the central interpolant using balanced truncations. The resulting, reduced-order transfer functions (recall that these correspond to the nondelayed portion of the model) are given by:

$$G_{nom} = [G_{nom,pan}\ G_{nom,tilt}]^T$$

$$G_{nom,pan}(z) = \frac{0.0206z^7 + 0.0765z^6 + 0.0551z^5}{1.0000z^7 + 0.412z^6 + 0.3976z^5} \frac{+0.0806z^4 + 0.0233z^3 + 0.0174z^2 + 0.045z + 0.045}{+0.1902z^4 + 0.2079z^3 + 0.0939z^2 + 0.0628z + 0.026}$$

$$G_{nom,tilt}(z) = \frac{0.0297z^5 + 0.0552z^4 + 0.0515z^3}{1.0000z^5 - 0.1554z^4 + 0.0375z^3} \frac{+0.0142z^2 - 0.0179z + 0.0358}{+0.0419z^2 + 0.0138z + 0.0023} \tag{7.18}$$

7.4.2 Existence of Variable Time Delays

Recall that the experimental data shows the existence of a variable time delay that fluctuates between two and three sampling periods[14], depending on the amount of time required to locate

[14] Sampling frequency of our system is 33 Hz.

the target in the image. This time delay cannot be captured by existing identification methods (since it cannot be modeled by a finite-dimensional transfer function). To circumvent this difficulty, we modeled this variable time delay as a combination of (i) a *fixed*, two sampling periods time delay, and (ii) multiplicative model uncertainty. This approach leads to a model of the form of Equation (7.16) where the weighting functions $W_{u,pan}(z)$ and $W_{u,tilt}(z)$ are increased to account for the additional terms of the form $|e^{-s\Delta T} - 1|$ arising from the time delays. The final weighting functions, taking into account both model and time–delay uncertainty are given below, with the corresponding Bode plots shown in Figures 7.9a and 7.9b.

$$W_{ut} = \begin{pmatrix} W_{ut,pan}(z) \\ \\ W_{ut,tilt}(z) \end{pmatrix} = \begin{pmatrix} \frac{3.9315z-3.4584}{z-0.3436} \\ \\ \frac{2.7045z-2.1925}{z-0.1944} \end{pmatrix} \tag{7.19}$$

7.4.3 Controller-implementation Considerations

Recall that in our setup the pan and tilt unit is controlled by a 10-channel δ-τ PMAC controller board hosted by a Sun UltraSPARC workstation. In order to avoid time delays associated with trajectory preplanning introduced by the PMAC controller board, the δ-τ PMAC board is driven at the servo level, *i.e.* by directly accessing its commanded position (CP) registers. A difficulty with this approach is that the relatively low sampling rate (33 Hz) can lead to jerky camera motions. This effect was prevented by distributing the set-point changes over the entire sampling period, resulting in smooth motions [61]. Distribution of these changes was accomplished by running a PMAC *plc0* background program that every 4 servo cycles (approximately 1.77 ms) increments the value of the CP registers an amount equal to the desired total set-point change divided by a suitable step size[15], leading to the block diagram shown in Figure 7.10.

It follows that the *plc0* program can be modeled as: $u(n) = u(n-1) + \hat{u}(n)$. Taking the z-transform of both sides yields the following transfer function:

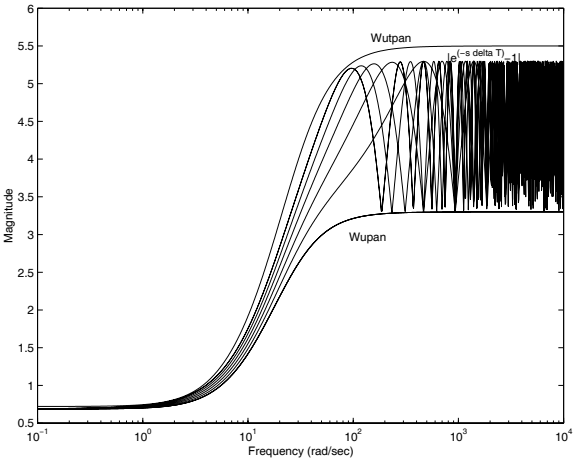
$$H_{plc}(z) = \frac{z}{z-1} \tag{7.20}$$

Hence, the net effect of the program is to add an *integrator* at the input to the system, thus smoothing out high-frequency vibrations and guaranteeing zero steady-state tracking error to step inputs. Combining this effect with the time delay and the transfer functions identified in Section 7.3 leads to the following overall model of the plant:

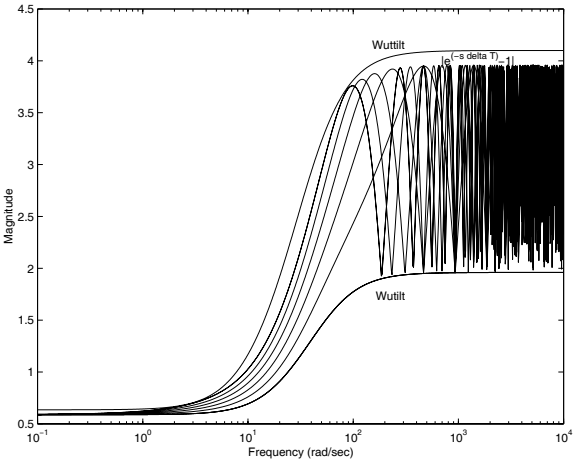
$$\begin{aligned} \hat{G}_{nom,i} &= \tfrac{1}{z(z-1)} G_{nom,i} \\ G_{nom,pan}(z) &= \tfrac{0.0206z^7+0.0765z^6+0.0551z^5+0.0806z^4+0.0233z^3+0.0174z^2+0.045z+0.045}{1.0000z^7+0.412z^6+0.3976z^5+0.1902z^4+0.2079z^3+0.0939z^2+0.0628z+0.026} \\ G_{nom,tilt}(z) &= \tfrac{0.0297z^5+0.0552z^4+0.0515z^3+0.0142z^2-0.0179z+0.0358}{1.0000z^5-0.1554z^4+0.0375z^3+0.0419z^2+0.0138z+0.0023} \end{aligned} \tag{7.21}$$

where $\hat{G}_{nom,i}$ denotes the identified, nondelayed transfer function corresponding to the axis under consideration.

[15] Our specific implementation uses 10 steps.



(a)



(b)

Fig. 7.9. Model uncertainty including the time delay and overall uncertainty weights. (a) pan axis. (b) tilt axis.

7.4.4 Limitations of Classical Controllers

Once a model of the plant has been identified, standard control techniques can be used to synthesize suitable controllers. Note, however, that when using these techniques, the time delay T_d must be taken into account in order to guarantee stability [71].

As shown in Figure 7.11b, an optimal LQG controller designed taking this time delay into account achieves good performance for the *nominal* system. However, these results also illustrate its fragility: a change in the focal length f destabilizes the closed-loop system [32]. A similar situation arises when using a PID controller, experimentally tuned to achieve a

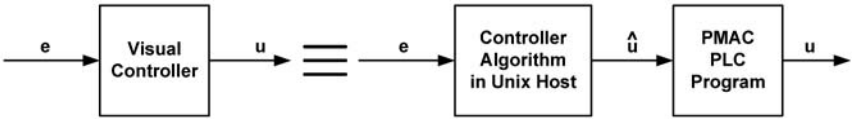
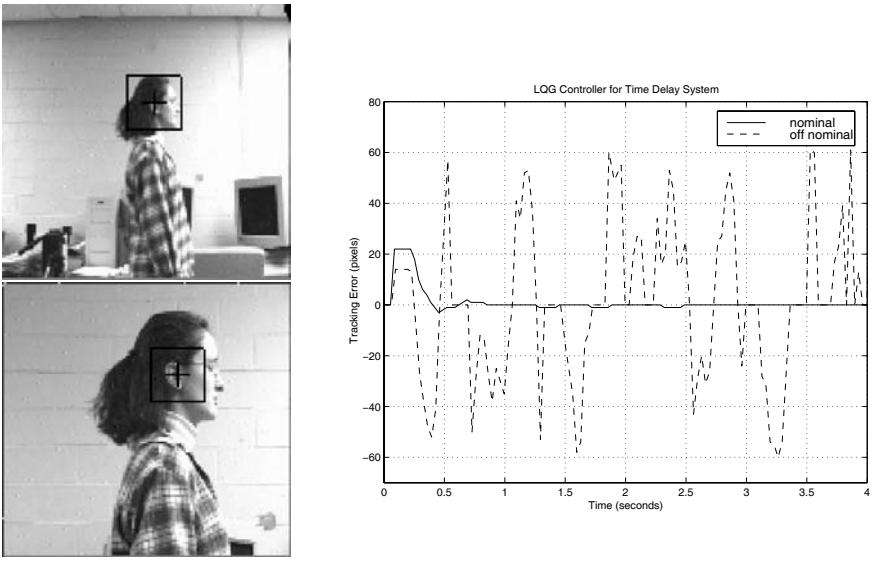


Fig. 7.10. Controller implementation to avoid excessive jerk



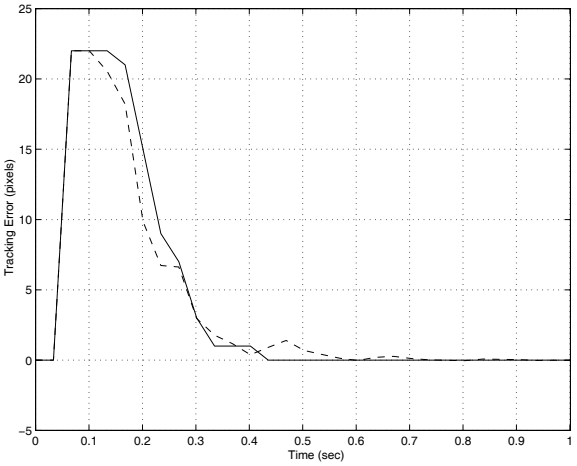
(a) (b)

Fig. 7.11. Smooth pursuit: (a) Target for different values of f. (b) Tracking error of an LQG controller (experimental).

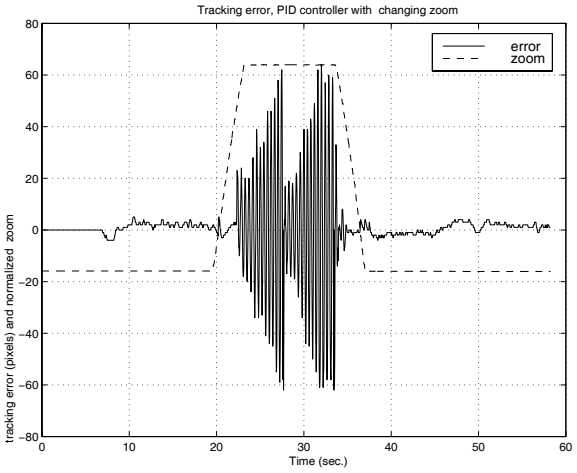
compromise between settling time and overshoot. As shown in Figure 7.12a this controller also achieves good performance for the nominal system. However, this is achieved by tuning the controller to the nominal model G_{nom}, and, as in the LQG case, this leads to potentially *fragile* closed-loop systems. This effect is illustrated in Figure 7.12b, showing the result of an experiment where the focal length f of the system is increased while tracking, to zoom-in a feature of interest, and then reduced to its previous value. As shown there, while the system exhibits good performance for the nominal value of f, it becomes unstable as f is changed.

The loss of stability can be easily explained from the experimental data shown in Figure 7.16, showing that (in the low-frequency range) f enters the model virtually as a static gain multiplying the control action. Thus, roughly speaking, the system becomes unstable when this "gain" exceeds the gain margin.

In principle, stability can be guaranteed by designing a PID controller for the maximum, rather than medium value of f, hence guaranteeing that the gain margin is never exceeded. However, this leads to very slow closed-loop systems for the medium and minimum values of f due to the small value of the gain that is required. As we show next, to a large extent this



(a)



(b)

Fig. 7.12. Pan-axis tracking error using a PID controller: (a) Nominal model, simulation (solid) and experimental (dashed). (b) Nominal *versus* off nominal f, experimental.

performance loss can be avoided by using the robust control tools described in Section 7.1 to synthesize controllers that guarantee a given performance level for all possible values of the parameters.

7.4.5 Robust Controller Design and Performance Weight Selection

In this section we describe how to combine the nominal model (7.21) and the associated uncertainty description (7.19) with μ-synthesis in order to obtain a controller that achieves robust

performance. The first step toward accomplishing this is to recast the plant into a suitable form and to select weighting functions that encapsulate the control objectives.

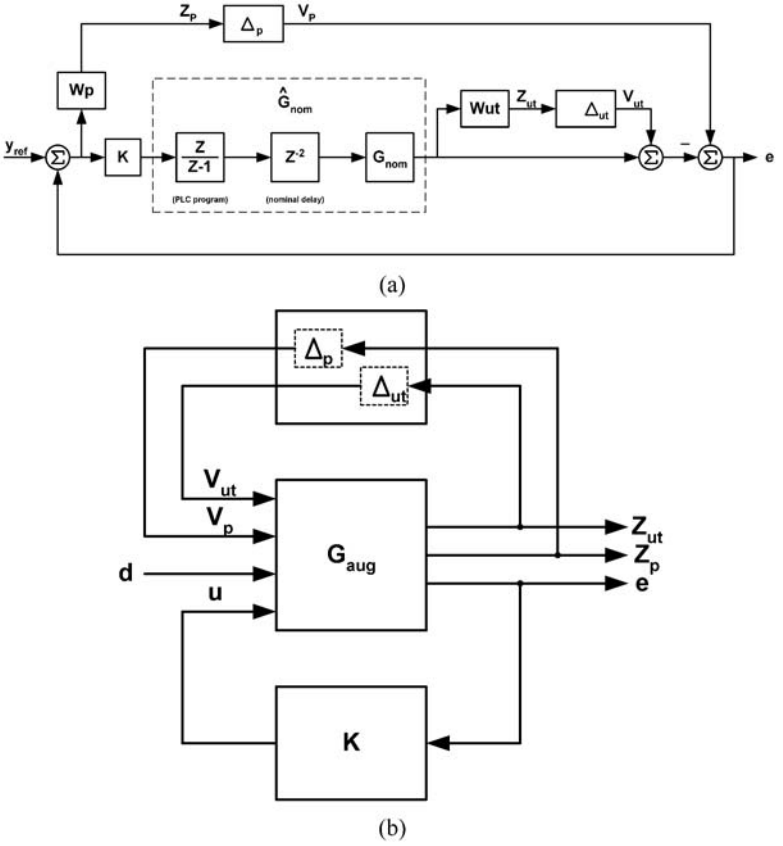


Fig. 7.13. (a) Block diagram of the augmented plant for each axis. (b) Block diagram with the uncertainty "pulled out" of the loop.

Figure 7.13a shows a block diagram of the active-vision system taking into account the uncertainty. As discussed in Section 7.1.3, the additional block Δ_p represents a fictitious perturbation block, which in conjunction with the weighting function W_p allows for imposing performance specifications on the tracking error $e(t)$. In particular, in our case W_p should be selected to achieve the following specifications (for all values of $0 \leq f \leq 70$ %):

1. Settling time to step inputs on the order of 1 s.
2. Fast rise time, with minimal overshoot, to avoid correlator walkoff problems.
3. Small control action to avoid actuator saturation.
4. Zero steady-state error tracking to step inputs[16].

[16] This specification is automatically satisfied as long as $W_p(1) \neq 0$, due to the integral action introduced by the *plc0* program.

Unfortunately, there is no explicit way to exactly map these time-domain performance specifications to frequency-domain weighting functions. Rather, there exist some well-established practical rules. Specifically, suppose that it is desired to achieve the following performance specifications:

- Steady-state error $\leq A$
- Closed-loop bandwidth $\geq W_B$
- High-frequency noise amplification $\leq P$

Then the magnitude Bode plot of the sensitivity function should resemble the plot shown in Figure 7.14a, which in turn leads to the performance weighting function shown in Figure 7.14b, [48].

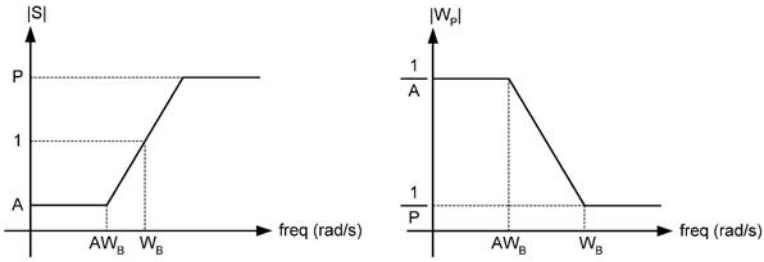


Fig. 7.14. (a) Desired sensitivity function. (b) Desired performance weight function.

$$W_p \cong \frac{1}{P}\left(\frac{\tau_{cl}s + P}{\tau_{cl}s + A}\right) \qquad \text{where} \quad \tau_{cl} = \frac{1}{W_B}$$

In addition, commonly used rules of thumb are to

- Penalize low frequencies for good steady-state and settling-time.
- Penalize high frequencies for good transient response.

Taken into account these considerations, a good compromise between the performance specifications was accomplished by using the following (experimentally tuned) function:

$$W_p(z) = \frac{0.2604z + 0.1412}{z + 0.2550} \tag{7.22}$$

As shown in the Bode plot, Figure 7.15, this function has a low-pass characteristic, penalizing the low-frequency components and leading to a closed-loop bandwidth larger than 30 Hz, in order to achieve the settling-time specifications.

Figure 7.13b shows a block diagram of the plant in the standard form used for μ-synthesis. Here, the augmented plant G_{aug} has the following realization:

$$\begin{bmatrix} Z_{ut} \\ Z_p \\ e \end{bmatrix} = \begin{bmatrix} 0 & 0 & W_{ut,i}\hat{G}_{nom,i} \\ -W_p & W_p & -W_p\hat{G}_{nom,i} \\ -1 & 1 & -\hat{G}_{nom,i} \end{bmatrix} \begin{bmatrix} V_{ut} \\ V_p \\ u \end{bmatrix} \tag{7.23}$$

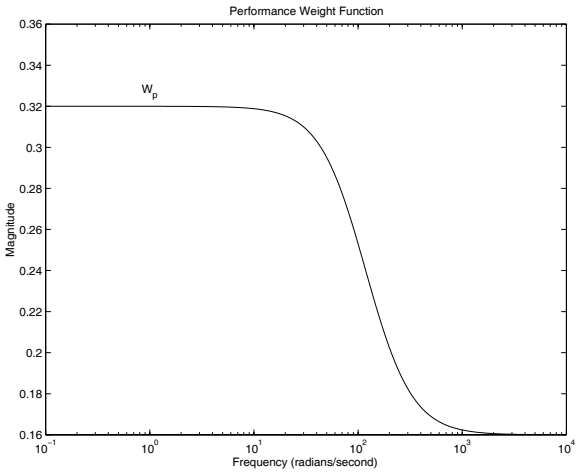


Fig. 7.15. Performance weight function

where $\hat{G}_{nom,i}$, the transfer function of the axis under consideration, and $W_{ut,i}$, the associated uncertainty description, are given in Equations (7.21) and (7.19), respectively. Using μ-synthesis with first-order scales, followed by a model reduction step leads to the following controllers:

$$K_{\mu_{pan}}(z) = \frac{1.4181z^6 - 0.7553z^5 + 0.5886z^4}{z^6 + 0.1265z^5 + 0.2633z^4} \\ \frac{-0.0262z^3 + 0.0439z^2 + 0.0519z + 0.0143}{+0.3171z^3 + 0.2723z^2 + 0.1087z + 0.3008} \tag{7.24}$$

$$K_{\mu_{tilt}}(z) = \frac{1.7939z^6 - 0.5281z^5 + 0.1520z^4}{z^6 + 0.6059z^5 + 0.4315z^4} \\ \frac{+0.0588z^3 + 0.0112z^2 + 0.0117z - 0.0052}{+0.2693z^3 + 0.1625z^2 + 0.1446z + 0.1123} \tag{7.25}$$

The values of μ corresponding to the full (solid line) and reduced order (+) controllers for the pan and tilt axes are shown in Figures 7.16a and 7.16b, respectively. Since in all cases we have that $\mu < 1$, it follows that these controllers achieve robust performance.

7.4.6 Experimental Validation

In order to validate the controllers, we performed the following sets of experiments:

- **Step Response:** The step responses (simulation and experimental) of the closed-loop system corresponding to several values of f are shown in Figures 7.17 and 7.18. As illustrated there, as opposed to both the LQG and PID controllers, these controllers achieve tracking throughout the entire range, without the need to sacrifice nominal performance or to detuning in order to guarantee stability. Similar results, omitted for space reasons, were obtained with other targets. Note in passing that there is very good agreement between the predictions of the linear simulation based on the models (7.18) and the actual experimental data.

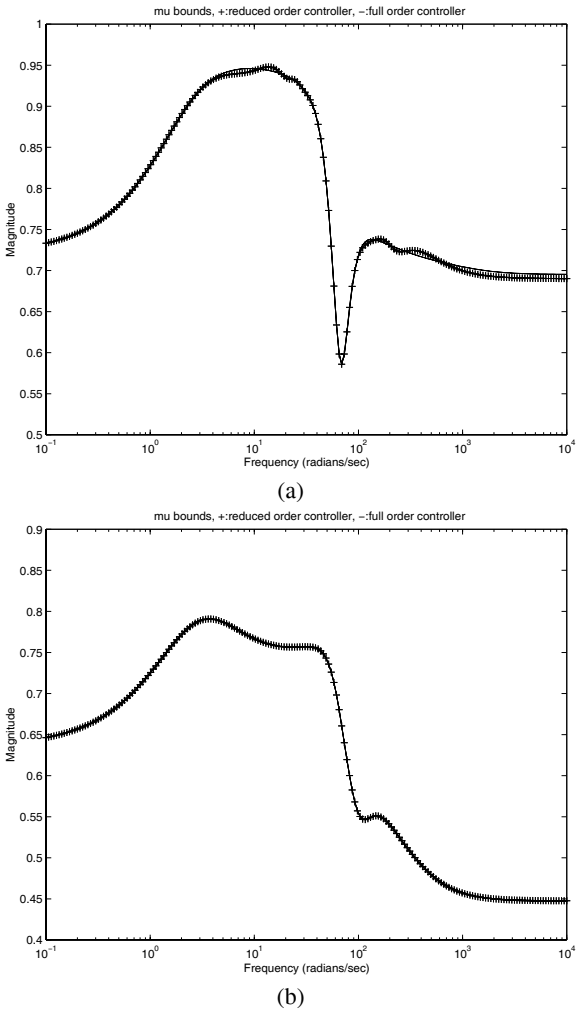
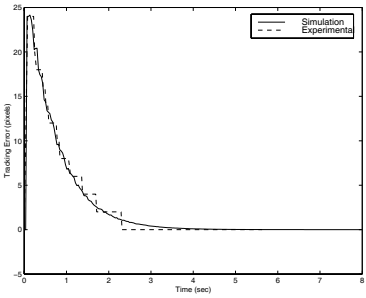
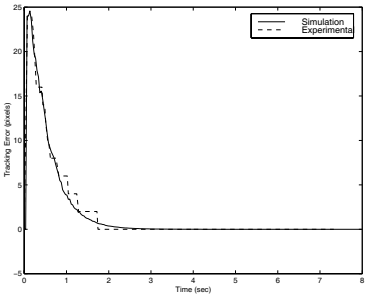


Fig. 7.16. (a) μ bounds for full (solid line) and reduced order (+) controllers. (a) Pan axis. (b) Tilt axis.

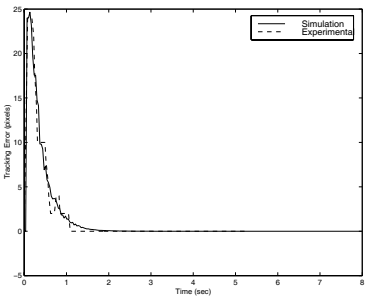
- **Smooth Pursuit:** Figures 7.20 and 7.21 show the results of experiments where the person shown in Figure 7.19 is tracked while zooming in and out of her features. As before, the robust controllers were able to achieve good tracking performance in spite of the substantial change in the dynamics of the plant due to the change in f, the uncertain time delay, and the presence of clutter.



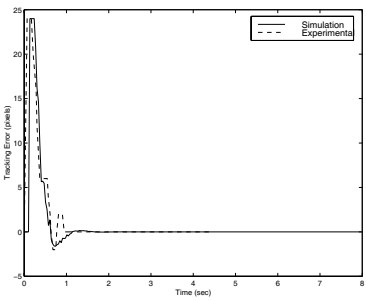
$f = 0\ \%$



$f = 20\ \%$

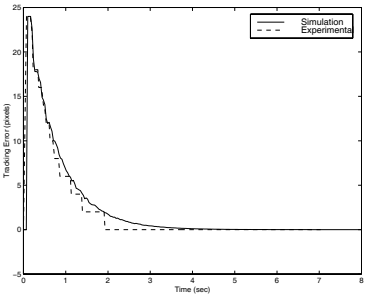


$f = 50\ \%$

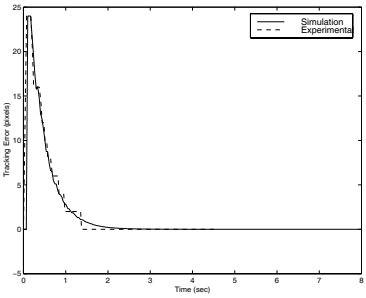


$f = 70\ \%$

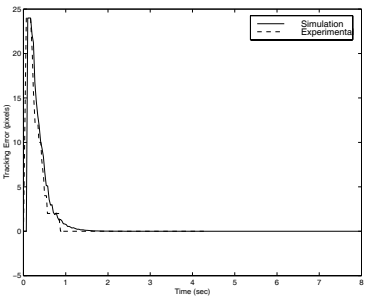
Fig. 7.17. Tracking error for different zoom values, step-response experiments: pan axis (solid: simulation, dashed: experimental).



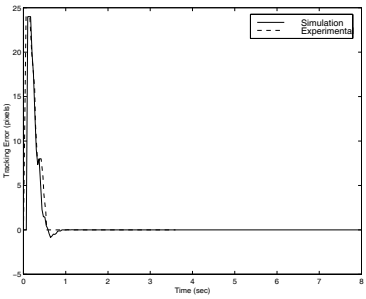
$f = 0\ \%$



$f = 20\ \%$

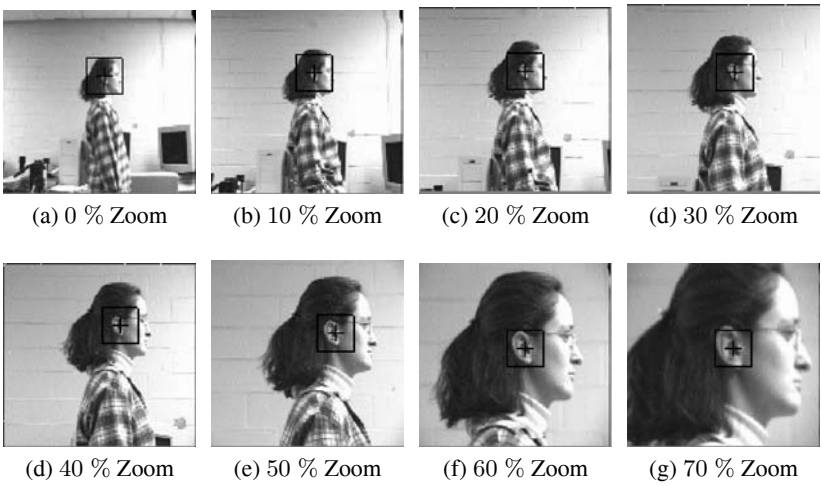


$f = 50\ \%$



$f = 70\ \%$

Fig. 7.18. Tracking error for different zoom values, step-response experiments: tilt axis (solid: simulation, dashed: experimental).



(a) 0 % Zoom (b) 10 % Zoom (c) 20 % Zoom (d) 30 % Zoom

(d) 40 % Zoom (e) 50 % Zoom (f) 60 % Zoom (g) 70 % Zoom

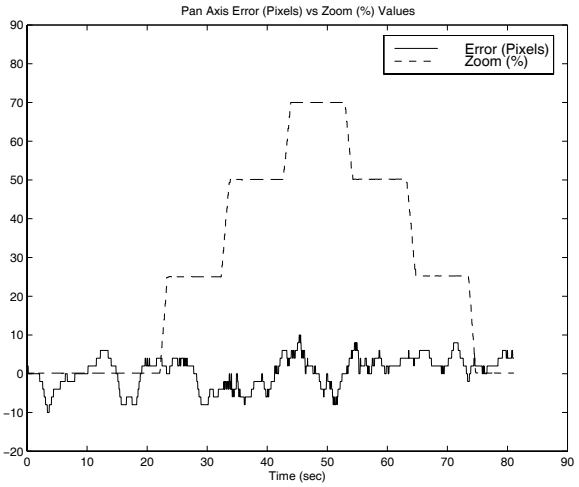
Fig. 7.19. Sample target images for zoom values 0 % through 70 %

7.5 Conclusions

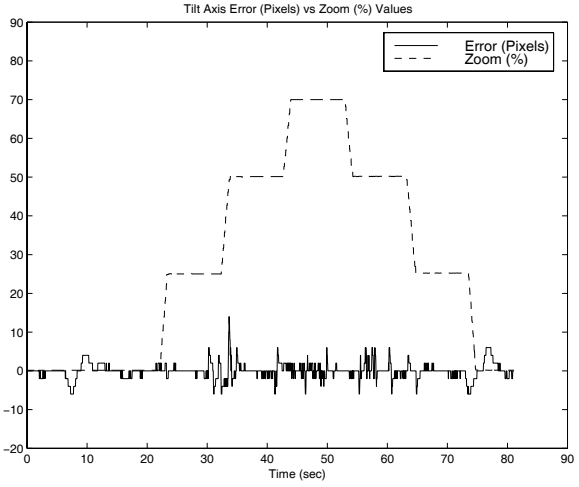
Recent hardware developments have opened up the possibility of applying active-vision techniques to a broad range of problems, such as intelligent vehicle highway systems, robotic-assisted surgery, MEMS microassembly and automated spacecraft docking. A salient feature common to all these applications is that using a feedback structure incorporating the visual information in the loop (as opposed to open-loop control) offers the possibility of achieving acceptable performance even in the presence of errors, stemming, for instance, from poorly calibrated cameras, blurring or only partially determined feature correspondences between images. Arguably, at this time one of the critical factors limiting widespread use of active-vision techniques is the fragility of the resulting systems. This lack of robustness can be traced, to a large extent, to substantial variations in the dynamics of the plant due to variations in its parameters, and to the existence of unmodeled dynamics.

In this chapter we illustrated how recently developed robust identification and model-(in)validation techniques can be brought to bear on the problem. Rather than producing a single model of the plant, these methods generate a nominal model as well as an uncertainty description guaranteed to cover both the effects of unmodeled dynamics, modeling errors and the variations due to changes in various parameters such as camera-zoom values. As shown in Section 7.4.5, this description can be directly combined with robust control techniques to synthesize controllers that guarantee acceptable performance for a large range of conditions. A salient feature of the proposed approach is that it requires very few assumptions on the system. For instance, neither the order nor the structure of the model needs to be fixed *a priori*, and it produces *a complete model that includes both the dynamics of the head and the computer-vision module without the need to separate these components (and thus potentially missing some of their coupling).* Finally, it leads to simple, computationally tractable LMI optimization problems.

The methodology was experimentally validated using an uncalibrated setup to successfully track an uncooperative target in a cluttered environment.



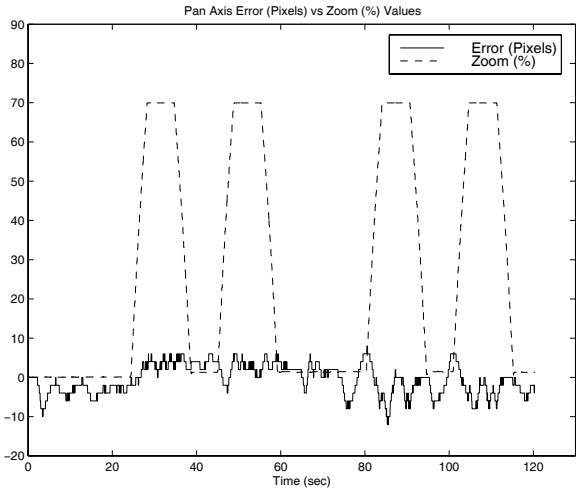
(a) pan-axis



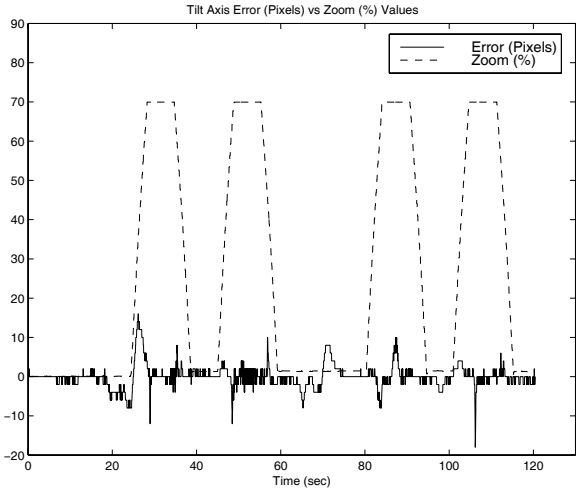
(b) tilt-axis

Fig. 7.20. Target is moving and zoom is being changed from 0 % to 70 % in 3 steps

It is worth noticing that, for each axis, a single controller is able to yield good tracking performance for a wide range of conditions. Naturally, this is achieved at the cost of mild degradation of nominal performance. In addition, the systems corresponding to lower values of f (and hence lower gain) exhibit slower responses than those corresponding to higher values. The results [73] indicate that these effects can be addressed by considering a family of parameter-varying robust controllers, indexed by the parameter f. Research in this direction is currently being explored.



(a) pan-axis



(b) tilt-axis

Fig. 7.21. Target is moving and zoom is being changed from 0 % to 70 % in 1 step

Acknowledgements

Support from NSF under grants ECS–0221562, IIS–0117387, and ITR–0312558 and AFOSR under grant FA9550–05–1–0437 is gratefully acknowledged.

References

[1] Akella, M. R. (2005). Vision-based adaptive tracking control of uncertain robot manipulators. *IEEE Transactions on Robotics* **21**(4), 747–753.

[2] Balas, G., Doyle, J. C., Glover, K., Packard, A., and Smith, R. (1991). *μ-Analysis and Synthesis Toolbox*. The MathWorks Inc.

[3] Ball, J., Gohberg, I., and Rodman, L. (1990). *Interpolation of Rational Matrix Functions, Operator Theory: Advances and Applications*, volume 45. Birkhäuser, Basel.

[4] Black, M. J. and Jepson, A. D. (1998). Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision* **26**(1), 63–84.

[5] Blake, A. and Isard, M. (1998). *Active Contours*. Springer Verlag, Berlin.

[6] Broggi, A., Cellario, M., Lombardi, P., and Porta, M. (2003). An evolutionary approach to visual sensing for vehicle navigation. *IEEE Transactions on Industrial Electronics* **50**(1), 18–29.

[7] Brown, C., Coombs, D., and Soong, J. (1993). Real time smooth pursuit tracking. A. Blake and A. Yuille, editors, *Artificial Intelligence*, pp. 123–136. MIT Press.

[8] Calabi, E., Olver, P. J., Shakiban, C., Tannenbaum, A., and Haker, S. (1998). Differential and numerically invariant signature curves applied to object recognition. *International Journal of Computer Vision* **26**(2), 107–135.

[9] Chaumette, F. and Hutchinson, S. (2006). Visual servo control part I: Basic approaches. *IEEE Robotics and Automation Magazine* **13**(4), 82–90.

[10] Chaumette, F. and Hutchinson, S. (2006). Visual servo control part II: Advanced approaches. In press.

[11] Chen, J., Dawson, D. M., Dixon, W. E., and Behal, A. (2005). Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension. *IEEE Transactions on Control Systems Technology* **13**(5), 814–825.

[12] Chen, J. and Gu, G. (2000). *Control Oriented System Identification, An $\mathcal{H}_\infty$ Approach*. John Wiley, New York.

[13] Chen, J. and Nett, C. (1995). The Carathéodory-Fejér problem and $\mathcal{H}_\infty/\ell^1$ identification. *IEEE Transactions on Automatic Control* **40**(4), 729–735.

[14] Chen, J. and Wang, S. (1996). Validation of linear fractional uncertain models: Solutions via matrix inequalities. *IEEE Transactions on Automatic Control* **41**(6), 844–849.

[15] Chesi, G. and Hashimoto, K. (2002). A self-calibrating technique for visual servoing. *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 2681–2686. Las Vegas, Nevada.

[16] Clark, J. J. and Ferrier, N. J. (1988). Modal control of an attentive vision system. *International Conference on Computer Vision*, pp. 514–523.

[17] Coombs, D. and Brown, C. (1993). Real-time binocular smooth pursuit. *International Journal of Computer Vision* **11**(2), 147–164.

[18] Corke, P. I. and Good, M. C. (1996). Dynamic effects in visual closed–loop systems. *IEEE Transactions on Robotics and Automation* **12**(5), 671–683.

[19] Doyle, J. (1982). Analysis of feedback systems with structured uncertainties. *IEE Proceedings, Part D* **129**, 242–250.

[20] Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation* **8**(3), 313–326.

[21] Fang, Y., Dixon, W. E., Dawson, D. M., and Chawda, P. (2005). Homography-based visual servo regulationof mobile robots. *IEEE Transactions on Systems, Man and Cybernetics, Part B* **35**(5), 1041–1050.

[22] Feddema, J. (1997). Microassembly of micro–electromechanical systems (MEMS) usingvisual servoing. *Block Island Workshop on Vision and Control*, pp. 257–272.

[23] Ferreira, A., Cassier, C., and Hirai, S. (2004). Automatic microassembly system assisted by vision servoing and virtual reality. *IEEE Transactions on Mechatronics* **9**(2), 321–333.

[24] Ferrier, N. J. and Clark, J. J. (1993). The Harvard binocular head. *International Journal of Pattern Recognition and Artificial Intelligence* **7**(1), 69–84.

[25] Finnefrock, M., Jiang, X., and Motai, Y. (2005). Visual-based assistance for electric vehicle driving. *IEEE Intelligent Vehicles Symposium*, pp. 656–661.

[26] Gans, N. R. and Hutchinson, S. A. (2003). An experimental study of hybrid switched system approaches to, visual servoing. *IEEE Int. Conf. on Robotics and Automation*, pp. 3061–3068. Taipei, Taiwan.

[27] Gu, L. and Su, J. (2006). Gaze control on humanoid robot head. *IEEE World Congress on Intelligent Control and Automation*, pp. 9144–9148. Dalian, China.

[28] Hager, G. (1997). A modular system for robust positioning using feedback from stereo vision. *IEEE Transactions on Robotics and Automation* **13**(4), 582–595.

[29] Hager, G. and Belhumeur, P. (1997). Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(10), 1025–1039.

[30] Hashimoto, K., Ebine, T., and Kimura, H. (1996). Visual servoing with hand–eye manipulator–optimal controlapproach. *IEEE Transactions on Robotics and Automation* **12**(5), 766–774.

[31] Hashimoto, K. and Kimura, H. (1993). LQ optimal and nonlinear approaches to visual servoing. K. Hashimoto, editor, *Visual Servoing*, volume 7, pp. 165–198. World Scientific Series in Robotics and Automated Systems, World Scientific, Singapore.

[32] Hayman, E., Reid, I., and Murray, D. W. (1996). Zooming while tracking using affine transfer. *British Machine Vision Conference*, pp. 395–404.

[33] Hill, J. and Park, W. T. (1979). Real time control of a robot with a mobile camera. $9th$ *ISIR*, pp. 233–246. Washington D.C.

[34] Hutchinson, S., Hager, G. D., and Corke, P. I. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation* **12**(5), 651–670.

[35] Hynes, P., Dodds, G. I., and Wilkinson, A. J. (2005). Uncalibrated visual-servoing of a dual-arm robot for surgical tasks. *IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, pp. 151–156. Espoo, Finland.

[36] Inanc, T. (2002). *A Novel Approach to Active Vision Systems: Modelling, Control and Real Time Tracking*. Ph.D. thesis, The Pennsylvania State University.

[37] Jain, R., Kasturi, R., and Schunck, B. G. (1995). *Machine Vision*. McGraw Hill.

[38] Jensfelt, P., Wijk, O., Austin, D. J., and Andersson, M. (2000). Experiments on augmenting condensation for mobile robot localization. *IEEE International Conference on Robotics and Automation*, pp. 2518–2524.

[39] Krupa, A., *et al.* (2003). Autonomous 3–d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing. *IEEE Transactions on Robotics and Automation* **19**(5), 842–853.

[40] Ljung, L. (1987). *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, N.J.

[41] Malis, E. (2004). Visual servoing invariant to changes in camera-intrinsic parameters. *IEEE Transactions on Robotics and Automation* **20**(1), 72–81.

[42] Martens, C., Ruchel, N., Lang, O., Ivlev, O., and Graser, A. (2001). A friend for assisting handicapped people. *IEEE Robotics and Automation Magazine* **8**(1), 57–65.

[43] Maybank, S. and Faugeras, O. (1992). A theory of self-calibration of a moving camera. *International Journal of Computer Vision* **8**(2), 123–151.

[44] Megretski, A. (1993). On the gap between structured singular values and their upper bounds. *Proc. 32nd IEEE CDC*, pp. 3461–3462. San Antonio, TX.

[45] Morari, M. and Zafiriou, E. (1989). *Robust Process Control*. Prentice Hall, New Jersey.

[46] Nageotte, F., Zanne, P., Doignon, C., and de Mathelin, M. (2006). Visual servoing-based endoscopic path following for robot-assisted laparoscopic surgery. *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 2364–2369. Beijing, China.

[47] North, B., Blake, A., Isard, M., and Rittscher, J. (2000). Learning and classification of complex dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(9), 1016–1034.

[48] P. Lundstrom, S. S. and Wang., Z. (1991). Performance weight selection for h_∞ and μ–control methods. *IEEE Transactions of the Institute of Measurements and Control* **13**(5), 241–252.

[49] P. Lundstrom, S. S. and Wang., Z. (1991). Uncertainty weight selection for $\mathcal{H}_\infty$ and μ–control methods. *IEEE Int. Conf. on Decision and Control*, pp. 1537–1542.

[50] Packard, A. and Doyle, J. (1993). The complex structured singular value. *Automatica* **29**(1), 71–110.

[51] Papanikolopoulos, N. P. (1995). Integrating computer vision and control for vision assisted robotics tasks. *1995 American Control Conference*, pp. 904–908. Seattle, WA.

[52] Papanikolopoulos, N. P., Khosla, P. K., and Kanade, T. (1993). Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Transactions on Robotics and Automation* **9**(1), 14–35.

[53] Parrilo, P. A., Sznaier, M., and Sánchez Peña, R. S. (1998). Mixed time/frequency domain based robust identification. *Automatica* **34**(11), 1375–1389.

[54] Pollefeys, M., Koch, R., and Gool, L. V. (1999). Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *International Journal of Computer Vision* **32**(1), 7–25.

[55] Ralis, S., Vikramaditya, B., and Nelson, B. (2000). Micropositioning of a weakly calibrated microassembly system using coarse-to-fine visual servoing strategies. *IEEE Transactions on Electronics Packaging Manufacturing* **23**(2), 123–131.

[56] Reid, I. D. and Murray, W. (1996). Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision* **18**(1), 41–60.

[57] Rotstein, H. and Rivlin, E. (1996). Optimal servoing for active foveated vision. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 177–182.

[58] Safonov, M. (1981). Stability margins of diagonally perturbed multivariable feedback systems. *IEEE Conf. on Decision and Control*, pp. 251–256.

[59] Safonov, M. and Chiang, R. (1988). *Robust Control Toolbox*. The MathWorks Inc.

[60] Sánchez Peña, R. S. and Sznaier, M. (1998). *Robust Systems Theory and Applications*. John Wiley, New Jersey.

[61] Seelen, U. M. C. V. (1997). *Performance Evaluation of An Active Vision System*. University of Pennsylvania.

[62] Shakernia, O., Ma, Y., Koo, T. J., and andS. S. Sastry, J. H. (1999). Vision guided landing of an unmanned air vehicle. *Proc. 1999 IEEE CDC*, pp. 4143–4148.

[63] Sharkey, P. M., Murray, D. W., Vandevelde, S., Reid, I. D., and McLauchlan, P. F. (1993). A modular head/eye platform for real-time reactive vision. *Mechatronics* **3**(4), 517–535.

[64] Skogestad, S., Morari, M., and Doyle, J. (1988). Robust control of ill-conditioned plants: High-purity distillation. *IEEE Transactions on Automatic Control* **33**(12), 1092–1105.

[65] Smith, C. E., Brandt, S. A., and Papanikolopoulos, N. P. (1997). Eye in hand robotic tasks in uncalibrated environments. *IEEE Transactions on Robotics and Automation* **13**(6), 903–914.

[66] Smith, C. E., Richards, C. A., Brandt, S. A., and P.Papanikolopoulos, N. (1996). Visual tracking for intelligent vehicle–highway systems. *IEEE Transactions on Vehicular Techology* **45**(4), 744–759.

[67] Song, W., Kim, J., and Bien, Z. (2000). Visual servoing for human-robot interaction in the wheelchair-based rehabilitation robot. *IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp. 1811–1816.

[68] Song, Y., Li, M., Sun, L., , and Ji, J. (2005). Global visual servoing of miniature mobile robot inside a micro-assembly station. *IEEE Int. Conf. on Mechatronics and Automation*, pp. 1586–1591. Niagara Falls, Canada.

[69] Starner, T. and Pentland, A. (1998). Real-time american sign language recognition using desk and wearable computerbased video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(12), 1371–1375.

[70] Sturm, P. (2002). Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length. *Image Vision Computation* **20**(5–6), 415–426.

[71] Sznaier, M. and Camps, O. (1998). Control issues in active vision: Open problems and some answers. *1998 IEEE Conference on Decision and Control*, pp. 3238–3244.

[72] Sznaier, M., Inanc, T., and Camps, O. (2000). Robust controller design for active vision systems. *2000 American Control Conference*, pp. 2013–2017.

[73] Sznaier, M., Murphy, B., and Camps, O. (2000). An LPV approach to synthesizing robust activevision systems. *2000 IEEE Conf. Dec. Control*, pp. 2545–2550.

[74] Taylor, C., Ostrowski, J., and Jung, S. H. (1999). Robust visual servoing based on relative orientation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 574–580.

[75] Taylor, C. J., Kosecka, J., Blasi, R., and Malik, J. (1999). Comparative study of vision-based lateral control strategies for autonomous highway driving. *International Journal of Robotics Research* **18**(5), 442–453.

[76] Thorhallsson, T. and Murray, D. W. (1999). Tensors of three affine views. *IEEE Computer Vision and Pattern Recognition*, pp. 450–456.

[77] Tsotsos, J. K., *et al.* (1998). PLAYBOT: A visually-guided robot for physically disabled children. *Image and Vision Computing* **16**(4), 275–292.

[78] Vitrani, M., Morel, G., and Ortmaier, T. (2005). Automatic guidance of a surgical instrument with ultrasound based visual servoing. *IEEE Int. Conf. on Robotics and Automation*, pp. 508–513. Barcelona, Spain.

[79] Wang, Y. F., Uecker, D. R., and Wang, Y. (1996). Choreographed scope maneuvering in robotically–assisted laparoscopy with active vision guidance. *3rd IEEE Workshop on Applications of Computer Vision*, p. 187. Sarasota.

[80] Xie, H., Chen, L., Sun, L., and Rong, W. (2005). Hybrid vision-force control for automatic assembly of miniaturized gear system. *IEEE Int. Conf. on Robotics and Automation*, pp. 1368–1373. Barcelona, Spain.

[81] Zergeroglu, E., Dawson, D. M., de Queiroz, M. S., and Behal, A. (2001). Vision-based nonlinear tracking controllers with uncertain robot-camera parameters. *IEEE Transactions on Mechatronics* **6**(3), 322–337.

8

Identification and Control Structure Design in Active (Acoustic) Noise Control

Miquel A. Cugueró[1], Bernardo Morcego[1], and Ricardo S. Sánchez Peña[2,1]

[1] Sistemes Avançats de Control, ESAII, Universitat Politècnica de Catalunya (UPC), Rbla. Sant Nebridi 10, Terrassa, 08222 Barcelona, Spain
{miquel.angel.cuguero,bernardo.morcego}@upc.edu
[2] Institució Catalana de Recerca i Estudis Avançats (ICREA), Barcelona, Spain
ricardo.sanchez-pena@upc.edu

Summary. The purpose of this chapter is to apply robust identification and control techniques to perform active noise control (ANC). The identification phase is based on a control-oriented robust identification approach that considers both parametric and nonparametric descriptions of the system, and quantifies the uncertainty. The controller design compares the feedback (FB), feedforward (FF) and hybrid (FB/FF) control structures. The feedback control is synthesized and evaluated in the robust control framework. It is designed using $\mathcal{H}_\infty$ optimal control as a mixed-sensitivity problem, and the robust performance of the loop is analyzed *via* the structured singular value (μ). The FF controller is an adaptive identifier, based on the robustly normalized σ-algorithm. Two approaches are developed to decide what control structure is more efficient on a 4-m duct example with broadband noise. In addition, the compromises between identification and control, the inherent limitations of feedback and implementation issues in ANC are explicitly pointed out. Relations between performance, controller order, parametric/nonparametric models and digital signal processor (DSP) implementation are discussed. Theoretical and experimental results on the duct are compared. Finally the gaps that still remain between theory and practice in this type of applications, are outlined.

Key words: Active noise control, $\mathcal{H}_\infty$ control, feedforward control, hybrid controller, structured singular value.

8.1 Background

8.1.1 Active Noise Control

Active noise control has been a very active research area for many years since the seminal ideas of [24]. Today, the underlying principles are well established ([23, 28]), but there are still many practical issues to be solved. The appearance of signal processors, *e.g.* DSP,has allowed ANC to become a feasible noise-suppression technology that has progressed from laboratory research to industrial implementation. A wide variety of applications that include aircraft engines, automobile interiors, heating, ventilation and air conditioning (HVAC) systems as well as household appliances have been produced. In general, ANC has been proven

as a viable method for noise suppression in low-frequency ranges, where traditional passive noise control devices become massive, bulky, or less effective.

The most widely used approach for ANC in practical applications is based on the use of adaptive FF algorithms because they do not need a previous identification stage (see details in Section 8.1.3). The FF–ANC efficiency usually depends on how accurately the online model adjusts the primary path transfer function, and could suffer from possible instability. As noticed in [3, 23], this does not preclude the use of feedback or hybrid (feedback/feedforward) control structures. The feedback control structure of an ANC system was first introduced in [29]. In this scheme, the system only requires the downstream error sensor. Unfortunately, this configuration provides poor broadband noise attenuation over a limited frequency range due to the spillover effect. A complete study of the limitations of feedback in ANC has been presented in [17], mainly based on the classical limitations of feedback loops ([14, 36]). However, if a nonacoustical reference signal is not available and/or the order of the system is small, feedback control should be a feasible approach ([34]). Furthermore, many applications can only deal with noise through a feedback control loop, *e.g.* headsets, windows ([21, 34]). Finally, hybrid control combines both FF and FB, which could potentially ([43]) add up the benefits of both approaches. Nevertheless, it is not always clear if this is so in practice. In this chapter two different approaches are developed that compare hybrid *versus* FF and FB controllers, as well as a systematic way of anticipating its possible benefits.

Plant uncertainty is one of the major contributing factors that affects performance and stability, in particular in ANC systems ([4]), as will be detailed throughout this chapter. It may be caused by modeling, computational, and/or measurement errors, or even perturbations in physical conditions. These factors lead to deviations of the plant from the nominal model, which should be considered at the control synthesis stage so that the closed loop is robust. Most robust control design methods dealing with uncertain models use a worst-case deterministic criterion to describe uncertainty, *e.g.* IMC or $\mathcal{H}_\infty$ optimal control ([3, 4, 21, 40]). Physical modeling can be used for simulations purposes, but in the end an experimental identification procedure is applied when control is the final objective, as concluded in [18]. Usually, models are obtained using classical parameter-estimation ([3]) and there is no systematic way to produce a deterministic uncertainty bound ([21]). The area of control-oriented and/or robust identification ([9], Chapter 10 of [35]) is instrumental in the design of robust controllers for ANC applications. Further explanations of the methodology used in the identification of the ANC problem presented in this chapter are given in the next section.

Here, adaptive FF, $\mathcal{H}_\infty$ FB and hybrid controller structures are systematically analyzed based upon the identification and controller robustness and performance experimental results, as suggested in [3]. Quantitative results at the identification stage using two different approaches, indicate if a hybrid controller adds extra benefits to a more standard FF adaptive one.

8.1.2 Robust Identification

General Background

The robust identification problem has been posed in [16] and has attracted considerable attention since then. It considers model uncertainty due to two different sources: measurement noise and lack of knowledge of the system itself due to the limited information supplied by the experimental data. Therefore, these identification procedures are based not only on the experimental data (*a posteriori* information), but also on the *a priori* assumptions on the class of

systems to be identified. The algorithms produce a nominal model based on the experimental information and a *worst-case* bound over the set of models defined by the *a priori* information.

From the different methods that use this approach, here we apply interpolatory algorithms, based on the rational interpolation theory developed in [6]:

Theorem 8.1.1 *There exists a transfer function $f(z) \in \mathcal{BH}_\infty$ ($\overline{\mathcal{B}}\mathcal{H}_\infty$) such that:*

$$\sum_{z_o \in \mathcal{D}} Res_{z=z_o} f(z) C_- (zI - A)^{-1} = C_+ \tag{8.1}$$

if and only if the following discrete-time Lyapunov equation has a unique positive (semi)-definite solution.

$$M = A^* M A + C_-^* C_- - C_+^* C_+ \tag{8.2}$$

where A, C_- and C_+ are constant complex matrices of appropriate dimensions. If $M > 0$ then the solution $f(z)$ is nonunique and the set of solutions can be parameterized in terms of $Q(z)$, an arbitrary element of $\overline{\mathcal{B}}\mathcal{H}_\infty$, as follows:

$$f(z) = \frac{T_{11}(z) q(z) + T_{12}(z)}{T_{21}(z) q(z) + T_{22}(z)} \tag{8.3}$$

$$= F_\ell \left[L(z), Q(z) \right] \tag{8.4}$$

$$T(z) = \begin{bmatrix} T_{11}(z) & T_{12}(z) \\ T_{21}(z) & T_{22}(z) \end{bmatrix} \tag{8.5}$$

where system $T(z)$ has the following state-space realization:

$$T(z) \equiv \left[\begin{array}{c|c} A_t & B_t \\ \hline C_t & D_t \end{array} \right] \tag{8.6}$$

$$\begin{aligned} A_t &= A \\ B_t &= M^{-1} \left(A^* - I \right)^{-1} \left[-C_+^* \; C_-^* \right] \\ C_t &= \begin{bmatrix} C_+ \\ C_- \end{bmatrix} (A - I) \\ D_t &= I + \begin{bmatrix} C_+ \\ C_- \end{bmatrix} M^{-1} * X \\ X &= \left(A^* - I \right)^{-1} \left[-C_+^* \; C_-^* \right] \end{aligned}$$

where $\mathcal{BH}_\infty$ ($\overline{\mathcal{B}}\mathcal{H}_\infty$) is the open (closed) ball of $\mathcal{H}_\infty = \left\{ \sup_{|z|<1} \|f(z)\|_\infty < \infty \right\}$. The classical frequency response (Nevanlinna–Pick) and time response (Carathéodory–Fejér) interpolation results can be obtained as special cases of the above, by a suitable selection of the matrices A, C_+, C_-. This leads to the solution of the problem of simultaneously interpolating time and frequency experimental information, which can be solved as a convex LMI optimization problem ([25, 32]).

In many cases, part of the model has a clear parametric structure. In these cases, disregarding this information may lead to conservative results. In addition, the model order in

classical interpolatory results duplicates the number of data points (in the case of frequency data), hence the use of parametric models (usually of order two) to fit the most significant frequency peaks, drastically reduces the total model order. This is usually the case in mechanical flexible structures and acoustic applications, which have a well-defined parametric model for the lower-frequency modes. On the other hand, the unknown higher-frequency behavior naturally leads to a nonparametric model.

The class of models traditionally considered in robust identification ([16]) is a subset of $\mathcal{H}_\infty$, denoted as $\mathcal{S}(K,\rho)$. It corresponds to exponentially stable systems (finite or infinite dimensional) that have a stability margin of $(\rho-1)$, a peak response to complex exponential inputs of K, and satisfy the time-domain bound

$$|h(k)| \leq K\rho^{-k}$$

Here, this set will be used to characterize the nonparametric part of the model: $H_{np}(z)$.

In the application developed in this chapter, the *a posteriori* (frequency-domain) experimental information and the *a priori* models and measurement noise sets are in the framework presented in [31, 32], as follows:

$$\mathcal{T} = \{H_{np}(z) + H_p(z) \mid H_{np} \in \mathcal{S},\ H_p \in \mathcal{P}\} \tag{8.7}$$

with

$$\begin{aligned}
\mathcal{S}(K,\rho) &= \left\{ H \in \mathcal{H}_\infty \mid \sup_{|z|<\rho} \|H(z)\|_\infty < K \right\} \\
\mathcal{P}(\mathbf{G}) &= \left\{ \mathbf{p}^T\mathbf{G}(z) \mid \mathbf{p} \in \mathbb{R}^{N_p},\ p_i \in [a_i, b_i] \right\} \\
\mathcal{N}_f(\epsilon_f) &= \left\{ \eta \in \mathbb{C}^{N_f} \mid |\eta_k| \leq \epsilon_f \right\} \\
\mathbf{y}^f &= \left\{ [y^f]_k = [\hat{h}]_k + [\eta]_k, k = 1,\ldots,N_f,\ [\hat{h}]_k = H(e^{j\Omega_k}),\ \eta \in \mathcal{N}_f \right\}
\end{aligned}$$

Here, the nonparametric *a priori* information characterizes only the smoothness and magnitude of the whole class of models in $\mathcal{S}$, but cannot distinguish among other properties, *e.g.* low-frequency model structure. Instead, the *a priori* parametric knowledge in $\mathcal{P}$ provides more "structured" information. This structure comes from the components $g_i(z)$ of vector $\mathbf{G}(z)$, which are known linearly independent functions. In this case, a finite set of Kautz orthonormal bases $B_i(z)$ has been selected, and tuned to the experimental information (see [5]). In addition, the experimental data are the N_f samples of the frequency response of the system at frequency values Ω_k, corrupted by realizations belonging to the noise set $\mathcal{N}_f$, collected in vector form $\mathbf{y}^f = \hat{h} + \eta \in \mathbb{C}^{N_f}$. The resulting identified model has the form:

$$H_{id}(z) = H_{np}(z) + \sum_{i=1}^{N_p} p_i B_i(z), \quad p_i \in [a_i, b_i], \quad H_{np}(z) \in \mathcal{S}$$

Consistency and Identification

The result in [32] solves two problems: establishing consistency of the data and obtaining a nominal model and bounds on the identification error. These are posed as a constrained finite-dimensional convex optimization problem, that can be efficiently solved using linear matrix inequalities techniques. In ([31]) this result is extended to parametric/nonparametric model structures, as the ones stated previously.

The important concept of consistency is defined in terms of the smallest set of models, according to all the available input data (*a priori* and *a posteriori*), which are indistinguishable from the point of view of the input information. This set should be "covered" when the final goal is to design a robust controller, which can be applied to the actual plant. For the case described before, it is as follows:

Definition 8.1.1 *The consistency set contains all possible models that could have produced the* a posteriori *information , in accordance with the class of measurement noise:*

$$\mathcal{T}(\mathbf{y}^f) \stackrel{\text{def}}{=} \left\{ H \in \mathcal{T} \mid (\mathbf{y}^f - \hat{h}) \in \mathcal{N}_f \right\}$$

This means that, there is no way to select a smaller set of candidate models with the knowledge of $(\mathbf{y}^f, \mathcal{T}, \mathcal{N}_f)$. Note that the above set is independent of the metric selected and depends on all the input information. From the above definition, the *a posteriori* and *a priori* information are consistent if $\mathcal{T}(\mathbf{y}^f) \neq \emptyset$ (Chapter 10 of [35]).

The optimization procedure that fits simultaneously the dynamic portion $H_{np}(z)$ and the parameters $\{p_i,\ i = 1, \ldots, N_p\}$, solves the following problem as a convex optimization one, *via* a set of LMI's.

Problem 1 *Given the experiments* $\mathbf{y}^f$ *and the* a priori *sets* $(\mathcal{T}, \mathcal{N}_f)$*, determine:*

(i) If the a priori *and* a posteriori *information are consistent,* i.e. *the consistency set is non-empty:* $\mathcal{T}(\mathbf{y}^f) \neq \emptyset$.
(ii) A nominal model that belongs to the consistency set $\mathcal{T}(\mathbf{y}^f)$.

The above *a priori* and *a posteriori* information allows a less conservative identification than in the cases where only time or frequency experiments are provided or when no knowledge of the parametric part of the model is used. Another interpretation can be made in terms of the "smaller" size of the consistency set, due to the fact that there is more experimental data and structured *a priori* information.

This consistency set can be determined by a set of LMIs using the result in [31] for the particular case where only frequency-domain experimental data is considered.

Theorem 8.1.2 *Given the experiment* $\mathbf{y}^f$*, it is consistent with the* a priori *information if and only if there exist two vectors:*

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{N_p} \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N_f} \end{bmatrix}$$

that solve the following LMI feasibility problem:

$$M = \begin{bmatrix} Q^{-1} & \frac{1}{K}X(\mathbf{w}) \\ \frac{1}{K}X^*(\mathbf{w}) & Q \end{bmatrix} > 0 \tag{8.8}$$

$$(\mathbf{y}^f - P_f\mathbf{p} - \mathbf{w}) \in \mathcal{N}_f \tag{8.9}$$

Here, $H(z_i) = w_i$, $i = 1, \ldots, N_f$ is considered, and the following matrices are defined:

$$P_f = \begin{bmatrix} G_1(z_1) & G_2(z_1) & \cdots & G_{N_p}(z_1) \\ G_1(z_2) & G_2(z_2) & \cdots & G_{N_p}(z_2) \\ \vdots & \vdots & \ddots & \vdots \\ G_1(z_{N_f}) & G_2(z_{N_f}) & \cdots & G_{N_p}(z_{N_f}) \end{bmatrix}$$

$$Q = \left[\frac{\rho^2}{\rho^2 - z_{i-1}^* z_{j-1}}\right]_{ij}, \qquad i, j = 1, \ldots, N_f$$

$$X(\mathbf{w}) = \text{diag}\left[\, w_1 \ \ldots \ w_{N_f} \,\right]$$

Once consistency is established, the second step of Problem 1 can be solved: generating a nominal model in the consistency set $\mathcal{T}(\mathbf{y}^f)$. The identification algorithm is based on the parameterization of all solutions of the generalized interpolation problem presented in in Theorem 8.1.1 ([6]). The algorithm can be summarized as follows.

1. Find feasible data vectors $\mathbf{p}, \mathbf{w}$ for the consistency problem, by solving the LMI feasibility problem given by Equations (8.8) and (8.9). Note that there is no need of any kind of optimality in the search for the feasible vectors. Instead, any pair $\mathbf{p}, \mathbf{w}$ in the admissible set will suffice.
2. Compute the Pick matrix M in Equation (8.8) (which should be positive-definite), corresponding to the vectors computed in step 1.
3. Use Theorem 8.1.1 to compute a model from the consistency set $\mathcal{T}$. Recall that all the models in $\mathcal{S}$ (*i.e.* all the solutions to the interpolation problem) can be parameterized as a linear fractional transformation (LFT) of a free parameter $Q(z) \in \overline{\mathcal{B}}\mathcal{H}_\infty$ as follows:

$$H(z) = F_\ell\left[L(z), Q(z)\right] \tag{8.10}$$

$$L(z) = \begin{bmatrix} T_{12}T_{22}^{-1} & T_{11} - T_{12}T_{22}^{-1}T_{21} \\ T_{22}^{-1} & -T_{22}^{-1}T_{21} \end{bmatrix} \tag{8.11}$$

with $T(z)$ in Equation (8.6). In particular, if the free parameter $Q(z)$ is chosen as a constant, then the nonparametric model order is less than or equal to $2N_f$.

Bases Selection

The procedure here follows the steps taken in [5], although more systematic procedures can be attempted (see for example [41]).

Let $\{F_k(z)\}_{k=1}^{\infty}$ be a set of orthonormal bases functions defined in $\mathcal{H}_2$, *i.e.* $\langle F_k, F_\ell \rangle = \delta_{k,\ell}$ with $\delta_{k,\ell}$ the Kronecker delta, and the inner product of every pair of functions $f,\ g \in \mathcal{H}_2$ is defined as

$$\langle f, g \rangle = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(e^{\jmath\Omega})\,\overline{g(e^{\jmath\Omega})}\,d\Omega \tag{8.12}$$

Then, every $G(z) \in \mathcal{H}_2$ can be represented by the series expansion:

$$G(z) = \sum_{k=1}^{\infty} g(k)\, F_k(z) \tag{8.13}$$

where $\sum_{k=1}^{\infty} |g(k)|^2$ converges in $\mathcal{H}_2$ and each coefficient $g(k)$ is computed from the values $G(z_1), \ldots, G(z_k)$. Therefore, truncating Equation (8.13) to the nth order can provide a low-order approximation $G_n(z)$ of $G(z)$ as follows,

$$G_n(z) = \sum_{k=1}^{n} g(k)\, F_k(z) \tag{8.14}$$

It is possible to build a set of orthonormal bases functions from a family of stable allpass filters with input normal realizations (see [27]). The functions $\{F_k(z)\}_{k=1}^{n}$ are computed from a set of scalar allpass filters $\{H_i\}_{i=1}^{m} \in \mathcal{RH}_2$, each one with an associated McMillan degree $\{n_i\}_{i=1}^{m}$, with $n = \sum_{i=1}^{m} n_i$. Their minimal realization is defined as $H_i(z) = C_i\,(zI - A_i)^{-1} B_i + D_i$, with the eigenvalues of A_i lying outside the unit circle.

The series connection has a state-space input normal realization of the form:

$$H(z) = \prod_{i=1}^{m} H_i(z) = C_n\,(zI - A_n)^{-1} B_n + D_n$$

with $n = \sum_{i=1}^{m} n_i$. Let $\mathbf{x}(t) = (\mathbf{x}_1(t), \ldots, \mathbf{x}_m(t))^T$, be the state of the filter $H(z)$, where each vector component $\mathbf{x}_i$ assumes the form $\mathbf{x}_i = (x_1, \ldots, x_{n_i})^T$ for $i = 1, \ldots, m$, then the set of orthonormal bases are defined by the transfer functions, $F_k(z)$, from the input $u(t)$ of the first allpass to each of the filter state components $x_k(t)$, with $k = 1, \ldots, n$,

$$x_k(t) = F_k(q)u(t) \tag{8.15}$$

$$F_k(z) = e_k\,(zI - A_n)^{-1} B_n, \qquad k = 1, \ldots, n \tag{8.16}$$

with q the forward shift operator: $qu(t) = u(t+1)$ and e_k the kth Euclidean bases vector in $\mathbb{R}^n$. It can be shown that the finite-dimensional subspace generated from the set of bases functions $X_n = \text{span}\,\{F_1(z), \ldots, F_n(z)\}$ generates a *complete model set* in $\mathcal{H}_2$, which can be used to build the approximate model $G_n(z)$ for $G(z)$ ([33]). Complete model sets currently in use are the Laguerre and Kautz models, the Cauchy kernel models or rational wavelets ([12]), and more recently *fundamental models sets* ([1]), among others.

In the application we make a broad use of the two parameter Kautz model as *a priori* parametric set information, to tune the resonant frequencies of the system's peaks. It has been shown in [42] and [39], that for lightly damped systems or systems with scattered poles as in the case of acoustic noise systems, Kautz models have optimal fixed denominator sets.

Finally, uniform boundedness in $\mathcal{H}_\infty$ for several complete sets in $\mathcal{H}_2$ was derived in [7] from the set of bases defined by equation,

$$B_i(z) = \frac{\sqrt{1 - |z_i|^2}}{1 - z_i^* z} \prod_{j=1}^{i-1} \left(\frac{z - z_j}{1 - z_j^* z} \right) \tag{8.17}$$

where z_i is the ith number in the sequence $(z_{1,1}, \ldots, z_{1,n_1}, z_{2,1}, \ldots, z_{1,n_2}, \ldots)$ being $z_{i,j}$ the jth pole of the ith allpass filter on the cascade.

For each resonant peak in the experimental frequency response curve – at frequency ω_k and with magnitude $m_k = |f(j\omega_k)|$ – we have to specify a pair of complex-conjugate

poles $[p_k, \bar{p}_k]$. One way to approach this problem is by fitting the following second-order continuous-time transfer function to an experimental peak:

$$f_k(s) = \frac{c_k}{s^2 + 2\omega_{n_k}\zeta_k s + \omega_{n_k}^2} \tag{8.18}$$

with poles located at:

$$[p_k, \bar{p}_k] = \omega_{n_k}\left(-\zeta_k \pm \jmath\sqrt{1-\zeta_k^2}\right)$$

The constant $c_k \leq 1$ is considered in Equation (8.18) to adjust the width of each peak. It is easy to verify that the constants ω_{n_k} and ζ_k are related to the frequency and the magnitude of the peak as follows:

$$\omega_k = \omega_{n_k}\sqrt{1-2\zeta_k^2}$$
$$m_k = \frac{c_k}{2\zeta_k\omega_{n_k}^2\sqrt{1-\zeta_k^2}}$$

Hence, given ω_k and m_k from the experimental curve, the resonant poles $[p_k, \bar{p}_k]$ associated to each pair of filters of the orthogonal bases set can be computed. The procedure for discrete-time experimental data can be made accordingly, transforming the continuous time data by a bilinear transformation.

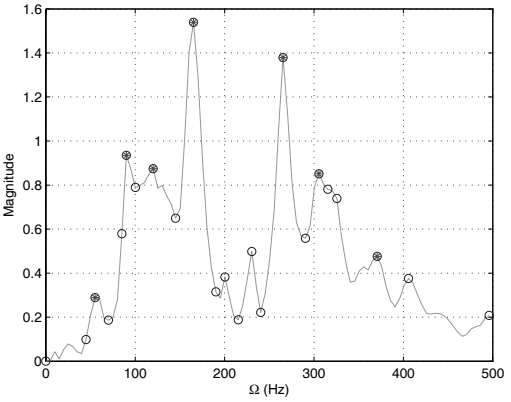


Fig. 8.1. Interpolation points and peak frequencies

As an example, Figure 8.1 illustrates the 23 interpolation points used to identify the secondary channel of the acoustic setup presented in next section, which amounts to a nonparametric model of order 46. In addition the seven asterisks indicate the peaks that will be fitted by second-order models (see Figure 8.2), which adds an extra 14th-order model. The number of peaks to be fitted depends on the application, and in particular in the frequency region

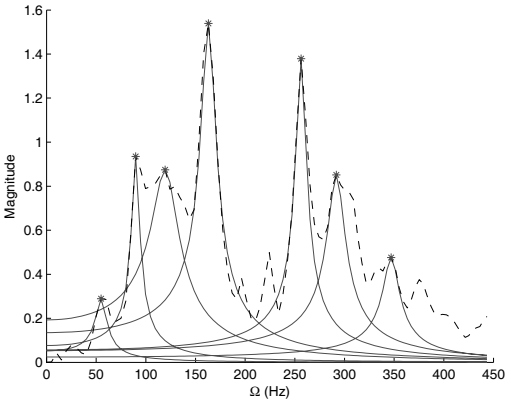


Fig. 8.2. Second-order bases fitted to the experimental data

where performance needs to be guaranteed. In Figure 8.3 the parametric and nonparametric components and model error of the acoustic system are presented, and in Figure 8.4 its complete identified model. The resulting 60th-order model could be excessive or even unacceptable, depending on the technology and the particular application. Further practical considerations should be taken into account when using it in an actual implementation of a feedforward/feedback controller for acoustic applications. These practical issues will be detailed further in this chapter.

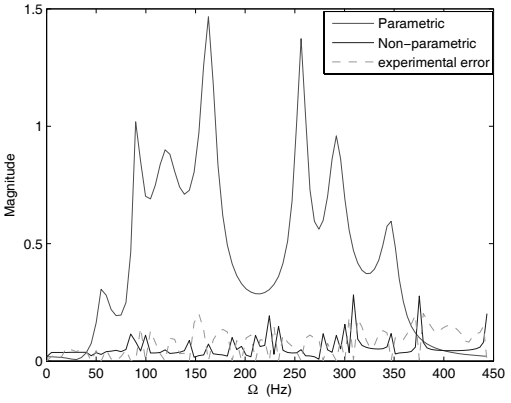


Fig. 8.3. Parametric, dynamic and model error for experimental data

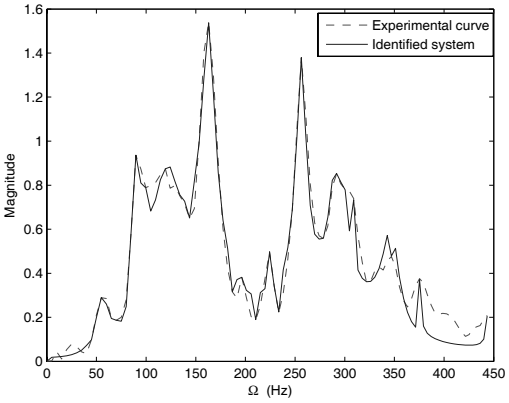


Fig. 8.4. Identified model of experimental data

Uncertainty Description

The objective of robust identification is not only to derive a nominal model but also to compute an upper bound for the worst-case identification error. This bound is *global* when it does not depend on the particular experiment, but only on the *a priori* information. Usually it is given in terms of the *radius* and *diameter* of information [10, 16] and is valid for *all* interpolatory algorithms, taking the available *a priori* and *a posteriori* information as inputs. In our case, due to the fact that the identified model is in set $\mathcal{T}(\mathbf{y}^f)$, its distance to the (Chebyshev) center of this consistency set is within the diameter of information. As a consequence, the algorithm is optimal up to a factor of two when compared with central strongly optimal procedures. For the same reason, it is also convergent and therefore the modeling error tends to zero as the information is completed. Further details of global errors in robust identification can be found in [35], and will not be explored here.

The computational requirements to obtain these bounds are minimal, and its main application is to prove algorithm convergence. Due to the fact that these type of global errors are independent of the particular experimental outcomes of the system, even for strongly optimal algorithms, they produce exceedingly conservative bounds. The practical application of these identification procedures requires alternative methods for computing uncertainty bounds ([15]). It is possible to obtain better ones by solving convex optimization problems using the experimental data. A more practical approach is to consider a local identification error, which depends on the particular experimental information at hand. In fact, even a frequency-dependent uncertainty upper bound is desirable, due to the fact that it agrees with the usual necessary and sufficient condition for model dynamic uncertainty and provides a less-conservative set of models to describe the physical system.

In addition, the fact that from this procedure we obtain a consistency set, allows different descriptions of the uncertainty. For instance, we can compute a bound that covers the whole

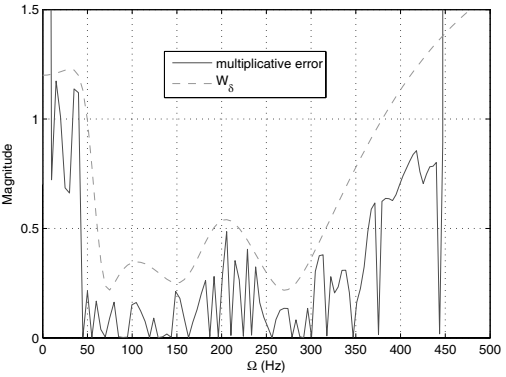


Fig. 8.5. Relative identification error and multiplicative uncertainty weight $W_\delta(z)$

set and includes both, the parametric and nonparametric uncertainty. This can also be interpreted as having "exact" nominal parameters and a global bound, due to the uncertainty of the nonparametric portion. Unless we have extra information, both interpretations are indistinguishable. In any case, the model set considered will be described as having multiplicative uncertainty and involves the whole physical system, as follows:

$$\mathcal{G} = \left\{ [I + \Delta W_\delta(z)]\, G(z) \mid \Delta \in \mathbb{C}^{r \times r}\ ,\ \bar{\sigma}(\Delta) < 1 \right\}$$

The nominal model $G(z)$ has been determined by the interpolatory procedure in a previous section. The uncertainty weight $W_\delta(z)$ should "cover" (in magnitude) the plot of the relative identification error:

$$e_r^k = \left[\mathbf{y}_k^f - G(e^{\jmath\Omega_k}) \right] G^{-1}(e^{\jmath\Omega_k}), \quad k = 1, \ldots, N_f \tag{8.19}$$

where $\mathbf{y}_k^f$ is the experimental response at frequency Ω_k (see Figure 8.5). Several practical issues concerning this computation will be detailed in the next section.

Finally, this model set perfectly fits the framework of $\mathcal{H}_\infty$ or μ-synthesis controller design.

8.1.3 Adaptive Identification

The term adaptive is usually related in the literature to online control techniques. The term identification, conversely, is related to offline parameter-estimation techniques. This section deals with online parameter-estimation techniques and makes use of results from both areas.

The application studied in this chapter, ANC, does not follow the typical closed-loop control scheme. Figure 8.6 shows a block diagram of the system and its control blocks. It can be observed that the error signal is located at the output of the system, while the input signal cannot be altered because it is the noise source. The controller's main function is to identify the dynamics of the sound-propagation system and to generate the opposite signal, thus producing as much silence as possible. Figure 8.6 also shows that the controller is laid out in a

feedforward fashion. The sound-propagation system, a duct in the present case, suffers from slow dynamics variations due to temperature and humidity changes. It also collects sudden external noise perturbations and pressure variations at its end. In view of this, the main task of an active noise controller is to adaptively identify the sound-propagation system. An important difficulty with this kind of control structure is to maintain and assure stability. One of the main concerns in this chapter is to deal with stability explicitly.

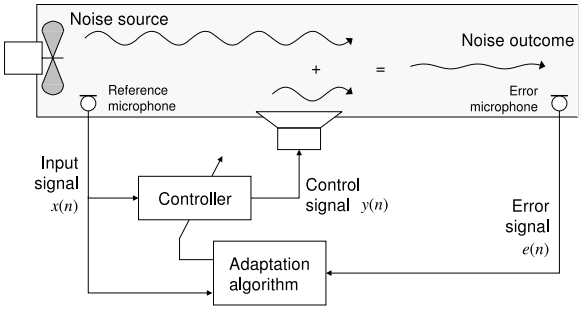


Fig. 8.6. Simplified conceptual system block diagram

Stability is an important issue for different reasons. On the one hand, there are physical reasons: the output loudspeaker is a nonminimum phase system and, as will be seen later, its transfer function needs to be inverted; in addition, one of the main characteristics of this kind of systems is their intrinsic delays, which also need to be accounted for, bringing up stability considerations. On the other hand, the algorithms and filter structures used for noise cancelation do not assure stability of the solution. The present chapter gives alternatives to the state-of-the-art algorithms that are intrinsically stable.

Classical Feedforward Structures

Feedforward active noise control systems are implemented using two different control structures, FXLMS and FULMS, standing for filtered X and filtered U least mean squared algorithms (see [23]). These control structures stem from different models or interpretations of the system, which will be explained in the following paragraphs.

The first *naive* model of the acoustic duct neglects the dynamics of the sensors and the actuator, and also neglects the propagation of the cancelation signal upstream and downstream of the duct. This model is shown in Figure 8.7.

The output signal $e = d - y = [G_{pri}(z) - W(z)]\, x$ becomes zero when $G_{pri}(z) = W(z)$. This means the controller $W(z)$ is exactly representing the dynamics of the duct, with a phase difference of 180 degrees.

The most widely used cancelation algorithm is FXLMS, based on the model shown in Figure 8.8. This model includes the secondary path, which is the propagation dynamics that modifies signal $y(n)$ through the cancelation speaker and the downstream portion of the duct to the error microphone. Refining slightly the previous model of the system one can distinguish several elements that were ignored. On the one hand, signals $x(n)$ and $e(n)$ are captured with the reference and the error microphones, respectively. These signals are then amplified, filtered and digitalized. On the other hand, signal $y(n)$ is converted to an analogical, amplified one and

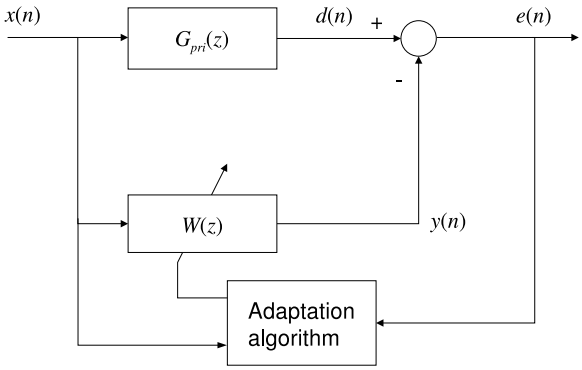


Fig. 8.7. *Naive* model of the acoustic duct with controller

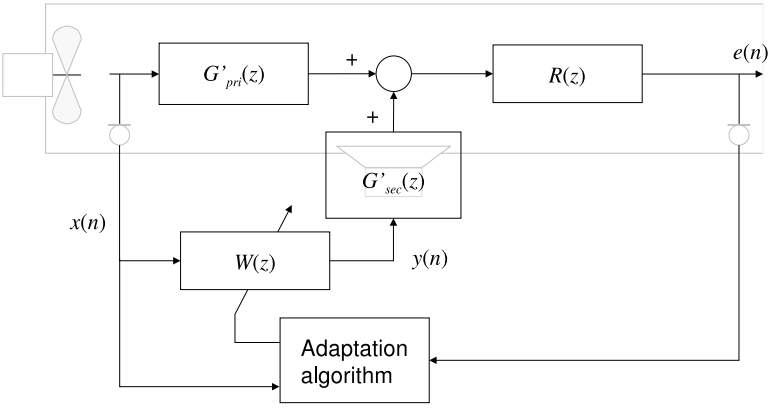


Fig. 8.8. Block diagram of the duct system including the secondary path

delivered by a loudspeaker. Finally, after signal $y(n)$ is in the duct, it must travel some distance to reach the error microphone. All these elements are captured in the transfer functions shown in Figure 8.8, but considering that both signals $G'_{pri}(z)x$ and $G'_{sec}(z)y$ pass through the same piece of duct, $R(z)$, assuming linear dynamics and using

$$\begin{aligned} G_{pri}(z) &= R(z)G'_{pri}(z) \\ G_{sec}(z) &= R(z)G'_{sec}(z) \end{aligned} \tag{8.20}$$

it is possible to represent the same model in a more convenient way, as in Figure 8.9. Considering

$$E(z) = R(z)[G'_{pri}(z) - G'_{sec}(z)W(z)]X(z)$$

and using Equation (8.20), it is easy to see that when the error signal, $e(n)$, goes to zero (that is when the filter $W(z)$ has converged) its transfer function will be:

$$W(z) = \frac{G_{pri}(z)}{G_{sec}(z)} \tag{8.21}$$

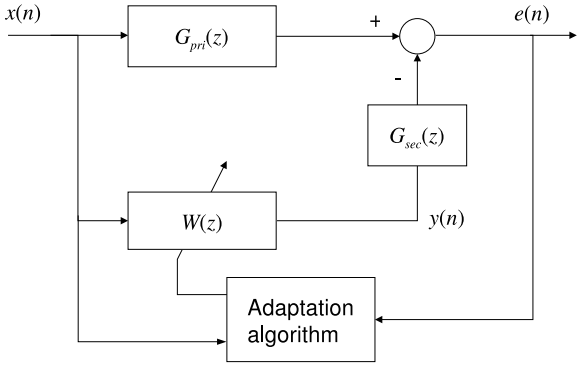


Fig. 8.9. Simplified block diagram of the duct system including the secondary path

This equation reveals two important details of the cancelation filter $W(z)$. First, if the overall delay of the secondary path is larger than the delay of the primary path the filter will not be realizable. This is the limiting causality constraint of FXLMS and FULMS control structures. Second, if the secondary-path transfer function is nonminimum phase, then $W(z)$ becomes unstable. In fact, this is the usual circumstance due to the dynamics of loudspeakers. Therefore, to avoid instability of $W(z)$, some modifications need to be done.

To understand the derivation of the FXLMS algorithm it is necessary to assume that the parameters in $W(z)$ are calculated with an LMS-like, or gradient descent, algorithm. In that case the error to minimize is

$$e = d - y = d - W(z)G_{sec}(z)x \tag{8.22}$$

but considering slow filter parameter variations, or linear dynamics, the order of $W(z)$ and $G_{sec}(z)$ in Equation (8.22) can be swapped over. Gradient descent algorithms need to calculate the gradient of the error with respect to the parameters,

$$\nabla e = G_{sec}(z)x \tag{8.23}$$

According to Equation (8.23), the parameter adaptation algorithm must be fed with a filtered version of signal $x(n)$. The filter should be the secondary-path transfer function, $G_{sec}(z)$, which is not available but can be estimated either online or offline. A more suitable expression of $\nabla e(n)$ would be

$$\nabla e = \hat{G}_{sec}(z)x$$

where $\hat{G}_{sec}(z)$ is the estimate of $G_{sec}(z)$. The previous transformations modify the control structure shown in Figure 8.9 leading to the one shown in Figure 8.10, where $W(z)$ is now fed with $x(n)$ and converges to $G_{pri}(z)$ rather than $\frac{G_{pri}(z)}{G_{sec}(z)}$, as stated previously in Equation (8.21). Of course, there are many important practical and theoretical considerations related to the accuracy of $\hat{G}_{sec}(z)$, *e.g.* [8, 13, 37], but they will not be taken into account here for the sake of brevity.

The FXLMS control structure, although being one of the most widely used, considers an incomplete model of the system. The output signal $y(n)$ is fed into the acoustic duct through the secondary loudspeaker and is meant to counteract signal $G_{pri}(z)x$. When a signal is inserted into the duct, it travels downstream and it also travels upstream, but the effects of $y(n)$

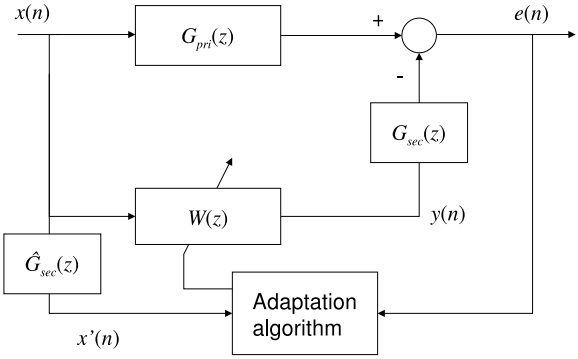


Fig. 8.10. Standard control scheme of FXLMS

upstream were ignored when FXLMS was generated. The control structure FULMS is meant to contemplate the effects of signal $y(n)$ upstream, which will be designated as acoustic feedback.

The acoustic feedback, shown in Figure 8.11 as transfer function $F(z)$, is the effect that signal $y(n)$ has over signal $u(n)$ when it is captured by the reference microphone. The transfer function $F(z)$ models the digital-to-analog conversion of signal $y(n)$, the upstream dynamics of the duct and the analog-to-digital conversion to obtain signal $x(n)$. Neglecting the effects

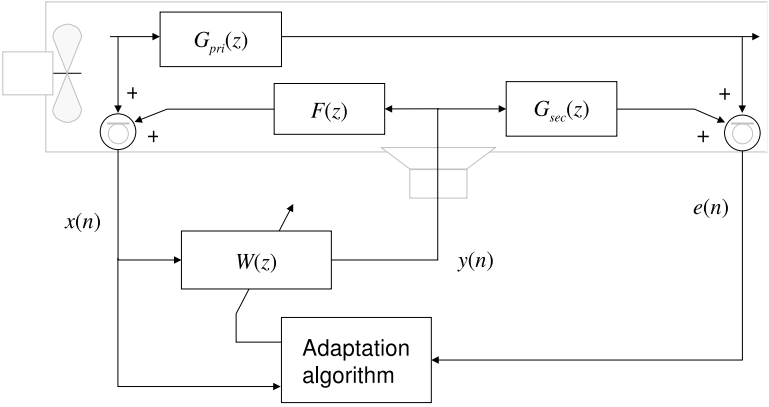


Fig. 8.11. Model of the duct with the secondary path and the acoustic feedback

of acoustic feedback is not as dangerous as neglecting secondary-path effects, but analyzing Figure 8.11 it is possible to conclude that the transfer function of the cancelation filter in Equation (8.24) could become unstable if the product $W(z)F(z)$ was larger than unity at some frequency.

$$H(z) = \frac{W(z)}{1 - W(z)F(z)} \tag{8.24}$$

The best way to counteract feedback effects is to minimize the acoustic feedback, for instance using a different transducer for the reference signal (like a piezoelectric sensor). Unfortunately, this is not always possible, hence it is useful to include an acoustic feedback cancelation filter in the control structure. This measure makes the overall control algorithm more complex, but it adds robustness and performance as compensation.

The FULMS control structure is shown in Figure 8.12. Its relation with FXLMS is very simple: the cancelation filter in FXLMS, an FIR filter, becomes an IIR filter un FULMS, *i.e.* a rational transfer function with poles.

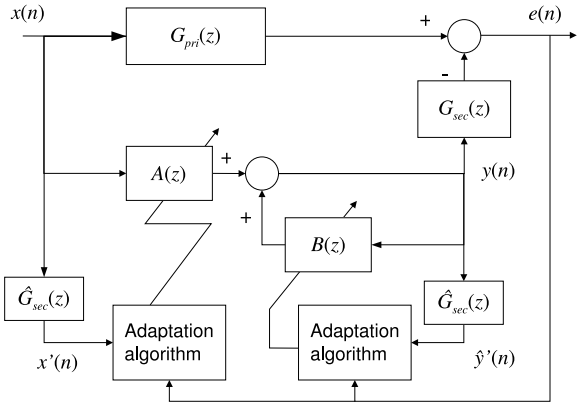


Fig. 8.12. FULMS control structure

From Figure 8.11 it is easy to verify that the optimal value of $W(z)$ is

$$W(z) = \frac{G_{pri}(z)}{G_{sec}(z) - G_{pri}(z)F(z)}$$

therefore, in order to adaptively eliminate the acoustic feedback, cancel the secondary-path effects and control the primary noise, $W(z)$ needs to be implemented with an IIR filter.

However, taking into account the gradient considerations that led to the FXLMS algorithm, the transfer function of $W(z)$ will have the same structure as $H(z)$ in Equation (8.24), *i.e.* a direct function modeling $G_{pri}(z)$ and a feedback function estimating the acoustic feedback,

$$\frac{Y(z)}{X(z)} = \frac{D(z)}{1 + D(z)\hat{F}(z)}$$

This transfer function can be arranged as the usual configuration of an IIR filter,

$$\frac{Y(z)}{X(z)} = \frac{A(z)}{1 - B(z)}$$

where $A(z) = D(z)$ and $B(z) = -\hat{F}(z)D(z)$. The control structure shown in Figure 8.12 includes filters $A(z)$ and $B(z)$, as well as the estimation of the secondary-path transfer function, $\hat{G}_{sec}(z)$, needed to eliminate the instability effects of modeling its inverse.

The main drawback of IIR filters with LMS-like adaptation rules is that neither convergence nor stability of the solution is assured. In fact, many practical situations give evidences that IIR filter can become unstable quite easily, *e.g.* [2].

Relation between Robust Adaptive Identification and Classical Feedforward Structures

As stated earlier, this chapter is about active noise cancelation, but a fundamental concern is to ensure the stability of the solution.

Most active noise-control solutions (commercial or not) are based on gradient-descent algorithms. In fact, the most widely used algorithm is LMS, mainly for speed of execution, which allows a large number of parameters to be applied in the cancelation filters. Many of those applications try to ensure the stability of their solutions by using proven control structures, such as FXLMS, or by carefully accomplishing physical laws that are known to maximize the effect of cancelation systems. However, there is little concern with the robustness of the algorithms employed.

Control researchers became very active in the 1980s in the area of robust adaptive control (see [30]), showing, for example, that an adaptive scheme designed for a plant model without disturbances considerations could go unstable in the presence of small disturbances ([19]). They also developed a considerable number of robust adaptive laws and proved, in each case, their robustness properties against different kinds of unstructured perturbations.

The adaptive algorithm used throughout this chapter in order to estimate the feedforward filter parameters is robust. The algorithm is the normalized LMS estimator with the σ-modification, which will be explained next (see [20] for a more detailed description).

The LMS algorithm, developed by Widrow [22], is a gradient-descent-based algorithm. The main equation of the gradient-descent algorithms is

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2}\nabla\xi(n) \tag{8.25}$$

where $\mathbf{w}$ is the parameter vector, $\nabla\xi(n)$ is the gradient of the error function with respect to $\mathbf{w}$ and μ is a convergence factor. This parameter-update law possesses many interesting statistical properties and is a proven algorithm. Its main problem is that obtaining $\nabla\xi(n)$ is often impractical and computationally intensive.

The simplest approximation to $\xi(n)$ is to use the instantaneous squared error instead. Then

$$\hat{\xi}(n) = e^2(n)$$

and the gradient used in the algorithm is

$$\nabla\hat{\xi}(n) = 2[\nabla e(n)]e(n) = -2\mathbf{x}(n)e(n)$$

When the estimate of the gradient is substituted in Equation (8.25) the well-known LMS rule is obtained:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\mathbf{x}(n)e(n) \tag{8.26}$$

This algorithm, as stated before, is simple, computationally inexpensive and effective. It is a good quality/price choice. One could argue, nevertheless, that its derivation was carried out in an ideal environment, with no consideration of model uncertainties or noise of any kind, which are necessary to guarantee boundedness properties of the parameters or the errors, or even convergence of the algorithm. Two modifications were then applied to Equation (8.26), in order to fulfill this lack of real considerations: robust normalization and σ-modification.

On the one hand, robust normalization is basically used to make the algorithm independent of signal-power changes. On the other hand, σ-modification guarantees boundedness of the parameters and its derivatives and boundedness of the estimation error against modeling errors, and also convergence of the algorithm (zero error) when there is no modeling error.

Robust normalization and σ-modification are summarized in the following equations:

$$\begin{aligned}\mathbf{w}(n+1) &= (1 - c\sigma(n))\mathbf{w}(n) + c\epsilon(n)\\ \epsilon(n) &= K[y(n) - \hat{y}(n)]\\ K &= \frac{r(n)}{1 + m_s(n)}\\ \hat{y}(n) &= \mathbf{w}^T(n)r(n)\end{aligned}$$

where c is related to the convergence rate and is chosen by the designer from $0 < c < 1$. The adaptive value $\sigma(n)$ plays an important role to avoid parameter drift, and it is defined as

$$\sigma(n) = \begin{cases} 0 & \text{if} \quad \|\mathbf{w}(n)\| \leq M \\ \left(\frac{\|\mathbf{w}(n)\|}{M} - 1\right)\sigma_0 & \text{if } M < \|\mathbf{w}(n)\| \leq 2M \\ \sigma_0 & \text{if} \quad \|\mathbf{w}(n)\| > 2M \end{cases}$$

The constant σ_0 is a positive value such that $\sigma_0 < (1 - c)/2$. The constant M is a bound of the Euclidean norm of the (unknown) true parameter set. M must be estimated beforehand and it allows bounding the parameters if their norm grows larger than M. The normalizing value $m_s(n)$, initialized to zero, is calculated with the following equation

$$m_s(n+1) = (1 - \delta_0)m_s(n) + x^2(n) + y^2(n)$$

where δ_0 is a parameter related to the frequency band, considering bounded uncertainty.

The parameters of the algorithm were set to the following values throughout all the experiments: $c = 0.5$, $\sigma_0 = (1 - c)/4 = 0.125$, $M = 0.5$ and $\delta_0 = 0.1$.

8.2 Acoustic-noise Experimental Setup

8.2.1 Description

The application is illustrated in Figure 8.13. It consists of a 4-m long square tube connected to a semianechoic[3] room (Figure 8.14), the other end connected to a noise generator, either a speaker or an industrial fan. This (primary) speaker generates noise in a certain controlled frequency bandwidth by means of a signal generator, and may physically simulate different noise sources. The advantage is that this noise source setup has a linear time invariant (LTI) behavior and precise experiments on the duct can be carried on. The industrial fan, however, is a real noise source with time-varying and nonlinear characteristics. Experimental results with both will be obtained and presented in Section 8.3.

There is also an error microphone near the control actuator (secondary speaker) and one near the noise source, known as the reference microphone. The microphones are omnidirectional BEHRINGER ECM8000 with linear frequency response within a bandwidth of 15 Hz to 20 kHz and -60 dB acoustic sensitivity. For this reason, the identification data below 15 Hz is

[3] These rooms have a nonabsorbing floor, hence the sound measurement also depends on the floor reflection. This is not a problem in this application.

unreliable and will not be considered in the uncertainty description (see details in Section 8.3). The speakers are BEYMA model 5 MP60/N of 5", 50 W, with a bandwidth of 50 Hz to 12 kHz.

The secondary acoustic circuit is the one related to the feedback-control section, with the control speaker as the input and the error microphone as the output. The perturbation signal that enters the error microphone coming from the acoustic path with its origin in the noise source is usually defined as the primary circuit. Both FF and FB controllers are implemented with a DSpace DSP-based on a Texas Instruments TMS320C40 over a DS1003.05 floating-point processor board. The sampling time for the identification experiments and control implementation is $T_s = 0.4$ ms, which is good enough for the real-time computations needed for both FF and FB controllers simultaneously. The complete signal-processing instrumentation is illustrated in Figure 8.15.

Fig. 8.13. Tube and input noise source (fan)

The control scheme applies the classical method ([24, 29]) of generating a signal as close as possible to the real noise but with opposite phase. This will be performed in three ways: by fixed $\mathcal{H}_\infty$ FB control, adaptive FF, and also combined as a hybrid controller (FF/FB), conceptually illustrated in Figure 8.16. As mentioned previously, a hybrid controller can be very effective in general ([43]), but the question if it really pays off in terms of performance and/or robustness needs to be answered. The present work continues that in [3] and produces a systematic approach to answer this question using two different approaches.

8.2.2 Physical Modeling

In this section, the main results on mathematical models for sound fields in rectangular tubes, such as the one proposed, are presented.

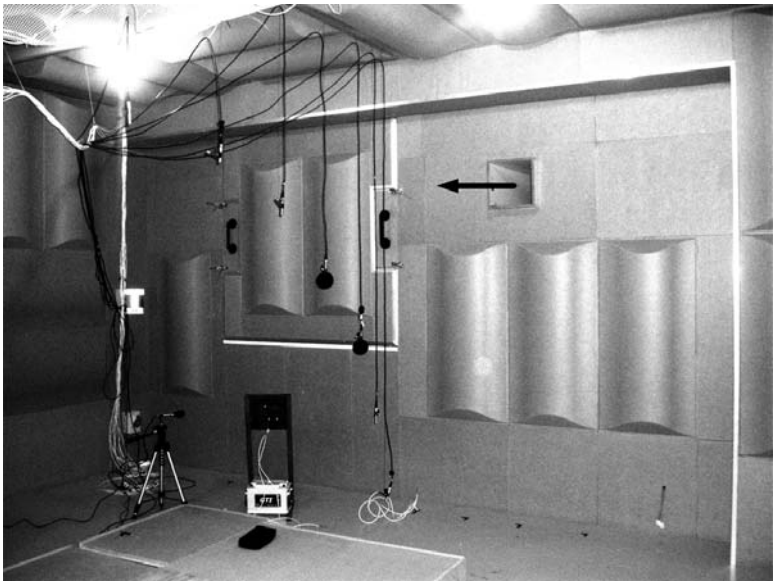


Fig. 8.14. Semianechoic room and tube output signal (arrow)

According to [18] and [4], below the cutoff frequency, the sound field in the duct can be treated as unidimensional with spatial coordinate $x \in [0, 1]$. The control loudspeaker is located at $x = x_s$, while the feedback microphone is located at $x = x_m$. A state-space model of the acoustic duct can be developed from three fundamental equations: the state equation, the continuity equation and the linearized inviscid force equation. Retaining r modes from such equations allows the duct state-space model to be derived with the following structure:

$$\begin{aligned} \dot{x}_d(t) &= A_d x_d(t) + B_d(t) u_s(t) \\ y_d(t) &= C_d x_d(t) \end{aligned} \tag{8.27}$$

where:

$$\begin{aligned} x_d(t) &= \left[q_1(t)\ \dot{q}_1(t)\ \cdots\ q_r(t)\ \dot{q}_r(t) \right]^T \\ A_d &= \text{block-diag}\left(\begin{bmatrix} 0 & 1 \\ -\omega_{n1}^2 & -2\xi_1\omega_{n1} \end{bmatrix}, \cdots, \begin{bmatrix} 0 & 1 \\ -\omega_{nr}^2 & -2\xi_1\omega_{nr} \end{bmatrix} \right) \\ B_d &= \left[0\ b_1\ \cdots\ 0\ b_r \right]^T; \quad C_d = \left[V_1(x_m)\ 0 \cdots V_r(x_m)\ 0 \right] \end{aligned}$$

This model can be extended even further by including the transfer function from the speaker voltage input V_s to the speaker baffle acceleration $\dot{v}_s$ given by:

$$\frac{s v_s(s)}{V_s(s)} = \frac{K_s s^2}{s^2 + 2\xi_s \omega_{ns} + \omega_{ns}^2} \tag{8.28}$$

that leads to a state-space vector of order $n = 2r + 2$.

Fig. 8.15. Signal-processing instrumentation composed by (a) DSpace system, (b) mixing console, (c) audio amplifier, (d) signal generator, (e) host PC, plus the oscilloscope

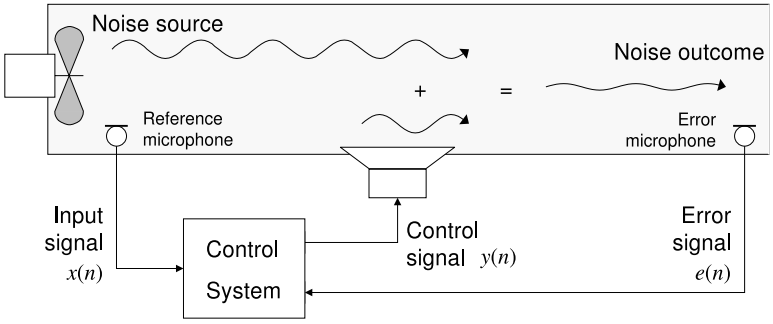


Fig. 8.16. Conceptual view of acoustic noise suppression

However, according to [18], the parameters of such a state-space model should be obtained by experimental identification. Therefore, in the next section a (control-oriented) identification technique fits a nominal model plus a frequency-dependent model uncertainty bound to the (noisy) frequency response with a worst-case criteria (usually the fitting is in a least-squares sense, see [18] and [4]). Furthermore, the previous mathematical model supports the fact that the nominal model should have parametric information based on second-order systems tuned

to the modal frequencies of the duct. In the approach described next, this is performed *via* a finite set of Kautz orthonormal bases, which also serves to keep the model order as low as possible ([31]).

8.3 Identification and Control Experiments

8.3.1 Objectives

In this section, a hybrid (FF/FB) controller is designed to cancel the acoustic noise in the tube described previously. The control system has a *filtered U* (FULMS) control structure as shown in Figure 8.12, which has been explained in greater detail in Section 8.1.3. This structure avoids the use of unstable filters for the cancelation of the nonminimum phase zeros of the secondary path, and compensates the acoustic feedback. The FF controller is based on the robustly normalized σ-algorithm and is adaptive in nature, and the FB is an $\mathcal{H}_\infty$ optimal controller.

The experiments are performed with three test inputs: the first one is a single tone of 105 Hz; the second one is a set of tones in the range $[95, 115]$ Hz that physically simulates an industrial fan, also called a *synthetic fan*. The third experiment is performed with the actual fan that has nonlinear and time-varying characteristics. The first two noise sources are produced by a signal generator connected to the primary speaker, as described in Section 8.2.

For both the FB controller and the FULMS structure, identified models of the secondary path are necessary. These are obtained in different ways depending on the experiment, by using the methodology described in Section 8.1.2. The two different identification approaches are based on different assumptions and explained next.

The first approach is focused on a *controlled LTI environment* (single tone and synthetic fan). The secondary path is identified with a standard multisinusoidal input, and separate designs for the FF and the FB are based on it. The hybrid controller is the combination of the FF and the FB obtained, and is tested with the hope that, if there is a close match between synthetic and real fans, it will work there as well.

The second approach is focused on the industrial fan, and is based on its nonlinear/time-varying characteristics. Using the previous secondary path, the FF is implemented and then the same secondary path is identified again, but with the actual signals in the loop. With this new model, the FB is designed and combined with the FF to produce the hybrid controller.

Therefore, the main objectives of this work are as follows:

Identification: The application of a robust (control-oriented) identification procedure to an ANC system with a parametric/nonparametric model structure. Two approaches are used to determine the applicability of an hybrid controller, based on two different identification steps.

Control: The design of an hybrid (FB/FF) controller for this application and a systematic procedure to justify its advantages, in terms of performance and robustness, with respect to FF or FB controllers.

Implementation: To explicitly point out the compromises and interplay between the identification and controller design for this ANC system. In addition, the feedback-loop limitations and the practical issues that arise are also pointed out.

8.3.2 Identification Approaches

The identification to design the hybrid controller for the single-tone and synthetic-fan experiments is performed with a multisinusoidal input, which is fed to the secondary speaker,

while the output is obtained from the error microphone. The identification method is the interpolatory robust one, based on a parametric/nonparametric model and on the experimental knowledge of low-frequency modes ([5, 31]), already explained in Section 8.1.2. The model computed is used as $\hat{G}_{sec}(z)$ in the FULMS control structure (see Figure 8.12) and as the nominal model $G_{sec}(z)$ in the FB design (see Figure 8.26). In the latter, also a worst-case deterministic multiplicative uncertainty bound $W_\delta(z)$ is obtained (see Section 8.1.2), which exactly fits the $\mathcal{H}_\infty$ optimal control method used to design the FB controller. This identification setup is similar to the one that generates the synthetic fan using the primary speaker. Therefore, due to the fact that the primary and secondary paths are essentially LTI, the FF and FB controllers can be designed separately, but connected together to form the hybrid one.

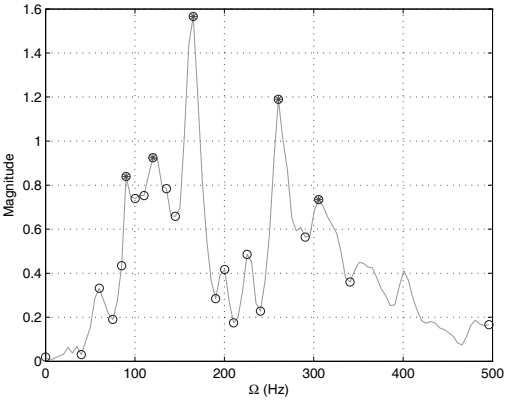


Fig. 8.17. Step 1: Interpolation points and peak frequencies

Instead, the identification to design the hybrid controller for the industrial-fan experiment is performed in two steps, as follows:

1. The same procedure as before is used to identify the secondary-path model $\hat{G}_{sec}(z)$ to be used in the FULMS control structure.
2. Once the FF control is designed with the FULMS structure, the secondary path is identified again, with the main acoustic noise source (fan) and the FF in place. Also in this case, not only is the nominal model obtained, but also a worst-case deterministic multiplicative uncertainty bound $W_\delta(z)$, both used in designing the FB controller.

The explanation for doing this consecutive identification is as follows. When including the actual fan in the experiment, a nonlinear/time-varying behavior in the setup is noticed. On the other hand, both the FF and FB need to work simultaneously, and the FF has its own real-time identification algorithm. The objective is to cancel a large part of the noise with the FF and add some extra performance and robustness through the FB controller, in this case in the same frequency range. Therefore, it seems more realistic to identify the model used to design the latter with the actual signals in the loop, *i.e.* with the noise source and the FF in place. Again, this is due to the nonlinear/time-varying nature of the physical setup that may cause

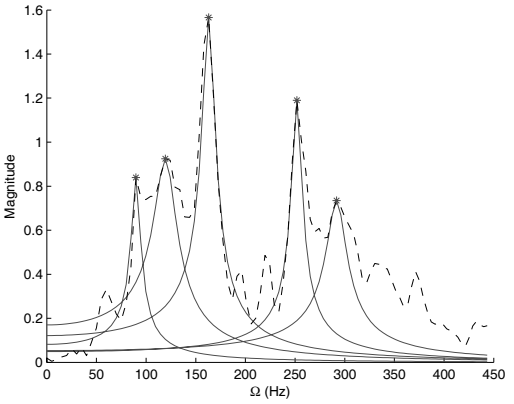


Fig. 8.18. Step 1: Second-order bases fitted to the experimental data

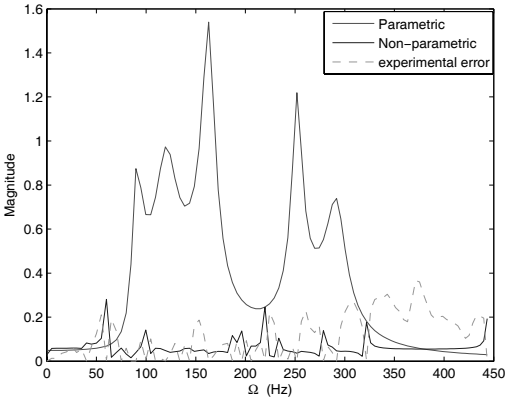


Fig. 8.19. Step 1: Parametric, dynamic and model error for experimental data

the model to change for different levels of the input signal. The input to the secondary path may be significantly lower when the FF is being used (there is already attenuation). Hence, the secondary path identified in this way could be more realistic, and consequently the FB controller that is based on it.

It could happen that the model obtained from this identification is *bad*, meaning that its relative uncertainty is above one for almost all frequencies of interest. Then, using the common (and controversial) adagio *"... let the data speak ..."*, we may conclude that the FB will not be able to add extra attenuation, besides the one already achieved by the FF. In this case, it is practical to use only the FF structure.

Finally, in identification step 2, the output of the primary path is also measured. This is useful to determine the weight $W_p(z)$, which represents the main frequencies to be attenuated when designing the FB $\mathcal{H}_\infty$ optimal controller (see Figure 8.25).

Identification Step 1

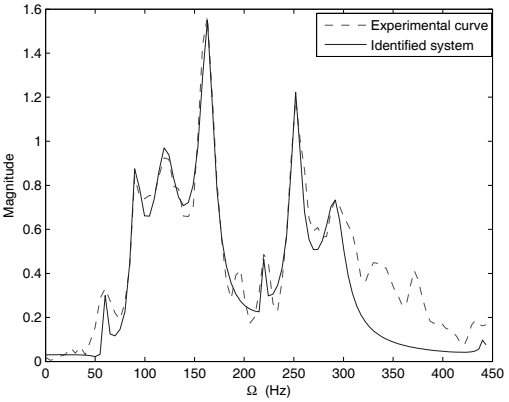


Fig. 8.20. Step 1: Identified model of experimental data

In the first step, using a set of sinusoidal inputs that cover the frequencies from 5 to 500 Hz on a 4-s experiment, the secondary path is identified. Figure 8.17 illustrates the 22 interpolation points used to identify the secondary channel of the acoustic setup presented in next section, which amounts to a nonparametric part of order 44. In addition, the five asterisks indicate the peaks that will be fitted by second order models (see Figure 8.18), which adds an extra 10th-order model. In Figure 8.19, the parametric and nonparametric components and model error of the acoustic system are presented, and in Figure 8.20 its complete identified model. The latter is a 16th-order model that has been obtained after balanced-truncation model reduction, based on the Hankel singular values. As explained previously, this model is used in all experiments as $\hat{G}_{sec}(z)$ in the FULMS control structure, and as the nominal model G_{sec} (see Figure 8.26) for the FB design in the single-tone and synthetic-fan experiments. In this last case, the multiplicative uncertainty weight $W_\delta(z)$ has also been computed, as illustrated in Figure 8.30 with the dashed curve.

Identification Step 2

The identified model in this case can be seen in Figures 8.21 to 8.23. Note the difference with similar figures for Step 1, especially the greater concentration of interpolation points in the relevant frequency range. This has to do with practical compromises, as will be explained in the last section. Also note that, in Step 2, these main frequencies have already been attenuated

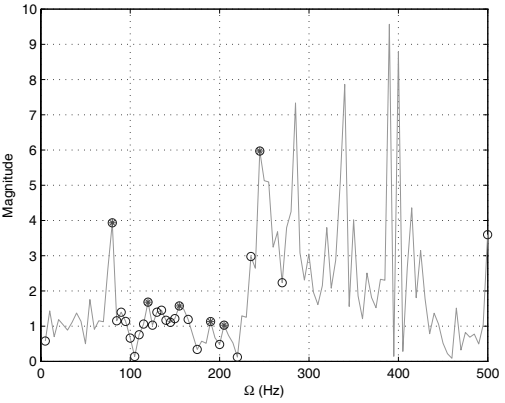


Fig. 8.21. Step 2: Interpolation points and peak frequencies

by the FF solution. From this procedure we also obtain the coverage of the relative identification error (8.19) that is used as the uncertainty weighting function $W_\delta(z)$ in the FB design, illustrated in Figure 8.24. Note the limited bandwidth where this weight is below one, when compared with the one in Figure 8.30. This will produce quite different performance results for the hybrid controller in both experiments, *i.e.* synthetic and industrial-fan inputs. Also, the achievable weight $W_p(z)$ is illustrated in Figure 8.27 and has been selected to attenuate the ideal frequency range, indicated in Figure 8.25.

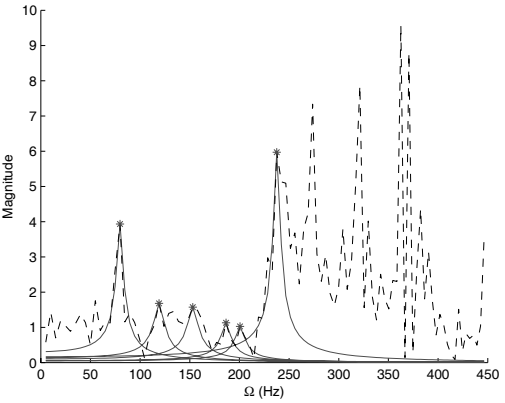


Fig. 8.22. Step 2: Second-order bases fitted to the experimental data

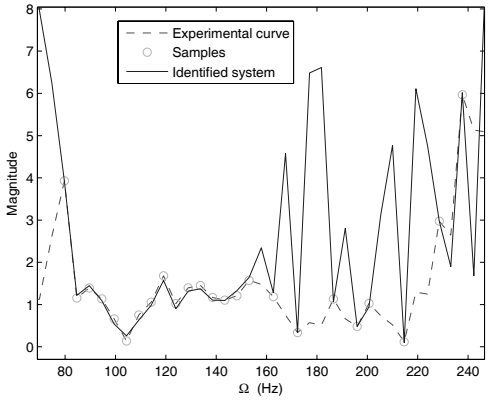


Fig. 8.23. Step 2: Identified model and experimental data

8.3.3 FB Controller Designs

The control objective is to minimize the acoustic-noise disturbance at the error microphone output, due to the noise source passing through the primary circuit, as illustrated by the block diagram in Figure 8.26. Therefore, the perturbations are a *set* of signals p in a certain frequency range, represented by the weight $W_p(z)$ in the same figure. If the energy of signal y at the error microphone has to be minimized, the approach from a worst-case perspective is to consider *all* weighted disturbances p in the set. Hence, nominal performance is defined as follows:

$$\min \|y\|_2 , \quad \forall \|p\|_2 < 1 \Longleftrightarrow \min_{i.s.K(z)} \|S(z)W_p(z)\|_\infty \tag{8.29}$$

where $S(z)$ is the sensitivity function, $\|\cdot\|_2$ represents the energy of the signal, and $i.s.$ stands for internally stabilizing. From Figure 8.26 it can be verified that the transfer function between p and y is precisely $S(z)W_p(z)$. The equivalent condition is simply a consequence of the induced norm in $\mathcal{L}_2$. This exactly fits the $\mathcal{H}_\infty$ optimal control procedure ([11, 35, 44]) applied to a set of perturbations p, whereas $\mathcal{H}_2$ optimal control ([34]) could have been used in the case of a single known perturbation.

In addition, a condition is added to take care of robustness, due to model uncertainty. Here, the system is represented by a global dynamic (multiplicative) set of models $\mathcal{G}$, and its equivalent condition for robust (internal) stability is as follows:

$$\begin{aligned} & i.s. \;\; K(z) \quad \forall\, G \in \mathcal{G} \stackrel{\text{def}}{=} \{[1 + W_\delta(z)\delta]\, G_{sec}(z),\ |\delta| < 1\} \\ \Longleftrightarrow\ & \|T(z)W_\delta(z)\|_\infty \leq 1 \end{aligned} \tag{8.30}$$

where $T(z)$ is the complementary sensitivity function and $W_\delta(z)$ weights the uncertainty among frequencies (see Figures 8.5, 8.24 and 8.26).

Performance in Equation (8.29) guaranteed for all models in the uncertainty set (8.30), defines robust performance, which in addition solves both conditions simultaneously with the same controller. The equivalent condition for robust performance is:

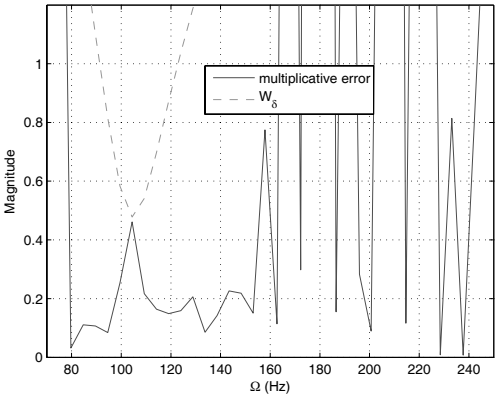


Fig. 8.24. Step 2: Relative identification error and multiplicative uncertainty weight for the industrial-fan experiment

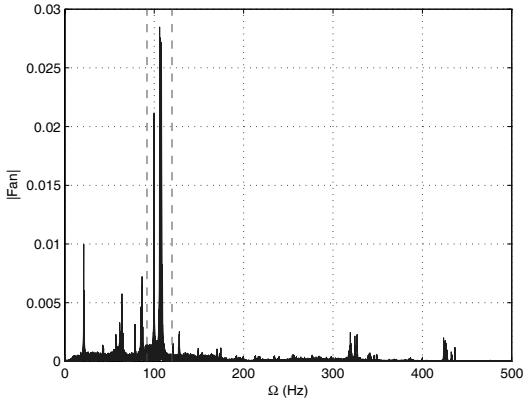


Fig. 8.25. Output of the primary path excited by the fan and ideal performance region (inside the dashed lines)

$$\begin{aligned} &\min \|y\|_2 , \quad \forall \|p\|_2 < 1 \quad \text{and} \quad G \in \mathcal{G} \\ \iff &|T(z)W_\delta(z)| + |S(z)W_p(z)| < 1, \quad \forall z = e^{\jmath \Omega T_s} \end{aligned} \tag{8.31}$$

This condition coincides with the μ (semi) norm for this SISO problem ([11, 35, 44]). In this case, instead of using μ-synthesis, the design has been solved as a mixed-sensitivity problem using $\mathcal{H}_\infty$ control,

$$\min_{i.s.K(z)} \left\| \begin{bmatrix} T(z)W_\delta(z) \\ S(z)W_p(z) \end{bmatrix} \right\|_\infty \tag{8.32}$$

This maintains the order of the controller limited to the order of the augmented plant's model (nominal model plus weights), which is important for the controller implementation.

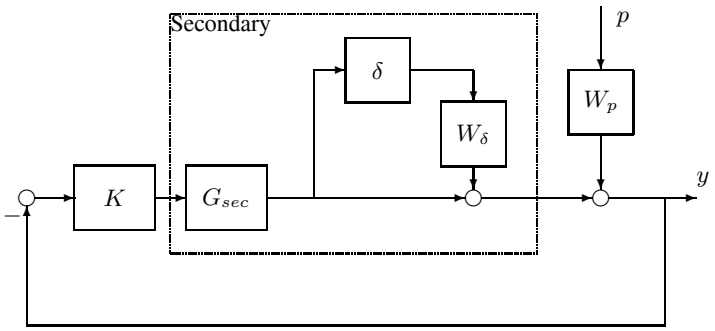


Fig. 8.26. Feedback (FB) design setup, with primary circuit perturbation (W_p) and secondary model with multiplicative uncertainty (W_δ)

Both the performance $W_p(z)$ and the robustness $W_\delta(z)$ weights are related to the perturbation and model identification data of the problem, respectively. The reasoning behind the weight selection is as follows.

The performance objective is to decrease the sensitivity of the system at the main frequencies of the disturbing (acoustic) signal, filtered by the primary path. The combination of primary circuit and noise-source response has the main peaks in the frequency range $[95, 115]$ Hz, both in the real and synthetic fan cases. Therefore, $W_p(z)$ has been selected so that it increases the performance at the ideal bandwidth, illustrated in Figure 8.25. The robustness weight $W_\delta(z)$ is computed so that it "covers" the relative model error-frequency response (see for example Figures 8.5 and 8.24). A tighter coverage of this modeling error should be achieved in the main range where acoustic noise needs to be attenuated. The reason for this is to improve performance in that frequency range, due to the relation between performance and robustness in Equation (8.31). In addition, the order of both weights needs to be limited due to the same fact pointed out before: they increase the order of the augmented plant and consequently, of the controller. For example, compare the actual weights W_δ in Figures 8.24 and 8.30 with an hypothetical better fitted weight from Figure 8.5, which clearly will have higher order.

Also, from a practical standpoint, the analysis of the robust performance condition (8.31) could be improved by replacing the uncertainty weight $W_\delta(z)$ by the actual frequency magnitude of the relative model error of Equation (8.19) (full line in Figure 8.29). This would provide a more realistic analysis condition, although in the design stage the weight $W_\delta(z)$ has to be used.

Another practical issue is actuator saturation, which could be considered in the design procedure. Again, in order not to increase unnecessarily the order of the controller, the actuator signal with the controller in place can be verified to be within limits, instead of adding an extra weight (at the controller output in Figure 8.26) to the design problem (8.32). This will also be mentioned as part of the experimental compromises in the last section.

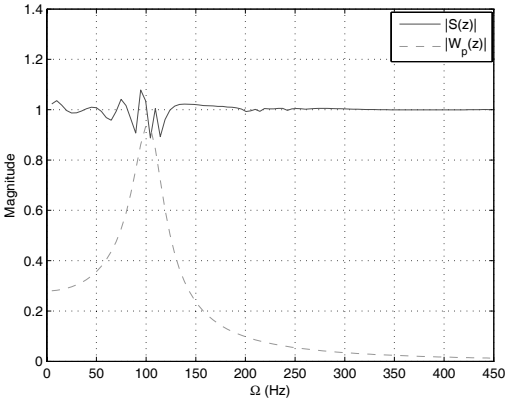


Fig. 8.27. Achievable performance weight and sensitivity function for the industrial-fan experiment

Design Cases

Two different designs have been made, according to the input noise source experiment. In the first case, for the synthetic-fan and single-tone experiments, only Step 1 identification has been used to design FF, FB and hybrid controllers. The uncertainty and performance weights, as well as the sensitivity, are illustrated in Figure 8.30. Note that $|W_p|$ is above one, and $|W_\delta|$ and the sensitivity magnitude are below one in the performance bandwidth $[95, 115]$ Hz. Therefore, these weights can accommodate reasonable performance and robustness results, as will be detailed in Section 8.3.4. Furthermore, the FB controller had order 12 and when combined with the FF (order 30) as a hybrid controller, was perfectly realizable with the actual DSP using $T_s = 0.4$ ms[4].

The second case is tuned for the industrial-fan experiment. The two-step identification has been performed, and the weights are shown in Figures 8.24 and 8.27. As indicated previously, in this case the uncertainty weight W_δ has a limited region where its magnitude is below one. In addition, to satisfy the robust performance condition (8.31), the achievable performance weight W_p magnifies the bandwidth $[95, 115]$ Hz, but is still below one. Note also that, in Figure 8.27, the sensitivity does not decrease significantly in the frequencies of interest, which means that there will be almost no noise attenuation. Therefore, it is reasonable to state that for this situation, the hybrid controller will not do a better job than the FF one. Besides, the robust performance condition was achieved with a FB controller of order 50, which combined with the FF control of order 30 cannot be implemented in this DSP. The analysis plots for this design are presented in Figures 8.27 to 8.29. In addition, note that when using the actual relative identification error in Figure 8.29, although robust performance is above one near 50 Hz, the design guarantees a *practical* robust stability condition.

The important point here is that the conclusions for both designs can be obtained solely by using the identification information, W_δ and W_p.

[4] Previously, a $T_s = 0.2$ ms was used, but it limited too much the order of the controllers due to implementation, particularly in the hybrid case.

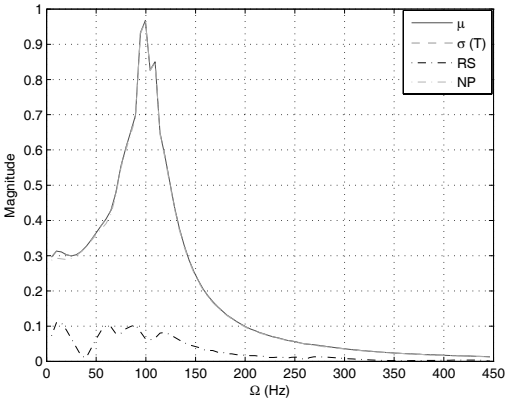


Fig. 8.28. Robust stability and nominal/robust performance conditions for the industrial-fan experiment

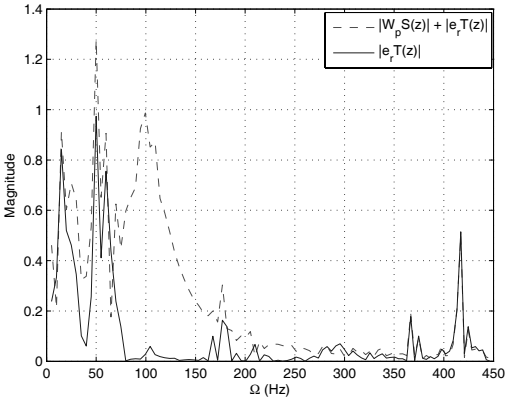


Fig. 8.29. Robust stability and performance experimental conditions using the relative identification error e_r for the industrial-fan experiment

8.3.4 Results

Three controllers have been designed: FF, FB and hybrid (FF/FB). These have been tested with three input noise sources (see Section 8.2.1):

- A speaker-generated single-frequency tone of $\Omega = 105$ Hz.
- A speaker-generated synthetic fan with bandwidth $\Omega \in [95, 115]$ Hz.
- The actual industrial fan.

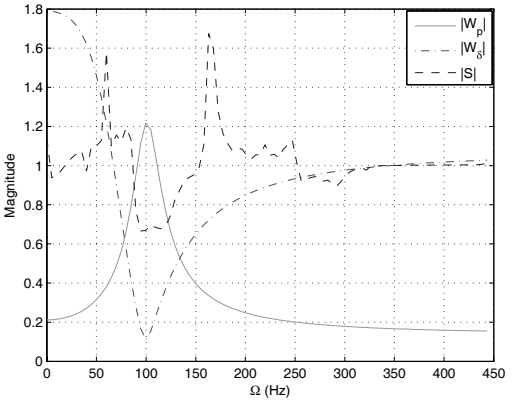


Fig. 8.30. Uncertainty (W_δ), performance (W_p) weights and sensitivity function S for the synthetic-fan experiment

For the first two inputs, only Step 1 identification of the secondary path has been used, as indicated in Section 8.3.2. The results are presented in Table 8.1. This first hybrid design was also tested with the real fan and shows a good agreement with the synthetic test (see values in the same Table). Figures 8.31, 8.32 and 8.33 compare the open-loop and closed-loop responses for the three different input noise sources, respectively. Figure 8.34 shows the attenuation attained by the hybrid (FF/FB) controller with the three different input signals. The attenuation values for the actual and the synthetic fan correspond to the performance bandwidth of interest.

A second hybrid design has been tuned to the actual fan input and hence, both identification steps have been applied. The analysis indicated that this controller would not perform better than the FF and, in addition the order (80) was too high to actually implement it in the DSP. Therefore, no results are shown in this case.

Table 8.1. Experimental characteristics and attenuation for different control structures

Structure	Identification	Order	Single tone	Synthetic fan	Industrial fan
FF	step 1	30	81.25 dB	9.56 dB	7.56 dB
FB	step 1	12	3.90 dB	4.15 dB	3.82 dB
Hybrid (1st)	step 1	30+12	82.36 dB	12.72 dB	10.1 dB

Some final comments on these results. The FF controller performs much better for a single tone. This seems less demanding for the adaptation algorithm than a set of different tones, for which the performance decreases significantly. Nevertheless, in this last case where a finite bandwidth is applied, the results are similar for both the synthetic and industrial fan.

The FB controller has clear performance limitations, mainly due to the nonminimum phase zeros, hence it does not perform as well as the FF. It has been designed based on a (conservative) set of models that represents the actual plant. This explains why this robust

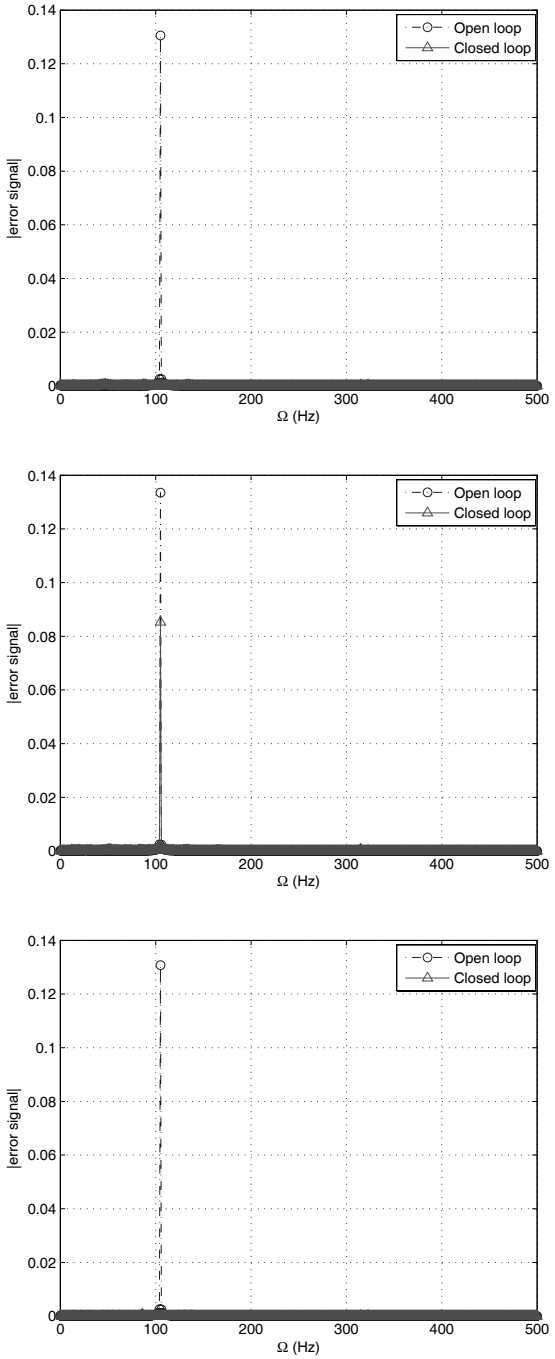


Fig. 8.31. FF, FB and hybrid controller attenuation in the case of a pure tone at $\Omega = 105$ Hz

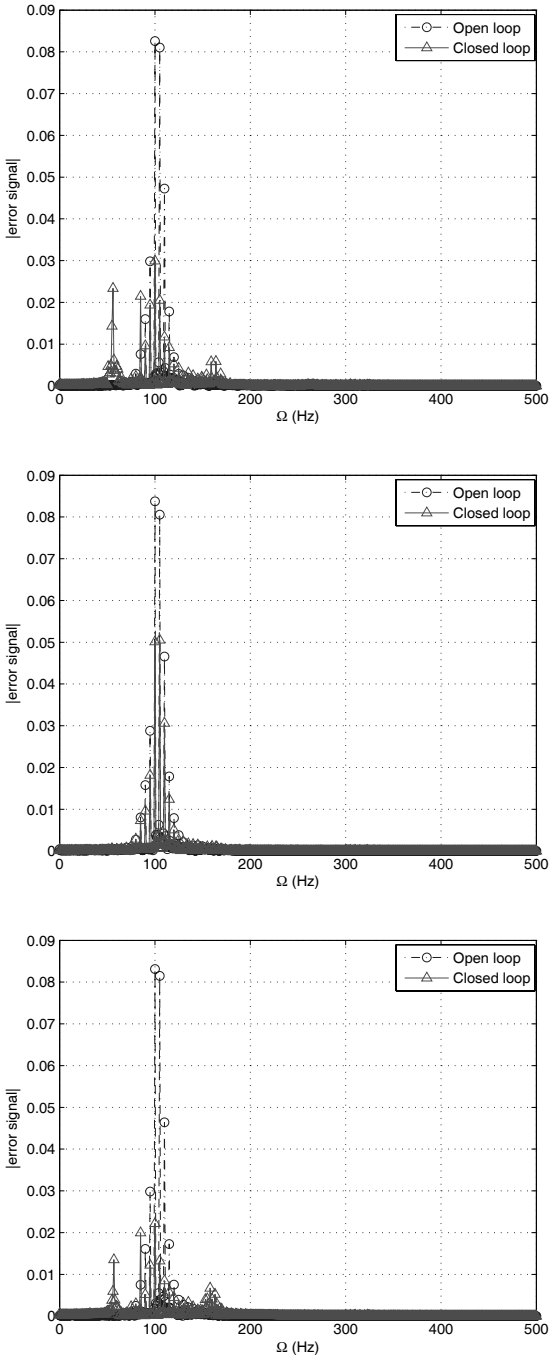


Fig. 8.32. FF, FB and hybrid controller attenuation in the case of the synthetic fan $\Omega \in [95, 115]$ Hz

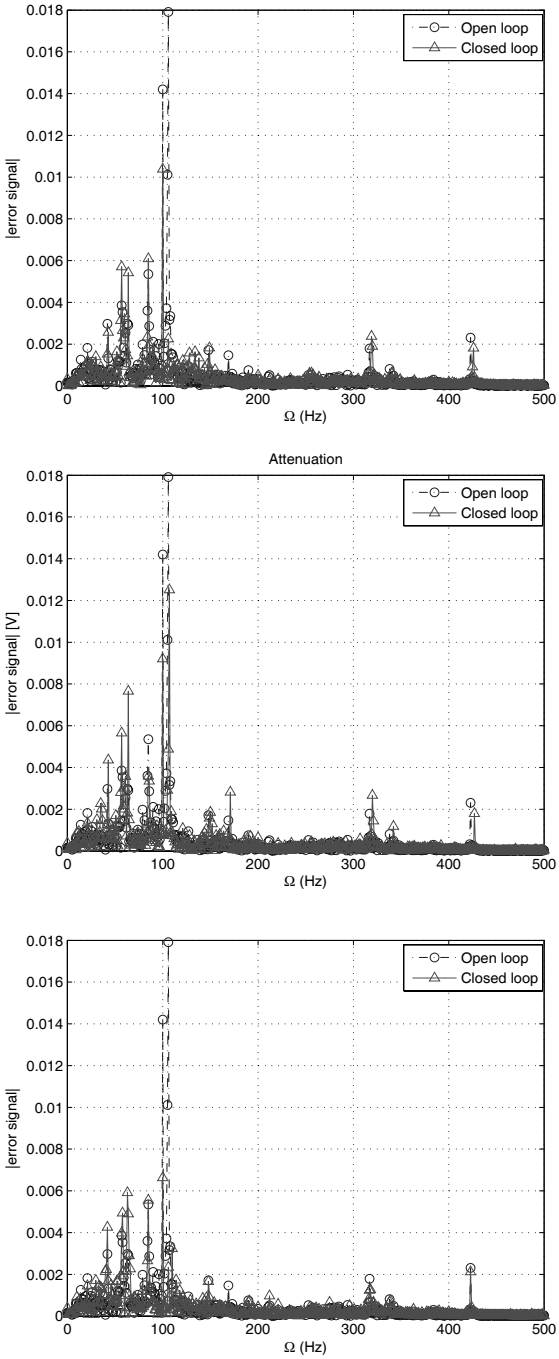


Fig. 8.33. FF, FB and hybrid controller attenuation in the case of the industrial fan

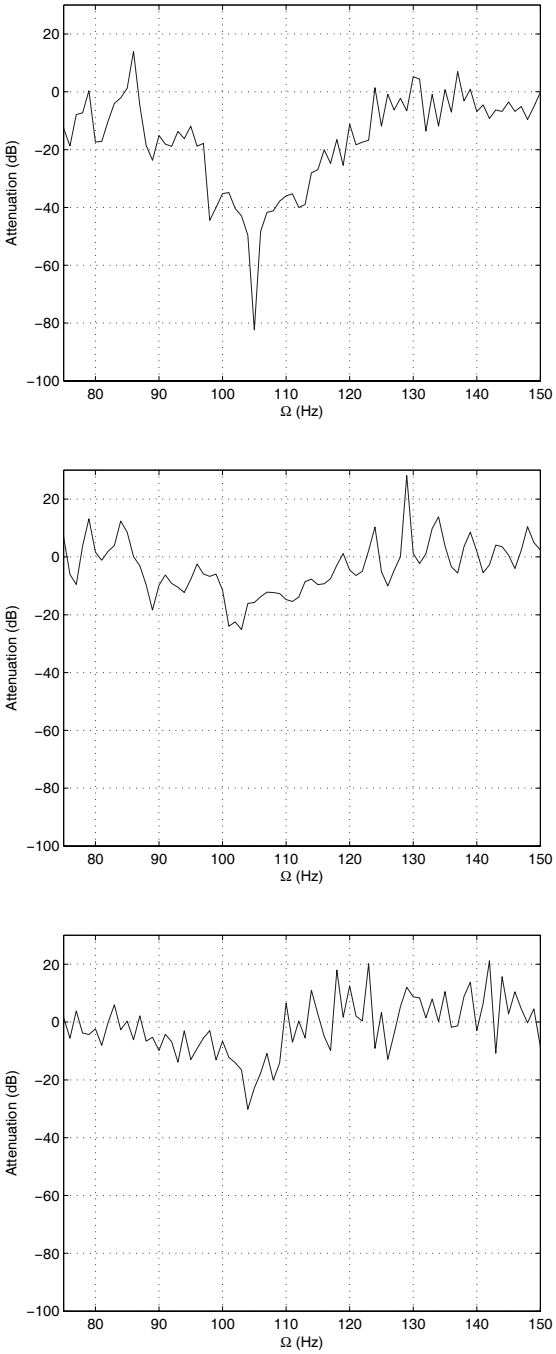


Fig. 8.34. Hybrid controller attenuation with the three different input signals

controller performs similarly for all inputs. Only with more information on the real fan, *i.e.* less uncertainty and a smaller model set, could the controller have performed better, probably with the extra cost of a higher order. Another step to increase performance would be to include the fan's nonlinear/time-varying characteristics in the model and uncertainty descriptions, and design a nonlinear controller.

In the same table, there is a good agreement in the attenuation produced by each controller separately, and combined together in a hybrid structure for all three inputs, *i.e.* the sum of both values add up to approximately the attenuation for the hybrid control. Although this design approach was tuned to the synthetic signal, *i.e.* not considering the two-step identification procedure described in Section 8.3.2, it performs very well when applied to the real fan, as indicated in Table 8.1. This result can be justified by the fact that model uncertainty covered both, synthetic and real fans, in the same set.

The methodology that uses a two-step identification procedure to design a hybrid controller is more conservative. It could be successful in certain situations where there is clearly a different behavior between real and synthetic inputs, due to nonlinearity and time variation. It seems particularly useful in applications where the FF reduces most of the broadband noise and the FB is designed to concentrate the attenuation in a certain (narrower) band. Clearly in this particular application the main perturbations are located already in a narrow band ($[95, 115]$ Hz) and the FF has eliminated most of it, hence the hybrid controller does not add any benefit. The model of the secondary path obtained from the 2nd identification step, is too uncertain at those frequencies and hence, no improvement can be achieved by adding a FB. An irrelevantly low attenuation would have been obtained (see Figure 8.27) using the resulting FB controller of order 50, which in fact could not be implemented in the DSP, combined with the 30th-order FF.

8.4 Compromises and Discrepancies between Theory and Practice

Practical Compromises

Besides maximizing performance and robustness, it is important to consider several practical constraints. They generate compromises in the controller design and possible identification iterations. These compromises arise from different sources: (i) feedback-loop constraints (see [14, 36], and in particular for ANC, see [17]), (ii) identification and control design interplay, (iii) implementation issues. These practical problems, which impose constraints in this application, have already been commented on this chapter, and are enumerated altogether next.

1. The nominal model of the secondary path has right half-plane zeros that limits performance in the FB case.
2. Frequency interpolation is used as an identification tool, therefore the model order is directly related to the number of interpolation points.
3. The usual robustness/performance compromise is a direct consequence in this case of Equation (8.31). This, in addition, forces an identification/control compromise, *i.e.* the multiplicative identification error should be lower than one in frequencies where noise attenuation (performance) is needed.
4. An additional problem is added related to the previous point. At certain frequencies where the magnitude of the model is small, the relative identification error increases; sometimes above unity.

5. Performance and robustness design weights increase the order of the augmented plant's model and therefore, of the controller.
6. The FB controller usually has poles very close to the unit circle in the design stage, which may lead to instability in the implementation, due to numerical issues.
7. The simplest way to represent the FB transfer function for controller implementation is numerator/denominator polynomials. This may lead to important pole distortion when it is implemented, specially with high-order polynomials.
8. Actuator saturation should be taken into account.
9. DSP implementation and sample time impose limitations on FF and FB controller orders.

Next, some solutions adopted in this application for the previous compromises are enumerated.

- Nonminimal phase models restrict performance in a well-known way. In fact they suffer from the *waterbed* effect, pointed out in [14, 36], which determines lower bounds in the size of the peaks of the sensitivity function magnitude $\left|S(e^{\jmath\Omega T_s})\right|$. It is clear from here that the lower the sensitivity will be in certain frequency bands, the higher it will increase in others. Hence, the performance weight $W_p(z)$ should reflect a decrease in the sensitivity **only** at frequency bands with the highest peaks of the error microphone output, *i.e.* in the range in $[95, 115]$ Hz. This frequency band, also called the "performance" range, determines the frequencies in which the sensitivity is below one, as indicated by Equation (8.29). The *waterbed* effect can be seen in Figure 8.30, where clearly the sensitivity is below one, and hence there is disturbance attenuation, only in the "performance" range $[95, 115]$ Hz. Instead, the controller will amplify signals with frequency content outside this range. This is not a problem when these signals have a small magnitude there, as illustrated in this case by the fan signal in Figure 8.25.
- From Equation (8.31), it is clear that only at frequencies where the multiplicative identification error is below one can robust performance be achieved. Therefore, $W_\delta(z)$ should be decreased as much as possible (while still bounding the relative identification error) *only* in frequencies where attenuation is needed, *i.e.* at the "performance" range pointed out previously. This has been solved by adding more interpolation points in this frequency range, keeping the total number of points as low as possible to avoid increasing the order (compare Figure 8.21 with Figure 8.17). This is a way of keeping track of the relation between robustness and performance, as well as the order of the controller at the identification stage, instead of only reducing the order of the controller at the end of the design.
- The order of the nominal model has also been decreased by eliminating interpolation points and concentrating them in the important "performance" bands, as mentioned before. In addition, frequency peaks that appear above 255 Hz in Figure 8.22, have not been fitted with Kautz bases for the same reason. If necessary at the end, a balanced model order-reduction step can be applied to the controller, based upon its Hankel singular values.
- The relative identification error e_r, defined in Equation (8.19) and applied to the nominal model of the secondary path $G_{sec}(z)$ in Figure 8.26, could be larger at frequencies where $\left|G_{sec}(e^{\jmath\Omega_k})\right|$ is small. To decrease it, either the nominal model can be changed locally or the noise set (and hence the identification error) can be made frequency dependent (see Section 8.1.2). In the latter case, the identification error bound ϵ_f^k is weighted to make it directly proportional to $\left|G_{sec}(e^{\jmath\Omega_k})\right|$.
- The orders of both weights, W_p and W_δ, have been kept as low as possible, while taking into account the performance and robustness features pointed out previously. For example, for the first design they have been chosen as follows:

$$W_p(z) = 0.1776\frac{z^2 - 1.464z + 0.538}{z^2 - 1.869z + 0.931} \tag{8.33}$$

$$W_\delta(z) = 0.978\frac{z^2 - 1.914z + 0.9756}{z^2 - 1.775z + 0.8086} \tag{8.34}$$

and illustrated in Figure 8.30.

- The FB controller has been finally implemented rescaling the magnitude of the poles closest to the unit circle, to avoid controller fragility. Clearly, this modifies the dynamics of the controller, therefore the analysis to assure robust performance must be rechecked. The analysis figures presented in this chapter were obtained using the implemented nonfragile FB controller.
- The previous controller was implemented using a series connection of second-order sections (SOS), which are numerically more efficient than the polynomial representation.
- Actuator saturation could be taken into account at the design stage by considering an extra weight at the controller's output. In this case, due to the fact that there was no clear saturation problem, the controller action was evaluated at the analysis and implementation stages, to avoid increasing the order of the augmented design model, and hence of the controller. According to the experimental results, the controller action was well within saturation limits.

Discrepancies between Theory and Practice

Active noise control is an active area of research and many different applications have been attempted, *e.g.* industrial and air-conditioning ducts, high-energy transformers, helmets, windows and airport surroundings. The main practical and commercial approaches consider linear models and adaptive FF solutions. There are clearly still many practical issues to be solved, for which more applied theory has to be developed. These issues increase when considering 3D environments like boxes, cabinets or even outdoor situations. A nonexhaustive list of the main issues that arise in this particular application, is presented next.

- Stability and robustness issues in adaptive FF controllers should be carefully considered. In practical situations, stability can be a potential (and fundamental) source of problems. Nonconservative nonlinear analysis tools should be used to solve this problem, so that stability and performance guarantees could be given.
- Control-oriented identification with a worst-case deterministic error bound that fits robust design methods can be very conservative in general. In the present case, the robust identification procedure used a local (experimental) identification error, instead of the global error. The more detailed the description of the system, the better performance the controller can obtain. Therefore, structured uncertainty, nonlinear and/or time-varying information could improve the system's description and hence the resulting performance. Very limited results have been developed for nonlinear and/or time-varying system identification with a worst-case deterministic error, *e.g.* [26, 38].
- High relative modeling errors (multiplicative uncertainty) in frequencies where the experimental data has smaller absolute values clearly limits robust stability and, as a consequence, the achievable robust performance at these frequencies. Control-oriented identification could consider this problem as part of the procedure to obtain the nominal model, as a way to minimize the uncertainty in certain frequency bandwidths.
- Controller order has to be limited due to its practical real-time implementation. Model order-reduction methods, as the ones based on Hankel singular values or using frequency-band weighting, are applied. At the different stages of the identification and controller

design, there are instances where controller order can be limited, *e.g.* selection and number of interpolation points, number of Kautz bases, order reduction of nominal model and weight-order selection. A clear analysis on how to decide on each instance could be very practical. In addition, design methods that consider the controller order as part of the formulation, would decrease the number of design iterations.

- Controller implementation is also an important issue, which could destroy the theoretical performance and robustness result. Numerical precision problems can arise, which produce a gap between the controller design and its implementation, derived in an internally fragile controller. Problems such as digital filter poles too close to the unit circle or pole distortion using numerator/denominator high-order polynomials, can be solved if they are taken into account at the design stage. To this end, numerically robust representations, *e.g.* series connection of SOS, can be used to implement the controllers in the DSP or microcomputer. Tools to take these numerical issues into account and avoid controller fragility, should be considered.

Future Research Issues

Finally, some issues that could be considered by interested researchers in this area are pointed out.

- Sensor and actuator allocation should be part of the identification and controller design, similarly to what has been done in aircraft design. This could provide a global optimization design environment and would certainly relieve some of the present limitations.
- A procedure to decide beforehand whether to use linear or nonlinear hybrid controllers, based on the desired performance, could be helpful. In some sense this would generalize the method followed in this chapter, in the case of hybrid *versus* FF controllers.

Acknowledgements

The authors wish to thank the support received by the Research Commission of the *Generalitat de Catalunya* (ref. 2005SGR00537), by the Spanish CICYT (ref. DPI2005-04722), and the last author by ICREA. We also appreciate the support provided by Prof. Jordi Romeu and his collaborators in the use of the acoustics laboratory.

References

[1] Akçay, H. and Ninness, B. (1995). *Rational basis functions for robust identification from frequency and time domain measurements*. Technical report, Royal Institute of Technology, Stockholm, Sweden.

[2] Anderson, B. (1985). Adaptive systems, lack of persistency of excitation and bursting phenomena. *Automatica* **21**(3), 247–58.

[3] Bai, M. and Lee, D. (1997). Comparison of active noise control structures in the presence of acoustical feedback by using the $\mathcal{H}_\infty$ synthesis technique. *Journal of Sound and Vibration* **206**(4), 453–471.

[4] Bai, M. and Lin, H. (1998). Plant uncertainty analysis in a duct active noise control problem by using the $\mathcal{H}_\infty$ theory. *Journal of Acoustical Society of America* **104**(1), 237–247.

[5] Baldelli, D., Mazzaro, M. C., and Sánchez Peña, R. S. (2001). Robust identification of lightly damped flexible structures by means of orthonormal bases. *IEEE Transactions on Control Systems Technology* **9**(5), 696–707.

[6] Ball, J., Gohberg, I., and Rodman, L. (1990). *Interpolation of Rational Matrix Functions,Operator Theory: Advances and Applications*, volume 45. Birkhäuser.

[7] Bodin, P., e Silva, T. O., and Wahlberg, B. (1995). *On the Construction of Orthonormal Basis Functions for System Identification*. Technical report, Royal Institute of Technology, Stockholm, Sweden.

[8] Boucher, C., Elliott, S., and Nelson, P. (1991). Effect of errors in the plant model on the performance of algorithms for adaptive feedforward control. *IEE Proceedings. F, Radar and Signal Processing* **138**(4), 313.

[9] Chen, J. and Gu, G. (2000). *Control-Oriented System Identification - An $\mathcal{H}_\infty$ Approach*. Wiley Interscience.

[10] Chen, J., Nett, C. N., and Fan, M. K. H. (1995). Worst-Case System Identification in $\mathcal{H}_\infty$: Validation of a priori Information, Essentially Optimal Algorithms, and Error Bounds. *IEEE Transactions on Automatic Control* **40**(7).

[11] Doyle, J. C., Francis, B., and Tannembaum, A. (1992). *Feedback Control Theory*. Maxwell/Macmillan.

[12] Dudley, N. F. and Partington, J. R. (1996). Robust identification in the disc algebra using rational wavelets and orthonormal basis functions. *International Journal of Control* **64**, 409–423.

[13] Feintuch, P., Bershad, N., and Lo, A. (1993). A frequency domain model for 'filtered' LMS algorithms-stability analysis, design, and elimination of the training mode. *IEEE Transactions on Signal Processing* **41**(4), 1518.

[14] Freudenberg, J. and Looze, D. (1985). Right half plane poles and zeros and design tradeoffs in feedback systems. *IEEE Transactions on Automatic Control* **30**(6), 555–565.

[15] Hakvoort, R. (1994). *System identification for robust process control*. Ph.D. thesis, Delft University.

[16] Helmicki, J., Jacobson, C., and Nett, C. (1991). Control Oriented System Identification: A worst case/deterministic approach in $\mathcal{H}_\infty$. *IEEE Transactions on Automatic Control* **36**(10).

[17] Hong, J. and Bernstein, D. (1998). Bode integral constraints, colocation, and spillover in active noise and vibration control. *IEEE Transactions on Control Systems Technology* **6**(1).

[18] Hong, J., *et al.* (1996). Modeling, identification and feedback control noise in an acoustic duct. *IEEE Transactions on Control Systems Technology* **4**(3), 283–291.

[19] Ioannou, P. and Kokotovic, P. (1984). Instability analysis and improvement of robustness of adaptive control. *Automatica* **20**(5), 583–94.

[20] Ioannou, P. and Sun, J. (1996). *Robust Adaptive Control*, chapter 8. Robust Adaptive Laws, pp. 530–633. Prentice Hall.

[21] Kaiser, O., Pietrzko, S., and Morari, M. (2003). Feedback control of sound transmission through a double glazed window. *Journal of Sound and Vibration* **263**(4), 775–795.

[22] Kalman, R. and DeClaris, N., editors (1970). *Aspects of Network and Systems Theory*, chapter "Adaptive Filters", pp. 563–587. Rinehart and Winston.

[23] Kuo, S. and Morgan, D. (1995). *Active Noise Control Systems: Algorithms and DSP Implementations*. John Wiley and Sons, New York.

[24] Lueg, P. (1934). *Process of silencing sound oscillations*. U.S. Patent 043,416.

[25] Mazzaro, M. C., Parrilo, P. A., and Sánchez Peña, R. S. (2004). Robust identification toolbox. *Latin American Applied Research* **34**(2), 91–100.

[26] Mazzaro, M. C. and Sznaier, M. (2004). Convex Necessary and Sufficient Conditions for Frequency Domain Model (In)Validation Under SLTV Structured Uncertainty. *IEEE Transactions on Automatic Control* **49**(10), 1683–92.

[27] Mullis, C. T. and Roberts, R. A. (1987). *Digital Signal Processing*. Addison-Wesley.

[28] Nelson, P. and Elliot, S. (1992). *Active Control of Sound*. Academic Press.

[29] Olson, H. and May, E. (1953). Electronic sound absorber. *Journal of Acoustical Society of America* **25**, 1130–1136.

[30] Ortega, R. and Tang, Y. (1989). Robustness of adaptive controllers-a survey. *Automatica* **25**(5), 651–77. ISSN 0005-1098.

[31] Parrilo, P. A., Sánchez Peña, R. S., and Sznaier, M. (1999). A parametric extension of mixed time/frequency robust identification. *IEEE Transactions on Automatic Control* **44**(2), 364–369.

[32] Parrilo, P. A., Sznaier, M., Sánchez Peña, R. S., and Inanc, T. (1998). Mixed time/frequency-domain based robust identification. *Automatica* **34**(11), 1375–1389.

[33] Partington, J. R. (1998). *Interpolation, Identification and Sampling*. Oxford Sciences Publications.

[34] Rafaely, B. and Elliot, S. (1999). H_2/H_∞ active control of sound in a headrest: Design and implementation. *IEEE Transactions on Control Systems Technology* **7**(1), 79–84.

[35] Sánchez Peña, R. S. and Sznaier, M. (1998). *Robust Systems Theory and Applications*. John Wiley & Sons, Inc.

[36] Serón, M., Braslavsky, J., and Goodwin, G. (1997). *Fundamental Limitations in Filtering and Control*. Springer.

[37] Snyder, S. and Hansen, C. (1994). Effect of transfer function estimation errors on the filtered-X LMS algorithm. *IEEE Transactions on Signal Processing* **42**(4), 950.

[38] Sznaier, M. and Mazzaro, M. C. (2003). An LMI Approach to Control-Oriented Identification and Model (In)Validation of LPV Systems. *IEEE Transactions on Automatic Control* **48**(9), 1619–24.

[39] Tøffner-Clausen, S. (1996). *Advances in Industrial Control*, chapter 10. System Identification and Robust Control: A case study approach, pp. 166–167. Springer-Verlag London Limited.

[40] Toochinda, V., Hollot, C., and Chait, Y. (2002). Disturbance attenuation in a SITO feedback control system. *American Control Conference*.

[41] Verboven, P., P-Guillaume, and Cauberghe, B. (2005). Multivariable frequency-response curve fitting with application to modal parameter estimation. *Automatica* **41**, 1773–1782.

[42] Wahlberg, B. (1994). System identification using Kautz models. *IEEE Transactions on Automatic Control* **39**(6), 1276–1281.

[43] Wu, J.-D. and Lee, T.-H. (2005). Application of $\mathcal{H}_\infty$ hybrid active controller for acoustic duct noise cancellation. *International Journal of Vehicle Noise and Vibration* **1**(3/4), 183–193.

[44] Zhou, K., Doyle, J. C., and Glover, K. (1996). *Robust and Optimal Control*. Prentice-Hall.

Part IV

Electromechanical

9

Iterative Identification and Control Design: Methodology and Applications

Pedro Albertos, Alicia Esparza, and Antonio Sala

Dept. Systems Engineering and Control, Universidad Politécnica de Valencia, P.O. Box. 22012, E-46071, Valencia, Spain. pedro@aii.upv.es, alespei@isa.upv.es, asala@isa.upv.es

Summary. When using model-based controller design methodologies in order to control a plant whose model is not *a priori* known, there are several practical issues to be taken into account. Theoretically, the procedure is to identify an accurate enough model of the plant and, then, apply any model-based controller design technique. But the *real world* is much more complex and issues as the presence of nonlinearities in the plant or/and in the actuators, the noise that always corrupts the data, the existing actuator constraints or computational limitations, *etc.* arise, imposing natural limitations over the achievable performance of the plant to be controlled as well as in the possible model to be experimentally obtained. In this contribution, an iterative framework has been used to overcome some of these problems. The proposed algorithm starts by using a very rough plant model and low-demanding specifications. These and the plant model are progressively improved only as needed, *i.e.* until the final desired performance is achieved or no improvements are attained in consecutive iterations. This procedure somehow avoids useless effort: on the one hand, the identification stage is only a tool for controller design, not aiming at achieving a very accurate model at any working condition but just at those frequencies that are interesting for control purposes; on the other hand, the bandwidth is increased as long as the plant allows it, avoiding undesirable experimental results trying to achieve what is not possible at all.

Key words: Iterative methods, iterative identification and control scheme, identification for control, model-based control, model-free control, low-order controller design

9.1 Background

9.1.1 Introduction and Motivation

When a designer faces the task of designing a controller for an unknown plant, he/she can chose among several available methodologies to be followed. One of the options is to use a model-based technique to design a controller on the basis of a previously identified model of the plant.

Usually, when using a model-based control design technique some assumptions are made:

- A perfect model of the plant is available. Otherwise, the uncertainty is also bounded and such a bound is known, as well as the uncertainty structure.

- The control requirements are completely stated. They could be expressed as either an index to be optimized or a set of behavioral parameters (defining performance bounds in the steady state and/or in the time or frequency domains) to be satisfied.
- There are many options to select the control structure and there are no constraints in the actuators.

But reality is rather different: when addressing the problem of designing a controller for a real plant, some issues are not at all completely defined and should be considered by the designer.

First, the task of the control system is to generate the appropriate plant inputs (control actions) to drive the plant into the required behavior, but:

1 Can the desired performance for the closed loop be achieved with a particular plant? We know that there are clear limitations in such a way that some (desired) performance goals cannot be achieved by any controller [35].
There are obvious power constraints, but there are also others regarding invertibility: for instance, if there is a pure delay in the output/input relationship in the plant, no feedback controller can counteract it (the physical controllers are causal).

In model-based control, the plant model is only used as a *tool* for controller design. This gives rise to what is usually called *identification for control*. This way of thinking realizes that the plant knowledge required to design the control in order to fulfill some goals very much depends on the goals themselves. Sometimes, just knowing the sign of the response may be enough to stabilize an unstable plant, by applying positive or negative feedback. Thus,

2 If the plant is available, what is a "good model" of it? That is, which are the plant properties to be captured that will allow the design of a controller achieving the desired closed-loop performance?
For instance, assume the control goal is to stabilize the temperature of a room. An on/off control (relay control), as is common in most home heating systems, is all that is needed, and a very rough model of the plant suffices. But, again, looking at the first of the issues under discussion (limitations), if the maximum capacity of the heat supply is not enough to warm up the plant (due to losses or unexpected values of disturbances, outside temperatures, for instance), no control system will achieve this simple goal.

And finally, if there is a mismatch between the actual plant and its model,

3 How can the designer be confident that the controller (computed from the plant model) achieves a closed-loop behavior in simulation close to the one achieved when operating on the real plant?
To illustrate this, let us set out an application example: Based on a perfect model of a mechanical system such as a CD drive, a controller can be designed. Initially, it works properly, but, after a number of hours of operation, the influence of temperature and wear will change the CD behavior. If this effect has not been taken into account in the plant model, the control performance will degrade. Furthermore, the same controller applied to another drive may also have a degraded performance.

All the above issues lead to a clear conclusion: to design a controller, the *control goals* should be achievable and the *plant model* should be appropriate. One way of addressing these matters is by using an *iterative* identification and control scheme. The approaches developed in this work try to give practical responses mainly to the first two previously discussed items.

Regarding the first one, if the final achievable performance is not *a priori* known, it seems reasonable to look for it in an iterative way, *i.e.* by first specifying a low-demanding performance, and by cautiously increasing it until no improvement is achieved in experiments.

The solution to the second issue can also be considered from the above iterative point of view: if the first desired performance is low demanding, the required model of the plant need not be excessively accurate. As long as the demands increase and the controllers designed upon the previously identified model do provide a good closed-loop behavior, the model is kept and used to synthesize new controllers. When the actually achieved closed-loop performance degrades, a new model is sought, tailoring the experiment in order for it to more accurately model the plant features that are important for the new closed-loop operation conditions (and that were not caught by the previous model).

The third item involves assessing model quality and robustness issues. In fact, the robustness can be introduced in this scheme by the *caution* in increasing the closed-loop desired performance[1].

9.1.2 High-order Model *versus* Low-order One

When using a model-based controller-design methodology, the identification step is crucial. Several questions arise about the experiment design, the model order and its structure. In theory, it could be advisable to identify a model as accurately as possible, in order to get a "close to perfect" plant model. But this decision usually carries lots of problems in practice.

At this point, it is worth recalling that the model is only used as a tool for controller design. So, why spend great effort in identifying a highly accurate and complex model if a simpler one could suffice for a particular set of control purposes? The plant model should be identified in accordance with the closed-loop specifications. That is, the *identification* is carried out *for control* purposes. This, of course, gives rise to the term identification for control, which has been the subject of much research during the 1990s [12, 13, 34, 36]. The conclusion is that what really matters in identification for control is that the achieved closed loop behaves as similar as possible to the designed one. It is not so significant to have a good fit between the real plant G and its identified model $\hat{G}$ when measured through open-loop properties such as step response, settling time, frequency response, *etc*.

To illustrate this statement, the following example is proposed. For a plant to be controlled, a 6th-order model G is assumed. The transfer function is given by:

$$G(s) = \frac{35.6(s^2 + 6s + 153)(s^2 + 8s + 500)}{(s^2 + 16s + 65)(s^2 + 4s + 104)(s^2 + 3.4s + 402.9)} \tag{9.1}$$

Now, consider a second-order transfer function, $\hat{G}$, such as:

$$\hat{G}(s) = \frac{34.79}{s^2 + 6.45s + 120} \tag{9.2}$$

Both, G and $\hat{G}$, are very different in open-loop step or frequency response. But in closed loop, with a 2nd-order controller $C(s) = \frac{7186(s^2+6.45s+120)}{s(s+1000)}$ designed to operate achieving a

[1] In order to provide simple and practical methods from an engineering point of view, we did not incorporate features such as "model error modeling" to obtain uncertainty bounds for validation; the reader is referred to [15–17] for details on identification of uncertainty models for controller validation; an alternative validation methodology based in "virtual" closed loops is also discussed in [32].

bandwidth of 500 rad/s, they have a very similar behavior. Figure 9.1 shows the Bode diagram of the open and closed loop of both G and $\hat{G}$. As can be observed, the frequency response of both models in closed loop are almost indistinguishable, whereas the open loop ones differ greatly. The trick, of course, is that they differ at frequencies for which the controller offers good robustness.

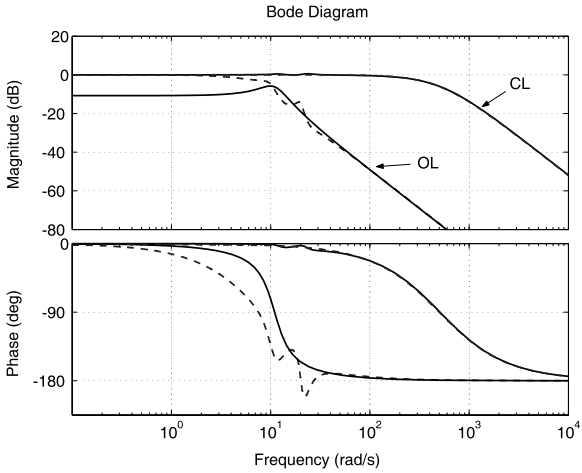


Fig. 9.1. Frequency response of both G (dashed line) and $\hat{G}$ (solid line) in open (OL) and closed (CL) loop

9.1.3 Iterative Identification and Control Scheme. Fundamentals

As previously mentioned, a reasonable solution to the controller design task for an unknown plant is to identify an accurate enough plant model and subsequently use any model-based controller-design technique. Both phases can be performed in an iterative framework, [3, 13], and this is the proposed approach in this work.

The problem of designing a controller for a plant whose initial model is not well known, and where the main goal is the maximization of a performance index, can be approached following an iterative scheme, such as:

> Given a plant model, design a controller improving the best performance achieved with older controllers, validate it against constraints and possible mismodeling, test on the real plant and use the new data to get a better model. Repeat until performance objectives are achieved.

Specifications are updated, for a given model, as the iterations proceed, as far as there is no constraint violation in the actual controlled plant. Under the current operating conditions, that is, on the limit of the constraints, new data should be gathered from the process and a more accurate model should be estimated.

The process ends if either there is no option to improve the model (because, for instance: (a) the signal-to-noise ratio is very low, (b) the model class cannot capture relevant details,

such as plant nonlinearity with linear models) or the last designed controller fulfills the requirements (or hits limiting constraints).

Two classical problems can illustrate this situation:

- Most mechanical systems present resonance phenomena, but the resonance only appears at high frequencies that are only excited if the required controlled-plant bandwidth is large enough. Thus, on the one hand, a plant model able to cope with the resonance can only be obtained from data gathered from the plant being excited in this range of frequencies. On the other hand, if the required bandwidth is not so large, the resonance does not need to be captured in the plant model.
- Given an unstable system there is no way to estimate its behavior around a desired equilibrium point if the system is not first stabilized. Once a stable system is achieved, experiments can be designed to estimate its behavior for small deviations around this equilibrium point. A linear/nonlinear (*rough*) controller is first designed and then a fine controller can be designed afterwards for better performances around the final target. Here, what matters is the signal's size. For instance, to just control the full opening/closing position of a piston stroke (an integrator), the friction and stickiness close to the final position are not relevant, and it can be controlled by an on/off controller. If the final position is required with a higher accuracy, now, some experiments can be carried out to determine the stroke behavior close to the equilibrium, for small displacements and reduced speed.

The decoupling of the identification (ID) and control design activities has been studied in different frameworks during the 1990s [14, 25, 27]. This methodology is rather general and different choices of identification, analysis, control design, and iteration algorithms may be applied.

The approach followed in this work is related to the features of the frequency-domain response of the plant. Based on the theoretical models of the resonant processes, the attention being focused on a simple set of such algorithms, in order to explore the main issues of this approach. Given an initial model of a plant, which is accurate only in a low-frequency range (it could be just a steady-state gain), the goal involves iteratively designing a controller that stabilizes the plant while increasing the closed-loop bandwidth (so reducing the settling time as much as possible). Low-complexity controllers (arising from low-order models) are sought.

Two iterative approaches have been simulated and also experimentally tested:

1. The first one is model based. Typical step responses, signal power spectrum analysis and/or input/output correlation by using PRBS signals are used to estimate the additional modes. Pole placement or just pure cancelation controllers are used to improve the designed controlled plant performance.
2. The second one is a model-free iterative version of virtual reference feedback tuning (VRFT) [7, 30, 32], where the controller is directly identified from filtered input–output data according to a desired performance M and some frequency filters.

Of course, other more sophisticated identification and/or control algorithms can be foreseen. For instance, in [4] a GPC controller with reduced weight in the control action to get a faster response is designed. A progressive reduction of the performance index can be achieved, as far as the improvement on the process model allows for stronger control actions.

9.1.4 Iterative Algorithm

The fundamental ideas of the iterative scheme, as used in this work, are detailed in [31]. Applications of particular variations of the iterative scheme can be found in [1, 2, 4].

The general algorithm can be summarized in the following steps. Given a real plant, determine the input/output pair of variables. Estimate the physical constraints based either on device limitations (such as saturations) or control requirements (overshoot, time response, ...). Then:

Step 0 Identify an initial model G_0 (it could suffice with just the static gain or the number of integrators) approximately valid at low-frequency ranges.

Step i Based on the current process model, G_i:

1. Design appropriate controllers by successively increasing the desired performance.
2. Before implementing each new controller, assess the closed-loop's constraints.
3. If the achieved performance is fully satisfactory, the design is complete. If the achieved performance differs too much from the designed goals, or if the real closed loop violates some constraints, then go to the next step.
4. At this step a new model G_{i+1} is identified. If G_{i+1} is relevant, the step i+1 is carried out. Otherwise stop.

Stop When either the achieved performance does not improve by using new plant models or the final requirements are met, the algorithm is stopped and the controller providing the last (best) performance is chosen.

In the following, some experimental applications will be presented so as to assess the capabilities of the proposed iterative scheme. First, in Section 9.2, the systems to be controlled will be described. In Section 9.3, the experimental designs will be exposed, helped in some cases by simulations of the closed loop to be implemented. This chapter will end with some comments on the attained experimental results.

The above-described iterative scheme will be refined/modified according to the particular requirements and techniques used in each of the two applications.

9.2 Description of the Applications

The above-mentioned iterative scheme has been applied to several mechanical systems, with different characteristics. Two illustrative systems are analyzed: a flexible transmission system and an active suspension one. Both have a common feature: they present resonant modes. This is a very typical situation that can be found in any flexible structure, like the one used in Chapter 6. Resonant modes are also present in other kind of systems, such as the acoustic noise model identified in Chapter 8.

Flexible Transmission System. A scheme of the flexible transmission system to be used for applying particular iterative schemes is depicted in Figure 9.2. It consists of three pulleys joined by two flexible bands. The movement of the whole system is produced by the rotation of a DC motor, which is located next to the first pulley and produces its axis movement. This movement is transmitted to the other pulleys by the flexible linking bands. The DC motor has a servo controller whose dynamics is very fast compared to those of the pulleys. The objective of the whole system is to control the angular position of the third pulley.

The control of the system is implemented by a computer, which communicates with the transmission system by means of an input–output acquisition board, its input being connected to the position transducer, and its output to the servo of the DC motor. The software used either to gather data or to apply the control action to the system is the Real Time Toolbox of MATLAB®. The sampling frequency considered for the actual experiments was 20 Hz.

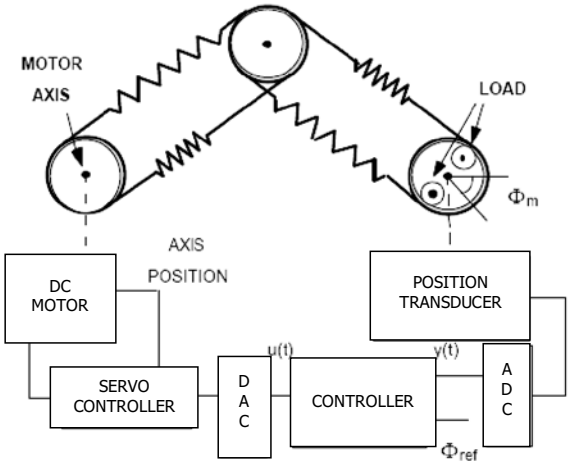


Fig. 9.2. Flexible transmission system

Active Suspension System. The active suspension system used for this contribution is shown in Figure 9.3 and located at the Laboratoire d'Automatique de Grenoble (INPG/CNRS)[2]. Its objective is to attenuate disturbances in a large frequency band. Its structure is depicted in Figure 9.4. The principal elements of this system are:

- An elastomer cone that encloses the main chamber filled with silicone oil (1).
- An inertia chamber enclosed with a flexible membrane (2).
- A piston (3) that is fixed on a DC motor; when the position of the piston is fixed, the suspension system is passive.
- An orifice (4) that allows oil flow between two chambers.

The main idea of the active suspension is to change the elasticity of the system in order to absorb the vibrations generated by the machine that it is expected to isolate. For the experimental purposes the machine is replaced by a shaker that is driven by a computer-generated control signal. The output of the system is the measured voltage corresponding to the residual force. The control input drives the position of the piston *via* an actuator. The system is controlled by a PC *via* a data-acquisition system. The sampling frequency is 800 Hz.

The block diagram of the active suspension system is depicted in Figure 9.5. The transfer function between the excitation of the shaker and the residual force is called the *primary path*. The *secondary path* is defined as the transfer function between the control input and the residual force.

The above-described systems are used as a benchmark for applying the iterative ID and control design scheme presented in Section 9.1. Different particular ID and controller design methodologies are used with each of them, to be detailed later.

[2] The authors would like to thank the staff of Laboratoire d'Automatique de Grenoble, and particularly Professor Ioan-Doré Landau, for their kindness and their help in allowing the execution of our experiments.

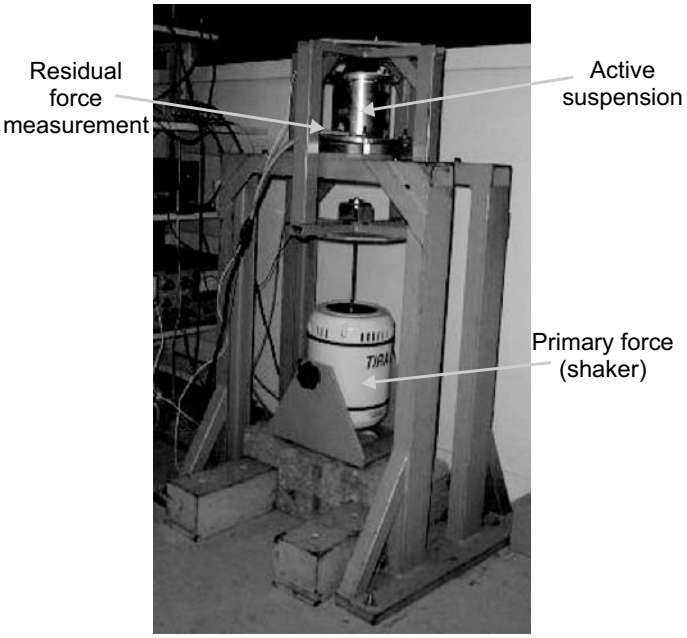


Fig. 9.3. Active suspension system, Laboratoire d'Automatique de Grenoble (INPG/CNRS)

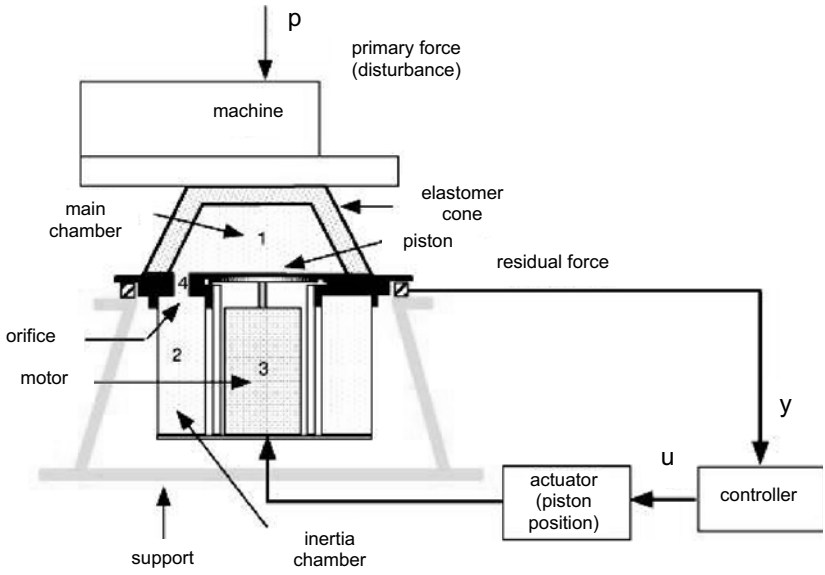


Fig. 9.4. Scheme of the active suspension system

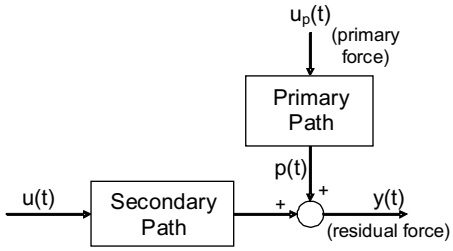


Fig. 9.5. Block diagram of the active suspension system

9.3 Simulations and Experimental Examples

Aiming at pointing out the existing differences between theory and "*real world*", the controller-design methodologies are first simulated using a high-order plant model, and will be applied to the real system later.

9.3.1 Iterative Controller Design for a Flexible Transmission System

In this section, the iterative methodology outlined in Section 9.1 will be used to design a low-order controller for a very flexible transmission system. The description of such a system is provided in Section 9.2.

As already mentioned, the iterative identification and controller design scheme can be applied using different identification algorithms and controller-design methodologies. In this particular case, two variations of such an iterative scheme are applied:

- The first application uses the iterative scheme presented in [31]. For this particular application, output-error (OE) algorithms are used in the identification step and the controllers are designed to match a reference model by cancelation (*i.e.* for a plant G, given a target closed-loop behavior M, the controller is $M(1-M)^{-1}G^{-1}$).
- The second one entails using in an iterative way the VRFT methodology, presented and extended in several contributions, such as [7, 8, 30, 32, 33]. The original methodology is a *one-shot* (*i.e.* requiring just one batch of data) controller-design technique. In order to apply it, it is necessary to select the desired reference model, as in the cancelation case. As argued in Section 9.1, the absence of knowledge about the final achievable bandwidth causes the iterations to appear.

As the final purpose of this chapter is to illustrate the differences between theoretical results and experimental ones, both applications will be previously simulated and finally carried out on the real plant. In the simulations, the following 8th-order model, G, which approximately fits most of the collected experimental data in prior experiments, will be assumed:

$$G(z) = \frac{0.025z^7+0.39z^6+1.38z^5+1.76z^4+0.92z^3+0.21z^2+0.042z-0.0015}{z^8+0.8z^7+0.32z^6+0.3z^5+0.17z^4+0.5z^3+1.08z^2+0.4z+0.102} \tag{9.3}$$

The sampling rate was 20 Hz, the same as the one programmed in the experimental setup.

Low-order Controller Design by Iterative ID and Control Design

The first application involves using the iterative algorithm proposed in [31] and summarized in Section 9.1.4, which can be detailed as:

1. Identify a first plant model G_0 at low frequencies (*e.g.*, a gain).
2. Design a controller with low-demanding goals. If the achieved behavior is similar to the desired one, slightly increase the desired bandwidth. If that is not the case, go to the next step.
3. Identify a new model G_i, using the latest data gathered from the plant, around the bandwidth that caused problems at the previous step (filtering input and output signals through a filter W with the required bandpass frequency characteristics).
4. Design a controller using a reference model featuring this bandwidth.
5. Before implementing any controller, apply a *robust-stability validation test* (testing for robustness using the previous models, which were accurate at lower frequencies).
   If the validation test is successfully passed, implement the controller and go to step **6**. If that is not the case, change the ID criterion (increasing low-frequency fit requirements) and reidentify a new model, going back to step **3**.
6. Compare the achieved response with the desired one. If they are similar, go to step **7**. If the misfit between them is not acceptable, then go to step **3** and increase the model order before carrying out the identification.
7. *Cautiously* increase the desired bandwidth and go to step **4**.

End The algorithm ends due to one of the following three reasons:

- When satisfactory performance is reached.
- When significant constraints' violation is unadvoidable.
- When further steps of model-order increments do not yield an improvement on the achieved performance.

Simulated Procedure

The model used as the "true plant" is the one in expression (9.3), using a sampling rate of 20 Hz. All the models are identified using data collected from the (noiseless) simulated plant in open loop.

For the first model G_0, a step is introduced to the simulated plant as the input, recording its output, and the system's gain is identified to obtain the model:

$$G_0(z) = 1.012 \tag{9.4}$$

With G_0, the controllers are designed from a reference model M_{0i} defined as:

$$M_{0i}(z) = \frac{1 - e^{-\lambda_{0i} T_s}}{1 - e^{-\lambda_{0i} T_s} z^{-1}}, \tag{9.5}$$

where λ_{0i} is the desired bandwidth and T_s is the sampling period, that is, 0.05 s.

The controllers are designed by cancelation, *i.e.*:

$$C_{ij}(z) = \frac{1}{G_i(z)} \frac{M_{ij}(z)}{1 - M_{ij}(z)} \tag{9.6}$$

In this way, two iterations were carried out, for bandwidths of $\lambda_{01} = 0.1$ and $\lambda_{02} = 0.2$ rad s^{-1}, obtaining the controllers C_{01} and C_{02}, respectively. As can be observed in Figure 9.6,

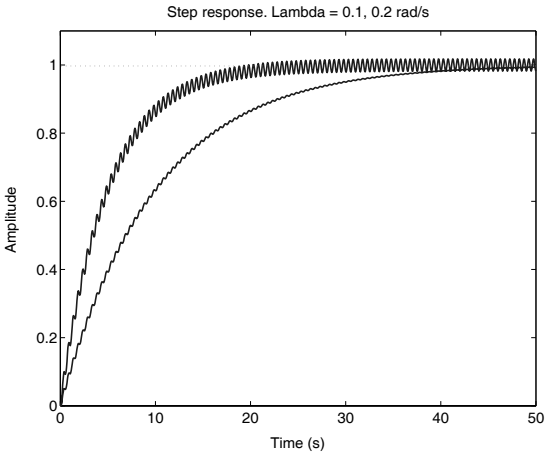


Fig. 9.6. Iterations carried out with G_0 for $\lambda_{0i} = 0.1$ and 0.2 rad s^{-1}

the second step response shows an evident oscillation, which is unacceptable and entails the need of identifying a new model of the plant, that is, G_1 that captures such oscillations arising from a resonant mode of the plant, up to now unmodeled by G_0.

For the new step, the identification experiment is done using two pseudorandom binary sequences (PRBS) of amplitude ± 1, generated by a 7-cell shift register and a frequency divisor of 4 (see [21] for more details). These two sequences are applied in a consecutive way, obtaining an input/output record of 1024 samples. The step responses in Figure 9.6 reveal that the resonant mode is around 12 rad s^{-1}, so the new model is identified by filtering the data through a passband prefilter with cutoff frequencies of 6 and 12.6 rad s^{-1}. Using an OE structure, this new model is:

$$G_1(z) = \frac{0.2274z + 0.21}{z^2 - 1.61z + 0.991} \tag{9.7}$$

The reference model used at this step is the following one:

$$M_{1i}(z) = \frac{(1 - e^{-\lambda_{1i}T_s})^2 z^{-2}}{(1 - e^{-\lambda_{1i}T_s} z^{-1})^2} \tag{9.8}$$

Using G_1 and $\lambda_{1i} = 1$, 2, 5, 10 and 15 rad s^{-1}, the controllers $\{C_{11}, \ldots, C_{15}\}$ are designed, respectively, yielding a step response depicted in Figure 9.7. The oscillating behavior when using C_{15} involves the need of a new plant model, if the bandwidth has to be increased. For information, the control action has also been plotted in Figure 9.7.

If this is the case, another step is carried out. This time, a 4th-order plant model is needed in order to catch the dynamics of the system that are important for closed-loop operation. This new model is G_2, defined as:

$$G_2(z) = \frac{0.31z^2 + 0.63z - 0.19}{z^4 - 1.34z^3 + 1.51z^2 - 1.26z + 0.9414} \tag{9.9}$$

Now, the reference model is defined by:

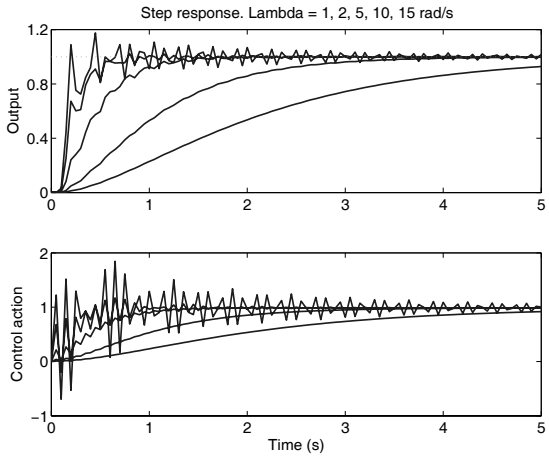


Fig. 9.7. Step response using G_1 for $\lambda_{1i} = 1$, 2, 5, 10 and 15 rad s^{-1}

$$M_{2i}(z) = \frac{(1-\beta z^{-1})(1-e^{-\lambda_{2i}T_s})^4 z^{-3}}{(1-\beta)(1-e^{-\lambda_{2i}T_s}z^{-1})^4} \tag{9.10}$$

where $\beta = -2.3$ is the nonminimum-phase zero of G_2, added to the reference model to avoid being canceled by the controller (which would have given rise to an internally unstable system).

Using G_2, several controllers were designed for increasing bandwidths (15, 30 and 60 rad s^{-1}), the closed-loop system responses being depicted in Figure 9.8.

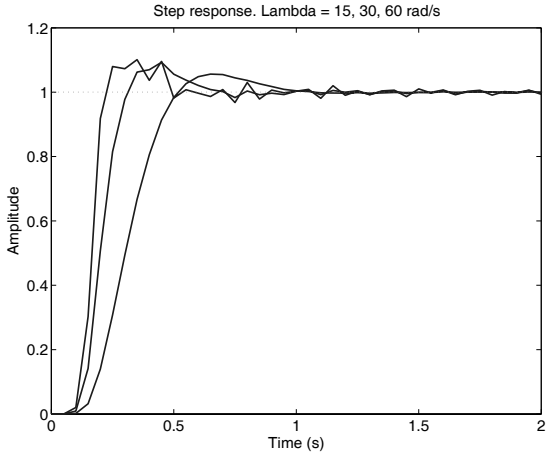


Fig. 9.8. Iterations carried out using G_2 for $\lambda_{2i} = 15$, 30 and 60 rad s^{-1}

To summarize, the last controllers being designed at each step are the following ones:

$$C_{02}(z) = \frac{0.01z}{z-1} \tag{9.11}$$

$$C_{15}(z) = \frac{1.224z^2 - 1.971z + 1.213}{z^3 - 0.03874z^2 - 0.9112z - 0.05} \tag{9.12}$$

$$C_{23}(z) = \frac{0.8z^4 - 1.068z^3 + 1.204z^2 - 1.011z + 0.7515}{z^5 - 0.468z^4 + 0.0684z^3 - 0.2512z^2 - 0.5021z + 0.1527} \tag{9.13}$$

The procedure does not go further, because the closed loop is approaching the Nyquist frequency (63 rad s^{-1}), so the effort spent at the identification step is not worth it just to slightly increase the final closed-loop bandwidth: the performance is deemed satisfactory at this stage.

From this simulated example, an important conclusion can be drawn: The (simulated) plant to be controlled is an 8th-order one (expression (9.3)). However, for low bandwidths just a gain or a second-order model suffices (expressions (9.4) and (9.7), respectively). If the specifications of the closed-loop increase, a fourth-order model will be enough (expression (9.9)). As a conclusion drawn from this simulated example, we could say that it is not worth spending too much effort at the identification step aiming at achieving a highly complex model if no information about the plant and/or its achievable bandwidth is available. Figure 9.9 shows the frequency response of the system G and its consecutive low-order models G_0, G_1 and G_2.

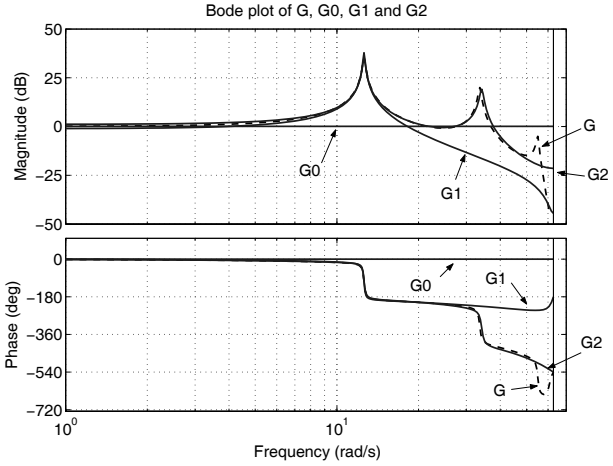


Fig. 9.9. Frequency response of the original plant G (dashed line) and its models G_0, G_1 and G_2 (solid lines)

Another conclusion from this example is the interest of using the sampling rate as a design parameter (not tested): the design of low-performance controllers does not require fast sampling, and possibly, the high-frequency oscillations appearing in, say, Figure 9.6 would be less manifest if the sampling rate had been lower than 2 Hz. Note, indeed, that the additional resonances in the full-order plant (Equation (9.3)) are not excited in Figure 9.8. Increasing the performance (if actuator power were available) would require both a more complex model and a faster sampling.

Experimental Application

Once the proposed iterative scheme under simulation is applied, it is time to put it into practice in an experimental setup. This was carried out using the system described in Section 9.2. The algorithm applied was the same as the one used for simulations.

The controlled plant, when using the controller being designed based on the simplest static model (gain close to 1) exhibits markedly oscillatory behavior with a frequency of about 2 Hz (12.6 rad s^{-1}), similar to the simulations in Figure 9.6, when the target settling time was below 30 s.

So the first decision is to define the identification frequency range. The previous oscillatory step response justifies the choice of a band between 6 and 12.6 rad s^{-1}. This determines the prefilters to be added in order to identify a new model.

The second decision is about the model complexity. To begin with, a second-order model is chosen in order to catch the first resonant mode dynamics.

The identification is then carried out using an open-loop set of data $\{u(t), y(t)\}$. The plant input $u(t)$ is a PRBS generated with a 7-cell shift register with a frequency divisor of 2. Both input and output signals are filtered through a 4th-order *Butterworth* filter. The used identification algorithm is the extended output error algorithm [21]. The obtained second-order model G_1 of the flexible transmission system is:

$$G_1(z) = \frac{0.229z + 0.1977}{z^2 - 1.608z + 0.9883} \tag{9.14}$$

Using this plant model and a model reference defined by Equation (9.8), consecutive cancelation controllers are designed up to $\lambda = 7$ rad s^{-1}, where oscillations begin to appear. The controller C_{IT} designed for $\lambda = 7$ rad s^{-1} is:

$$C_{IT}(z) = \frac{0.217z^3 - 0.173z^2 - 0.06826z + 0.1738}{z^3 - 0.629z^2 - 0.796z + 0.425} \tag{9.15}$$

Figure 9.10 shows the comparison between the simulated and the experimental step responses with C_{IT}. For the simulation, the model in Equation (9.3) has been used.

At this moment, due to the appearance of the oscillations, a new model has to be identified. However, as new identifications reveal that an order increment is needed so as to go on increasing the closed-loop bandwidth, the whole process is stopped to avoid increasing the controller's complexity, as the achieved behavior is deemed satisfactory. The final controller is then C_{IT} in Equation (9.15).

Restricted-complexity Controller Design by Iteratively Applying VRFT

This second example tries to illustrate a different usage of an iterative scheme. As explained in Section 9.1, the general (usual) iterative scheme consists of several interleaved plant identification and controller design steps. However, this basic scheme can also be modified to account for different controller design methodologies that are not based on the availability of a plant model (nonmodel-based controllers). Instead, a model of the controller is iteratively estimated by directly using input/output data.

This is the case, for example, of the virtual reference feedback tuning methodology. This is a *direct* restricted complexity controller design. VRFT was introduced in [7], and further developed in [32], and extended to nonlinear settings in [5, 30].

The basic idea is to directly identify the controller using a set of data collected from the real plant in open or in closed loop, and to specify the desired performance by suitably

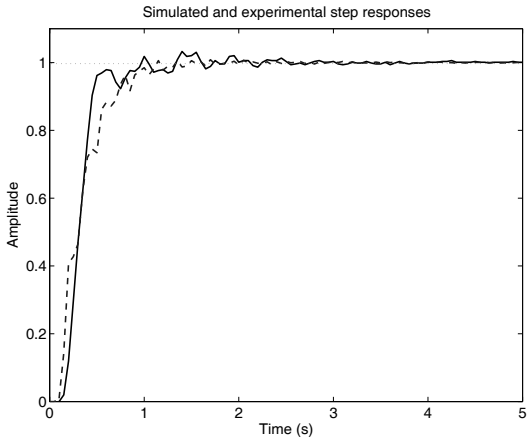


Fig. 9.10. Comparison between the simulated (dashed line) and the achieved (solid line) step response using the controller designed for $\lambda = 7$ rad s^{-1}

choosing the identification criterion. This is carried out by computing some *virtual* signals from the experimental data: the *virtual reference* (r_v) and the *virtual error* (e_v). If the desired performance is defined as a reference model (M) in the complementary sensitivity function, these signals are computed as:

$$r_v = M^{-1} y_{ex} \tag{9.16}$$

$$e_v = (M^{-1} - 1) y_{ex} \tag{9.17}$$

where y_{ex} is the experimental output when u_{ex} is applied to the plant's input.

The so-defined *virtual error* is the input to the to-be-designed controller in a *virtual* closed loop having r_v as the reference, y_{ex} as the plant's output and u_{ex} as the controller's output (see Figure 9.11).

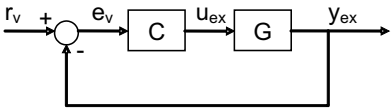


Fig. 9.11. *Virtual loop* built from the experimental data $\{u_{ex}, y_{ex}\}$ and the reference model M

Once the virtual error is computed, the controller is designed by identification given its input, e_v and its output u_{ex}.The reader is referred to [32] to see different options in this setting.

Iterative VRFT

The VRFT methodology involves the definition of a reference model. This can be imposed over the complementary sensitivity function [6], the input or output sensitivity function [8, 23]

or both the sensitivity and the complementary sensitivity functions [24]. In this particular application, the complementary sensitivity function reference model, called M, is used. The selection of M entails the definition of the *desired* level of performance, *i.e.* the desired bandwidth. But this is a hazardous task, as, if no information about the plant is available, we cannot guess the final achievable bandwidth. This, of course, suggests to first choose a low bandwidth and to increase it as more knowledge about the plant is gained. Therefore the proposed iterative VRFT scheme is summarized as:

1. Collect a set of experimental data in open loop: $\{u_{ex}, y_{ex}\}$.
2. Define a low-bandwidth reference model M_i and compute the *virtual* signals r_v and e_v with the most recent experimental data.
3. Identify the controller C_i between u_{ex} and e_v, using a low-order model structure (in this application, the OE algorithm is used).
4. Check the closed-loop stability before implementing the controller. To do this, in reference [32], the identification of the *achieved* input sensitivity function is proposed. If this transfer function is stable, the controller is validated, so go to *Step i-d*. If this is not the case, increase the controller's order and go back to step **3**.
5. Implement the controller. If the achieved performance is not so different from the designed one, then increase the reference model's bandwidth and go to step **2**. If the misfit between the achieved and the designed performances is unacceptable, then increase the controller's order and go back to step **3**.

End The algorithm ends due to one or even to a combination of several reasons:

- The achieved final performance is satisfactory.
- The presence of noise and/or nonlinearities hinders the identification step.
- Further order increases do not provide a significant improve in the closed-loop's behavior.

Application to the Flexible Transmission System

Next, this iterative VRFT scheme is applied to the system under consideration.

The methodology is first tested in simulation using the model in expression (9.3). So the input u_{ex} is introduced to the model in open loop, and its output y_{ex} is recorded. This input is the same sequence as that used for simulations at the previous section (two consecutive PRBS of amplitude ± 1 generated by a 7-cell shift register and a frequency divisor of 4).

The consecutive reference models M_{ij} are designed using the following expression (recalling that λ_{ij} and T_s are the current bandwidth and the sampling period, respectively):

$$M_{ij}(z) = \frac{(1 - e^{-\lambda_{ij} T_s})^2 z^{-3}}{(1 - e^{-\lambda_{ij} T_s} z^{-1})^2} \tag{9.18}$$

In the expression above, the index i stands for the group of iterations considering the same order for the controller. When the controller's complexity has to be increased, the index i increases by one. On the other hand, the index j indicates the number of the iteration within each group.

The first batch of iterations is carried out using a second-order model for the controller. This complexity is enough for bandwidths up to 1.5 rad s^{-1}. In fact, in Figure 9.12, the step responses of the closed loop when operating the controllers C_{11} and C_{12} designed for the bandwidths $\lambda_{11} = 1$ rad s^{-1} and $\lambda_{12} = 1.5$ rad s^{-1}, respectively, are depicted. For information, the identified controller C_{12} has a transfer function given by:

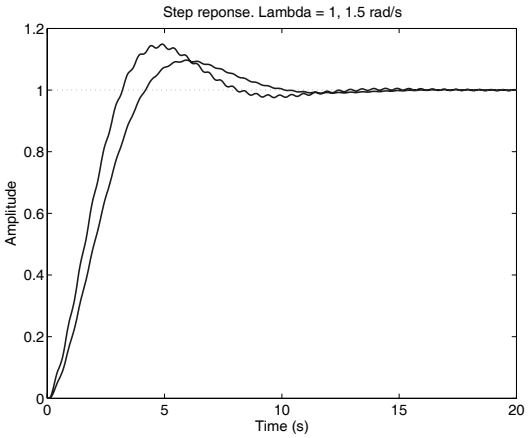


Fig. 9.12. Iterations with a second-order controller

$$C_{12}(z) = \frac{0.00253z^2 + 0.000671z - 0.001858}{z^2 - 1.965z + 0.9653} \tag{9.19}$$

As the controller's complexity is not enough for achieving higher performances, its order is increased by one, *i.e.* third-order controllers are then identified using VRFT methodology. With this complexity, the iterations depicted in Figure 9.13 are carried on. As can be observed, some oscillations appear from $\lambda = 10$ rad s^{-1}, being bigger as the bandwidth increases. The identification of controllers of higher order did not provide significant improvement in the step response, so the algorithm stopped here.

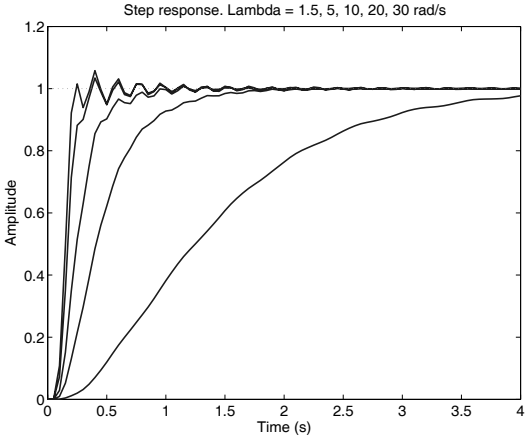


Fig. 9.13. Iterations with a third-order controller

To summarize, the last controllers of each group of iterations have been C_{12} in Equation (9.19) and:

$$C_{25}(z) = \frac{0.2764z^3 - 0.1676z^2 - 0.1711z + 0.273}{z^3 - 1.397z^2 + 0.5382z - 0.1415} \tag{9.20}$$

Experiments. The results when applying the above-explained iterative procedure to the true flexible transmission system are now displayed. The model reference is also defined by the expression (9.18).

First- and second-order controllers behaved similarly to the above simulations. The final iterations were carried out by identifying third-order controllers, based on the experimental set of data $\{u_{ex}, y_{ex}\}$ obtained from the plant with a lower-order controller.

In this way, the best performance achieved by the true plant is for $\lambda = 12$ rad s^{-1}. The controller identified for achieving this bandwidth is called C_{VRFT}, and its expression is:

$$C_{VRFT}(z) = \frac{0.0788z^3 - 0.048z^2 - 0.04884z + 0.078}{z^3 - 2.1188z^2 + 1.615z - 0.4963} \tag{9.21}$$

Figure 9.14 shows the comparison between the simulated and the experimental step responses, using again for simulations the model (9.3). The control action for the final controller has also been plotted.

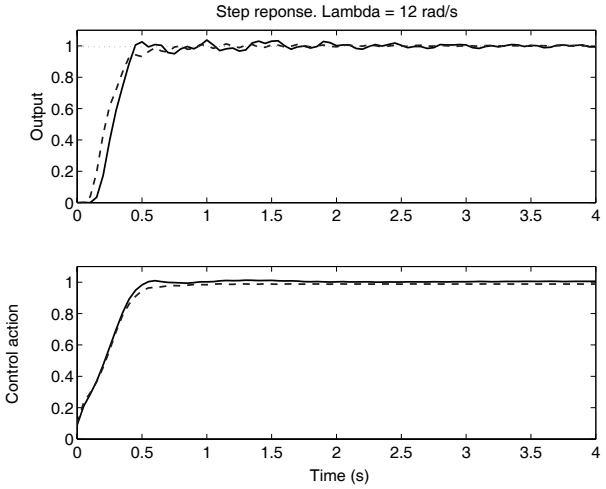


Fig. 9.14. Comparison between the simulated (dashed line) and the experimental (solid line) step response for $\lambda = 12$ rad s^{-1}

Comparison between Both Approaches

To end with this first application, both previous iterative approaches are compared. Figure 9.15 shows both simulated step responses: the one using C_{IT} and the other when operating C_{VRFT} (recall that these controllers were designed for $\lambda = 7$ and 12 rad s^{-1}, respectively).

Finally, Figure 9.16 depicts the experimental step response achieved when implementing both C_{IT} and C_{VRFT}.

Note that, in practice, the actually achieved performance is similar for both methodologies.

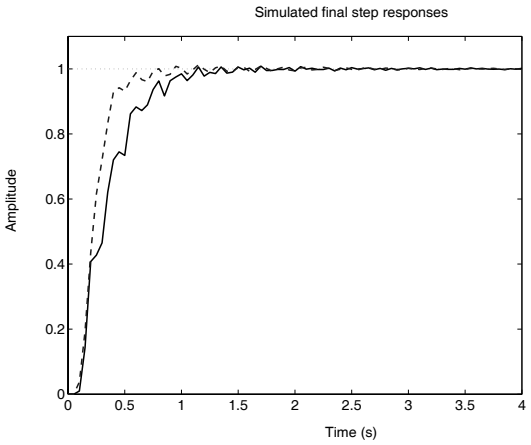


Fig. 9.15. Comparison between the simulated step response when both C_{IT} (solid line) and C_{VRFT} (dashed line) are operating

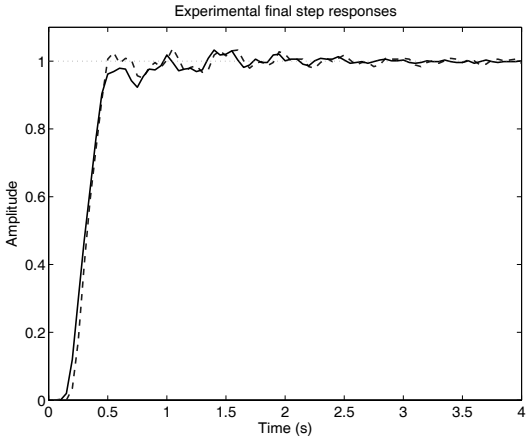


Fig. 9.16. Comparison between the experimental step response when using C_{IT} (solid line) and C_{VRFT} (dashed line)

9.3.2 Low-order Controller Design for an Active Suspension System

The second application is the iterative low-order controller design for an active suspension system. Configuration and operation of the system are detailed in Section 9.2. The specifications of the feedback loop to be designed are analogous to those of the benchmark proposed in [20], where the reader may consult experimental results of alternative control-design approaches on the same system.

The active suspension system is a complex plant, with several resonant modes. So, an accurate model will present a high complexity. If the controller-design methodology is based on such a model, it will deliver a high-order controller. In order to try to avoid such a complexity, there are two possible paths to follow:

1. An indirect approach, starting with ID of a high-order model. There are two options:
   - Designing such a high-complexity controller and then using any controller-reduction techniques [26].
   - Reducing the model complexity (using any model-reduction technique); subsequently, designing a low-order controller from that reduced-order plant model.
2. Directly designing a low-order controller using the experimental data. To achieve this, the controller-design methodology can be either:
   - *Model based*: If the model is a high-order one, the restricted-complexity controllers may be directly designed by optimization [9, 10, 19, 22]. If, on the contrary, a low-order plant model is directly identified from the experimental data, the controllers can be designed using any model-based methodology [31]. This is one of the paths that will be followed in this application.
   - *Model free:* there are nonmodel-based controller design methodologies that obtain the controller directly from the experimental data, without identifying any plant model [7, 18, 29].

In this section, two approaches for controller design being tested will be used:

1. Indirect low-order controller design by identification in closed loop [11, 28].
2. Direct *iterative* low-order controller design using the scheme outlined on page 255 and detailed in [31].

The reason for comparing these two different low-order controller design approaches is that, as we shall see, theoretical and practical results do not totally agree. What *a priori* seemed a more accurate way of addressing the problem (the *indirect* method, as will be discussed later), has somewhat poorer results when put into practice.

Of course, following the same line as in the previous application (Section 9.3.1), both methodologies are previously simulated, using a full-order model of the active suspension system.

Indirect Low-order Controller Design

The first methodology used to design a low-order controller is the same as in [11, 28]. The steps followed in these references are:

1. To identify a good model of the plant. This identification is carried out in open loop.
2. To design a controller based on the previously identified model. This controller is obtained using a modified pole-placement methodology that also incorporates restrictions on the shape of some sensitivity functions [11, 28].
3. With this controller, a new identification is done, but this time in closed loop.
4. The new (high-order) model obtained in step 3 is used to design a second controller by pole placement, considering the same specifications as for the design done in step 2.
5. Finally, this controller is reduced using identification in closed loop.

The controller design for an active suspension system problem was addressed in [20]. In this reference, a benchmark was proposed. The objective was to design a controller for this system keeping its complexity as low as possible.

In this benchmark, a high-order model for the active suspension was provided to be used by all authors. However, later experiments developed onsite by one of the present authors obtained significant differences; therefore, the need for a new model for the active suspension plant became evident in order to achieve good performance designs.

The model used for simulations (called G) is the result of the application of the algorithm stated above; in particular, it is the one identified in closed loop at step 3: it has been identified in closed loop when operating a controller that, in turn, was designed at step 2 for the required performance. This helps the accuracy around the control-relevant frequencies. The model's structure is a SISO 16th-order transfer function. For brevity, it is not reproduced here because it does not introduce any additional information, but its Bode diagram can be observed in Figure 9.17.

This plant model is used to simulate the designed closed loop before implementing it. Therefore, the simulated results can be used in two ways: first, they assess the closed-loop performance (and even its stability); second, they can be used for comparison with the experimental ones.

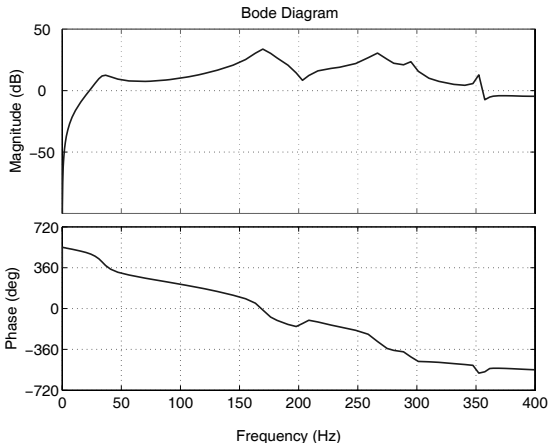


Fig. 9.17. Identified model for the active suspension system

The required controllers of steps 2 and 4 are designed by the modified pole-placement methodology using the MATLAB®'s toolbox *ppmaster*[3]. The specifications imposed on the design are that both input and output sensitivity functions (S_{yp} and S_{up}) must be below the templates of Figure 9.18.

In summary, when applying the above-outlined procedure, the full-order controller obtained at step 4, denoted as C_{FULL}, has order 24, and meets the specifications when simulated. However, some discrepancies appear (around 20 Hz and at higher frequencies) with the experimentally achieved output sensitivity function (Figure 9.19).

Controller-order Reduction

The methodology used to reduce the order of the above-designed controller is based on controller identification in closed loop. In particular, the closed-loop input matching (CLIM) algorithm is applied [21]. In some sense, CLIM is related to the previously used VRFT methodology [30].

[3] This toolbox can be found in the website of the book [21] *http://landau-bookic.lag.ensieg.inpg.fr/english/index_ENG_Fonctions_liste.htm*

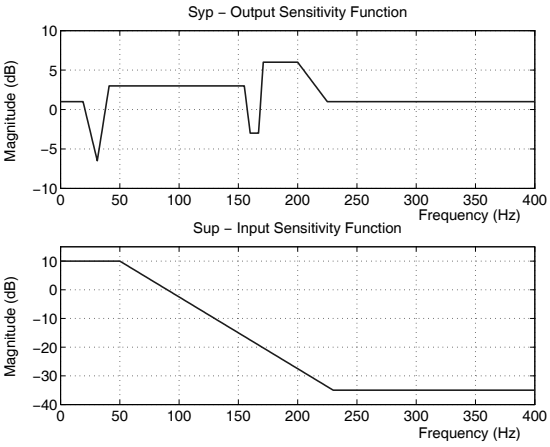


Fig. 9.18. Templates for the sensitivity functions S_{up} and S_{yp}

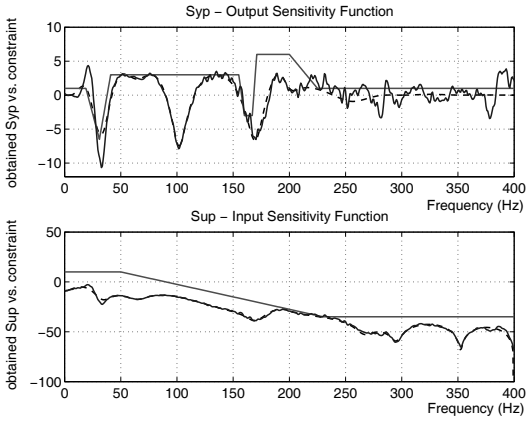


Fig. 9.19. Simulated (dashed line) and achieved (solid line) behavior with C_{FULL} operating in the closed loop

Using this algorithm, a PRBS sequence generated by a 10-cell shift register is introduced as the input r of the closed loop. Then, a new controller (C_{RED}) is obtained (of 9th order) so that the closed-loop behavior does not deteriorate too much. Figure 9.20 shows a comparison between the simulated closed loops when C_{FULL} and C_{RED} are applied. In addition, Figure 9.21 compares the real and simulated behavior of the closed loop with C_{RED}.

Iterative Low-order Controller Design

Next, a low-order controller will be designed for the active suspension system following the same iterative scheme as in Section 9.3.1, detailed on page 255. The performance evaluation, however, is now based on frequency-domain characteristics, instead of the time-domain ones used in the previously discussed flexible-link application.

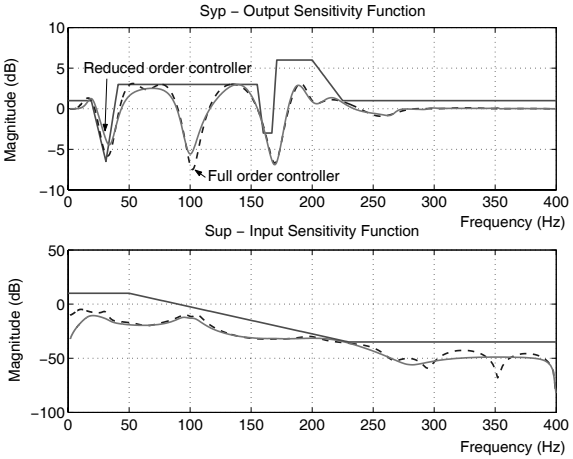


Fig. 9.20. Comparison between the theoretically expected behavior of the closed loop when operating C_{FULL} (dashed line) and C_{RED} (solid line)

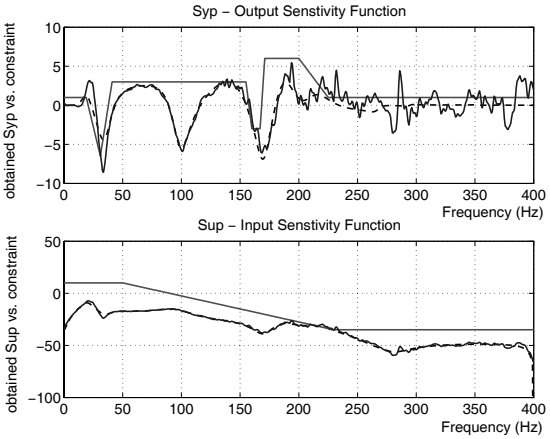


Fig. 9.21. Simulated (dashed line) and real (solid line) closed-loop behavior with the reduced-order controller C_{RED} in operation

Defining the Specifications to be Achieved by the Closed Loop

As previously discussed, the methodology involves progressively increasing the specifications to be achieved by the closed loop. In this case, the desired final behavior is defined in Figure 9.18.

Performance increase will be understood as achieving progressively smaller values of the output sensitivity, S_{yp}.

Figure 9.22 shows S_{yp1}, S_{yp2}, S_{yp3} and S_{yp}, *i.e.* the consecutive templates to be considered over the output sensitivity function. Note that the last one, called S_{yp}, is the final target template.

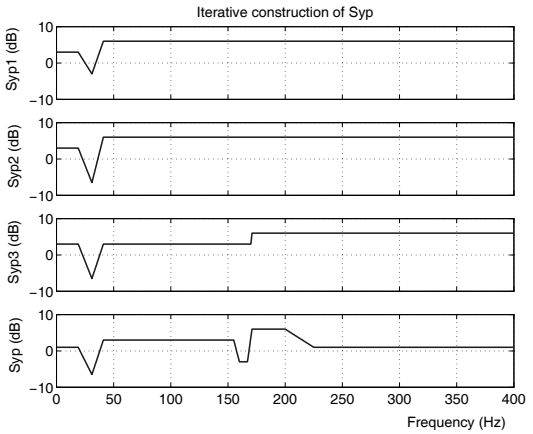


Fig. 9.22. Output sensitivity function templates to be considered

Iterative-controller Design

The first step in this procedure is to identify a low-order model. The first iteration, based on a very simple model capturing the lower-frequency resonant peak, shows unacceptable performance. Thus, as the experimental responses reveals the existence of two significant resonant peaks, and as the measurements (acceleration) include a double derivator, a sixth-order model is sought.

Identification is carried out using the experimental plant's output collected when the input to the secondary path is a PRBS generated by a 10-cell shift register, applied in open loop. First, these data are filtered through a bandpass *Butterworth* filter with cutoff frequencies of 1 and 400 Hz, to avoid high-frequency noise. The model is identified using an extended output error algorithm, incorporating the prior knowledge of the double derivator. The resulting model is:

$$\begin{aligned} G_1(z) &= \frac{-2.36\cdot10^{-7}z^3+2.25\cdot10^{-6}z^2-4.72\cdot10^{-6}z+5.8\cdot10^{-6}}{z^4-2.36z^3+2.81z^2-2.206z+0.856}\cdot\left(\frac{z-1}{zT_s}\right)^2 \\ &= \frac{-0.15z^5+1.74z^4-6.04z^3+11.19z^2-10.44z+3.71}{z^6-2.36z^5+2.81z^4-2.21z^3+0.86z^2} \end{aligned} \tag{9.22}$$

where T_s is the sampling period, *i.e.* 1.25 ms.

All controllers have been designed by pole placement, using again the software *ppmaster*. The different graphs in Figure 9.23 show the simulated system behavior when operating the different controllers. Recall that the full-order model G previously identified in closed loop is used for simulation. In this figure, the dashed lines represent the frequency response of the low-order model G_1 and the solid ones show the behavior of the full-order model G, both operating in closed loop with the controllers designed below.

The first controller is C_{11} (Figure 9.23a), designed from G_1 to satisfy S_{yp1} and being of 6th order:

$$C_{11}(z) = \frac{0.02z^6-0.0073z^5+0.00038z^4+0.012z^3-0.016z^2}{z^6-2.52z^5+2.67z^4-1.52z^3+0.47z^2-0.16z+0.076} \tag{9.23}$$

C_{12} is designed from G_1 with the same order as before (Figure 9.23b), using the template S_{yp2} and obtaining:

$$C_{12}(z) = \frac{0.024z^6+0.02z^5+0.014z^4+0.026z^3+0.0083z^2}{z^6-1.4z^5+0.95z^4-0.31z^3+0.07z^2-0.11z-0.036} \tag{9.24}$$

Then, the output sensitivity template is changed to S_{yp3} and C_{13a} is designed from G_1, again with the same order. As the behavior is not as expected at low frequencies (the achieved output sensitivity function does not fit the specifications around 20–30 Hz, see Figure 9.23c), the order has been increased to 8 in C_{13b} (Figure 9.23d):

$$C_{13b}(z) = \frac{0.0071z^8-0.018z7+0.014z^6+0.0007z^5-0.014z^4+0.017z^3-0.007z^2}{z^8-4.95z^7+10.47z^6-12.3z^5+8.64z^4-3.65z^3+0.92z^2-0.17z+0.03} \tag{9.25}$$

Finally, the output sensitivity function's template considered is the final one (S_{yp}) and a first 6th-order controller C_{14a} is designed from G_1. Again, as the specifications are not fulfilled (Figure 9.23e), C_{14b} is designed as an 8th-order controller. This is the final controller, as the specifications are almost met (Figure 9.23f), and its transfer function is:

$$C_{14b}(z) = \frac{0.03z^8-0.07z^7+0.062z^6+0.003z^5-0.067z^4+0.065z^3-0.02z^2}{z^8-3.9z^7+6.7z^6-6.4z^5+3.71z^4-1.5z^3+0.68z^2-0.36z+0.1} \tag{9.26}$$

Further attempts have been made considering both controller and model-order increments, but the results have not improved substantially. Therefore, C_{14b} has been applied to the real system, obtaining the result depicted in Figure 9.24.

To end with this application, let us pay attention to Figure 9.25. In this figure, the achieved behavior of the closed loop when implemented C_{RED} (9th order) and C_{14b} (8th order) is depicted. Observe that C_{14b} provides a better high-frequency behavior besides being of lower order.

9.4 Discrepancies/Misfits between Theory and Experimental Results

This work has presented simulations and experiments of iterative identification and control design techniques for mechanical systems, putting emphasis on the use of low-order models and controllers when the required performance goals are not stringent. As the optimal control goals themselves are uncertain when the controlled plant is uncertain, the suggested procedures increase, cautiously, the required performance as far as it is not fully satisfactory and basic constraints are not violated. Remodeling decisions are made when performance improvements appear in simulation but not in practice.

A typical discussion about this approach is the possibility of getting, from the very beginning, a complete full-order model of the plant. From a theoretical point of view this is, of course, a simpler approach, but from a practical perspective, the modeling effort can be drastically reduced and tailored for the specific requirements.

In fact, all obtained models will be "reduced-complexity" ones as no system is exactly linear, time invariant and finite order in practice. Hence, if assumedly "full-order" techniques are used, it will be necessary, in fact, to consider the issues related to experiment design, signal-to-noise ratio, performance goals, *etc*.

From the experience gained in the work described here and other trials, it seems that the simulations, when all conditions prescribed in theory are fulfilled, allow a larger number of iterations than practice does: other phenomena, unaccounted for in transfer-function models, limit the achievable performance in practice earlier than in the simulations, as discussed below.

Indeed, noise-free simulations yield, when the order is increased, the "true" model and hence, perfect performance can be achieved (with no nonlinear saturation constraints, of course).

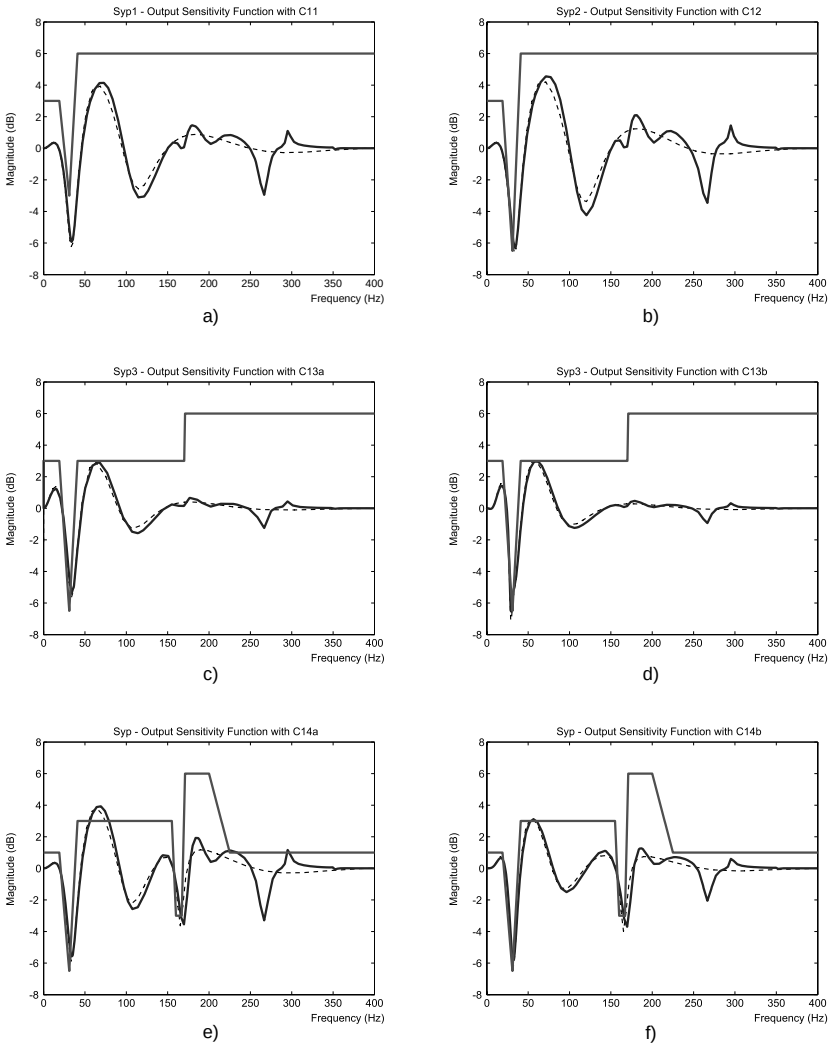


Fig. 9.23. Output sensitivity when applying a) C_{11}, b) C_{12}, c) C_{13a}, d) C_{13b}, e) C_{14a}, f) C_{14b}

In contrast, at least for the relatively simple mechanical systems under study in this work, reasonable performance can be achieved with low-order regulators in one or two "modeling-control-(re)modeling-…" iterations. Higher-order models and controllers do not seem to substantially improve the achieved performance: indeed, there is a poorer signal-to-noise ratio at high frequencies and, importantly, nonlinearities (saturation, gear backlash, friction) start to be relevant for high-performance control of mechanical systems (not being captured in linear models, whatever its order is). Saturation, in particular, seems to be a key limiting factor. So, it seems that, in mechanical systems subject to linear control, the decision of using low-order models and controllers capturing just the first (and maybe the second) resonant modes is wise and, in fact, iteration-stopping criteria do suggest it.

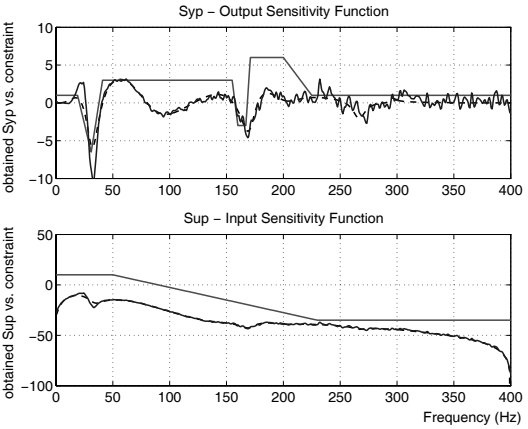


Fig. 9.24. S_{yp} and S_{up} achieved by C_{14b}: simulated (dashed line) and experimental result (solid line)

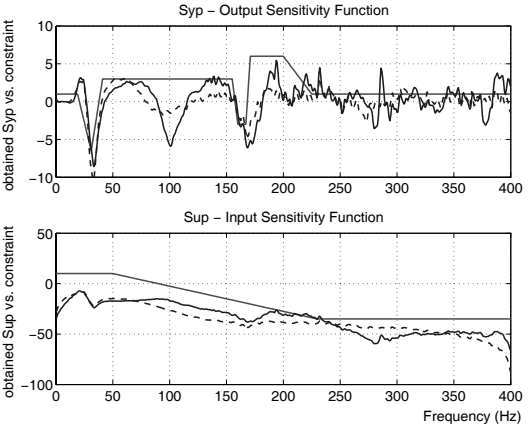


Fig. 9.25. Comparison between the behavior of the closed loop when using C_{RED} (solid line) and C_{14b} (dashed line)

Note that the proposed schemes may be conceived as part of one including nonlinear control, where increased complexity of the models might mean not only increasing the order of the dynamics but increasing the number of parameters for nonlinear function approximators, if any. However, the nonlinear methodologies have not been explored in this work, requiring further research, as the frequency-response issues used for identification experiment design and robustness validation no longer hold.

As a conclusion of the work, two remarks are relevant: (a) the "full-model" approach never exists in practice and approximations must always be made; (b) it is very difficult for any "one-shot" theoretical procedure to succeed to find a high-performance design in a totally unknown plant. Hence, identification and control design in practice will nearly always need some sort of "iteration" because the attainable goals depend on the plant, and the plant modeling criteria

depend on the goals of the entities that will afterwards use the model (*i.e.* controller design). Time variations (aging) may force the need of retuning, repeating some of the steps.

Acknowledgements

This work was partially sponsored by the Research Incentive Program (PPI-00-05) of the Universidad Politécnica de Valencia (Spain).

References

[1] Albertos, P. and Esparza, A. (2006). Some issues about the control of resonant processes. *Proceedings of the 12th IEEE Methods and Models in Automation and Robotics, Miedzyzdroje, Poland*, pp. 1–10.

[2] Albertos, P., Esparza, A., and Romero, J. (2000). Model-based iterative control desgin. *Proceedings of the American Control Conference, Chicago, IL, USA*, pp. 2578–2582.

[3] Albertos, P. and Sala, A., editors (2002). *Iterative Identification and Control: Advances in Theory and Applications*. Springer, London.

[4] Albertos, P., Valera, A., Romero, J., and Esparza, A. (2005). An application of iterative identification and control in the robotics field. *Current Trends in Nonlinear Systems and Control*, pp. 217–229. Birkhäuser.

[5] Campi, M. and Savaresi, S. (2006). Direct nonlinear control design: The virtual reference feedback tuning (VRFT) approach. *IEEE Transactions on Automatic Control* **51**(1), 14–27.

[6] Campi, M. C., Lecchini, A., and Savaresi, S. M. (2000). Virtual reference feedback tunning (VRFT): A new direct approach to the design of feedback controllers. *Proceedings of the 39th Conference on Decision and Control, Sydney*, pp. 623 – 628.

[7] Campi, M. C., Lecchini, A., and Savaresi, S. M. (2002). Virtual reference feedback tuning: A direct method for the design of feedback controllers. *Automatica* **38**, 1337–1346.

[8] Campi, M. C., Lecchini, A., and Savaresi, S. M. (2003). An application of the virtual reference feedback tuning method to a benchmark problem. *European Journal of Control* **9**, 66–76.

[9] Campos-Delgado, D., Fermal, R., and Ruiz-Velázquez, E. (2003). Design of reduced-order controllers via H_∞ and parametric optimisation: Comparison for an active suspension system. *European Journal of Control* **9**(1), 48–60.

[10] Cao, Y. and Yan, W. (2003). Multi-objective optimisation of restricted complexity controllers. *European Journal of Control* **9**(1), 61–65.

[11] Constantinescu, A. and Landau, I. D. (2003). Controller order reduction by identification in closed-loop applied to a benchmark problem. *Europen Journal of Control* **9**(1), 84–99.

[12] Forssell, U. (1999). *Closed-Loop Identification. Methods, Theory, and Applications*. Ph.D. thesis, Department of Electrical Engineering, Linköping University, Sweden.

[13] Gevers, M. (1993). Towards a joint design of identification and control. H. Trentelman and J. Willems, editors, *Essays on Control: perspectives in the theory and its applications*, pp. 111–151. Birkhäuser, Boston.

[14] Gevers, M. (1996). Modelling, identification and control. *Communications, Computing Control and Signal Processing 2000*, pp. 1–15. Kluwer Academic Publishers, Dordrecht.

[15] Gevers, M. (2002). Identification and validation for robust control. *Iterative Identification and Control: Advances in Theory and Applications*, pp. 185–208. Springer.

[16] Gevers, M., Bombois, X., Codrons, B., Scorletti, G., and Anderson, B. (2003a). Model validation for control and controller validation in a prediction error identification framework - part i: Theory. *Automatica* **39**, 403–415.

[17] Gevers, M., Bombois, X., Codrons, B., Scorletti, G., and Anderson, B. (2003b). Model validation for control and controller validation in a prediction error identification framework - part II: Illustrations. *Automatica* **39**, 417–427.

[18] Hjalmarsson, H., Gevers, M., Gunnarson, S., and Lequin, O. (1998). Iterative feedback tuning: Theory and applications. *IEEE Control Systems Magazine* **18**(4), 26–41.

[19] Hol, C., Scherer, C., Van der Meché, E., and Bosgra, O. (2003). A nonlinear SDP approach to fixed-order controller synthesis and comparison with two other methods applied to an active suspension system. *European Journal of Control* **9**(1), 13–28.

[20] Landau, I. D., Karimi, A., Miskovic, L., and Prochazka, H. (2003). Control of an active suspension system as a benchmark for design and optimisation of restricted-complexity controllers. *European Journal of Control* **9**, 3–12.

[21] Landau, I. D. and Zito, G. (2005). *Digital Control Systems: Design, Identification and Implementation*. Springer.

[22] Le Mauff, F. and Duc, G. (2003). Designing a low order robust controller for an active suspension system thank LMI, genetic algorithm and gradient search. *European Journal of Control* **9**(1), 29–38.

[23] Lecchini, A., Campi, M., and Savaresi, S. (2001). Sensitivity shaping via virtual reference feedback tuning. *Proceedings of the 40th Conference on Decision and Control, Orlando, Florida, USA*, pp. 750–755.

[24] Lecchini, A., Campi, M., and Savaresi, S. (2002). Virtual reference feedback tuning for two degree of freedom controllers. *Int. J. Adapt. Control and Signal Processing* **16**, 355–371.

[25] Lee, W., Anderson, B., Kosut, R., and Mareels, I. (1993). On robust performance improvement through the windsurfer approach to adaptive robust control. *Proceedings of the 32nd Conference on Decision and Control, San Antonio, Texas, USA*, pp. 2821–2827.

[26] Obinata, G. and Anderson, B. (2001). *Model Reduction for Control System Design*. Springer-Verlag, London.

[27] Partanen, A. and Bitmead, R. (1993). Two stage iterative identification/controller design and direct experimental controller refinement. *Proceedings con the 32nd Conference on Decision and Control, San Antonio, Texas, USA*, pp. 2833–2838.

[28] Prochazka, H. and Landau, I. D. (2003). Letter to the Editor on the Paper "Controller Order Reduction by Identification in Closed Loop Applied to a Benchmark Problem" by I. D Landau and A. Constantinescu. *European Journal of Control* **9**(5), 528–532.

[29] Safonov, M. (2001). Unfalsified control: A behavioral approach to learning and adaptaton. *Proceedings of the 40th IEEE Conference on Decision and Control, Orlando, Florida, USA*, pp. 2682–2685.

[30] Sala, A. (2007). Integrating virtual reference feedback tuning into a unified closed-loop identification framework. *Automatica* **43**(1), 178–183.

[31] Sala, A. and Esparza, A. (2003). Reduced-order controller design via iterative identification and control. *European Journal of Control* **9**, 105–117.

[32] Sala, A. and Esparza, A. (2005). Extensions to "Virtual Reference Feedback Tunning: A Direct Method for the Design of Feedback Controllers". *Automatica* **41**(8), 1473–1476.

[33] Sala, A. and Esparza, A. (2005). Virtual reference feedback tuning in restricted complexity controller design of non-minimum phase systems. *Proceedings of the* 16*th IFAC World Congress, Prague*, pp. 1063–1069.

[34] Schrama, R. (1992). Accurate identification for control: The necessity of an iterative scheme. *IEEE Transactions on Automatic Control* **37**, 991–994.

[35] Seron, M. M., Braslavsky, J. H., and Goodwin, G. C. (1997). *Fundamental Limitations in Filtering and Control*. Springer.

[36] Van Den Hof, P. M. J. and Schrama, R. J. P. (1995). Identification and control - closed-loop issues. *Automatica* **31**, 1751–1770.

10

Classical, Robust and LPV Control of a Magnetic-bearing Experiment

Alejandro S. Ghersin[1,2], Roy S. Smith[3], and Ricardo S. Sánchez Peña[4,5]

[1] Ingeniería en Automatización y Control, Departamento de Ciencia y Tecnología, Universidad Nacional de Quilmes (UNQ), Argentina
[2] Grupo de Identificación y Control Robusto (GICOR), Departamento de Electrónica, Facultad de Ingeniería, Universidad de Buenos Aires (FIUBA), Argentina
aghersin@unq.edu.ar
[3] Department of Electrical & Computer Engineering, University of California, Santa Barbara, CA 93106, USA roy@ece.ucsb.edu
[4] Sistemes Avançats de Control, ESAII, Universitat Politècnica de Catalunya (UPC), Rbla. Sant Nebridi 10, Terrassa, 08222 Barcelona, Spain
[5] Institució Catalana de Recerca i Estudis Avançats (ICREA), Barcelona, Spain
ricardo.sanchez-pena@upc.edu

Summary. This chapter reports on the modeling, identification and control system design of a magnetic-bearing system. The model used in order to carry out the design of the control system is first obtained in a process with two steps. In the first one, a nonlinear model based upon physical laws is derived. After a linearization step, and with a set of frequency-response samples in hand, an *ad hoc* identification process follows that renders a linear model that adequately fits the experimental frequency-response samples.

The control-system design problem is addressed through three different techniques. The first one resorts to simplifying the multiple input multiple output (MIMO) 4 degrees of freedom (DOF) dynamics of the magnetic-bearing system to a set of single input single output (SISO) transfer functions, one per DOF. A classical phase-lead controller is designed, its performance being evaluated in terms of the phase margin. The second technique is $\mathcal{H}_\infty$ robust control. Through *ad hoc* considerations about the high-frequency behavior of the system, in particular the rotating beam's flexible modes, a family of transfer functions is derived that represents the uncertain systems' dynamics. The design deals with the problem of robustly rejecting sinusoidal disturbance *signals* that actually represent the beam's imbalance. Finally, evolving from the $\mathcal{H}_\infty$ design, the parameter variation nature of the system (as the rotation rate of the beam changes) is introduced in order to derive a linear parameter varying (LPV) model. This approach seeks rejection of the sinusoidal imbalance force carrying out an LPV adaptation to the online measured rotation rate. The design process is thoroughly covered with remarks concerning the discrepancies between theory and practice that appear when applying the different methodologies.

Key words: Magnetic bearings, LPV control.

10.1 Introduction and Background

10.1.1 Motivation

In this chapter different techniques have been used in order to design the active magnetic-bearing (AMB) system of an MBC500 experimental magnetic-bearing system. The MBC500 is a product designed and manufactured for academic research by *Magnetic Moments*, a division of *LaunchPoint Technologies, LLC* (see [16] for a detailed description and the company's website: `http://www.launchpnt.com`). The particular MBC500 used for the experiments includes the "Turbo 500" option that allows for controlled rotation of the beam.

The addressed control problem deals with the stabilization of the machine's rotating shaft. No matter how well balanced the rotor may be, there is always an uncertain amount of eccentricity in it. This means that the axis of inertia is not exactly the geometric one, and as a result, imbalance forces appear. As a consequence, controlling the vibration of the rotor due to imbalance is within the main goals.

In practice, imbalance can be modeled as "external" forces representing the eccentricity, while considering the rotor as a rigid body with its inertia and geometric axis being the same. As functions of time, these forces are considered sinusoidal (while the machine rotates), with uncertain but bounded magnitudes and phases, and measurable frequency (rotor's rotation speed).

From a practical point of view, the motivation derives from applications where very high speed rotation should be controlled. This is the case in aerospace applications of momentum wheels used in satellites as a sensor and control element and in unmanned aerial vehicles (UAV) as an (electromechanical) energy-saving device. In both applications, weight is strongly penalized, therefore if mechanical momentum is to be maximized a higher rotation speed is a more efficient choice.

This work was originally based on the problem solved in [13]. In this chapter a loop-shaping $\mathcal{H}_\infty$ control for a motor with magnetic bearings is developed. Initially neglecting the rotor's rotation and imbalance, stabilization is achieved. Through an additional block (linear-fractionally connected), the imbalance vibration is passed through a notch filter with a scheduled central frequency (filter's central frequency varies with rotor speed). This classical *gain-scheduling* approach was tested in practice. Nevertheless, certain performance parameters are not taken into account at the initial design stage, such as the maximum rate of change in rotation speed. It also suffers the usual problems of gain-scheduled controllers that motivates the use of LPV control: there is no guarantee of stability of the time-varying closed-loop system while switching among controllers.

Here we have considered the problem of choosing alternative control strategies for this kind of application. First, the classical approach used in this kind of system consisting of a lead controller (or PD) was designed seeking a good phase margin for each of the loops involved in the system (one per DOF). With this design in hand the objective has been to investigate if, for this application, a simpler and easily implementable controller can withstand the robustness and time-varying stability requirements. On the other hand, we have considered taking into account the rate of change of the varying parameter, the uncertainty due to high-order flexible modes of the rotor (neglected by the rigid-body model), as well as to guarantee stability of the time-varying closed-loop system. To this end we have considered $\mathcal{H}_\infty$ and LPV control as design tools. Therefore, we can account for the relative influence of uncertainty and the time-varying nature of the model in the performance of the system.

In particular, the LPV control theory gives an elegant background for the well-known classical gain scheduling. In general, LPV models can arise from the linearization of a nonlinear

model at various parameter-dependent operating points or simply from natural LPV dynamics. This technique, produces a time-varying controller that schedules with the same varying parameters as the plant, which are measured in real time. In some sense it is an adaptive controller, but without the delays added by the identification stage of classical adaptive control. On the other hand, it can incorporate uncertainty in the usual linear fractional transformation (LFT) framework, as in robust control. The stability of the complete time-varying system is taken into account, considering also the bound on the parameter-variation rate. The controllers give sufficient conditions for stability and performance throughout the whole parameter-variation space.

When the scheduling parameters enter affinely into the system matrices a simplified version of LPV controllers can be designed [4]. Considering the range of parameter variations as a hypervolume defined by its vertices this methodology solves the controller in terms of a finite number of "vertex" controllers. The controller is computed as a convex combination of the vertex controllers giving a smooth scheduling as the parameter changes. In the following, "parameter" will be taken as parameter vector. These results will be discussed in the following section.

To address robustness concerns, the modeling approach is intended to deal with the flexible dynamics of the MCB500's rotor, covering them with global dynamic uncertainty frequency bounds. Simple experimental tests have shown that due to the limited bandwith of the current amplifier of the system, only the first two bending modes of the rotor show up in the responses of the system to sinusoidal signals introduced at the voltage control inputs of the current amplifiers.

The background section assumes knowledge of robust control theory, $\mathcal{H}_\infty$ and μ-synthesis methods, as well as modeling with dynamic uncertainty ([7, 22, 26, 27]) and covers the essential results on LPV Controllers, as developed in [4], which are used for synthesis throughout the application example.

10.1.2 Gain Scheduling and LPV control

Gain scheduling is a technique that has been extensively used in control in the past (see [19, 20] and references therein). Let us show a basic example that motivates gain scheduling. Suppose you have a dynamic system modeled through the following nonlinear equation:

$$\dot{x}(t) = f\left[x(t), u(t), \pi(t)\right] \tag{10.1}$$

We call the system modeled by Equation (10.1) a parameter-dependent system with the following remark: the state equation depends on the time "t" through the parameter vector $\pi(t)$ that is unknown *a priori* but can be measured in real time. For a specific *trajectory* of the parameter π, Equation (10.1) turns into the model of a time-varying system.

Having $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^{n_u}$ and $\pi(t) \in \mathbb{R}^s$ classical gain scheduling consists of gridding the $\mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R}^s$ real space. For each point in the grid, a linearization is carried out on Equation (10.1) and a linear time invariant (LTI) controller is designed. The gain-scheduling technique switches between controllers as the system evolves in $\mathbb{R}^n \times \mathbb{R}^{n_u} \times \mathbb{R}^s$.

The design and implementation of this kind of control strategy, involves intensive simulation in order to empirically establish two important aspects concerning the response of the final closed-loop system, namely system stability and maximum rate of change between operating points in the grid. It must be pointed out that the local LTI controllers give no information about these issues.

On the other hand, LPV control theory provides stability and performance analyses for a kind of linear system whose state-space matrices depend on a parameter $\pi(t)$ ranging in a compact set $\mathcal{P} \subset \mathbb{R}^s$. This section uses as main references classical work on LPV control theory ([4, 25]). In [4] stability and performance of LPV systems is established both for synthesis and analysis based upon quadratic Lyapunov functions. Although this approach is essentially conservative because of the kind of Lyapunov function it is based upon, it is still appealing as it renders very simple and computable solutions that are similar to the ones found in [8] for $\mathcal{H}_\infty$ control.

Reference [25] (a continuation of [4] in a way), bases its study of performance and stability of LPV systems upon parameter-dependent Lyapunov functions (PDLF), a less conservative approach.

This section is organized as follows. An analysis condition based upon a PDLF is presented first. The way this analysis condition changes for a single quadratic Lyapunov function (SQLF) is shown afterwards according to [4]. A synthesis theorem is presented later, which is an extension of the $\mathcal{H}_\infty$ control synthesis condition presented in [8]. A corollary of the synthesis theorem is presented next for affine LPV systems whose parameter-variation set is a compact polytope in $\mathbb{R}^s$. This corollary is the tool used for synthesis in the MBC500 application of this chapter and is appealing for practice, as it renders synthesis conditions based upon a linear matrix inequality (LMI) feasibility problem with a finite number of LMIs. The closing subsections show extensions of this theory to pole-placement conditions along the lines of [6] but for LPV systems (see also [11]).

10.1.3 Parameter-dependent Analysis

Definition 1 (Set of ξ-trajectories – $\mathcal{F}_{\mathcal{P}}^{\xi}$.). *Let $\mathcal{P} \subset \mathbb{R}^s$ be a compact set and $\xi \in \mathbb{R}^s$ a vector such that $\xi = \begin{bmatrix}\xi_1 & \xi_2 & \dots & \xi_s\end{bmatrix}^T$. $\mathcal{C}^1(\mathbb{R}, \mathbb{R}^s)$ is the set of continuous and differentiable real functions with image in $\mathbb{R}^s$. The set of ξ-trajectories of the parameter is defined as*

$$\mathcal{F}_{\mathcal{P}}^{\xi} \triangleq \{\pi \in \mathcal{C}^1(\mathbb{R}, \mathbb{R}^s) : \pi(t) \in \mathcal{P}, |\dot{\pi}_i(t)| \leq \xi_i, i = 1, 2, ..., s \quad \forall t \in \mathbb{R}_+\}$$

with $\mathbb{R}_+$ being the $[0, \infty)$ interval.

Definition 2 (LPV System). *The following four continuous matrix functions represent an n-dimensional LPV system:*
$A : \mathbb{R}^s \to \mathbb{R}^{n \times n}$
$B : \mathbb{R}^s \to \mathbb{R}^{n \times n_w}$
$C : \mathbb{R}^s \to \mathbb{R}^{n_z \times n}$
$D : \mathbb{R}^s \to \mathbb{R}^{n_z \times n}$.
Let $\pi \in \mathcal{F}_{\mathcal{P}}^{\xi}$ be a specific trajectory. The system evolves according to the following state-space equation:

$$\begin{bmatrix}\dot{x} \\ z\end{bmatrix} = \begin{bmatrix}A(\pi) & B(\pi) \\ C(\pi) & D(\pi)\end{bmatrix} \begin{bmatrix}x \\ w\end{bmatrix} \tag{10.2}$$

with $x(t) \in \mathbb{R}^n$, $z(t) \in \mathbb{R}^{n_z}$ and $w(t) \in \mathbb{R}^{n_w}$. The notation $G_{\mathcal{P}_\xi}$

$$G_{\mathcal{P}_\xi} \equiv \left[\begin{array}{c|c} A(\cdot) & B(\cdot) \\ \hline C(\cdot) & D(\cdot) \end{array}\right] \tag{10.3}$$

will be used to denote the LPV system defined by the A, B, C and D matrix functions with the set of parameter ξ-trajectories given by the compact set $\mathcal{P}$ and the ξ vector.

Let $\pi \in \mathcal{F}_{\mathcal{P}}^{\xi}$ be a specific trajectory of the parameter. G_π

$$G_\pi \equiv \left[\begin{array}{c|c} A(\pi) & B(\pi) \\ \hline C(\pi) & D(\pi) \end{array}\right] \tag{10.4}$$

will denote the linear time-varying (LTV) system whose dynamics are described by Equation (10.2). For zero initial conditions, with $\Phi(t, t_0)$ being the corresponding transition matrix, the system of Equation (10.2), G_π represents a linear operator as follows:

$$\begin{aligned} G_\pi :&\mathcal{L}_2 \to \mathcal{L}_2 \\ z =&G_\pi w \\ z(t) = &\int_{t_0}^{t} C(\pi(\tau))\Phi(\tau, t_0)B(\pi(\tau))w(\tau)d\tau + Dw(t) \end{aligned} \tag{10.5}$$

From these definitions, the LPV system is defined as the set of LTV systems that make the LPV:

$$G_{\mathcal{F}_{\mathcal{P}}^{\xi}} = \{G_\pi : \pi \in \mathcal{F}_{\mathcal{P}}^{\xi}\}$$

Finally, the S matrix function will be defined as follows. Given a trajectory $\pi \in \mathcal{F}_{\mathcal{P}}^{\xi}$, $S(\pi(t)) \in \mathbb{R}^{(n+n_z)\times(n+n_w)}$. In all cases the S matrix will be defined by the A, B, C and D matrices arranged as follows:

$$S \equiv \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{10.6}$$

S_π will be to G_π what $S_{\mathcal{P}_\xi}$ is to $G_{\mathcal{P}_\xi}$. We remark that an LPV system is a set of LTV systems.

π-Quadratic Stability

Consider now the LPV dynamics without input:

$$\dot{x} = A(\pi)x \qquad\qquad x(0) = x_0 \tag{10.7}$$

with $\pi \in \mathcal{F}_{\mathcal{P}}^{\xi}$. A functional V is defined now as $V(x) = x^T P(\pi)x$ with $P(\pi(t)) \in \mathbb{R}^{n\times n}$ and with $P(z)$ being an symmetric positive-definite matrix for all $z \in \mathcal{P}$. For any $\pi \in \mathcal{F}_{\mathcal{P}}^{\xi}$, being $x(t)$ a state trajectory for Equation (10.7), $v(t) = V(x(t))$ is the functional V evaluated on the trajectory of the state. With this notation in hand, the time derivative of v will then be

$$\dot{v} = x^T[A^T(\pi)P(\pi) + P(\pi)A(\pi)]x + \sum_1^s \dot{\pi}_i \frac{\partial P(\pi)}{\partial \pi_i}$$

The dependence of v, x and π on time has not been made explicit for the sake of clarity.

Definition 3. *The function A is π-quadratically stable over $\mathcal{P}_\xi$ (or π-QS over $\mathcal{P}_\xi$) if there exists a matrix function $P : \mathbb{R}^s \mapsto \mathbb{R}^{n\times n}$ such that for all $\pi \in \mathcal{P} \subset \mathbb{R}^s$ and for any $\nu = \begin{bmatrix}\nu_1 & \nu_2 & \dots & \nu_s\end{bmatrix}$, $|\nu_i| \leq \xi_i$ the following inequality is satisfied:*

$$x^T[A^T(\pi)P(\pi) + P(\pi)A(\pi)]x + \sum_1^s \nu_i \frac{\partial P}{\partial \pi_i} < 0$$

It can be proved ([25]), that if the last inequality is satisfied, the system of Equation (10.7) is exponentially stable for any $\pi \in \mathcal{F}_{\mathcal{P}}^{\xi}$.

Given the linear operator defined by an LTV system (see Equation (10.5)), and based upon the standard definition of the $\mathcal{L}_2$ induced norm for an operator, the supremum over all parameter trajectories is defined in order to use it as a tool to evaluate performance of LPV systems. The following lemma is presented first.

Lemma 1. *Given a strictly proper (its D matrix is zero) π-QS LPV system $G_{\mathcal{P}_\xi}$, there exists a positive number M, such that*

$$\sup_{\pi \in \mathcal{F}_{\mathcal{P}}^{\xi}} \|G_\pi\|_\infty \leq M < \infty$$
$$\|G_\pi\|_\infty = \sup_{\|w\|_2=1} \|G_\pi.w\|_2$$

Proof. See [25].

Definition 4 (Norm of an LPV system). *The norm of a π-QS LPV system is defined as*

$$\|G_{\mathcal{P}_\xi}\| = \sup_{\pi \in \mathcal{F}_{\mathcal{P}}^{\xi}} \sup_{\|w\|_2=1} \|G_\pi.w\|_2$$

Next a lemma that gives a sufficient analysis condition for the stability and performance of LPV systems based upon a PDLF is presented.

Lemma 2. *[Bounded real lemma for LPV systems] Let $\mathcal{P} \subset \mathbb{R}^s$ be a compact set and let $\{\xi_i\}_1^s$ be a set of non-negative numbers. If there exists a continuously differentiable function $X : \mathbb{R}^s \mapsto \mathbb{R}^{n \times n}$ with $X(\pi)$ symmetric and positive-definite for all $\pi \in \mathcal{P}$ and if the following inequality is satisfied*

$$\begin{bmatrix} A^T(\pi)X(\pi) + X(\pi)A(\pi) + \sum_1^s \nu_i \frac{\partial X}{\partial \pi_i} & X(\pi)B(\pi) & C^T(\pi) \\ B^T(\pi)X(\pi) & -\gamma I & D^T(\pi) \\ C(\pi) & D(\pi) & -\gamma I \end{bmatrix} < 0 \tag{10.8}$$

for all $\pi \in \mathcal{P}$ and for all $|\nu_i| \leq \xi_i$, with $i = 1, 2, \ldots, s$, then the LPV system $G_{\mathcal{P}_\xi}$ is π-QS. Moreover the LPV norm of the system is $\|G_{\mathcal{P}_\xi}\| < \gamma$.

Proof. See [25].

10.1.4 The Control Synthesis Problem

Let the plant G be a finite n-dimensional LPV dynamical system, mapping disturbance inputs w and control inputs u to performance outputs z and measured outputs y, its dynamics being ruled by the following state-space equation:

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A(\pi) & B_1(\pi) & B_2(\pi) \\ C_1(\pi) & D_{11}(\pi) & D_{12}(\pi) \\ C_2(\pi) & D_{21}(\pi) & D_{22}(\pi) \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix} \tag{10.9}$$

Without lost of generality it is assumed that the D_{22} matrix is zero, a condition that can be overcome through standard *loop-shifting* techniques (see [3]).

The γ-performance LPV control problem seeks an LPV controller $K_{\mathcal{P}_\xi}$,

$$K_{\mathcal{P}_\xi} \equiv \left[\begin{array}{c|c} A_k(\cdot) & B_k(\cdot) \\ \hline C_k(\cdot) & D_k(\cdot) \end{array}\right] \tag{10.10}$$

such that for the closed-loop system

$$G^{cl}_{\mathcal{P}_\xi} \equiv \left[\begin{array}{c|c} A_{cl}(\cdot) & B_{cl}(\cdot) \\ \hline C_{cl}(\cdot) & D_{cl}(\cdot) \end{array}\right] \tag{10.11}$$

with

$$\begin{aligned} A_{cl} &= \begin{bmatrix} A + B_2 D_k C_2 & B_2 C_k \\ B_k C_2 & A_k \end{bmatrix} \\ B_{cl} &= \begin{bmatrix} B_1 + C_2 D_k D_{21} \\ B_k D_{21} \end{bmatrix} \\ C_{cl} &= \begin{bmatrix} C_1 + D_{12} D_k C_2 & D_{12} C_k \end{bmatrix} \\ D_{cl} &= D_{11} + D_{12} D_k D_{21} \end{aligned}$$

the following analysis LMI is feasible

$$\begin{bmatrix} A_{cl}^T(\pi) X_\infty(\pi) + X_\infty(\pi) A_{cl}(\pi) + \sum_1^s \nu_i \frac{\partial X_\infty}{\partial \pi_i} & X_\infty(\pi) B_{cl}(\pi) & C_{cl}^T(\pi) \\ B_{cl}^T(\pi) X_\infty(\pi) & -\gamma I & D_{cl}^T(\pi) \\ C_{cl}(\pi) & D_{cl}(\pi) & -\gamma I \end{bmatrix} < 0$$

for some matrix function $X_\infty : \mathbb{R}^s \to \mathbb{R}^{(n+n_k)\times(n+n_k)}$ with $X_\infty(\pi) > 0 \; \forall \pi \in \mathbb{R}^s$. Based upon Lemma 2, if this function exists, the closed-loop system is π-QS and the norm of the system is $\|G^{cl}_{\mathcal{P}_\xi}\| < \gamma$ (see Figure 10.1).

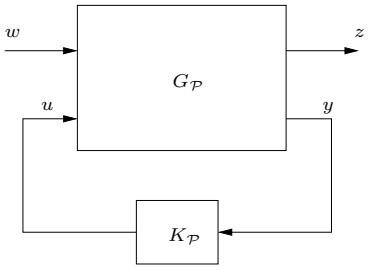


Fig. 10.1. Feedback interconnection of plant and controller

Solution

The problem has been stated in [25] where it is defined as *γ-performance and ξ-variation LPV synthesis problem*. From the analysis condition for the closed-loop system and through algebraic manipulation, sufficient synthesis conditions can be derived. This passage from the single LMI analysis condition to the three LMI synthesis condition based upon the open-loop model, is carried out similarly in four different references.

1. In [8] the discussion is for LTI systems. Beginning with the necessary and sufficient condition for closed-loop internal stability with $\mathcal{H}_\infty$ norm less than γ (bounded real lemma), the derivation reaches a necessary and sufficient condition based upon the open-loop data, for the existence of a controller solving the problem. The approach makes use of a quadratic Lyapunov function of the form $V(x) = x^T X_\infty x$ with $X_\infty \in \mathbb{R}^{n_{cl} \times n_{cl}}$ being a symmetric and positive-definite matrix with n_{cl} being the order of the closed-loop system. The requirements imposed on the open-loop plant are (see [8]) that the (A, B_2) pair be stabilizable, and the (A, C_2) pair be detectable with $D_{22} = 0$.
2. LPV systems with parameter trajectories in a set like $\mathcal{F}_\mathcal{P}^\xi$ are addressed in [25]. Beginning with a sufficient condition for the exponential internal stability of the set of LTV closed-loop mappings with $\mathcal{L}_2$-induced norm less than γ, a sufficient condition is reached that solves the problem of controller existence based upon the open-loop information. The approach is based upon a family of quadratic PDLFs of the form $V(x) = x^T X_\infty(\pi) x$ with $X_\infty(\pi(t)) \in \mathbb{R}^{n_{cl} \times n_{cl}}$ symmetric and positive-definite. It must be pointed out that the bound on the maximum rate of variation of the parameter is taken into account.
3. Reference [4], which is prior to [25], is similar to the former as it concerns the algebraic manipulations that are performed to derive the synthesis LMIs starting from a closed-loop analysis LMI. The main difference between the two is in the kind of Lyapunov function this approach employs. The fact that a SQLF as $V(x) = x^T X_\infty x$ is used, with X_∞ being a constant matrix, lead this approach to make no use of any bounds available on the maximum rate of change of the parameter, technically allowing for the parameter trajectories to be just piecewise continuous. The appealing aspect in such a Lyapunov function stems from the fact that it turns out to be easily tractable from the controller computation point of view. One the other hand, controller computation with the PDLF approach is more difficult. As will be shown in the corollary of Theorem 1, if the dependence of the state-space matrices of the open-loop augmented plant with the parameter is affine, and if the compact set $\mathcal{P}$ is a polytope defined by a finite number of vertices, the synthesis of LPV controllers can be carried out by solving a feasibility problem with a finite number of LMIs.
4. A number of software packages that solve LMI problems are available (see, for instance, [9, 23, 24]). In [9], the approach adopted is also through a SQLF. With the algebraic manipulations used to obtain the synthesis LMIs, being similar to those in [8] for LTI systems, it is necessary to restrict the B_2, C_2, D_{12} and D_{21} matrices of the open-loop augmented plant to be constant. This does not pose any serious restriction as it can be overcome through low-pass filtering at the plant's input or output at the expense of increasing the number of states of the augmented plant ([9]).

In the following section, the synthesis theorem that follows the approach of [9] is presented as it is among the tools employed for the design of the controller for the magnetic-bearing system.

10.1.5 Single Lyapunov Function Analysis

Definition 5 ($\mathcal{F}_\mathcal{P}$ trajectory set.). *Let $\mathcal{P} \subset \mathbb{R}^s$ be a compact set. $\mathcal{C}(\mathbb{R}, \mathbb{R}^s)$ is the set of piecewise continuous functions with domain in $\mathbb{R}$ and codomain in $\mathbb{R}^s$. The parameter trajectory set $\mathcal{F}_\mathcal{P}$ is defined as*

$$\mathcal{F}_\mathcal{P} \triangleq \{\pi \in \mathcal{C}(\mathbb{R}, \mathbb{R}^s) : \pi(t) \in \mathcal{P} \quad \forall t \in \mathbb{R}_+\}$$

Definition 2 is valid in this context with $\pi \in \mathcal{F}_\mathcal{P}$. The notation $G_\mathcal{P}$ is equivalent to $G_{\mathcal{P}_\xi}$ of Equation (10.3). Let π be a trajectory with $\pi \in \mathcal{F}_\mathcal{P}$. In this context we refer to G_π in the same way as in Equation (10.4). G_π represents a linear operator in the same way as in Equation (10.5). Again, we have the set of LTV systems

$$G_{\mathcal{F}_\mathcal{P}} = \{G_\pi : \pi \in \mathcal{F}_\mathcal{P}\}$$

S is as in Equation (10.6) but with $\pi \in \mathcal{F}_\mathcal{P}$.

Quadratic Stability

Consider now the LPV system with no input of Equation (10.7) but with $\pi \in \mathcal{F}_\mathcal{P}$. The functional V is defined as $V(x) = x^T P x$ with $P \in \mathbb{R}^{n\times n}$ symmetric and positive-definite. For any $\pi \in \mathcal{F}_\mathcal{P}$, with $x(t)$ a state trajectory of (10.7), $v(t) = V(x(t))$ denotes again the V functional evaluated on the state trajectory. The time derivative of v is as a consequence

$$\dot{v} = x^T[A^T(\pi)P + PA(\pi)]x$$

Definition 6. *The function A is quadratically stable on $\mathcal{P}$ (or QS on $\mathcal{P}$) if there exists a matrix $P \in \mathbb{R}^{n\times n}$ with $P = P^T > 0$, such that for all $\pi \in \mathcal{P} \subset \mathbb{R}^s$ the following inequality*

$$x^T[A^T(\pi)P + PA(\pi)]x < 0$$

is satisfied.

It can be proved that if the last inequality is satisfied, the system (10.7) is exponentially stable for any $\pi \in \mathcal{F}_\mathcal{P}$.

From the standard definition of the $\mathcal{L}_2$-induced norm for a linear operator given by an LTV system, a norm is defined based upon the supremum over all parameter trajectories possible for an LPV system. The following lemma parallels Lemma 1 in this context.

Lemma 3. *Given a strictly proper QS LPV system $G_\mathcal{P}$, there exists a positive number M such that*

$$\sup_{\pi\in\mathcal{F}_\mathcal{P}} \|G_\pi\|_\infty \le M < \infty$$
$$\|G_\pi\|_\infty = \sup_{\|w\|_2=1} \|G_\pi\ w\|_2$$

Definition 7 (QS LPV system norm). *The norm of a QS LPV system is defined as*

$$\|G_\mathcal{P}\| = \sup_{\pi\in\mathcal{F}_\mathcal{P}} \sup_{\|w\|_2=1} \|G_\pi.w\|_2$$

Lemma 4. *[Bounded real lemma for QS LPV systems] Let $\mathcal{P} \subset \mathbb{R}^s$ be a compact set. If there exists a symmetric and positive-definite matrix $X \in \mathbb{R}^{n\times n}$ such that the following inequality is satisfied*

$$\begin{bmatrix} A^T(\pi)X + XA(\pi) & XB(\pi) & C^T(\pi) \\ B^T(\pi)X & -\gamma I & D^T(\pi) \\ C(\pi) & D(\pi) & -\gamma I \end{bmatrix} < 0 \tag{10.12}$$

for all $\pi \in \mathcal{P}$, then the LPV system $G_\mathcal{P}$ is QS over $\mathcal{P}$. Moreover, the norm is $\|G_\mathcal{P}\| < \gamma$.

It must be pointed out that the sufficiency condition for stability and norm boundedness given by Lemma 4 is a particular case of Lemma 2, restricting the Lyapunov matrix to be constant. We note in passing that in the corresponding LMI, the condition on the differential of X disappears. It stems from this and from the definition of the $\mathcal{F}_\mathcal{P}$ trajectory set, that the stability and performance condition obtained is strong, accepting discontinuous parameter trajectories.

10.1.6 Control-problem Statement

The statement of Section 10.1.4 is repeated here. The plant G is an n-dimensional LPV dynamical system, mapping disturbance input w and control action u onto performance output z and measured output y. It is described in state-space form by Equation (10.9). The following assumptions are made on the plant:

A1. $D_{22} = 0$.
A2. B_2, C_2, D_{12} y D_{21} are constant.

Let π be the parameter trajectory with $\pi \in \mathcal{F}_{\mathcal{P}}$. The γ-performance LPV control problem consists in finding an LPV controller $K_{\mathcal{P}}$ such as the one of Equation (10.10) such that for the closed-loop system $G_{\mathcal{P}}^{cl}$ of Equation (10.11) the following analysis LMI,

$$\begin{bmatrix} A_{cl}^T(\pi)X_\infty + X_\infty A_{cl}(\pi) & X_\infty B_{cl}(\pi) & C_{cl}^T(\pi) \\ B_{cl}^T(\pi)X_\infty & -\gamma I & D_{cl}^T(\pi) \\ C_{cl}(\pi) & D_{cl}(\pi) & -\gamma I \end{bmatrix} < 0 \tag{10.13}$$

is feasible for some symmetric matrix $X_\infty \in \mathbb{R}^{(n+n_k)\times(n+n_k)}$ with $X_\infty > 0$. Because of the analysis theorem, if there exists such a matrix the closed-loop system is QS and its norm is $\|G_{\mathcal{P}}^{cl}\| < \gamma$.

10.1.7 Solution of the γ-performance Problem

Theorem 1. *Let G be the plant of Equation (10.9) that satisfies requirements **A1** and **A2**. $\mathcal{N}_R$ is a basis of the kernel of the $\begin{bmatrix} B_2^T & D_{12}^T & 0 \end{bmatrix}$ matrix and $\mathcal{N}_S$ is a basis of the kernel of the $\begin{bmatrix} C_2 & D_{21} & 0 \end{bmatrix}$ matrix. If there exist two symmetric positive-definite matrices, R and $S \in \mathbb{R}^{n\times n}$ such that the following LMI system*

$$\mathcal{N}_R^T \begin{bmatrix} AR + RA^T & RC_1^T & B_1 \\ C_1 R & -\gamma I & D_{11} \\ B_1^T & D_{11}^T & -\gamma I \end{bmatrix} \mathcal{N}_R < 0 \tag{10.14}$$

$$\mathcal{N}_S^T \begin{bmatrix} A^T S + SA & SB_1 & C_1^T \\ B_1^T S & -\gamma I & D_{11}^T \\ C_1 & D_{11} & -\gamma I \end{bmatrix} \mathcal{N}_S < 0 \tag{10.15}$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \geq 0 \tag{10.16}$$

is satisfied for all $\pi \in \mathbb{R}^s$, then there exists an LPV controller $K_{\mathcal{P}}$ such as the one in Equation (10.10) such that the closed-loop system $G_{\mathcal{P}}^{cl}$ is QS. Moreover, the norm of the closed-loop system is $\|G_{\mathcal{P}}^{cl}\| < \gamma$.

Comments on the Proof. In [8] the equivalence between LMI (10.13) and the system of LMIs (10.14) to (10.16) is proved for the LTI case, *i.e.* for each point in $\mathcal{P}$ that renders sufficiency for the analysis lemma. In [4] the following is pointed out:

LMIs (10.14), (10.15) and (10.16) are convex constraints on the R and S matrices. Moreover, given that the dependence of these inequalities on $\pi \in \mathcal{P}$ is continuous, the solution to the LPV γ-performance problem ends up stated in terms of a convex LMI feasibility problem

with an infinite number of constraints (as $\mathcal{P}$ is infinite). Gridding the parameter set $\mathcal{P}$ is the typical approach employed in order to solve a finite approximation to this problem with an infinite number of constraints. If a solution cannot be found on a finite grid of $\mathcal{P}$, it is clear that there is no solution for all $\mathcal{P}$. On the other hand, if there exists a solution for the grid, there is no guarantee that it is valid for all $\mathcal{P}$. In this case, the practical approach intends to have the LMI problem solved on a dense grid of $\mathcal{P}$. As the number of parameters grows, the number of points in the grid grows exponentially. Reference [4] points out that reasonable results are expected for no more than four parameters, *i.e.* $\pi \in \mathcal{P} \subset \mathbb{R}^4$.

10.1.8 Affine Systems

As previously anticipated, additional constraints render a convex feasibility problem with a finite number of LMIs. In fact, this is the case if the $\mathcal{P}$ set is restricted to be a convex polytope with a finite number of vertices and if the dependence of the augmented open-loop plant's state-space matrices on the parameter is affine. This means that the corresponding system matrix $S(\pi)$ (see Equation (10.6)) depends on the parameter as follows:

$$\begin{aligned} S(\pi) &= S_0 + \pi_1 S_1 + \ldots + \pi_s S_s \\ \pi(t) &\in \Theta = \mathcal{C}_o\{\pi_1, \ldots, \pi_p\} \end{aligned}$$

with $\mathcal{C}_o$ being the convex hull of the set of vertices that define $\mathcal{P}$ and with the S_i system matrices being constant.

The affine LPV system can be equivalently given by the "*vertex systems*", *i.e.* the images of the p vertices that make up the set $\mathcal{P}$ as follows:

$$\begin{aligned} S(\pi(t)) &\in \mathcal{C}_o\{S_1, \ldots, S_p\} \\ S_i &= S(\pi_i) \qquad\qquad i = 1, \ldots, p \end{aligned}$$

in a way such that for every t the system matrix can be expressed as

$$\begin{aligned} S(\pi(t)) &= \sum_1^p S_i \eta_i(t) \quad \text{with} \\ \pi(t) &= \sum_1^p \pi_i \eta_i(t) \quad \text{and} \quad \sum_1^p \eta_i(t) = 1 \end{aligned} \tag{10.17}$$

Systems given in this form are referred to as polytopic systems with the η_i numbers depending on π being the polytopic coordinates of S. Having introduced affine systems the following corollary is presented.

Theorem 2. *Let $\mathcal{P} \subset \mathbb{R}^s$ be a convex polytope defined by its p vertices. G is the augmented plant of Equation (10.9) that satisfies conditions **A1** and **A2**. Moreover, the plant G is a polytopic system whose S matrix can be given as in Equation (10.17). $\mathcal{N}_R$ and $\mathcal{N}_S$ are the same as in Theorem 1. If there exist two symmetric and positive-definite matrices R and $S \in \mathbb{R}^{n \times n}$ such that the following system of $(2p+1)$ LMIs:*

$$\mathcal{N}_R^T \begin{bmatrix} A_i R + R A_i^T & R C_{1i}^T & B_{1i} \\ C_{1i} R & -\gamma I & D_{11i} \\ B_{1i}^T & D_{11i}^T & -\gamma I \end{bmatrix} \mathcal{N}_R < 0 \tag{10.18}$$

$$\mathcal{N}_S^T \begin{bmatrix} A_i^T S + S A_i & S B_{1i} & C_{1i}^T \\ B_{1i}^T S & -\gamma I & D_{11i}^T \\ C_{1i} & D_{11i} & -\gamma I \end{bmatrix} \mathcal{N}_S < 0 \tag{10.19}$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \geq 0 \tag{10.20}$$

is satisfied for $i = 1, \ldots, p$, then there exists an LPV polytopic controller $K_{\mathcal{P}}$ such that the closed-loop system is QS and such that the norm of the LPV system is $\|G_{\mathcal{P}}^{cl}\| < \gamma$.

The fact that this is an LMI convex feasibility problem with a finite number of inequalities is noted. The synthesis of LPV controllers based upon this theorem can be carried out with any of the aforementioned software packages ([9, 23, 24]). As far as the resulting controller is concerned, having the R and S matrices allows a Lyapunov matrix to be obtained for the closed-loop system. In turn, for each vertex, having the closed-loop Lyapunov matrix allows an LMI to be solved (see [8], page. 435, Equation (55)) the solution being a *vertex controller* (in [8] the solution is the LTI controller). When the LMI system is such that the LMI (10.20) is positive-definite, then the order of the controller is that of the augmented plant. As a consequence, with R and S in hand, a direct computation of the vertex controllers is possible without solving any LMI.

For every t, having a real-time measured value of the parameter π, and as a consequence, its η_i polytopic coordinates, the LPV controller state-space matrices are obtained by convexly combining the vertex controllers. This allows for digital implementation as follows:

$$\pi(t) = \sum_1^p \eta_i(t)\pi_i$$

$$S_k(\pi(t)) = \sum_1^p \eta_i(t) S_{k_i}$$

with S_{k_i} being the vertex components of the controller and π_i the vertices of $\mathcal{P}$.

The fact that controller system matrix $S_k(\pi(t))$ is updated in real time with the measured parameter π in hand is noted. We also remark that on the contrary, the vertex systems S_{k_i} are computed offline. This form of adaptation makes LPV control appealing for *gain scheduling*.

10.1.9 LPV Pole Placement

As shown in [6], LMIs allow for the statement of constraints on the location of the eigenvalues of square matrices. This allows us to state control problems with mixed objectives. The theory developed in [6] for LTI systems, deals in particular with mixed $\mathcal{H}_\infty$, $\mathcal{H}_2$ and pole-placement problems.

An extension of this approach to LPV systems will be presented in this section. It must be clear beforehand, that there is *no* such thing as *poles* of LTV systems. Nevertheless, the eigenvalue-clustering technique of [6] applied to LPV dynamics has proved useful in the solution of a particular kind of problem that eventually shows up when solving LPV control problems. The kind of inconvenience is sometimes referred to as the *fast poles problem* and is described in the following.

Suppose we have the LPV system given by the linear fractional interconnection of an augmented plant $G_{\mathcal{P}}$ and its corresponding controller $K_{\mathcal{P}}$, with $\mathcal{P}$ being the parameter-variation

set (see Figure 10.1). For each point in $\mathcal{P}$, the LTI system resulting from holding the parameter π fixed, sometimes presents a usually small number of poles (two or four) whose real-part modulus is much greater than the corresponding quantity of the rest of the closed-loop poles (the condition of "*fast*" with respect to a pole depends on the particular application and the practical know-how there is about it).

Fast poles bring up two problems from the practical point of view. Both problems have to do with the numerical integration of a differential equation with both slow and fast states. For the simulation phase as well as for the digital implementation of the controller, integration is necessary and becomes difficult with these fast dynamics.

The approach used to deal with these difficulties is *LPV pole placement*. Through LMI constraints, its objective is to keep the poles of each LTI closed-loop system resulting from holding the parameter fixed at each point of the parameter-variation set, in a prescribed region of the complex plane.

Analysis Conditions

We begin with the definitions of LMI region and matrix $\mathcal{D}$-stability.

Definition 8. ***LMI Region*** *A region in the complex plane $\mathcal{D}$, is called an* LMI region *if there exists a symmetric matrix $\alpha = [\alpha_{kl}] \in \mathbb{R}^{m\times m}$ and a matrix $\beta = [\beta_{kl}] \in \mathbb{R}^{m\times m}$ such that*

$$\mathcal{D} = \{z \in \mathbb{C} : \ f_{\mathcal{D}}(z) < 0\}$$

with

$$f_{\mathcal{D}}(z) = \alpha + z\beta + \bar{z}\beta^T = [\alpha_{kl} + \beta_{kl} z + \beta_{lk}\bar{z}]_{1\leq k,l\leq m} \tag{10.21}$$

A more detailed discussion on LMI regions can be found in [6]. It must be pointed out that these regions are a dense subset in the set of regions of the complex plane that are symmetric with respect to the real axis. This condition makes them usefull to specify requirements on the location of the eigenvalues of any real square matrix. The definition of $\mathcal{D}$-*Stability* and a lemma that gives an LMI condition to check $\mathcal{D}$-stability of a square matrix are next.

Definition 9. *A matrix $A \in \mathbb{R}^{n\times n}$ is* $\mathcal{D}$-stable *if and only if its eigenvalues are in an LMI region $\mathcal{D}$*

Lemma 5. *A matrix $A \in \mathbb{R}^{n\times n}$ is $\mathcal{D}$-stable if and only if there exists a symmetric positive-definite matrix $X_{\mathcal{D}}$ such that*

$$\left[\alpha_{kl} X_{\mathcal{D}} + \beta_{kl} A X_{\mathcal{D}} + \beta_{lk} X_{\mathcal{D}} A^T\right]_{1\leq k,l\leq m} < 0$$

Now we present a lemma for the $\mathcal{D}$-stability of a matrix function $A : \mathcal{P} \subset \mathbb{R}^s \to \mathbb{R}^{n\times n}$ in each point of $\mathcal{P}$.

Lemma 6. *A matrix function $A : \mathcal{P} \subset \mathbb{R}^s \to \mathbb{R}^{n\times n}$ is $\mathcal{D}$-stable for all $\pi \in \mathcal{P}$ if there exists a matrix function $X_{\mathcal{D}} : \mathcal{P} \subset \mathbb{R}^s \to \mathbb{R}^{n\times n}$ with $X_{\mathcal{D}}(\pi)$ symmetric and positive-definite for each $\pi \in \mathcal{P}$ such that*

$$\left[\alpha_{kl} X_{\mathcal{D}}(\pi) + \beta_{kl} A(\pi) X_{\mathcal{D}}(\pi) + \beta_{lk} X_{\mathcal{D}}(\pi) A(\pi)^T\right]_{1\leq n,m\leq n} < 0 \quad \forall \pi \in \mathcal{P}$$

It is noted that this lemma gives an LMI-based analysis condition for the transient response of an LPV system. With this in hand and with a set of bounds on the parameter-variation rate, aspects concerning the implementation phase can be evaluated as the necessary sampling rate of the digital controller. Next, the LPV γ-performance problem with pole-placement constraints is presented.

Let $\mathcal{D}$ be a region of the open left-hand side complex plane defined by Equation (10.21) and let γ be a positive number. Given the LPV plant of Equation (10.9), an LPV controller such as the one in Equation (10.10) is sought with the closed-loop system being such as the one in Equation (10.11). If there is a controller such that there exist two matrix functions $X_\infty : \mathbb{R}^s \to \mathbb{R}^{(n+n_k)\times(n+n_k)}$ and $X_\mathcal{D} : \mathbb{R}^s \to \mathbb{R}^{(n+n_k)\times(n+n_k)}$, such that for all $\pi \in \mathcal{P}$ the following LMIs

$$\begin{bmatrix} A_{cl}^T(\pi)X_\infty(\pi) + X_\infty(\pi)A_{cl}(\pi) + \sum_1^s \nu_i \frac{\partial X_\infty}{\partial \pi_i} & X_\infty(\pi)B_{cl}(\pi) & C_{cl}^T(\pi) \\ B_{cl}^T(\pi)X_\infty(\pi) & -\gamma I & D_{cl}^T(\pi) \\ C_{cl}(\pi) & D_{cl}(\pi) & -\gamma I \end{bmatrix} < 0$$

$$\left[\alpha_{kl}X_\mathcal{D}(\pi) + \beta_{kl}A_{cl}(\pi)X_\mathcal{D}(\pi) + \beta_{lk}X_\mathcal{D}(\pi)A_{cl}(\pi)^T\right] < 0 \tag{10.22}$$

$$\forall k, l \in [1, m]$$

are satisfied, then the closed-loop system is π-QS with its norm being less than γ. Moreover, the closed-loop system is $\mathcal{D}$-stable for all $\pi \in \mathcal{P}$.

As well as in Section 10.1.6, a single quadratic Lyapunov function approach is used in this case, in order to have a tractable problem. As in the case of passing from the LPV π-QS control problem to the LPV QS control problem and following the arguments in [6] for LTI systems, the analysis problem is simplified by forcing the functions

$$X_\infty : \mathbb{R}^s \to \mathbb{R}^{(n+n_k)\times(n+n_k)} \quad y \quad X_\mathcal{D} : \mathbb{R}^s \to \mathbb{R}^{(n+n_k)\times(n+n_k)}$$

to be the same constant matrix,

$$X_\infty(\pi) = X_\mathcal{D}(\pi) = X_{cl} \in \mathbb{R}^{(n+n_k)\times(n+n_k)} \qquad \forall \pi \in \mathcal{P}$$

It is assumed again (with almost no loss of generality) that the B_2, C_2, D_{12} and D_{21} matrices are constant for convexity and that $D_{22} = 0$.

The X_{cl} and X_{cl}^{-1} matrices are block-partitioned as follows

$$X_{cl} = \begin{bmatrix} R & M \\ M^T & U \end{bmatrix} \quad , \quad X_{cl}^{-1} = \begin{bmatrix} S & N \\ N^T & V \end{bmatrix}$$

A change of variables in the controller matrices is carried out as follows:

$$\begin{aligned} \mathcal{B}_k(\pi) =& NB_k(\pi) + SB_2D_k(\pi) \\ \mathcal{C}_k(\pi) =& C_k(\pi)M^T + D_k(\pi)C_2R \\ \mathcal{A}_k(\pi) =& NA_k(\pi)M^T + NB_k(\pi)C_2R \\ &+ SB_2C_k(\pi)M^T + S(A(\pi) + B_2D_k(\pi)C_2)R \end{aligned} \tag{10.23}$$

Note that having restricted the B_2, C_2, D_{12} and D_{12} matrices to be constant, the functional dependence of the new $\mathcal{A}_k$, $\mathcal{B}_k$ and $\mathcal{C}_k$ variables with the parameter is similar to that of the

original A_k, B_k and C_k variables. With affine systems in mind, this means that if the latter depend affinely on π so will the former.

If, as mentioned in [6] for the LTI case, M and N are invertible, given the $\mathcal{A}_k(\cdot)$, $\mathcal{B}_k(\cdot)$ $\mathcal{C}_k(\cdot)$ and $D_k(\cdot)$ functions with R and S matrices, the controller is uniquely defined. Next, the synthesis theorem is presented.

Theorem 3. *Let $\mathcal{D}$ be an LMI region in the open left-hand side complex plane, with its characteristic function as in Equation (10.21). The LPV γ-performance problem with pole-placement constraints has a solution if the following LMI system*

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} > 0 \tag{10.24}$$

$$\left[\alpha_{kl} \begin{bmatrix} R & I \\ I & S \end{bmatrix} + \beta_{kl}\Phi(\pi) + \beta_{lk}\Phi^T(\pi)\right] < 0 \qquad \forall\, k, l = 1, ..., m \tag{10.25}$$

$$\begin{bmatrix} \Psi_{11}(\pi) & \Psi_{21}^T(\pi) \\ \Psi_{21}(\pi) & \Psi_{22}(\pi) \end{bmatrix} < 0 \tag{10.26}$$

is feasible for some set of matrix functions
$\mathcal{A}_k : \mathbb{R}^s \to \mathbb{R}^{n_k \times n_k}$
$\mathcal{B}_k : \mathbb{R}^s \to \mathbb{R}^{n_k \times n_y}$
$\mathcal{C}_k : \mathbb{R}^s \to \mathbb{R}^{n_u \times n_k}$
$D_k : \mathbb{R}^s \to \mathbb{R}^{n_u \times n_y}$
and a pair of matrices R and S, for all $\pi \in \mathcal{P}$. R and S are symmetric and positive-definite in $\mathbb{R}^{n \times n}$. The $\Phi(\cdot)$, $\Psi_{11}(\cdot)$, $\Psi_{21}(\cdot)$ and $\Psi_{22}(\cdot)$ matrix functions are as follows:

$$\Phi = \begin{bmatrix} AR + B_2C_k & A + B_2D_kC_2 \\ \mathcal{A}_k & SA + \mathcal{B}_kC_2 \end{bmatrix} \tag{10.27}$$

$$\Psi_{11} = \begin{bmatrix} AR + RA^T + B_2\mathcal{C}_k + (B_2\mathcal{C}_k)^T & B_1 + B_2D_kD_{21} \\ (B_1 + B_2D_kD_{21})^T & -\gamma I \end{bmatrix}$$

$$\Psi_{21} = \begin{bmatrix} \mathcal{A}_k + (A + B_2D_kC_2)^T & SB_1 + \mathcal{B}D_{21} \\ C_1R + D_{12}\mathcal{C}_k & D_{11} + D_{12}D_kD_{21} \end{bmatrix}$$

$$\Psi_{22} = \begin{bmatrix} SA + A^TS + \mathcal{B}_kC_2 + (\mathcal{B}_kC_2)^T & (C_1 + D_{12}D_kC_2)^T \\ C_1 + D_{12}D_kC_2 & -\gamma I \end{bmatrix}$$

where the dependence of the corresponding matrices with the parameter π has been omitted for the sake of clear notation. The resulting controller is of order $n_k = n$.

As in the case of the LPV γ-performance control problem, if the augmented open-loop plant's matrices depend affinely on the parameter and if $\mathcal{P}$ is a convex polytope, the problem can be stated in terms of the vertex systems that make up the affine representation. The resulting optimization problem is in terms of a finite number of LMIs. The complete LMI problem consists in LMI (10.24) plus the $m^2 + 1$ LMIs in Equations (10.25) and (10.26) for each vertex. This could be a very large number.

The solution sought will be in terms of a pair of constant matrices R and S, as well as a set of matrices $\{\mathcal{A}_i, \mathcal{B}_i, \mathcal{C}_i, D_i\}$ per vertex. Given that LMI (10.24) is positive-definite, the sets of matrices $\{\mathcal{A}_i, \mathcal{B}_i, \mathcal{C}_i, D_i\}$ directly define the *vertex controllers* that make up the polytopic controller.

10.2 The Magnetic-bearing Experiment

In this section, a model for the dynamics of the rotor of an MBC500 is presented. A photograph of the MBC500 can be seen in Figure 10.2.

The rotor whose dynamics we intend to describe is the horizontal shaft that can be seen in the photograph. In this system, the stainless steel shaft or rotor can be levitated using eight "horseshoe" electromagnets, four at each end of the rotor. This is described in the diagram of Figure 10.3 for the horizontal dynamics. Hall-effect sensors placed just outside of the electromagnets at each end of the rotor measure the rotor-end displacement. As the diagram shows (Figure 10.3), the displacement from equilibrium measured by the sensors is given by x_1 (y_1) and x_2 (y_2), and the displacement of the rotor with respect to the electromagnets is given by the X_1 (Y_1) and X_2 (Y_2) for the horizontal (vertical) dynamics. The values of the parameters of the shaft according to [12, 16] are reproduced in Table 10.1.

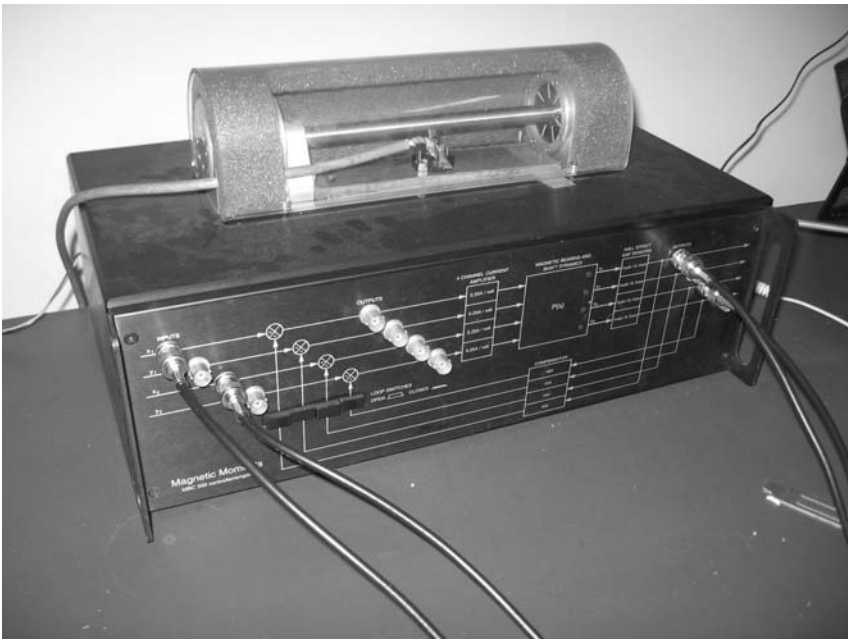


Fig. 10.2. The MBC500

First, the 6-DOF dynamics of the rotor as a rigid body will be considered to derive afterwards the 4-DOF model for control, taking into account the equations describing the sensors' behavior as well as the magnetic forces acting on the rotor to stabilize it.

Table 10.1. Parameters of the MBC500

Symbol	Description	Value
L	Total length of the rotor	0.269 m
l	Distance from each bearing to the end of the rotor	0.024 m
l_2	Distance from each Hall-effect sensor to the end of the rotor	0.0028 m
I	Moment of inertia of the rotor with respect to rotation about the x- and y-axes	1.5884×10^{-3} kg m^2
m	Mass of the rotor	0.2629 kg

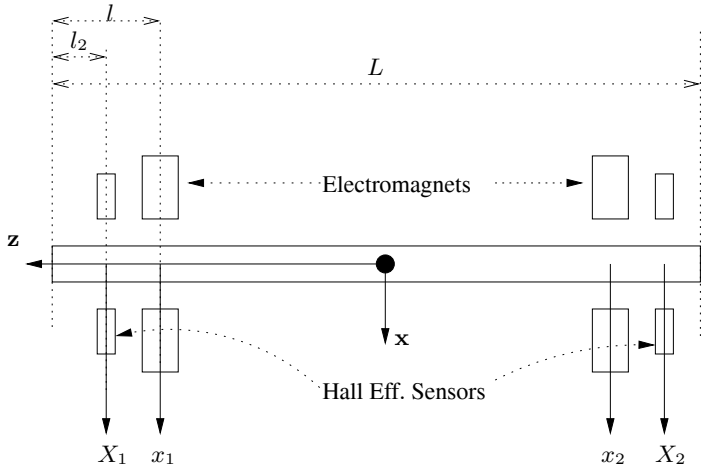


Fig. 10.3. Top view of the MBC500

10.2.1 4-DOF Rigid Model

In order to define the kinematic variables of the rotor, we begin with the definition of reference frame $\mathcal{I}$. This frame is fixed with respect to the MBC500's body. In the ideal case, when the rotor is at equilibrium, the output of the sensors is zero and the rotor's CM coincides with the origin of frame $\mathcal{I}$. In this case the z-axis of the frame is aligned with the longitudinal axis of the rotor as well. The x-axis completes the definition of the horizontal plane, and the y-axis points upwards.

Frame $\mathcal{B}$ is fixed with respect to the rotor's body. The rotation and displacement of frame $\mathcal{B}$ with respect to frame $\mathcal{I}$ completely describes the state of the rotor. As the roll angle of the rotor does not matter to the problem, frame $\tilde{\mathcal{B}}$ is defined with its origin being at the rotor's CM. The z-axes of frames $\tilde{\mathcal{B}}$ and $\mathcal{B}$ coincide (see Figure 10.4). Frames $\tilde{\mathcal{B}}$ and $\mathcal{B}$ are just rotated with respect to one another over the longitudinal axis of the rotor. In equilibrium, frames $\tilde{\mathcal{B}}$ and $\mathcal{I}$

coincide even if the rotor is spinning with respect to its z-axis. The orientation of frame $\tilde{\mathcal{B}}$ with respect to frame $\mathcal{I}$ (*i.e.* the orientation of the revolving body) will be described using just two Euler angles (Figure 10.5 shows displacement and rotation of frame $\tilde{\mathcal{B}}$ with respect to frame $\mathcal{I}$ for the horizontal dynamics). The ϕ (θ) angle and the horizontal (vertical) displacement of the CM x_o (y_o) can be determined from the output of the horizontal (vertical) gap sensors. As the rotor's CM motion along its z-axis is limited by the bearings themselves only two

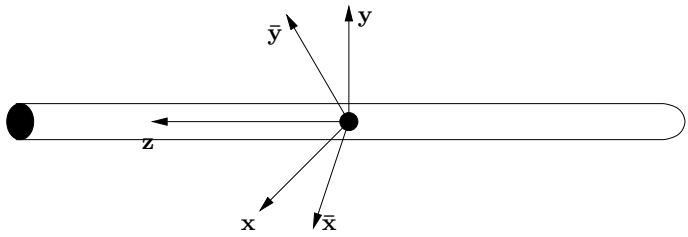


Fig. 10.4. Frames $\tilde{\mathcal{B}}$ and $\mathcal{B}$

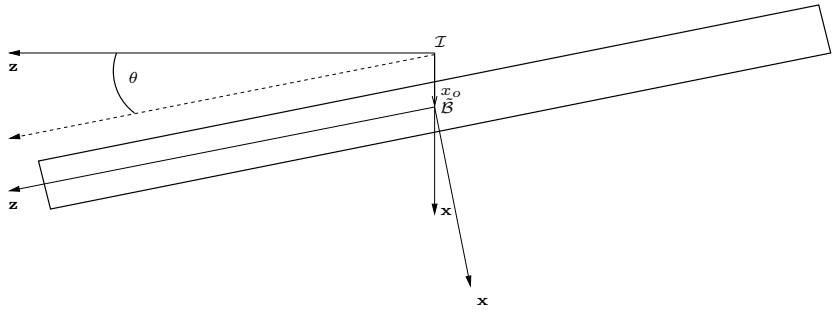


Fig. 10.5. Frames $\tilde{\mathcal{B}}$ and $\mathcal{I}$. Top view.

displacement variables are necessary for the equations.

As is known, angular momentum and angular velocity are related in frame $\mathcal{B}$ by the inertia tensor J:

$$h^b = J\omega_{bi}^b \tag{10.28}$$

where h^b is the angular momentum, ω_{bi}^b is the angular velocity of the rotor with respect to frame $\mathcal{I}$. Due to the symmetry of the rotor with respect to rotations in the x- and y-axes, it can be shown that the inertia with respect to frame $\tilde{\mathcal{B}}$ is constant and equals the inertia tensor with respect to frame $\mathcal{B}$. As a consequence, Equation (10.28) is valid in frame $\tilde{\mathcal{B}}$ as well:

$$h^{\tilde{b}} = J\omega_{bi}^{\tilde{b}}$$

Let $\frac{d}{dt^{\mathcal{I}}}$ denote the derivative with respect to frame $\mathcal{I}$. The reader is assumed to be familiar with derivatives with respect to different frames. The following expression relates the derivatives with respect to frame $\mathcal{B}$ and $\mathcal{I}$. Given any physical vector u,

$$\frac{du}{dt^{\mathcal{B}}} = \frac{du}{dt^{\mathcal{I}}} + \omega_{bi} \times u \tag{10.29}$$

Let M be the resulting moment of the forces applied on the rotor with respect to its CM. The standard equation that relates M to the derivative of the body's angular momentum is:

$$M = \frac{dh}{dt^{\mathcal{I}}} = \frac{dh}{dt^{\tilde{\mathcal{B}}}} + \omega_{\tilde{b}i} \times h = J\frac{d\omega_{bi}}{dt^{\tilde{\mathcal{B}}}} + \omega_{\tilde{b}i} \times J\omega_{bi} \tag{10.30}$$

From the definition of the frames it follows that:

$$\omega_{bi} = \omega_{b\tilde{b}} + \omega_{\tilde{b}i}$$

Specifying the equations in frame $\tilde{\mathcal{B}}$ Equation (10.30) turns out to be:

$$\begin{aligned} M^{\tilde{b}} &= (\frac{dh}{dt^{\tilde{\mathcal{B}}}})^{\tilde{b}} + \omega_{\tilde{b}i}^{\tilde{b}} \times h^{\tilde{b}} \\ &= J\left(\frac{d\omega_{bi}}{dt^{\tilde{\mathcal{B}}}}\right)^{\tilde{b}} + \omega_{\tilde{b}i}^{\tilde{b}} \times J\omega_{bi}^{\tilde{b}} \\ &= J\left(\frac{d\omega_{bi}}{dt^{\tilde{\mathcal{B}}}}\right)^{\tilde{b}} + \omega_{\tilde{b}i}^{\tilde{b}} \times J\omega_{b\tilde{b}}^{\tilde{b}} \end{aligned} \tag{10.31}$$

From the definition of the frames we have:

$$\begin{aligned} \omega_{b\tilde{b}}^{\tilde{b}} &= \begin{bmatrix} 0 & 0 & r \end{bmatrix}^T \\ \omega_{\tilde{b}i}^{\tilde{b}} &= \begin{bmatrix} p & q & 0 \end{bmatrix}^T \\ M^{\tilde{b}} &= \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^T \end{aligned}$$

Hence, Equation (10.31) can be rewritten as:

$$\begin{aligned} M^{\tilde{b}} &= J\left(\frac{d\omega_{bi}}{dt^{\tilde{\mathcal{B}}}}\right)^{\tilde{b}} + \omega_{\tilde{b}i}^{\tilde{b}} \times J\omega_{b\tilde{b}}^{\tilde{b}} \\ \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} &= J\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} q \\ -p \\ 0 \end{bmatrix} I_z r \end{aligned} \tag{10.32}$$

Further manipulation of Equation (10.32) renders the following expression that is suitable for state-space treatment:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & r\frac{I_z}{I} & 0 \\ -r\frac{I_z}{I} & 0 & \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \frac{1}{I} & 0 & 0 \\ 0 & \frac{1}{I} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}$$

Newton's law for the spindle's CM renders the other equation for the dynamics:

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} - \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix}$$

with $v = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ being the CM's velocity in frame $\mathcal{I}$. $r_0 = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T$ is the CM's position in frame $\mathcal{I}$ with $\dot{r}_0 = v$. The rotation ${}^{\tilde{b}}C_i$ is given in terms of the quaternion $\begin{bmatrix} {}^{i\tilde{b}}q_0 & {}^{i\tilde{b}}q^{\tilde{b}} \end{bmatrix}$ by

$${}^{\tilde{b}}C_i = I_3 - 2\,({}^{i\tilde{b}}q_0)({}^{i\tilde{b}}q\times) + 2\,({}^{i\tilde{b}}q\times)({}^{i\tilde{b}}q\times)$$

The equation for the evolution of the quaternion is:

$$\begin{aligned} {}^{i\tilde{b}}\dot{q}_0 &= -\frac{1}{2}{}^{i\tilde{b}}q^{\tilde{b}} \cdot \omega_{\tilde{b}i}^{\tilde{b}} \\ {}^{i\tilde{b}}\dot{q}^{\tilde{b}} &= \frac{1}{2}({}^{i\tilde{b}}q_0 I + {}^{i\tilde{b}}q^{\tilde{b}}\times)\omega_{\tilde{b}i}^{\tilde{b}} \end{aligned}$$

Taking into account an hypothesis of small angles, we make use of the corresponding Euler angles with the following equivalences and approximations:

$$\begin{aligned} {}^{i\tilde{b}}q_0 &\approx 1 \\ {}^{i\tilde{b}}q &\approx \frac{1}{2}\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \\ \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T &= \begin{bmatrix} p & q & r \end{bmatrix}^T \end{aligned}$$

Gathering the previous expressions the state-space equation turns out to be:

$$\begin{aligned} \dot{x_0} &= v_x \\ \dot{\theta} &= q \\ \dot{v_x} &= \frac{1}{m}F_x \\ \dot{q} &= \frac{1}{I}M_y - r\frac{I_z}{I}p \end{aligned} \tag{10.33}$$

$$\begin{aligned} \dot{y_0} &= v_y \\ \dot{\phi} &= p \\ \dot{v_y} &= \frac{1}{m}F_y \\ \dot{p} &= \frac{1}{I}M_x + r\frac{I_z}{I}q \end{aligned} \tag{10.34}$$

It can be noted that the first four lines starting with Equation (10.33) describe the horizontal dynamics, whereas the last four starting with Equation (10.34) describe the vertical dynamics. Clearly, while the shaft is not spinning ($r = 0$), the horizontal and vertical dynamics are decoupled.

10.2.2 Forces, Moments and Input/Output Voltages

The equations for measurements, forces and moments follow from those in [14]. A few sign modifications take place, which account for discrepancies between the model of the cited reference and measurements taken from the particular MBC500 used here.

As we have mentioned previously, the x_1, x_2, y_1 and y_2 variables give the displacement of the rotor with respect to equilibrium at the ends of the shaft where the magnetic forces are applied (see Figure 10.3).

The variables in capital letters (X_1, X_2, Y_1, Y_2) refer to the displacement of the rotor with respect to equilibrium at the point where the Hall-effect sensors are placed. These variables are related to the kinematic variables as follows:

$$\begin{aligned} x_1 &= x_0 + (\frac{L}{2} - l)\sin(\theta) \\ x_2 &= -x_0 + (\frac{L}{2} - l)\sin(\theta) \\ X_1 &= x_0 + (\frac{L}{2} - l_2)\sin(\theta) \\ X_2 &= -x_0 + (\frac{L}{2} - l_2)\sin(\theta) \end{aligned} \tag{10.35}$$

$$\begin{aligned} y_1 &= y_0 - (\frac{L}{2} - l)\sin(\phi) \\ y_2 &= y_0 + (\frac{L}{2} - l)\sin(\phi) \\ Y_1 &= y_0 - (\frac{L}{2} - l_2)\sin(\phi) \\ Y_2 &= y_0 + (\frac{L}{2} - l_2)\sin(\phi) \end{aligned} \tag{10.36}$$

Let i_{x_1} , i_{x_2}, i_{y_1} and i_{y_2} be the currents of the electromagnets that exert the magnetic forces on the shaft. According to [12, 14, 16] the forces exerted by the electromagnets are given by the following expressions:

$$\begin{aligned} F_{x_1} &= +k\left[\left(\frac{i_{x_1}+0.5}{x_1 - 4\times10^{-4}}\right)^2 - \left(\frac{i_{x_1}-0.5}{x_1 + 4\times10^{-4}}\right)^2\right] \\ F_{x_2} &= -k\left[\left(\frac{i_{x_2}+0.5}{x_2 - 4\times10^{-4}}\right)^2 - \left(\frac{i_{x_2}-0.5}{x_2 + 4\times10^{-4}}\right)^2\right] \\ F_{y_1} &= +k\left[\left(\frac{i_{y_1}+0.5}{y_1 - 4\times10^{-4}}\right)^2 - \left(\frac{i_{y_1}-0.5}{y_1 + 4\times10^{-4}}\right)^2\right] \\ F_{y_2} &= +k\left[\left(\frac{i_{y_2}+0.5}{y_2 - 4\times10^{-4}}\right)^2 - \left(\frac{i_{y_2}-0.5}{y_2 + 4\times10^{-4}}\right)^2\right] \end{aligned} \tag{10.37}$$

where $k = 2.8\times10^{-7}\frac{\mathrm{N\,m^2}}{\mathrm{A^2}}$. As a result of the latter expression, the forces and moments exerted on the spindle are as follows:

$$\begin{aligned} F_x &= F_{x_1} + F_{x_2} \\ F_y &= F_{y_1} + F_{y_2} \\ M_y &= (\frac{L}{2} - l)F_{x_1} - (\frac{L}{2} - l)F_{x_2} \\ M_x &= -(\frac{L}{2} - l)F_{y_1} + (\frac{L}{2} - l)F_{y_2} \end{aligned} \tag{10.38}$$

According to [12, 14, 16] as well, the output voltages from the Hall-effect sensors are related to the displacement variables X_i and Y_i ($i = 1, 2$) by the following equality:

$$\begin{aligned} V^o_{x_i} &= 5 \times 10^3 X_i + 25 \times 10^9 X_i^3 \qquad i = 1, 2 \\ V^o_{y_i} &= 5 \times 10^3 Y_i + 25 \times 10^9 Y_i^3 \end{aligned} \tag{10.39}$$

10.2.3 Linearization and Change of Variables

In order to obtain a linear model suitable for control design, Equations (10.33) to (10.44) are gathered and linearized as follows. A small-angles approximation ($\sin(x) \simeq x$) is carried out in Equations (10.35) and (10.36). As a consequence, the following linear relations are established:

$$\begin{bmatrix} x_o \\ \theta \end{bmatrix} = T_{1h} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{10.40}$$

$$\begin{bmatrix} y_o \\ \phi \end{bmatrix} = T_{1v} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = T_{2h} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{10.41}$$

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = T_{2v} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

with

$$\begin{aligned} T_{1h} &= \begin{bmatrix} 1 & (\frac{L}{2} - l) \\ -1 & (\frac{L}{2} - l) \end{bmatrix}^{-1} \\ T_{1v} &= \begin{bmatrix} 1 & (\frac{L}{2} - l) \\ 1 & -(\frac{L}{2} - l) \end{bmatrix}^{-1} \\ T_{2h} &= \begin{bmatrix} 1 & (\frac{L}{2} - l_2) \\ -1 & (\frac{L}{2} - l_2) \end{bmatrix} T_{1h} \\ T_{2v} &= \begin{bmatrix} 1 & (\frac{L}{2} - l_2) \\ 1 & -(\frac{L}{2} - l_2) \end{bmatrix} T_{1v} \end{aligned} \tag{10.42}$$

We remark that the change of variables of Equation (10.40) is valid for the corresponding time derivatives as follows:

$$\begin{bmatrix} v_x \\ q \end{bmatrix} = T_{1h} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$

$$\begin{bmatrix} v_y \\ p \end{bmatrix} = T_{1v} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix}$$

To finish the process, we take partial derivatives in Equation (10.37) with respect to the currents i_{x_1} , i_{x_2}, i_{y_1} and i_{y_2} and the displacements x_1, x_2, y_1 and y_2. In Equation (10.39), first of all take partial derivatives with respect to X_1, X_2, Y_1 and Y_2 and then apply the change of variables (10.41) to obtain a linear approximation of Equation (10.39) in terms of x_1, x_2, y_1 and y_2.

The above depicted procedure on Equations (10.33) to (10.44), was carried with the help of a symbolic math software package. As a result, the linearized equations turn out to be as follows. Let the state variable x, be split into horizontal and vertical states:

$$x = \begin{bmatrix} x_h \\ x_v \end{bmatrix} \qquad x_h = \begin{bmatrix} x_1 \\ x_2 \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} \qquad x_v = \begin{bmatrix} y_1 \\ y_2 \\ \dot{y}_1 \\ \dot{y}_2 \end{bmatrix}$$

The input and output variables are also split,

$$i = \begin{bmatrix} i_h \\ i_v \end{bmatrix} \qquad i_h = \begin{bmatrix} i_{x_1} \\ i_{x_2} \end{bmatrix} \qquad i_v = \begin{bmatrix} i_{y_1} \\ i_{y_2} \end{bmatrix}$$
$$y = \begin{bmatrix} y_h \\ y_v \end{bmatrix} \qquad y_h = \begin{bmatrix} V^o_{x1} \\ V^o_{x_2} \end{bmatrix} \qquad y_v = \begin{bmatrix} V^o_{y1} \\ V^o_{y_2} \end{bmatrix}$$

With this notation in hand the linear model is as follows:

$$\begin{bmatrix} \dot{x_h} \\ \dot{x_v} \end{bmatrix} = \begin{bmatrix} A_h & rA_{hv} \\ rA_{vh} & A_v \end{bmatrix} \begin{bmatrix} x_h \\ x_v \end{bmatrix} + \begin{bmatrix} B_h & 0 \\ 0 & B_v \end{bmatrix} \begin{bmatrix} i_h \\ i_v \end{bmatrix} \tag{10.43}$$
$$\begin{bmatrix} y_h \\ y_v \end{bmatrix} = \begin{bmatrix} C_h & 0 \\ 0 & C_v \end{bmatrix} \begin{bmatrix} x_h \\ x_v \end{bmatrix}$$

with the matrices being, respectively,

$$A_h = \begin{bmatrix} 0_2 & I_2 \\ A_{hh} & 0_2 \end{bmatrix} \qquad B_h = \begin{bmatrix} 0_2 \\ B_{hh} \end{bmatrix} \qquad C_h = \begin{bmatrix} C_{hh} & 0_2 \end{bmatrix}$$
$$A_v = \begin{bmatrix} 0_2 & I_2 \\ A_{vv} & 0_2 \end{bmatrix} \qquad B_v = \begin{bmatrix} 0_2 \\ B_{vv} \end{bmatrix} \qquad C_v = \begin{bmatrix} C_{vv} & 0_2 \end{bmatrix}$$

The A_{hh}, A_{vv}, B_{hh}, B_{vv}, C_{hh} and C_{vv} matrices are determined by the partial derivatives of the Equations (10.37) and (10.39) with respect to x_h, x_v, u_h and u_v. The A_{hv} and A_{vh} matrices are as follows:

$$A_{hv} = T_{1h}^{-1} \begin{bmatrix} 0_3 & 0_{3\times 1} \\ 0_{1\times 3} & -\frac{I_z}{I} \end{bmatrix} T_{1v}$$
$$A_{vh} = T_{1v}^{-1} \begin{bmatrix} 0_3 & 0_{3\times 1} \\ 0_{1\times 3} & -\frac{I_z}{I} \end{bmatrix} T_{1h}$$

There is a current amplifier at the input of the system that regulates the current into the bearings. The dynamics of this LR circuit are as follows:

$$\frac{di_{x_i}}{dt} = \frac{1}{2.2 \times 10^{-4}} \left(-i_{x_i} + \frac{1}{4} V^i_{x_i} \right) \; i = 1, 2 \tag{10.44}$$
$$\frac{di_{y_i}}{dt} = \frac{1}{2.2 \times 10^{-4}} \left(-i_{y_i} + \frac{1}{4} V^i_{y_i} \right)$$

With this notation in hand,

$$u = \begin{bmatrix} u_h \\ u_v \end{bmatrix} \qquad u_h = V_h = \begin{bmatrix} V^i_{x_1} \\ V^i_{x_2} \end{bmatrix} \qquad u_v = V_v = \begin{bmatrix} V^i_{y_1} \\ V^i_{y_2} \end{bmatrix}$$

the linear dynamics of the LR network, can be expressed in terms of the following transfer function:

$$i_{x_i}(s) = \frac{k_a}{1 + \frac{s}{2\pi\ f_c}} V^i_{x_i}(s) \qquad f_c = 723.43 \text{ Hz} \quad k_a = 0.25 \tag{10.45}$$
$$i_{y_i}(s) = \frac{k_a}{1 + \frac{s}{2\pi\ f_c}} V^i_{y_i}(s) \qquad i = 1, 2$$

In the following section, the parameter-fitting process carried out on the system model will be presented. Samples of the magnetic-bearing system's frequency response were taken at different frequency ranges. In order to do this *ad hoc* identification process of the rigid-body dynamics, samples in the low frequency range (10 to 90 Hz) were used. As can be seen from Equation (10.45), for that range the relation between voltage and current can be simplified to $i_h = k_a u_h$ and $i_v = k_a u_v$.

10.2.4 Experimental Data and Parameter Fitting

In order to obtain a linear model that adequately suits for controller design, we make a few observations at this point. As will be introduced in the following pages, the rotor actually suffers disturbances that are neglected by the rigid-body model. These disturbances are, as was mentioned in the introduction of the chapter, imbalance and bending. Nevertheless, we point out that imbalance is a much worse "disturbance" to the nominal nonspinning dynamics than the gyroscopic effect is, as the dependence of the imbalance force with the rotation rate of the shaft is quadratic, whereas the gyroscopic effect is linear. As a consequence, for the remainder of this chapter we will neglect the horizontal–vertical coupling, *i.e.* we will assume $A_{hv} = 0$ and $A_{vh} = 0$. Keeping this in mind, we further point out, that addressing the identification and control of the horizontal *subsystem*, has proved to be enough with the purpose of drawing conclusions from this experimentation. Keeping in mind that only the horizontal dynamics will be addressed, a change in the notation is made. Because of the fact that the equations presented so far model the rotor as a rigid body, in the next equation the notation changes as follows with respect to Equation (10.43):

$$\begin{array}{lll} A_r = A_{hh} & B_r = B_{hh} & C_r = C_{hh} \\ x_r = x_h & y_r = y_h & u = u_h \end{array} \tag{10.46}$$

As a consequence, Equation (10.43) is rewritten as follows, with the change in the notation and bringing forward the fact that we neglect the horizontal–vertical coupling:

$$\begin{aligned} \dot{x_r} &= \begin{bmatrix} 0_2 & I_2 \\ A_r & 0_2 \end{bmatrix} x_r + \begin{bmatrix} 0_2 \\ \tilde{B}_r \end{bmatrix} u \\ y_r &= \begin{bmatrix} C_r & 0_2 \end{bmatrix} x_r \end{aligned} \tag{10.47}$$

with $\tilde{B}_r = B_r k_a$. It can be said that the qualitative outcome of the modeling process is about the shape of the above equation, whereas its quantitative counterpart is in the figures that make up the A^o_r, $\tilde{B}^o_r$ and C^o_r matrices that play the role of initial values for the parameter-optimization process that we describe next.

Given that the magnetic bearing and shaft dynamics are open-loop unstable, no responses can be obtained without a stabilizing compensator connected. The MBC500 comes with an internal compensator, that allows for the spindle to levitate. This compensator was designed with classical techniques neglecting all coupling between sides of the rotor. According to [14, 16], the mapping from the sensed voltages to the control voltages is given by the following transfer function:

$$c(s) = k_c \frac{\frac{s}{\omega_z} + 1}{(\frac{s}{\omega_p} + 1)(\frac{s}{\omega_{hf}} + 1)} \cong k_c \frac{\frac{s}{\omega_z} + 1}{\frac{s}{\omega_p} + 1}$$

with

$$\begin{aligned} f_z &= 178.63 \text{ Hz} & \omega_z &= 2\pi\ f_z \\ f_p &= 482.29 \text{ Hz} & \omega_p &= 2\pi\ f_p \\ f_{hf} &= 21.9 \text{ KHz} & \omega_{hf} &= 2\pi\ f_{hf} \\ k_c &= 1.41. \end{aligned} \tag{10.48}$$

The magnetic-bearing system has four inputs and eight outputs (see Figure 10.6). Four switches at the front panel allow for operation with the internal compensator engaged (closed loop) or disengaged (open loop).

The inputs placed on the front panel of the MBC500 have been used for two purposes throughout this work. The first of them is injecting excitation signals in order to carry out system identification with the internal compensator connected. The second one is injection of control signals, with the internal compensator disconnected. A slight distinction with some abuse in the notation is carried out as a consequence of using this front-panel inputs for different uses.

For the identification phase (internal compensator engaged) this input signal is called $w = \begin{bmatrix} w_{x_1} \\ w_{x_2} \end{bmatrix}$, whereas u is kept for the actual input signal to the current amplifier. In the control phase, u denotes the input signal as well as the signal injected on the front-panel connector, hence, the externally connected controller output signal.

For the identification phase, assuming linearity and time invariance ($r = 0$), the input-sensitivity transfer function from w to u is denoted as $S_r(s)$ and the transfer function from w to y is denoted as $GS_r(s)$ with

$$\begin{aligned} S_r(s) &= [I + C(s)G_r(s)]^{-1} \qquad C(s) = \begin{bmatrix} c(s) & 0 \\ 0 & c(s) \end{bmatrix} \\ GS_r(s) &= G_r(s)S_r(s) \end{aligned}$$

Frequency samples of the $S_r(s)$ and $GS_r(s)$ transfer matrices were taken for n different frequencies from 10 to 90 Hz with $n = 81$. The process of acquiring the frequency *samples* was carried out in a SISO fashion. This means that for each input–output pair of the two-input–two-output transfer matrices, and for each of the frequency points, a sinusoidal input was injected, taking a time response. A spectral decomposition was then carried out on this acquired signal keeping the component corresponding to the input frequency as a result. A frequency sweep was completed for all input–output pairs, for both of the transfer matrices. With this data in hand we calculate "*samples*" of the frequency response of the open-loop rigid dynamics $G_r(s)$. This procedure for the acquisition of open-loop "*samples*" is based upon previous work of the second author (see [21] and references therein). For each of the n frequency points ω_l with $l = 1, ..., n$, the samples $G_r(\jmath\omega_l)$ are

$$G_r(\jmath\omega_l) = GS_r(\jmath\omega_l)S_r(\jmath\omega_l)^{-1} = \begin{bmatrix} G_{r_{11}}(\jmath\omega_l) & G_{r_{12}}(\jmath\omega_l) \\ G_{r_{21}}(\jmath\omega_l) & G_{r_{22}}(\jmath\omega_l) \end{bmatrix} \tag{10.49}$$

For each frequency, the sample of the response consists of four complex values, one for each entry of the two-input–two-output transfer matrix, as shown in Equation (10.49). These four values are appended to form the ϑ_j vector in $\mathbb{C}^4$. All these ϑ_l vectors in $\mathbb{C}^4$, are appended

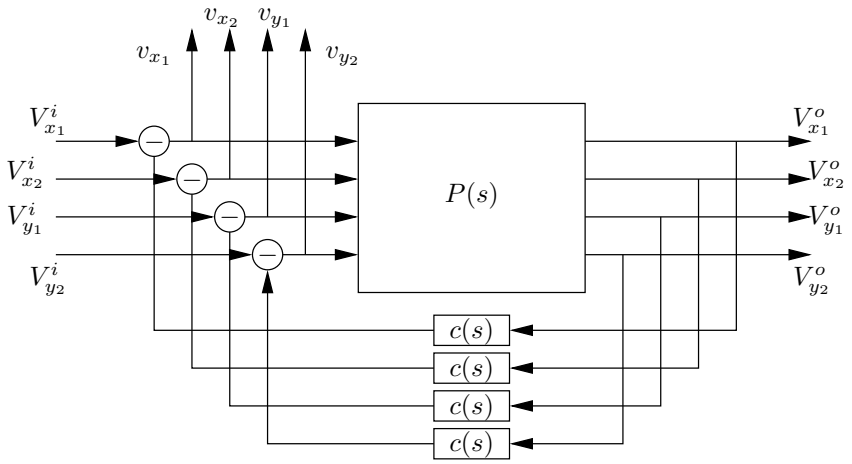


Fig. 10.6. MBC500 with the compensator

as well, rendering the Θ vector in $\mathbb{C}^{n\times 4}$ as can be seen in Equation (10.50), which contains *experimental* data.

$$\begin{bmatrix} G_{r_{11}}(\jmath\omega_l) & G_{r_{12}}(\jmath\omega_l) \\ G_{r_{21}}(\jmath\omega_l) & G_{r_{22}}(\jmath\omega_l) \end{bmatrix} \longrightarrow \vartheta_l = \begin{bmatrix} G_{r_{11}}(\jmath\omega_l) \\ G_{r_{12}}(\jmath\omega_l) \\ G_{r_{21}}(\jmath\omega_l) \\ G_{h_{22}}(\jmath\omega_l) \end{bmatrix} \longrightarrow \Theta = \begin{bmatrix} \vartheta_1 \\ \vdots \\ \vartheta_n \end{bmatrix} \tag{10.50}$$

A process of optimization was carried out on the A_r, $\tilde{B}_r$ and C_r matrices, so that the frequency response of the resulting model would fit the "*experimental*" samples with an error as small as possible according to the following optimization criterion. For each of the n frequencies, the frequency response of the model is given by the following calculation:

$$G^c_r(\jmath\omega_l) = C_r(\jmath\omega_l I - A_r)^{-1}\tilde{B}_r = \begin{bmatrix} G^c_{r_{11}}(\jmath\omega_l) & G^c_{r_{12}}(\jmath\omega_l) \\ G^c_{r_{21}}(\jmath\omega_l) & G^c_{r_{22}}(\jmath\omega_l) \end{bmatrix} \tag{10.51}$$

where the ^{c} superscript denotes *calculated*. The Θ^c vector containing the frequency response of the model is assembled as can be seen in Equation (10.52):

$$\begin{bmatrix} G^c_{r_{11}}(\jmath\omega_l) & G^c_{r_{12}}(\jmath\omega_l) \\ G^c_{r_{21}}(\jmath\omega_l) & G^c_{r_{22}}(\jmath\omega_l) \end{bmatrix} \longrightarrow \vartheta^c_l = \begin{bmatrix} G^c_{r_{11}}(\jmath\omega_l) \\ G^c_{r_{12}}(\jmath\omega_l) \\ G^c_{r_{21}}(\jmath\omega_l) \\ G^c_{r_{22}}(\jmath\omega_l) \end{bmatrix} \longrightarrow \Theta^c = \begin{bmatrix} \vartheta^c_1 \\ \vdots \\ \vartheta^c_n \end{bmatrix} \tag{10.52}$$

A nonlinear least squares algorithm (see [5]) minimized the following quantity:

$$(\Theta^c - \Theta)^*(\Theta^c - \Theta)$$

where the * symbol denotes the transpose complex conjugate.

Equation (10.53) shows the A_r^o, $\tilde{B}_r^o$ and C_r^o matrices that are the matrices resulting from the linearization process and according to the parameters of the MBC500, L, l, l_2, m, I and the force constant k ([12, 16]). These were used by the algorithm as initial conditions for the optimization process. The A_r, $\tilde{B}_r$ and C_r matrices resulting from the optimization process are shown as well.

$$
\begin{aligned}
A_r^o &= 10^4 \begin{bmatrix} 5.0273 & 1.6990 \\ 1.6990 & 5.0273 \end{bmatrix} & A_r &= 10^5 \begin{bmatrix} 0.9460 & 0.6307 \\ 0.0118 & 1.1214 \end{bmatrix} \\
\tilde{B}_r^o &= 10^0 \begin{bmatrix} 40.2180 & 13.5919 \\ 13.5919 & 40.2180 \end{bmatrix} & \tilde{B}_r &= 10^0 \begin{bmatrix} 20.0832 & 8.8209 \\ 1.7109 & 30.2025 \end{bmatrix} \\
C_r^o &= 10^3 \begin{bmatrix} 5.4796 & 0.4796 \\ 0.4796 & 5.4796 \end{bmatrix} & C_r &= 10^4 \begin{bmatrix} 1.1115 & 0.3419 \\ 0.1667 & 1.0206 \end{bmatrix}
\end{aligned}
\tag{10.53}
$$

The frequency response of the *fitted* system can be seen with the frequency *samples* in Figure 10.7.

10.2.5 Compensator Identification

An analysis of the documentation of the magnetic-bearing system ([12]), shows that the values of the k_c gains in the transfer functions of the compensators (see Equation (10.48)) are likely to be inaccurate. A look into the schematics of the analogic circuits that make up the compensators shows that the f_z, f_p and f_{hf} frequencies are determined by the values of resistors and capacitors that are fixed, so these values can be *trusted*. On the other hand, a variable resistor is present at the input of the compensator to adjust the compensator gain.

As a consequence, and based upon a method similar to the one described in the previous section, a *fitting* process was carried out on the compensator gains. For this purpose the recently fitted model G_r transfer function was used, in order to carry out calculations. The optimization criterion is exactly the same as in the previous section. The difference resides in the fact that G_r is now a *trusted* transfer function and optimization was carried out on two gains (one per compensator) to fit the frequency samples with the fitted compensator. The fitted values are $k_{c_1} = 0.6647$ for the *left* compensator and $k_{c_2} = 0.6395$ for the *right* compensator.

10.2.6 Flexible Modes

A thorough analysis of the rotor's bending modes is not within the scope of this chapter. Nevertheless, including a model of the first two bending modes has proved important as far as these dynamics are likely to receive excitation especially due to the digital implementation of the designed controllers. The reason for not including any modes higher than the second, is that because of the bandwidth of the actuator's current amplifiers in the electromagnet's L/R circuit, it is practically impossible that these modes would receive excitation. In [14] a model with four states for the two first bending modes is presented. The resemblance of these dynamics with the dynamics of a spring–mass system is stressed. In this chapter, taking after [2], a differential equation with four states is used as well, to model the first two bending modes.

As far as a spectral analysis of the rotor's dynamics is concerned, a clear separation exists between the flexible modes and the rigid-body modes. The rigid dynamics show up in the frequency band from DC to about 100 Hz where it rolls off, and the flexible dynamics show

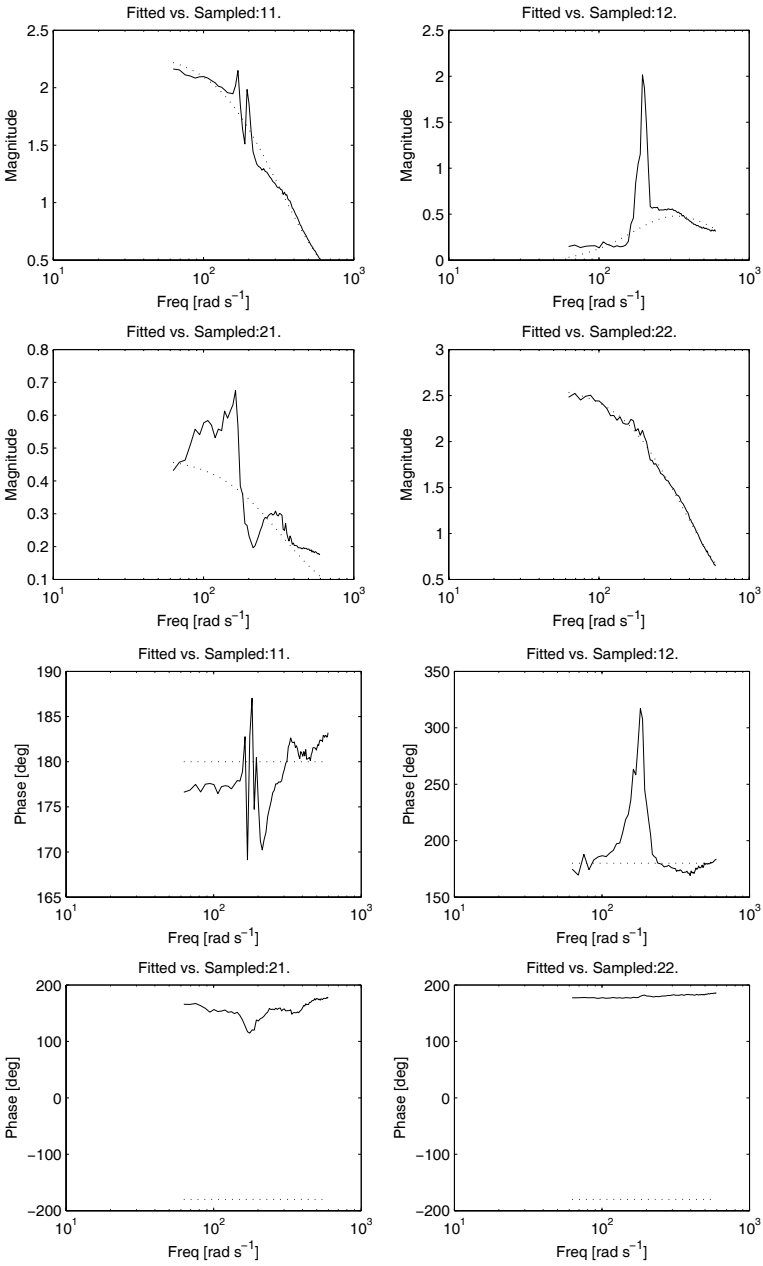


Fig. 10.7. Fitted (dotted line) *versus* sampled (solid line) frequency responses – rigid dynamics 10 to 90 Hz. Magnitude and phase for all input–output pairs.

up at 777 Hz for the first bending mode and 2065 Hz for the second. A practical approach for the final model, which reflects the rigid and flexible modes, consists in the superposition of three linear blocks as shown in Figure 10.8.

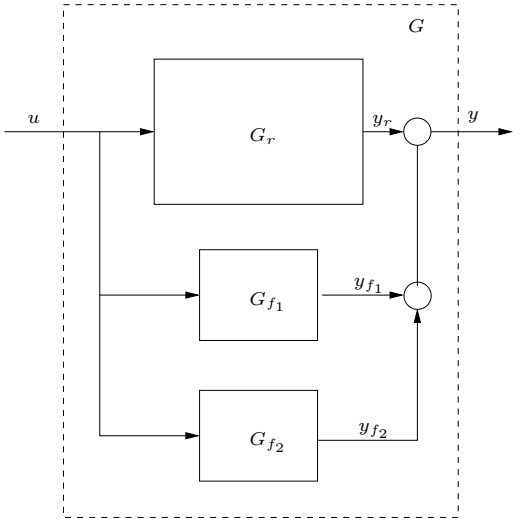


Fig. 10.8. Superposition of rigid and flexible dynamics

Let $x_{f_i} = \begin{bmatrix} x_{f_i}^1 \\ x_{f_i}^2 \end{bmatrix}$ be the state of the flexible blocks G_{f_i} in Figure 10.8, with $i = 1, 2$. The form of the state-space equations for each of the blocks is as follows:

$$\dot{x}_{f_i} = A_{f_i} x_{f_i} + B_{f_i} u$$
$$y_{f_i} = C_{f_i} x_{f_i}$$

with

$$A_{f_i} = \begin{bmatrix} 0 & \omega_{f_i} \\ \omega_{f_i} & 0 \end{bmatrix} \qquad B_{f_i} = \begin{bmatrix} b_1^i & b_2^i \\ 0 & 0 \end{bmatrix} \qquad C_{f_i} = \begin{bmatrix} 0 & c_1^i \\ 0 & c_2^i \end{bmatrix}$$

Frequency samples of the GS_r transfer function were obtained in a way similar to the one described in Section 10.2.4. In this case the information of the identified compensator was used. The parameter fitting on the A_{f_i}, B_{f_i} and C_{f_i} matrices with $i = 1, 2$ was carried out seeking an adequate fitting of experimental and calculated frequency responses. The final values of the model parameters for the bending modes are the following:

$$\omega_{f_1} = 2\pi\ 777\ \frac{\text{rad}}{\text{s}} \qquad b_1^1 = 5.5 \qquad b_2^1 = -8 \qquad c_1^1 = 7 \qquad c_2^1 = 7$$
$$\omega_{f_2} = 2\pi\ 2065\ \frac{\text{rad}}{\text{s}} \qquad b_1^2 = 3 \qquad b_2^2 = 3 \qquad c_1^2 = -3 \qquad c_2^2 = -5$$

In Figures 10.9 and 10.10, the corresponding frequency responses of models and samples can be seen.

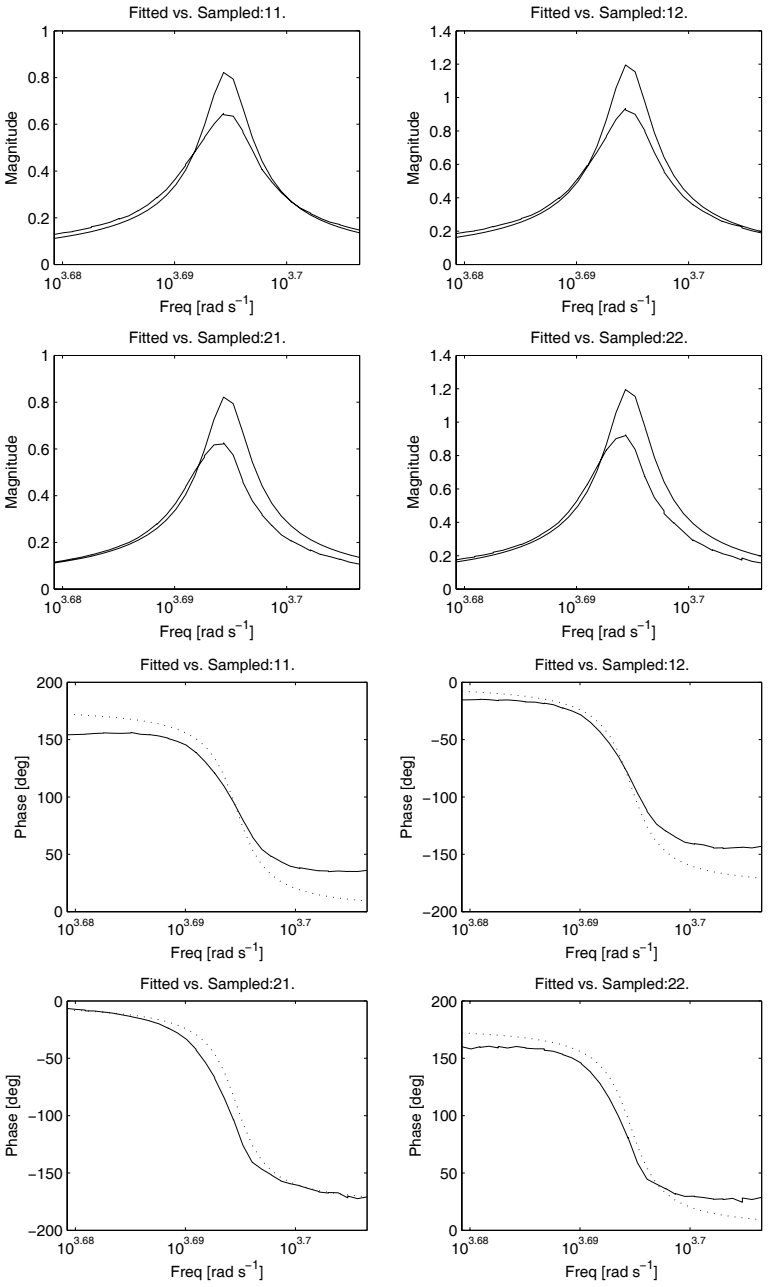


Fig. 10.9. Fitted (dotted line) *versus* sampled (solid line) – first bending mode – GS_τ transfer function

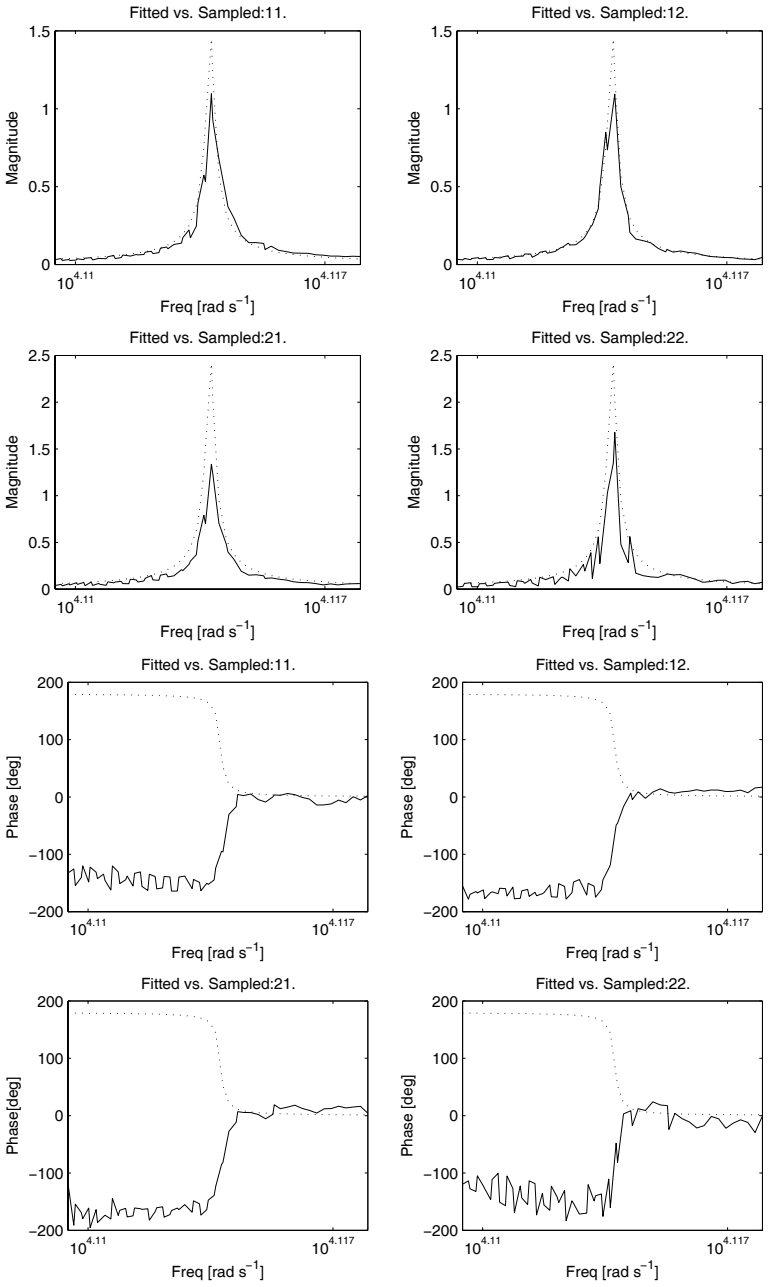


Fig. 10.10. Fitted (dotted line) *versus* sampled (solid line) – second bending mode – GS_r transfer function

10.2.7 Rotor Eccentricity

As was previously mentioned, the eccentricity is taken into account as *external* forces acting on the the rotor. Rather than a fine identification of the magnitude and direction of the source of the imbalance, we seek a gross estimate of this force. Another simplification carried out concerning eccentricity is the fact that variations in the rotation rate of the shaft are neglected, hence the eccentricity force of the model only depends on the square of the rotation rate. As a consequence, a very simple calculation is carried out for this static case that slightly modifies Equation (10.47) in order to take imbalance into account in the rigid dynamics as follows:

$$\begin{aligned} \dot{x_r} &= \begin{bmatrix} 0_2 & I_2 \\ A_r & 0_2 \end{bmatrix} x_r + \begin{bmatrix} 0_2 \\ \tilde{B}_r \end{bmatrix} u + r^2 \begin{bmatrix} 0_2 \\ k_{im} I_2 \end{bmatrix} w_{im} \\ y_r &= \begin{bmatrix} C_r & 0_2 \end{bmatrix} x_r \end{aligned} \tag{10.54}$$

with

$$k_{im} = \frac{1}{2}\frac{1}{100} r_o \qquad r_o = \frac{1}{2} d_o \qquad d_o = 12 \times 10^{-3} m$$

where d_o is the diameter of the shaft. The $\frac{1}{100}$ factor stems from the assumption that the distance from the true CM to the geometrical axis of the rotor, is in the order of one per cent of the shaft's radius (an assumption which is somewhat pessimistic – see Figure 10.11). The $\frac{1}{2}$ factor stems from the fact that two bearings cope with the eccentricity.

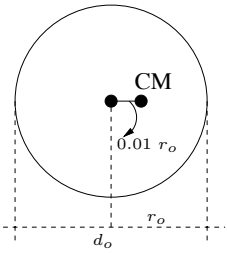


Fig. 10.11. Diagram for the eccentricity model

To give an example that accounts for the soundness of this gross estimate, take for instance a 10 000 rpm rotation rate ($r_{max} = 2\pi\ 10\ 000/60\ \frac{\text{rad}}{\text{s}}$). In this case, the magnitude of the disturbance on the $\ddot{x_i}$ ($i = 1, 2$) accelerations would be $r_o r_{max}^2 = 43.86\ \frac{\text{m}}{\text{s}^2}$ that is, according to the $\tilde{B}_r$ matrix, a disturbance in the order of magnitude the electromagnets can handle with an input voltage of a couple of volts.

As the G_{f_1} and G_{f_2} blocks are concerned, there is *a priori* knowledge of the frequency band where the w_{im} signal appears. This is directly related to the range of rotation rate of the rotor, and it is assumed that this signal does not present an excitation to the bending modes, because of the fact that the maximum rotation rate of the machine is 10 000 rpm, *i.e.* 166 Hz, while the bending modes are in 777 Hz and 2065 Hz.

A final comment concerns the neglecting of the pole of the L/R circuit of the electromagnet. No serious discrepancies have appeared because of this, especially considering that the design approach focuses on the rigid dynamics of the rotor treating the bending modes as additive uncertainty. As a consequence, the relation between the current of the electromagnets and the input voltage signals has been modeled as $i = k_a u$.

10.3 Control and Simulation Results

In this section the results of the control-system design are presented with simulations and experimental results. The solution of the control problem aims towards the following goals. Stabilize the shaft through a closed-loop controller, avoid excitation of bending modes (which can actually and eventually be heard in practice) and render adequate imbalance rejection in a range of rotating rates as broad as possible.

In order to evaluate performance of the designed controllers, frequency responses of the output-sensitivity functions will be presented in all cases. In this problem, good tracking is not the main goal, especially since it would demand a high loop gain at low frequency, which in turn would lead to a peak in the sensitivity function at a frequency close to the crossover frequency. This is because of fundamental limitations of closed-loop systems in the presence of unstable poles, a subject beyond the scope of this chapter (see [15, 18]). Another way to evaluate performance will be through simulated time responses to step inputs, which should show no excitation of the bending modes and acceptable transient behavior. In particular, these responses will be compared with the transient behavior of the internal compensator. Finally, in the experimental phase, the spinning of the shaft will be the ultimate test, whereas time responses to the equivalent sinusoidal inputs will be observed in the simulations.

10.3.1 SISO Synthesis

Based upon the G_r transfer function (Equation (10.47)), a SISO design is carried out, further simplifying this model. Let $x_r = \begin{bmatrix} x_1 & x_2 & \dot{x}_1 & \dot{x}_2 \end{bmatrix}$ be the state variable defined in Equation (10.46). Neglecting the left (x_1, $\dot{x}_1$) and right states (x_2, $\dot{x}_2$) coupling of Equation (10.47), a couple of SISO transfer functions are obtained. Let the A_r, B_r and C_r matrices be given elementwise as $A_r = \{A_r^{ij}\}$, $\tilde{B}_r = \{\tilde{B}_r^{ij}\}$ and $C_r = \{C_r^{ij}\}$ with $i, j = 1, 2$. The left- and right-side transfer functions are described in state space as follows:

$$\dot{x}_L = \begin{bmatrix} 0 & 1 \\ A_r^{11} & 0 \end{bmatrix} x_L + \begin{bmatrix} 0 \\ \tilde{B}_r^{11} \end{bmatrix} u_L, \qquad y_1 = \begin{bmatrix} C_r^{11} & 0 \end{bmatrix} x_L \qquad \text{with } x_L = \begin{bmatrix} x_1 \\ \dot{x}_1 \end{bmatrix}$$

$$\dot{x}_R = \begin{bmatrix} 0 & 1 \\ A_r^{22} & 0 \end{bmatrix} x_R + \begin{bmatrix} 0 \\ \tilde{B}_r^{22} \end{bmatrix} u_R, \qquad y_2 = \begin{bmatrix} C_r^{22} & 0 \end{bmatrix} x_R \qquad \text{with } x_R = \begin{bmatrix} x_2 \\ \dot{x}_2 \end{bmatrix}$$

In transfer-function format the latter turns out to be as follows:

$$g_1(s) = \frac{2.36}{\frac{s^2}{\omega_L^2} - 1} \qquad \omega_L = 2\pi\, f_L \qquad f_L = 48.95 \text{ Hz}$$

$$g_2(s) = \frac{2.75}{\frac{s^2}{\omega_R^2} - 1} \qquad \omega_R = 2\pi\, f_R \qquad f_R = 53.28 \text{ Hz}$$

With these transfer functions in hand, a SISO compensator was designed based upon simple objectives, namely, to obtain a good phase margin for stability, to avoid too high loop gains at low frequency, and to avoid excitation of the bending modes due to the digital implementation.

Following the approach of [14], a phase-lead controller was proposed with changes in the frequencies of the zeros and poles with respect to values of the internal compensator. These parameters, as well as the gain of the compensator, were adjusted by hand, in an iterative process until satisfactory phase margins were obtained, as well as simulated and experimental time responses. In order to avoid excitation of the bending modes, an *ad hoc* approach was adopted, consisting in placing notch filters tuned to the frequencies of the bending modes in a series interconnection with the phase-lead transfer function. The reason to report on the design of this controller, apart from the internal compensator bundled in the MBC500's box, was to look into issues concerning the digital implementation. The transfer function of the SISO compensator is as follows:

$$
\begin{aligned}
k(s) &= k_1(s)h_1(s)h_2(s) \\
k_1(s) &= k_s \frac{\frac{s}{\omega_z} + 1}{(\frac{s}{\omega_{p_1}} + 1)(\frac{s}{\omega_{p_2}} + 1)} \\
h_i(s) &= 1 - \frac{\zeta w_{f_i} s}{s^2 + \zeta w_{f_i} s + w_{f_i}^2}
\end{aligned}
$$

with $i = 1, 2$ and

$$
\begin{aligned}
k_s &= 0.65 & \omega_z &= 2\pi\ 60\ \frac{\text{rad}}{\text{s}} & \omega_{p_1} &= 2\pi\ 240\ \frac{\text{rad}}{\text{s}} \\
\omega_{p_1} &= 2\pi\ 480\ \frac{\text{rad}}{\text{s}} & \omega_{f_1} &= 2\pi\ 777\ \frac{\text{rad}}{\text{s}} & \omega_{f_2} &= 2\pi\ 2065\ \frac{\text{rad}}{\text{s}}
\end{aligned}
$$

10.3.2 Robust $\mathcal{H}_\infty$ and LPV designs

These two approaches are presented together, given that the statement of the problem is the same for both. The details of the LPV approach are presented first, with remarks concerning the feasible parameter-variation set. Regarding the Robust $\mathcal{H}_\infty$ methodology, designs are presented that aim towards the operation of the rotating machine with a fixed rotation rate. These designs, carried out at rotating rates of 5000 and 10 000 rpm, give a reference of the desirable γ performance factor for the LPV design and this, in turn, is related to the subject of the size of the parameter-variation set.

Anticipating the subject of the discrepancies between theory and practice, the conclusion concerning the parameter-variation set, is not satisfactory for this application. Even though a *natural* parameter-variation set exists ($\mathcal{P}_n$ = 0–10 000 rpm), which is given by the manufacturer of the magnetic-bearing experiment, it was not possible for this range to obtain a feasible solution to the γ-performance LPV control problem (Theorem 1) as stated here.

Different parameter-variation sets were tried in order to establish in which cases a solution to the control problem existed, and how performance was affected by the size of the parameter-variation set (given feasibility). Before turning to the details of this trial and error process, the statement of the control problem, *i.e.* the augmented plant with the weighting functions, will be presented.

The diagram of Figure 10.12 shows the blocks involved in the augmented plant. In order

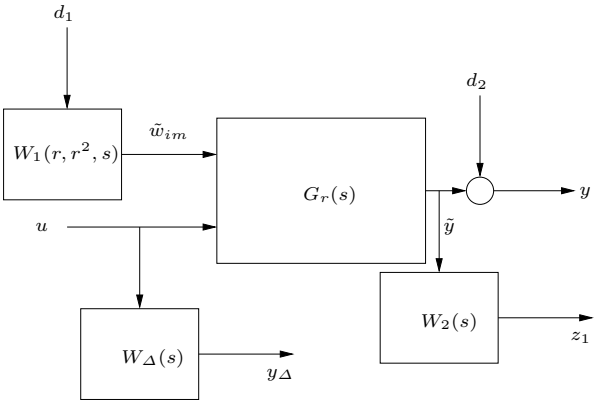


Fig. 10.12. Augmented plant

to simplify the problem statement and with the objective of keeping the parameter-varying part of the problem circumscribed to the W_1 block, a minor modification is carried out on Equation (10.54) as follows:

$$\begin{aligned} \dot{x_r} &= \begin{bmatrix} 0_2 & I_2 \\ A_r & 0_2 \end{bmatrix} x_r + \begin{bmatrix} 0_2 \\ \tilde{B}_r \end{bmatrix} u + \begin{bmatrix} 0_2 \\ k_{im} I_2 \end{bmatrix} \tilde{w}_{im} \\ y_r &= \begin{bmatrix} C_r & 0_2 \end{bmatrix} x_r \end{aligned} \tag{10.55}$$

rendering the G_r transfer matrix as follows:

$$G_r(s) \equiv = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C & 0 & 0 \end{array}\right] \tag{10.56}$$

with

$$A = \begin{bmatrix} 0_2 & I_2 \\ A_r & 0_2 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 0_2 \\ k_{im} I_2 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0_2 \\ \tilde{B}_r \end{bmatrix} \qquad C = \begin{bmatrix} C_r & 0_2 \end{bmatrix}$$

In the modification turning Equation (10.54) into Equation (10.55), notice that the r^2 factor that accounts for the increase in the magnitude of the imbalance force (as the rotation rate grows), has been removed. As the $\mathcal{H}_\infty$ and LPV methods used for synthesis in this application consider all disturbance inputs to be signals in $\mathcal{L}_2$ (*i.e.* $d_1, d_2 \in \mathcal{L}_2$), we resorted to including the W_1 block, a variable gain, variable-frequency bandpass prefilter, that accounts for the fact the true w_{im} signal (see Equation (10.54)) is a sinusoidal signal of known frequency whose amplitude depends on r^2. As was previously mentioned, this *disturbance* signal serves the purpose of modeling the imbalance forces acting on the rotor. As a sinusoidal signal, its frequency is given by the rotation rate of the shaft. Given a fixed rotation rate r, the transfer function of the W_1 filter turns out to be as follows:

$$W_1(s) = \begin{bmatrix} w_1(s) & 0 \\ 0 & w_1(s) \end{bmatrix}$$

$$w_1(s) = r^2 \frac{rs}{s^2 + 2\zeta r s + r^2} \equiv \left[\begin{array}{cc|c} -2\zeta r & -r^2 & r^2 \\ 1 & 0 & 0 \\ \hline r & 0 & 0 \end{array}\right] \tag{10.57}$$

Notice that the state-space representation of Equation (10.57) can be rewritten so that the affine dependence of the state-space matrices of the filter on r and r^2 is clear:

$$w_1(s) \equiv \left[\begin{array}{cc|c} 0 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right] + r \left[\begin{array}{cc|c} -2\zeta & 0 & 0 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 0 \end{array}\right] + r^2 \left[\begin{array}{cc|c} 0 & -1 & 1 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right]$$

When setting up the LPV synthesis problem, the existing relation between r and r^2 will be neglected. As far as the synthesis method is concerned, the w_1 weighting function depends affinely on two independent parameters p_1 and p_2 as follows:

$$w_1(s) \equiv \left[\begin{array}{cc|c} 0 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right] + p_1 \left[\begin{array}{cc|c} -2\zeta & 0 & 0 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 0 \end{array}\right] + p_2 \left[\begin{array}{cc|c} 0 & -1 & 1 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right]$$

No advantage is derived from knowing that $p_2 = p_1^2$ and in that way, the approach is slightly conservative.

On the other hand, the $W_\Delta(s)$ filter is shaped in frequency with two objectives in mind. In the first place, penalize any control action above a given ω_h frequency and secondly, give the synthesis algorithm the frequency information concerning the bending modes. The transfer function of this weighting functions is:

$$W_\Delta(s) = \begin{bmatrix} w_\Delta(s) & 0 \\ 0 & w_\Delta(s) \end{bmatrix}$$

$$w_\Delta(s) = w_h(s) w_{b_1}(s) w_{b_2}(s)$$

$$w_h(s) = k_h \frac{s}{s + \omega_h}$$

$$w_{b_i}(s) = \frac{s^2 + \omega_{b_i} s + \omega_{b_i}^2}{s^2 + \zeta_{b_i}\omega_{b_i} s + \omega_{b_i}^2} \text{ with } i = 1, 2$$

and with

$$\omega_h = 2\pi\ 500\ \frac{\text{rad}}{\text{s}} \qquad k_h = 1.3$$

$$\omega_{b_1} = 2\pi\ 777\ \frac{\text{rad}}{\text{s}} \qquad \zeta_{b_1} = 0.02$$

$$\omega_{b_2} = 2\pi\ 2065\ \frac{\text{rad}}{\text{s}} \qquad \zeta_{b_1} = 0.02$$

$W_\Delta(s)$ can be thought of as a combination of a weighting function for a family of plants with additive uncertainty and a weighting function penalizing control action at high frequency. Additive uncertainty is taken into account, as the norm reduction of the mapping from d_2 to y_Δ (see Figure 10.12) is addressed by the synthesis algorithm minimizing the norm of the mixed-sensitivities operator mapping $\begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$ to $\begin{bmatrix} y_\Delta \\ z_1 \end{bmatrix}$.

The $W_2(s)$ weighting transfer function for the controlled output is as follows:

$$W_2(s) = \begin{bmatrix} w_2(s) & 0 \\ 0 & w_2(s) \end{bmatrix}$$
$$w_2(s) = k_l \frac{1}{\frac{s}{\omega_l} + 1}$$

with

$$\omega_l = 2\pi\ 60\ \frac{\text{rad}}{\text{s}} \qquad\qquad k_l = 2$$

The shape of the w_2 and w_h filters follows typical guidelines for the choice of closed-loop bandwidth and control bandwidth. Avoiding a high peak in the magnitude of the frequency response of the sensitivity transfer function is sought as well (see [27], Chapter 6 and [26]). An iterative process was carried out as well for the final fine tuning of these values.

Parameter-variation Set

Table 10.2. γ Performance index and parameter-variation set

Method	Range[rpm]	γ Index
$\mathcal{H}_\infty$	$r = 5000$	2.572
LPV	$r \in [4000, 5000]$	6.1483
LPV	$r \in [3500, 5000]$	infeasible
$\mathcal{H}_\infty$	$r = 10\ 000$	20.42
LPV	$r \in [8000, 10\ 000]$	49.598
LPV	$r \in [7000, 10\ 000]$	infeasible

The general observation concerning the parameter-variation set is that given a maximum value for the rotation rate r_{max}, the reasonable parameter-variation set for the two-parameter (r, r^2) LPV system is

$$\mathcal{P} = \{(p_1, p_2) \in [r_{min}, r_{max}] \times [r_{min}^2, r_{max}^2]\}$$

with $r_{min} = 0.8\ r_{max}$. This rule was empirically established following the aforementioned criteria of feasibility of the corresponding LMI problem and γ performance index within two times the γ performance index of the robust $\mathcal{H}_\infty$ problem resulting from holding the parameter r fixed at $r = r_{max}$.

Table 10.2 shows the different $\mathcal{H}_\infty$ and LPV control designs that were attempted. Because of the infeasibility of the optimization problem for the whole $[0, 10\ \ 000]$ rpm range, two ranges were tried in order to draw conclusions. Namely $[4000, 5000]$ rpm and $[8000, 10\ 000]$ rpm. The $\mathcal{H}_\infty$ designs carried out, correspond to $r = 5000$ rpm in one case and $r = 10\ 000$ rpm in the other. As can be seen in the table, greater intervals were infeasible.

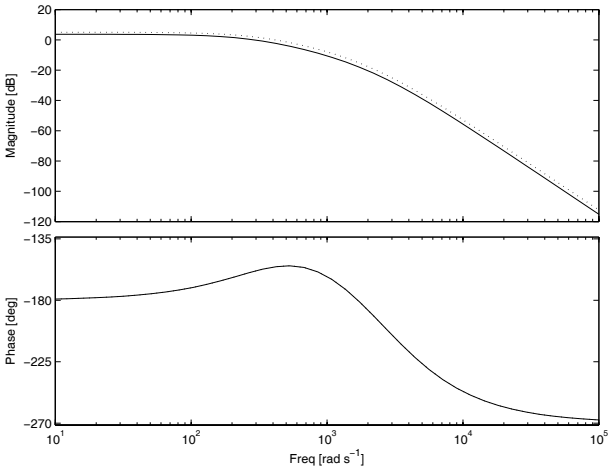


Fig. 10.13. Phase margin of the SISO internal compensator. Solid line: left-hand side transfer function $k(s)g_1(s)$. Dotted line: left-hand side transfer function $k(s)g_2(s)$.

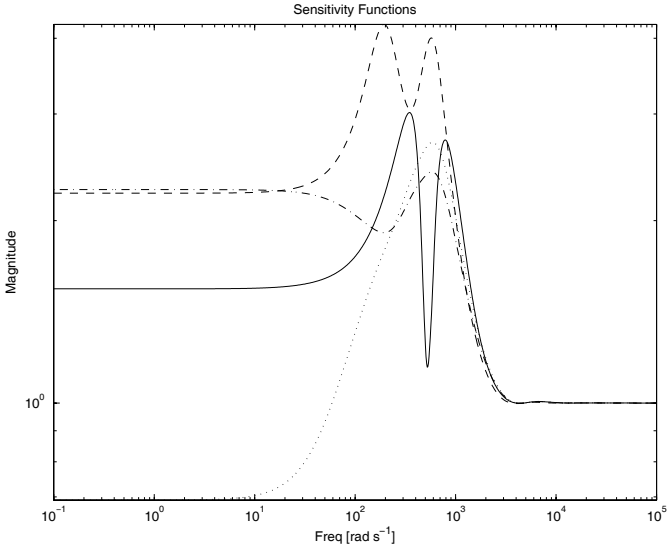


Fig. 10.14. Frequency response of the sensitivities. $r_{max} = 5000$ rpm. Solid line: $\mathcal{H}_\infty$, dashed line: SISO, dash-dot line: LPV(r_{max}), dotted line: LPV(r_{min}).

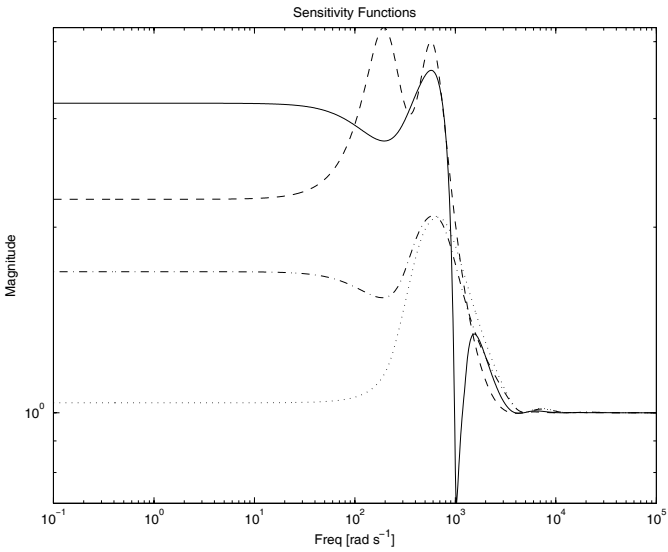


Fig. 10.15. Frequency response of the sensitivities. $r_{max} = 10\ 000$ rpm. Solid line: $\mathcal{H}_\infty$, dashed line: SISO, dash-dot line: LPV(r_{max}), dotted line: LPV(r_{min}).

10.3.3 Frequency Responses and Simulation Results

In the present section frequency-response graphics are shown, as well as simulated time responses. In Figure 10.13, the frequency response for the SISO open-loop transfer functions of the SISO design can be seen. The sole purpose of this figure is to show the phase margin of 25 degrees that was part of the criteria to evaluate this design.

Figures 10.14 and 10.15 show the frequency response of the maximum singular value of the output-sensitivity transfer function. Notice that the $\mathcal{H}_\infty$ design, specifically designed for a fixed rotation rate, renders a reduction of the sensitivity at the expected rotation-rate frequency. On the other hand, the frequency responses of the LPV controller calculated for $r = r_{max}$ and $r = r_{min}$, show higher gains at low frequency than the $\mathcal{H}_\infty$. Notice as well that the reduction in the magnitude of the sensitivity at low frequency for the $\mathcal{H}_\infty$ controller designed for $r = 10\ 000$ rpm, is poor.

Figures 10.16 and 10.17 show time responses to step inputs to be followed by the x_1 output. Recall that good tracking is not a must, but it cannot be too poor either. If that were the case, the system might fail to stabilize the rotor for $r = 0$ rpm.

Figures 10.18, 10.19, 10.20 and 10.21 show time responses to sinusoidal inputs injected on both sides that simulate imbalance forces. These are to be rejected. Notice that the $\mathcal{H}_\infty$ design performs better than the rest at the specific frequency it was designed for ($r_{max} = 5000$ and $10\ 000$ rpm) while achieving a decent result for the low rotation rate the LPV was designed for ($r_{max} = 4000$ and 8000 rpm). Naturally, the LPV performs better than the $\mathcal{H}_\infty$ in this case, but not by much.

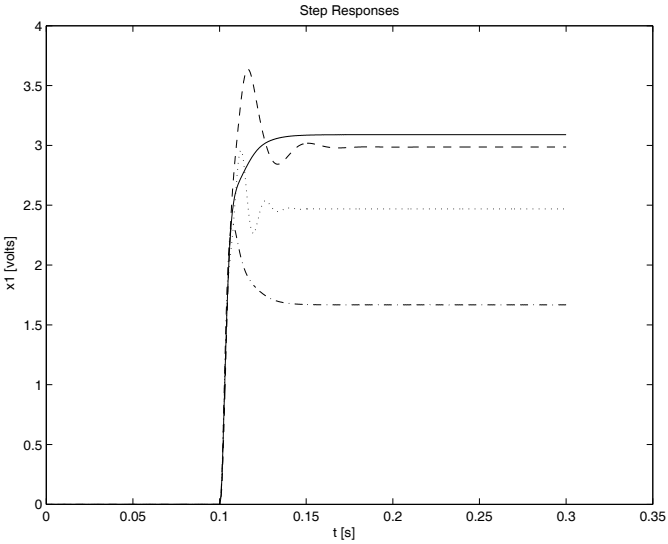


Fig. 10.16. Step responses. $r_{max} = 5000$ rpm. Solid line: $\mathcal{H}_\infty$, dashed line: SISO, dash-dot line:LPV(r_{max}), dotted line: LPV(r_{min}).

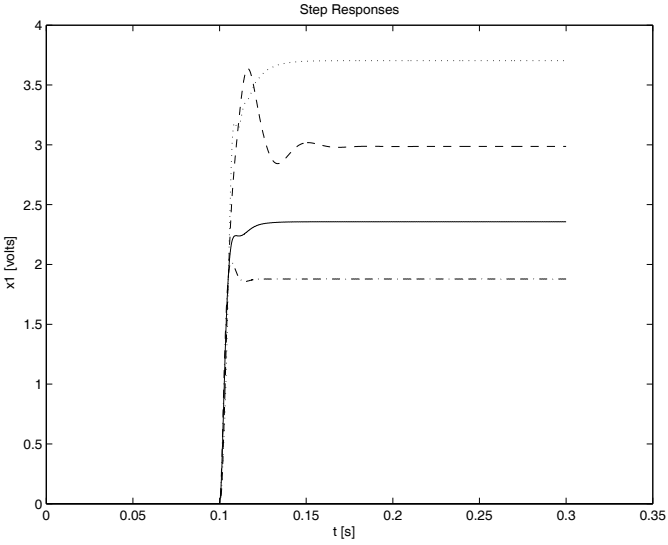


Fig. 10.17. Step responses. $r_{max} = 10\ 000$ rpm. Solid line: $\mathcal{H}_\infty$, dashed line: SISO, dash-dot line: LPV(r_{max}), dotted line: LPV(r_{min}).

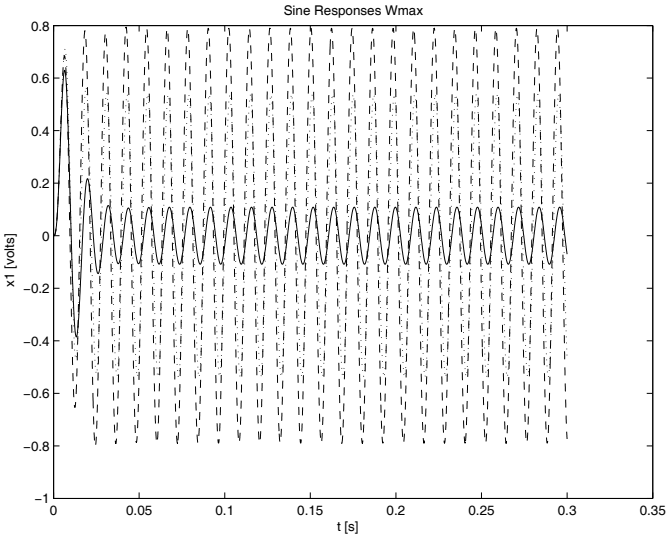


Fig. 10.18. Responses to sinusoidal force of frequency $r_{max} = 5000$ rpm. Solid line: $\mathcal{H}_\infty$, dashed line: SISO, dash-dot line: LPV(r_{max}).

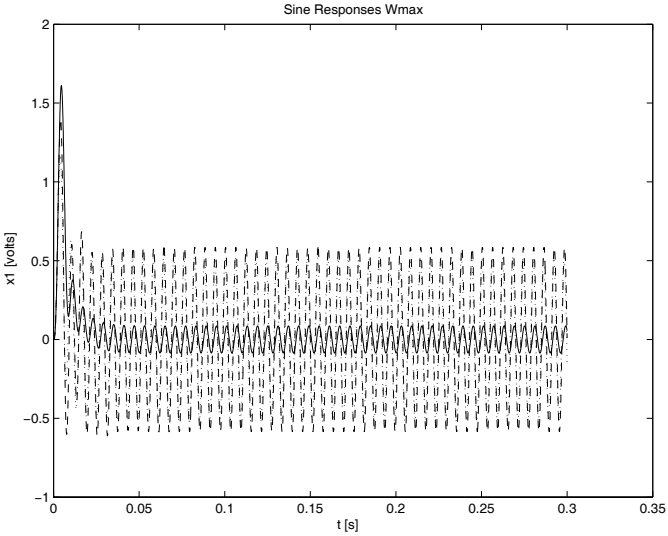


Fig. 10.19. Responses to sinusoidal force of frequency $r_{max} = 10\,000$ rpm. Solid line: $\mathcal{H}_\infty$, dashed line: SISO, dash-dot line: LPV(r_{max}).

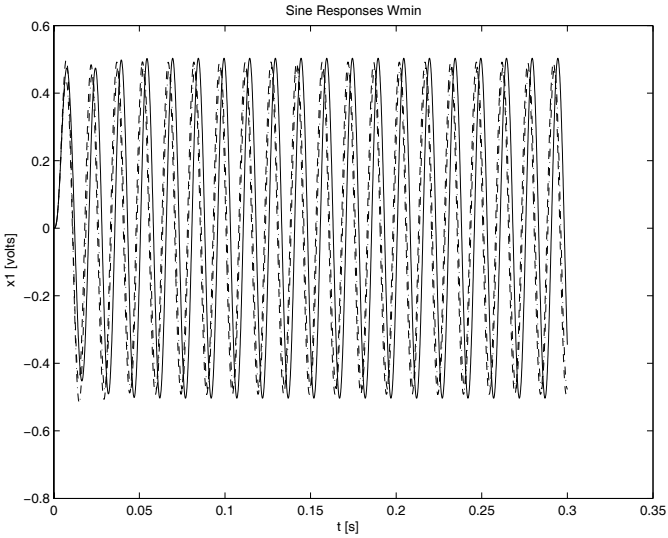


Fig. 10.20. Responses to sinusoidal force of frequency $r_{max} = 4000$ rpm. Solid line: $\mathcal{H}_\infty$, dashed line: SISO, dotted line: LPV(r_{min}).

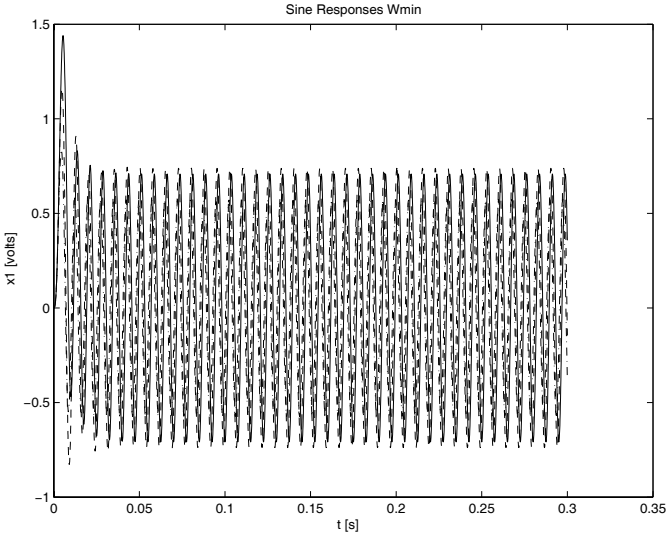


Fig. 10.21. Responses to sinusoidal force of frequency $r_{max} = 8000$ rpm. Solid line: $\mathcal{H}_\infty$, dashed line: SISO, dotted line: LPV(r_{min}).

10.3.4 Experimental Results

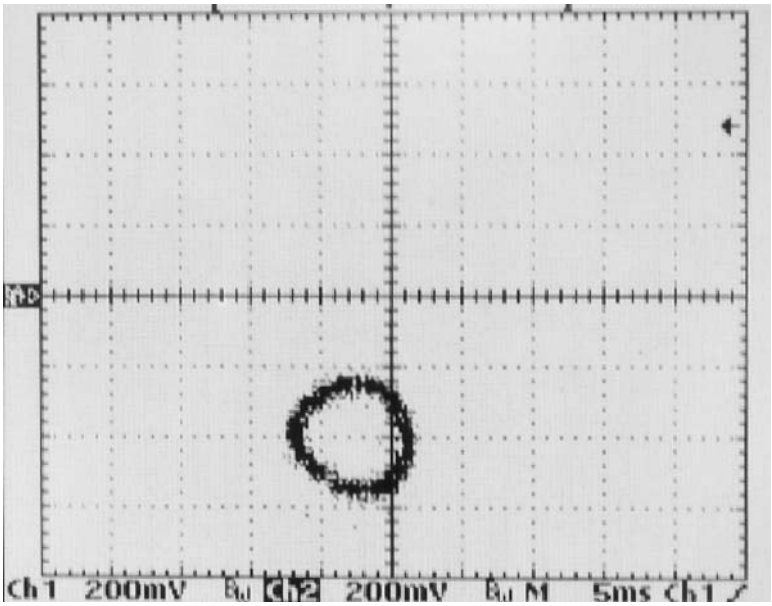


Fig. 10.22. Experimental responses to 5000 rpm. x- and y-axes: internal compensator.

In this section, the experimental results are presented. The first thing that must be pointed out, is that the LPV controller was not implemented because of the fast poles problem mentioned in Section 10.1.9. The material of that section has been included in the chapter for reference, but in spite of the fact that a solution of the LPV synthesis problem with eigenvalue clustering was tried, the outcome was not acceptable. When that algorithm was used, while the optimization problem was feasible, the resulting controller was unstable, hence not a good choice for implementation. The LMI region used was the following:

$$\mathcal{D} = \{z \in \mathbb{C} : \Re(z) > -2\pi\ 2100\}$$

which in terms of the α and β matrices of Equation (10.21) gives scalars $\alpha = -2\pi\ 2100$ and $\beta = -1$. In spite of the fact that this is a very simple LMI region as far the size of α and β is concerned, the computation time of the LPV controller is considerably increased. Another remark regards the $\mathcal{H}_\infty$ controller designed for the 10 000 rpm rate; this controller did not fulfill the basic stabilization requirement due to its low gain at low frequency.

Figures 10.22 to 10.27 show the response of the experimental equipment for three different controllers with the machine rotating at 5000 and 10 000 rpm respectively. The measured signals, displayed in an oscilloscope in XY mode, are the horizontal (x_1) and vertical (y_1) displacements of the rotor on the left-hand side of the machine. The displayed voltage signals

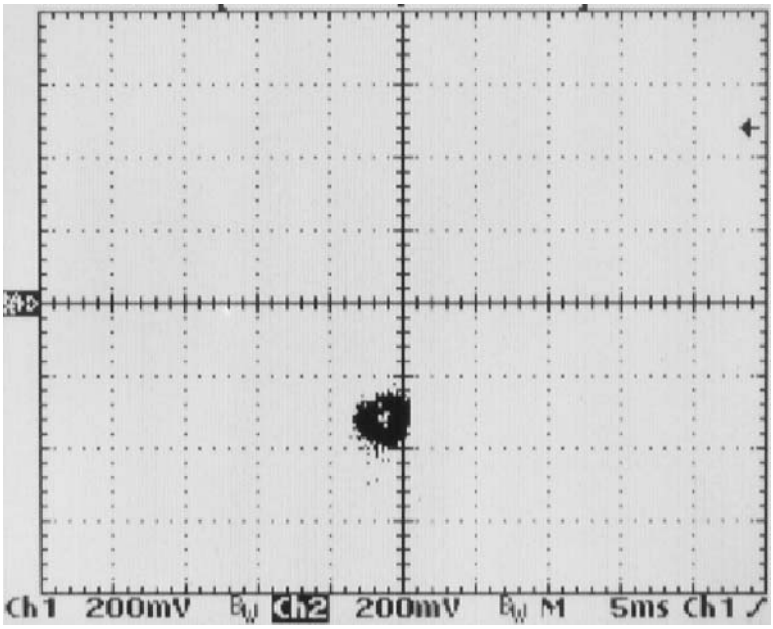


Fig. 10.23. Experimental responses to 10 000 rpm. x- and y-axes: internal compensator.

correspond with displacement with a factor of 1 mm per volt (*i.e.* 100 mV is 0.1 mm). The scale of the horizontal and vertical axes for all figures is 200 mV/div. Notice that only the excursion in the x-axis matters in order to evaluate the performance of the designed controllers. The control of the vertical axes y_1 and y_2 was left to the internal compensator. Figures 10.22 and 10.23 show the responses for the internal compensator of the MBC500, Figures 10.24 and 10.25 for the designed SISO compensator, and Figures 10.26 and 10.27 for the Robust $\mathcal{H}_\infty$ controller designed for imbalance forces at 5000 rpm (in the following $\mathcal{H}_\infty^{5000}$). Figure 10.26 shows the filtering of imbalance forces that the $\mathcal{H}_\infty^{5000}$ controller does at 5000 rpm. On the other hand, the performance of the controller at higher rotation rates is comparable to that of the internal compensator. The $\mathcal{H}_\infty^{5000}$ also showed better behavior in the transition from 0 to 5000 rpm than the internal compensator as well as the designed SISO controller compared to the internal compensator that shows little imbalance rejection at low rotation rates. The behavior of the SISO design at high rates was quite poor.

10.4 Discrepancies between Theory and Practice

The use of LPV synthesis methods for this application is *a priori* appealing, motivated by references such as [13] where traditional gain scheduling is used and by the results obtained in simulations carried out for this chapter. In spite of the benefits it provides in terms of smooth scheduling and guaranteed stability, the method has shown its problems in the implementation

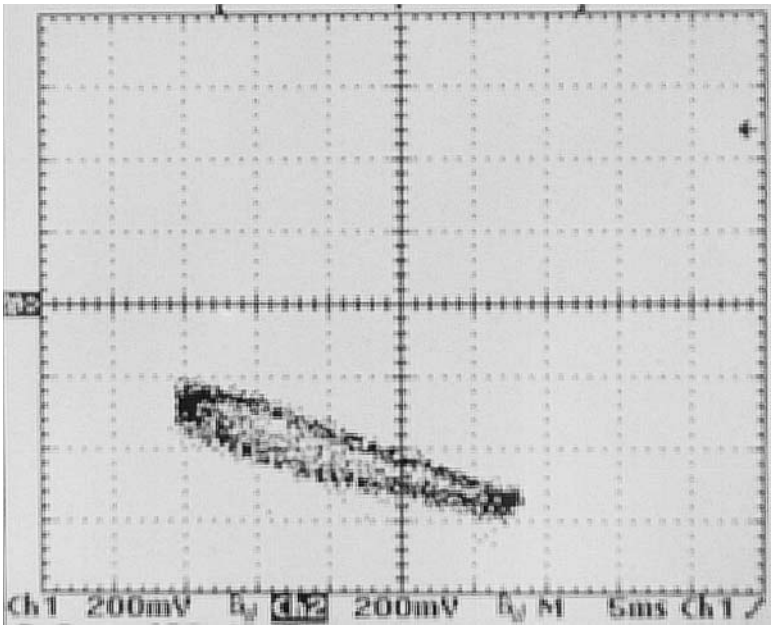


Fig. 10.24. Experimental responses to 5000 rpm. x-axis controller: SISO, y-axis controller: internal.

phase too. The matter of the limited range the LPV method was capable of suggests the use of less conservative techniques such as LPV synthesis based upon PDLFs ([25], see also [1]). The use of more advanced LFT techniques, which give less conservative solutions, such as [17] is another possible direction for LPV redesign. Regarding the techniques of [17], preliminary research of the authors on the subject has shown their practical implementation is yet not a fully solved problem (see [10]).

The matter of fast dynamics that prevents an easy implementation of controllers has been already addressed. The LPV techniques of Theorem 2, which are pretty simple as the complexity in their application is not greater than that of $\mathcal{H}_\infty$ control, even rendering an LMI problem with a finite number of constraints (at the expense of being conservative, perhaps), showed limited applicability in this particular example. Its simplicity is appealing, but results should be improved.

A word on the required hardware for the implementation of the controllers must be said too. The internal analog compensator is a very simple solution with low cost, simple theory and robust performance. Nevertheless, classical design is unlikely to address satisfactorily stricter requirements regarding imbalance rejection and moreover, more challenging problems such as avoiding the excitation of bending modes due to imbalance. For this chapter, sophisticated PC hardware was used with expensive acquisition and control boards in order to implement the more complex control techniques. In spite of this, experience exists in the implementation of digital controllers with field-programmable gate array integrated circuits (FPGA ICs) for this kind of application. While the PC hardware (AMD Athlon XP 2500+, 1800 MHz proces-

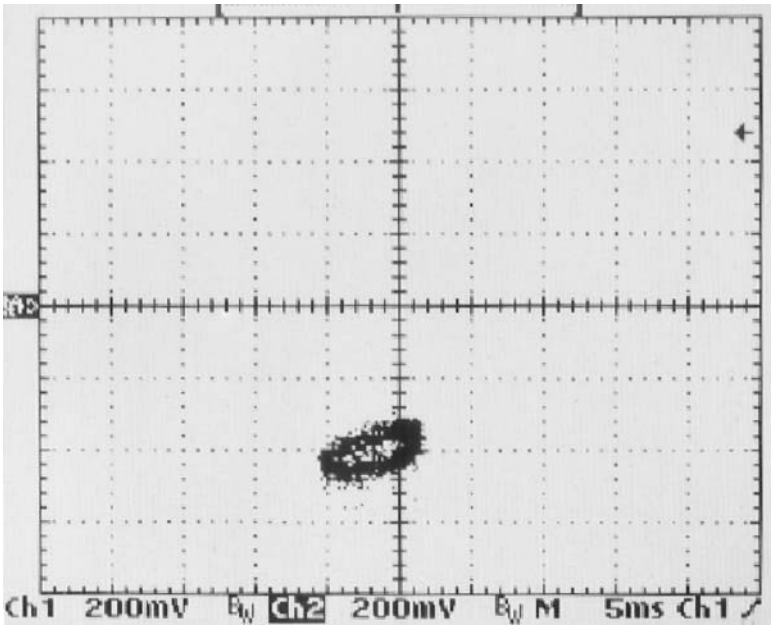


Fig. 10.25. Experimental responses to 10 000 rpm. x-axis controller: SISO, y-axis controller: internal.

sor) with the PCI boards used in this application (National Instruments PCI-6289 A/D - D/A board) were capable of achieving at most 16 kHz sampling rates for an LTI digitalized state-space controller with twenty states, the FPGA technology, which tends to have lower cost and greater computation power, is known to easily achieve sampling rates of 50 kHz for the same case, yet providing the needed flexibility for a control laboratory in order to carry out an iterative design process. Moreover, this kind of circuitry could provide affordable solutions for magnetic-bearing machines manufactured with their controllers embedded and implementing sophisticated techniques.

Acknowledgments

The research work reported in this chapter was partially supported by the *Agencia Nacional de Promoción Científica y Tecnológica* of Argentina through grants PICT 97' BID OC/AR 1758 and PICT 99' BID OC/AR 7263 and by the University of Buenos Aires through grant PC 98'-00' TI34. The last author was supported by the Research Commission of the *Generalitat de Catalunya* (ref. 2005SGR00537), by the Spanish CICYT (ref. DPI2005-04722), and by ICREA. The experimental work of this research was carried out at the Space Access Program Lab at the National Commission of Space Activities of Argentina (CONAE), a facility jointly operated with the GICOR–FIUBA. The first and last authors wish to thank Roberto Yasielski,

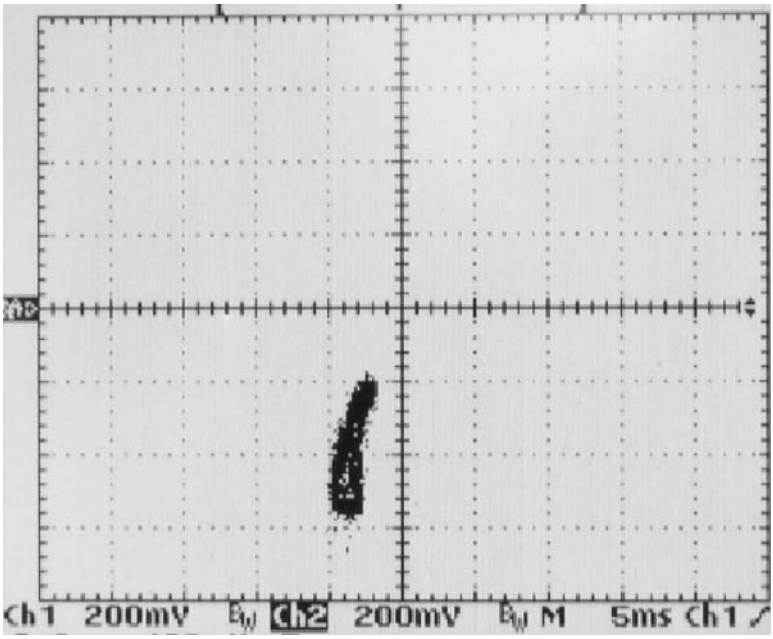


Fig. 10.26. Experimental responses to 5000 rpm. x-axis controller: $\mathcal{H}_\infty^{5000}$, y-axis controller: internal.

José Astigueta and Daniel Caruso from CONAE, for their support with the use of the Lab and the acquisition hardware. The first author also wishes to thank Professors Mario Lozano, Guillermo Casas and Virginia Mazzone from UNQ for their help, and his colleagues from GICOR, Juan Giribet and Alberto Fraguío for their helpful discussions.

References

[1] Apkarian, P. and Adams, R. J. (1998). Advanced gain-scheduled techniques for uncertain systems. *IEEE Transactions on Control Systems Technology* **6**(1), 21–32.

[2] Arredondo, I., Jugo, J., and Etxebarria, V. (2004). Modelización y control de un eje sustentado mediante levitación magnética activa (in spanish). *Proceedings of the XXV Jornadas de Automática*. CEA-IFAC.

[3] Becker, G. S. (1993). *Quadratic Stability and Performance of Linear Parameter Dependent Systems*. Ph.D. thesis, University of California at Berkeley.

[4] Becker, G. S. and Packard, A. (1994). Robust performance of LPV systems using parametrically-dependent linear feedback. *Systems and Control Letters* **23**, 205–215.

[5] Branch, M. A. and Grace, A. (1996). *Matlab Optimiztion Toolbox*. The Mathworks, Inc.

[6] Chilali, M. and Gahinet, P. (1996). $\mathcal{H}_\infty$ control design with pole placement constraints: An LMI approach. *IEEE Transactions on Automatic Control* **41**(3), 358–367.

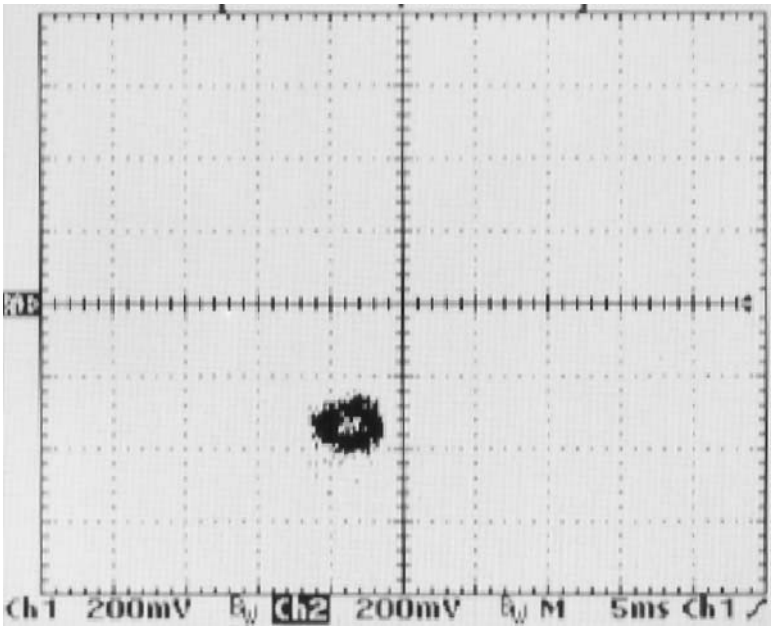


Fig. 10.27. Experimental responses to 10 000 rpm. x-axis controller: $\mathcal{H}_\infty^{5000}$, y-axis controller: internal.

[7] Dullerud, G. E. and Paganini, F. (2000). *A Course in Robust Control Theory*. Springer-Verlag.

[8] Gahinet, P. and Apkarian, P. (1994). An LMI apporach to $\mathcal{H}_\infty$ control. *International Journal of Robust and Nonlinear Control* **4**(8), 421–448.

[9] Gahinet, P., Nemirovski, A., Laub, A. J., and Chilali, M. (1995). *LMI Control Toolbox*. The Mathworks, Inc.

[10] Ghersin, A. S. (2006). Applicability of LPV synthesis with full block multipliers. *Proceedings of the XX Congreso Argentino de Control Automático – AADECA 2006, Buenos Aires - Argentina*. AADECA.

[11] Ghersin, A. S. and Sánchez Peña, R. S. (2002). LPV control of a 6 DOF vehicle. *IEEE Transactions on Control Systems Technology* **10**(6).

[12] Magnetic Moments, LLC (1999). *MBC500 Manual*.

[13] Matsumura, F., Namerikawa, T., Hagiwara, K., and Fujita, M. (1996). Application of gain scheduled $\mathcal{H}_\infty$ robust controllers to a magnetic bearing. *IEEE Transactions on Control Systems Technology* **4**(5), 484–493.

[14] Morse, N., Smith, R., and Paden, B. (1996). Magnetic bearing lab # 1: Analytical modeling of a magnetic bearing system.

[15] Morse Thibeault, N. and Smith, R. S. (2002). Magnetic bearing measurement configurations and associated robustness and performance limitations. *ASME Journal of Dynamic Systems, Measurement and Control* **124**(4), 589–598.

[16] Paden, B., Morse, N., and Smith, R. (1996). Magnetic bearing experiment for integrated teaching and research laboratories. *Proceedings of the 1996 IEEE Conference on Control Applications*.

[17] Scherer, C. W. (2001). LPV control and full block multipliers. *Automatica* **37**(3), 361–375.

[18] Seron, M. M., Braslavsky, J. H., and Goodwin, G. C. (1997). *Fundamental Limitations in Filtering and Control*. Springer-Verlag, London.

[19] Shamma, J. (1988). *Analysis and Design of Gain Scheduled Control Systems*. Ph.D. thesis, Dept. of Mechanical Engineering, MIT.

[20] Shamma, J. and Athans, M. (1992). Gain scheduling: Potential hazards and possible remedies. *IEEE Control Systems Magazine* **12**(3), 101–107.

[21] Smith, R. S. (1998). Closed-loop identification of flexible structures: An experimental example. *AIAA Journal of Guidance, Control and Dynamics* **21**(3), 435–440.

[22] Sánchez Peña, R. S. and Sznaier, M. (1998). *Robust Systems Theory and Applications*. John Wiley & Sons, Inc.

[23] Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software* **11–12**, 625–653. Version 1.1R3 available from `http://sedumi.mcmaster.ca`.

[24] Toh, K. C., Tütüncü, R. H., and Todd, M. J. (2007). SDPT^3 - a MATLAB software package for semidefinite-quadratic-linear programming. `http://www.math.cmu.edu/~reha/sdpt3.html`.

[25] Wu, F., Yang, X. H., Packard, A., and Becker, G. S. (1996). Induced $\mathcal{L}_2$–norm control for LPV systems with bounded parameter variation rates. *International Journal of Nonlinear and Robust Control* **6**(9), 983–998.

[26] Zhou, K. (1996). *Robust and Optimal Control*. Prentice–Hall.

[27] Zhou, K. (1998). *Essentials of Robust Control*. Prentice–Hall.

Index