

Advanced Textbooks in Control and Signal Processing

Da-Wei Gu
Petko H. Petkov
Mihail M. Konstantinov

Robust Control Design with MATLAB®

Second Edition



Springer

Advanced Textbooks in Control and Signal Processing

Editors

M.J. Grimble

M.A. Johnson

Glasgow, UK

For further volumes:

www.springer.com/series/4045

This page intentionally left blank

Da-Wei Gu • Petko H. Petkov •
Mihail M. Konstantinov

Robust Control Design with MATLAB[®]

Second Edition

 Springer

Da-Wei Gu
Department of Engineering
University of Leicester
Leicester, UK

Mihail M. Konstantinov
Civil Engineering and Geodesy
University of Architecture
Sofia, Bulgaria

Petko H. Petkov
Department of Automatics
Technical University of Sofia
Sofia, Bulgaria

Additional material to this book can be downloaded from <http://extras.springer.com>.

ISSN 1439-2232 Advanced Textbooks in Control and Signal Processing
ISBN 978-1-4471-4681-0 ISBN 978-1-4471-4682-7 (eBook)
DOI 10.1007/978-1-4471-4682-7
Springer London Heidelberg New York Dordrecht

Library of Congress Control Number: 2013938042

© Springer-Verlag London 2005, 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

To our families

This page intentionally left blank

Series Editors' Foreword

The topics of control engineering and signal processing continue to flourish and develop. In common with general scientific investigation, new ideas, concepts, and interpretations emerge quite spontaneously and these are then discussed, used, discarded or subsumed into the prevailing subject paradigm. Sometimes these innovative concepts coalesce into a new sub-discipline within the broad subject tapestry of control and signal processing. This preliminary battle between old and new usually takes place at conferences, through the Internet and in the journals of the discipline. After a little more maturity has been acquired by the new concepts then archival publication as a scientific or engineering monograph may occur.

A new concept in control and signal processing is known to have arrived when sufficient material has evolved for the topic to be taught as a specialized tutorial workshop or as a course to undergraduate, graduate or industrial engineers. *Advanced Textbooks in Control and Signal Processing* are designed as a vehicle for the systematic presentation of course material for both popular and innovative topics in the discipline. It is hoped that prospective authors will welcome the opportunity to publish a structured and systematic presentation of some of the newer emerging control and signal processing technologies in the textbook series.

It is always interesting to look back and review how a particular control systems theory developed. Intimations of paradigm change often seem to be associated with short sharp papers that make the academic community think again about the prevailing theoretical orthodoxy; so it was with robust control theory in which the conference papers of George Zames and colleagues (circa 1980) and the short paper of Doyle on the robustness of control systems using the very popular linear quadratic Gaussian design method now stand as landmark contributions that initiated a new control design paradigm. All through the 1980s came a steady stream of papers that began to change the very language of control itself with terms like robustness, system uncertainty, H_∞ control design and μ synthesis entering the control lexicon. This evolutionary development was vividly captured by Professor Christos G. Cassandras writing in the *IEEE Control Systems Magazine* (Vol. 32, No. 3, pp. 10–13, 2012) in a short informal study article on papers submitted to the premier control theory conference, the CDC; he wrote “The term “robust” appears in zero titles or

abstracts in 1971 before showing up in 13–14 % of papers in 1991 and 2001". In fact in 1981, the proportion was just 3.4 %, and this was around the time that the academic community was beginning to take note that change was at hand.

Throughout the 1960s and 1970s the formalism of state space and the methods of optimal control were in the ascendant for system modeling and academic control system design. One feature of this framework is that it is able to provide solutions for multivariable system problems. However, it was found that despite clever use of advanced matrix analysis and computer-aided design packages, classical control frequency domain techniques did not easily generalize to multivariable systems; a missing part of the control systems toolbox was really successful frequency domain design methods for multivariable systems. Thus, from the 1980s onward the tools of robust control for multivariable systems were developed and refined, and MATLAB[®] routines were developed to facilitate the implementation of the new design methods. As befits a "unified" theory, one nice feature of the new robust control methods is that it subsumes the terminology of classical control into the general framework. With this development of the robust control paradigm, the publication of the book *Robust Control Design with MATLAB[®]* by Da-Wei Gu, Petko H. Petkov, and Mihail Konstantinov in the *Advanced Textbooks in Control and Signal Processing* series in 2005 was very timely and the book was considered to be an excellent addition to this series of advanced control textbooks.

With the benefit of hindsight, with the perspective of seven more years of consolidation in robust control theory and practice, and with a new MATLAB Robust Control Toolbox 3 available, Da-Wei Gu, Petko H. Petkov, and Mihail Konstantinov have now produced this second edition of their book. The revised text has grown from 13 chapters and 389 pages to 19 chapters and just over 460 pages. But most significantly the textbook has a slightly different and improved structure that makes the MATLAB material more accessible and focused. Part I still looks at the different and basic elements of robust control design procedures. However, there is a useful new chapter on general results that use Linear Matrix Inequalities (LMIs). The newly introduced Part II presents chapters focusing on the use of tools and routines from the Robust Control Toolbox Version 3. Thus, whereas in Part I the theory of the robust control procedures is outlined, in this new part, the MATLAB implementations are presented and demonstrated. Finally, in Part III we find six in-depth design studies, two of which are entirely new. The case studies carried over from the first edition of the textbook are the benchmark system examples, namely, hard-disc-drive control, the triple inverted pendulum system, a high-purity distillation system, and a flexible manipulator system, which is a distributed-parameter system. The two new case studies are a twin rotor aerodynamical system and a self-balancing two-wheeled robot; both systems that are often found as test rigs in the control engineering teaching laboratory; a feature of these two new design studies is the presentation of experimental results from laboratory test-rigs to demonstrate and assess the performance of the robust control designs.

We have now passed through some 30 years of consolidation since the academic control community began to look again at the foundations of control system design and formulated and developed the robust control paradigm. The fact that the

MATLAB Robust Control Toolbox has reached “Version 3” demonstrates how robust control and MATLAB remain pervasive tools for the academic and industrial control engineering community. Thus the Editors are very pleased to welcome this second edition of *Robust Control Design with MATLAB*[®]. It is a valuable contribution to the list of publications in the *Advanced Textbooks in Control and Signal Processing* series; the authors' new structure and material brings additional pedagogical value to the new edition as an advanced textbook for teaching on courses and for self-study by the control engineer.

Glasgow, Scotland, UK
July 2012

M.J. Grimble
M.A. Johnson

This page intentionally left blank

Preface to the Second Edition

It has been encouraging to receive comments and suggestions on this book of *Robust Control Design with MATLAB*[®] since it was published in 2005. And, we are now very pleased to take the opportunity to offer our readers the 2nd edition of the book.

Robustness is of crucial importance in control systems design, because real engineering systems are vulnerable to external disturbance and measurement noise, and there are always discrepancies between mathematical models used for design and the actual system in practice. Typically a control engineer is required to design a controller which will make the closed loop system stable and achieve certain performance levels in the presence of disturbance signals, noise interference, unmodeled plant dynamics and plant parameter variations. The purpose of this book is to help post-graduate students and control engineers learn how to use well-developed, advanced robust control system design methods and the state-of-the-art MATLAB[®] tools in practical design cases.

A major feature of the second edition is the inclusion of an introduction to the use of the Robust Control Toolbox[®]3, as a new Part II between “Basic Methods and Theory” (Part I) and “Design Examples” (now Part III). Main function routines available in Robust Control Toolbox[®]3, ranging from building dynamic plant models and uncertainty models, systems stability and performance analysis to design of \mathcal{H}_∞ and μ as well as parameter-dependent controllers, are introduced in this new Part II. With relatively simple examples, such as the *mass–spring–damper* system, which is widely used in teaching laboratories of control engineering courses, usage of those MATLAB[®] routines is explained and illustrated at a level easier to be understood by readers, which of course follows the guidance in writing this book. In addition to this new Part II and correction of a few minor errors in the first edition, there is also a new chapter in Part I on introduction to Linear Matrix Inequalities (LMI). Although LMI is not directly used in those examples presented in Part III, it is indirectly used in some controller solution procedures. With its value in control systems analysis and design and its popularity, a brief introduction on LMI’s basic properties and use in control engineering would be welcomed by users, we hope. There are also some changes in Part III. The *mass–spring–damper* sys-

tem case has now been moved to Part II for illustration of using Robust Control Toolbox[®]3 function routines. The *robust control of a rocket* case has been removed. Instead, we now include two other design examples, namely *robust control of a twin-rotor aerodynamic system* and *robust control of self-balancing two-wheeled robot*. These two new examples are very popular experimental devices widely available in control teaching laboratories. Inclusion of these examples will better help readers to understand robust control design methods and to practice their own design which can be tested and verified in their own experiments. All design examples in Part III are realistic case studies. They are presented in Part III in detail. These design exercises are all conducted using Robust Control Toolbox[®]3. It is the authors' hope that studying these examples with attached programs used in all designs, with minimum exposure of theory and formulas in earlier parts of the book, will help readers obtain essential ideas and useful insights of several important robust control systems design approaches, including \mathcal{H}_∞ and related methods and μ -synthesis methods, and develop their own skills to design real-world industrial, robust control systems.

The authors are indebted to several people and institutions who helped them in the preparation of the book. We are particularly grateful to The MathWorks, Inc. for their continuous support, to Professor Sigurd Skogestad of Norwegian University of Science and Technology who kindly provided the nonlinear model of the Distillation Column, to Associate Professor Georgi Lehov from Technical University of Russe, Bulgaria, who developed the uncertainty model of the Flexible-Link Manipulator and to Associate Professor Tsonyo Slavov from Technical University of Sofia, Bulgaria, who developed the Simulink[®] model of the Hydro Turbine Gain Scheduling Control. As always, help from the Springer Editors and comments from the Series Editors in preparing this second edition are highly appreciated.

Using the Downloadable Material

For readers who purchased this book, supporting material, comprised of six folders with 170 M- and MDL-files intended for design, analysis, and simulation of the six design examples in Part III, plus five folders containing 35 files with the examples from Part II, plus a Readme.m file and a pdf, hypertext version of the book can be downloaded from <http://www.springer.com/978-1-85233-938-8>.

In order to use the M- and MDL-files the reader should have at his/her disposition MATLAB[®] version R2011a or higher, with Robust Control Toolbox[®]3, Control System Toolbox v9.1 and Simulink[®] v7.7. The design and experiments with the last two examples (*Twin-Rotor Aerodynamic System* and *Self-Balancing Two-Wheeled Robot*) are performed with MATLAB[®] version R2007b in order to have compatibility with the corresponding third-party software for real-time control. Please note also that the programs downloadable are different from those contained on the CD ROM distributed with the first edition of the book, for the convenience of readers to

use the later version of Robust Control Toolbox instead of using functions from the obsolete μ -toolbox.

Leicester, UK
Sofia, Bulgaria
Sofia, Bulgaria

Da-Wei Gu
Petko H. Petkov
Mihail M. Konstantinov

This page intentionally left blank

Preface to the First Edition

Robustness has been an important issue in control-systems design ever since 1769 when James Watt developed his flyball governor. A successfully designed control system should be always able to maintain a stability and performance level in spite of uncertainties in system dynamics and/or in the working environment to a certain degree. Design requirements such as gain margin and phase margin in using classical frequency-domain techniques are solely for the purpose of robustness. The robustness issue was not that prominently considered during the period of 1960s and 1970s when system models could be much more accurately described and design methods were mainly mathematical optimizations in the time domain. Due to its importance, however, the research on robust design has been going on all the time. A breakthrough came in the late 1970s and early 1980s with the pioneering work by Zames [192] and Zames and Francis [193] on the theory, now known as the \mathcal{H}_∞ optimal control theory. The \mathcal{H}_∞ optimization approach and the μ -synthesis/analysis method are well developed and elegant. They provide systematic design procedures of robust controllers for linear systems, though the extension into nonlinear cases is being actively researched.

Many books have since been published on \mathcal{H}_∞ and related theories and methods [34, 45, 73, 150, 155, 158, 195, 196]. The algorithms to implement the design methods are readily available in software packages such as MATLAB® and Slicot [78]. However, from our experience in teaching and research projects, we have felt that a reasonable percentage of people, students as well as practicing engineers, still have difficulties in applying the \mathcal{H}_∞ and related theory and in using MATLAB® routines. The mathematics behind the theory is quite involved. It is not straightforward to formulate a practical design problem, which is usually nonlinear, into the \mathcal{H}_∞ or μ design framework and then apply MATLAB® routines. This hinders the application of such a powerful theory. It also motivated us to prepare this book.

This book is for people who want to learn how to deal with robust control-system design problems but may not want to research the relevant theoretic developments. Methods and solution formulas are introduced in the first part of the book, but kept to a minimum. The majority of the book is devoted to several practical design case studies (Part II). These design examples, ranging from teaching laboratory experi-

ments such as a mass–damper–spring system to complex systems such as a supersonic rocket autopilot and a flexible-link manipulator, are discussed with detailed presentations. The design exercises are all conducted using the new *Robust Control Toolbox v3.0* and are in a hands-on, tutorial manner. Studying these examples with the attached MATLAB[®] and Simulink[®] programs (170 plus M- and MDL-files) used in all designs will help the readers learn how to deal with nonlinearities involved in the system, how to parameterize dynamic uncertainties and how to use MATLAB[®] routines in the analysis and design, etc. It is also hoped that by going through these exercises the readers will understand the essence of robust control system design and develop their own skills to design real, industrial, robust control systems.

The readership of this book is postgraduates and control engineers, though senior undergraduates may use it for their final-year projects. The material included in the book has been adopted in recent years for M.Sc. and Ph.D. engineering students at Leicester University and at the Technical University of Sofia. The design examples are independent of each other. They have been used extensively in the laboratory projects on the course *Robust and Optimal Control Systems* taught in a masters programme in the Technical University of Sofia.

The authors are indebted to several people and institutions who helped them in the preparation of the book. We are particularly grateful to The MathWorks, Inc. for their continuous support, to Professor Sigurd Skogestad of Norwegian University of Science and Technology who kindly provided the nonlinear model of the Distillation Column and to Associate Professor Georgi Lehov from Technical University of Russe, Bulgaria, who developed the uncertainty model of the Flexible-Link Manipulator.

Using the CD ROM

The attached CD ROM contains six folders with M- and MDL-files intended for design, analysis, and simulation of the six design examples, plus a pdf file with color hypertext version of the book. In order to use the M- and MDL-files the reader should have at his (her) disposition MATLAB[®] v7.0.2 with Robust Control Toolbox v 3.0, Control System Toolbox v6.1 and Simulink[®] v6.1. Further information on the use of the files can be found in the file Readme.m on the disc.

Contents

Part I Basic Methods and Theory

1	Introduction	3
1.1	Control-System Representations	4
1.2	System Stabilities	5
1.3	Coprime Factorization and Stabilizing Controllers	7
1.4	Signals and System Norms	9
1.4.1	Vector Norms and Signal Norms	9
1.4.2	System Norms	10
2	Modeling of Uncertain Systems	13
2.1	Unstructured Uncertainties	13
2.2	Parametric Uncertainty	17
2.3	Linear Fractional Transformations	19
2.4	Structured Uncertainties	21
3	Robust Design Specifications	23
3.1	Small-Gain Theorem and Robust Stabilization	23
3.2	Performance Considerations	26
3.3	Structured Singular Values	27
4	\mathcal{H}_∞ Design	31
4.1	Mixed Sensitivity \mathcal{H}_∞ Optimization	31
4.2	2-Degree-of-Freedom \mathcal{H}_∞ Design	33
4.3	\mathcal{H}_∞ Suboptimal Solutions	34
4.3.1	Solution Formulas for Normalized Systems	35
4.3.2	Solution to S -over- $K S$ Design	38
4.3.3	The Case of $D_{22} \neq 0$	39
4.3.4	Normalization Transformations	40
4.3.5	Direct Formulas for \mathcal{H}_∞ Suboptimal Central Controller	41
4.4	Formulas for Discrete-Time Cases	45

5	\mathcal{H}_∞ Loop-Shaping Design Procedures	49
5.1	Robust Stabilization Against Normalized Coprime Factor Perturbations	49
5.2	Loop-Shaping Design Procedures	52
5.3	Formulas for the Discrete-Time Case	54
5.3.1	Normalized Coprime Factorization of Discrete-Time Plant	54
5.3.2	Robust Controller Formulas	56
5.3.3	The Strictly Proper Case	57
5.3.4	On the Three DARE Solutions	58
5.4	A Mixed Optimization Design Method with LSDP	60
6	μ-Analysis and Synthesis	65
6.1	Consideration of Robust Performance	65
6.2	μ -Synthesis: D - K Iteration Method	68
6.3	μ -Synthesis: μ - K Iteration Method	70
7	Lower-Order Controllers	73
7.1	Absolute-Error Approximation Methods	74
7.1.1	Balanced Truncation Method	75
7.1.2	Singular Perturbation Approximation	76
7.1.3	Hankel-Norm Approximation	77
7.2	Reduction via Fractional Factors	80
7.2.1	Fractional Balanced Truncation (FBT) Method	82
7.2.2	Fractional Singular Perturbation Approximation (FSPA) Method	82
7.3	Relative-Error Approximation Methods	84
7.4	Frequency-Weighted Approximation Methods	86
7.4.1	Frequency-Weighted Balanced Truncation (FWBT)	88
7.4.2	Frequency-Weighted Singular Perturbation Approximation (FWSPA)	89
7.4.3	Frequency-Weighted Moduli Truncation Method (FWMT)	89
8	LMI Approach	93
8.1	Basics About LMI	93
8.2	Control Problems Using LMI	94
8.2.1	Lyapunov Stability Criterion	94
8.2.2	Stabilization by State Feedback	94
8.2.3	Computation of \mathcal{L}_2 Norm	95
8.2.4	Computation of \mathcal{H}_∞ Norm	97
8.2.5	Formulation of LQR in LMI	97
8.3	A Few More Properties Concerning LMI	99
8.3.1	Congruence Transformation	99
8.3.2	Schur Complements for Nonstrict Inequalities	100
8.3.3	Projection and Finsler's Lemmas	100
8.3.4	The S-Procedure for Quadratic Functions	102
8.3.5	Dualization Lemma	102

Part II Introduction to Robust Control Toolbox v3

9	Building Uncertain Models	107
9.1	LTI Models	107
9.2	Structured Uncertainty Models	121
9.2.1	Uncertain Real Parameters	123
9.2.2	Uncertain State-Space Systems	124
9.2.3	Properties of Uncertain Systems	125
9.2.4	Other Functions to Build Uncertain Models	130
9.2.5	Decomposing Uncertain Objects	131
9.3	Building Uncertain Models Using <code>iconnect</code> and <code>sysic</code>	132
9.4	Unstructured Uncertainty Models	134
9.4.1	Models with Additive Uncertainty	135
9.4.2	Models with Multiplicative Uncertainty	138
9.4.3	Unmodeled Dynamics	140
9.4.4	Multivariable Plants with Unstructured Uncertainty	142
9.5	Exercises	143
10	Robust Stability and Performance	145
10.1	Robust Stability Analysis	145
10.2	Robust Performance Analysis	154
10.3	Worst-Case Gain	165
10.4	Exercises	170
11	\mathcal{H}_∞ Design	173
11.1	\mathcal{H}_∞ Loop-Shaping Design	173
11.2	Mixed Sensitivity Design	179
11.3	Other Versions of \mathcal{H}_∞ Design	187
11.3.1	\mathcal{H}_∞ Control with Models	187
11.3.2	Two-Degree-of-Freedom \mathcal{H}_∞ Control	194
11.4	Exercises	200
12	μ-Synthesis	203
12.1	The μ -Synthesis Problem	203
12.2	μ -Synthesis by D - K Iterations	205
12.3	Versions of μ -Synthesis	212
12.3.1	μ -Synthesis with Model	212
12.3.2	μ -Synthesis of 2-Degree-of-Freedom Controller	214
12.4	Practical Aspects of μ -Analysis and μ -Synthesis	216
12.5	Exercises	219
13	Analysis and Design of Parameter-Dependent Systems	221
13.1	Representation of Parameter-Dependent Systems	221
13.1.1	SYSTEM Matrix	221
13.1.2	Affine Parameter-Dependent Models	223
13.1.3	Polytopic Models	227
13.2	Analysis of Parameter-Dependent Systems	229

13.3	Gain Scheduling Design for Parameter-Dependent Systems . . .	234
13.4	Exercises	246

Part III Design Examples

14	Robust Control of a Hard Disk Drive	249
14.1	Hard Disk Drive Servo System	249
14.2	Derivation of Uncertainty Model	255
14.3	Closed-Loop System Design Specifications	258
14.3.1	Nominal Performance	260
14.3.2	Robust Stability	260
14.3.3	Robust Performance	260
14.4	System Interconnections	261
14.5	Controller Design in Continuous-Time	262
14.5.1	μ -Design	263
14.5.2	\mathcal{H}_∞ Design	269
14.5.3	\mathcal{H}_∞ Loop-Shaping Design	269
14.6	Comparison of Designed Controllers	270
14.7	Controller-Order Reduction	277
14.8	Design of Discrete-Time Controller	279
14.9	Nonlinear System Simulation	284
14.10	Conclusions	289
14.11	Notes and References	289
15	A Triple Inverted Pendulum Control System Design	291
15.1	System Description	292
15.2	Modeling of Uncertainties	295
15.3	Design Specifications	302
15.3.1	Robust Stability	304
15.3.2	Nominal Performance	304
15.3.3	Robust Performance	304
15.4	System Interconnections	306
15.5	\mathcal{H}_∞ Design	308
15.6	μ -Synthesis	313
15.7	Nonlinear System Simulation	322
15.8	Conclusions	322
15.9	Notes and References	325
16	Robust Control of a Distillation Column	327
16.1	Introduction	327
16.2	Dynamic Model of the Distillation Column	328
16.3	Uncertainty Modeling	332
16.4	Closed-Loop System Performance Specifications	334
16.5	Open-Loop and Closed-Loop System Interconnections	339
16.6	Controller Design	342
16.6.1	Loop-Shaping Design	343
16.6.2	μ -Synthesis	346

16.7	Nonlinear System Simulation	362
16.8	Conclusions	364
16.9	Notes and References	365
17	Robust Control of a Flexible-Link Manipulator	367
17.1	Dynamic Model of the Flexible Manipulator	368
17.2	A Linear Model of the Uncertain System	371
17.3	System Performance Specifications	378
17.4	System Interconnections	381
17.5	Controller Design and Analysis	383
17.6	Nonlinear System Simulations	394
17.7	Conclusions	395
17.8	Notes and References	399
18	Robust Control of a Twin-Rotor Aerodynamic System	401
18.1	Twin-Rotor Aerodynamic System	401
18.2	Nonlinear System Model	402
18.3	Linearized System Model	407
18.4	Uncertainty Modeling	410
18.5	Closed-Loop System Performance Requirements	412
	18.5.1 Robust Stability	414
	18.5.2 Nominal Performance	414
	18.5.3 Robust Performance	414
18.6	System Interconnections	415
18.7	μ -Synthesis	415
18.8	Nonlinear System Simulation	427
18.9	Experimental Results	428
18.10	Conclusions	428
	18.10.1 Notes and References	433
19	Robust Control of Self-balancing Two-Wheeled Robot	435
19.1	Introduction	435
19.2	Uncertain Model of the Two-Wheeled Robot	436
19.3	Design of Robust Controller	440
19.4	Closed-Loop System Properties	444
19.5	Experimental Results	450
19.6	Conclusions	452
19.7	Notes and References	453
	References	455
	Index	465

This page intentionally left blank

Part I

Basic Methods and Theory

This page intentionally left blank

Chapter 1

Introduction

Robustness is of crucial importance in control-system design because real engineering systems are vulnerable to external disturbance and measurement noise and there are always differences between mathematical models used for design and the actual system. Typically, a control engineer is required to design a controller that will stabilize a plant, if it is not stable originally, and satisfy certain performance levels in the presence of disturbance signals, noise interference, unmodeled plant dynamics and plant-parameter variations. These design objectives are best realized via the feedback control mechanism, although it introduces in the issues of high cost (the use of sensors), system complexity (implementation and safety) and more concerns on stability (thus internal stability and stabilizing controllers).

Though always having been appreciated, the need and importance of robustness in control-systems design has been particularly brought into the limelight during the last two decades. In classical single-input single-output control, robustness is achieved by ensuring good gain and phase margins. Designing for good stability margins usually also results in good, well-damped time responses, i.e. good performance. When multivariable design techniques were first developed in the 1960s, the emphasis was placed on achieving good performance, and not on robustness. These multivariable techniques were based on linear quadratic performance criteria and Gaussian disturbances, and proved to be successful in many aerospace applications where accurate mathematical models can be obtained, and descriptions for external disturbances/noise based on white noise are considered appropriate. However, application of such methods, commonly referred to as the linear quadratic Gaussian (LQG) methods, to other industrial problems made apparent the poor robustness properties exhibited by LQG controllers. This led to a substantial research effort to develop a theory that could explicitly address the robustness issue in feedback design. The pioneering work in the development of the forthcoming theory, now known as the \mathcal{H}_∞ optimal control theory, was conducted in the early 1980s by Zames [192] and Zames and Francis [193]. In the \mathcal{H}_∞ approach, the designer from the outset specifies a model of system uncertainty, such as additive perturbation and/or output disturbance (details in Chap. 2), which is most suited to the problem at hand. A constrained optimization is then performed to maximize the robust stabil-

ity of the closed-loop system to the type of uncertainty chosen, the constraint being the internal stability of the feedback system. In most cases, it would be sufficient to seek a feasible controller such that the closed-loop system achieves certain robust stability. Performance objectives can also be included in the optimization cost function. Elegant solution formulas have been developed, which are based on the solutions of certain algebraic Riccati equations, and are readily available in software packages such as Slicot [78] and MATLAB®.

Despite the mature theory [34, 45, 73, 150, 158, 195, 196] and availability of software packages, commercial or licensed freeware, many people have experienced difficulties in solving industrial control-systems design problems with these \mathcal{H}_∞ and related methods, due to the complex mathematics of the advanced approaches and numerous presentations of formulas as well as adequate translations of industrial design into relevant configurations. This book aims at bridging the gap between the theory and applications. By sharing the experience in industrial case studies with minimum exposure to the theory and formulas, the authors hope readers will obtain insight into robust industrial control-system designs using major \mathcal{H}_∞ optimization and related methods.

In this chapter, the basic concepts and representations of systems and signals will be discussed.

1.1 Control-System Representations

A control system or plant or process is an interconnection of components to perform certain tasks and to yield a desired response, i.e. to generate desired signal (the output), when it is driven by manipulating signal (the input). A control system is a causal, dynamic system, i.e. the output depends not only the present input but also the input at the previous time.

In general, there are two categories of control systems, the open-loop systems and closed-loop systems. An open-loop system uses a controller or control actuator to obtain the design response. In an open-loop system, the output has no effect on the input. In contrast to an open-loop system, a closed-loop control system uses sensors to measure the actual output to adjust the input in order to achieve desired output. The measure of the output is called the feedback signal, and a closed-loop system is also called a feedback system. It will be shown in this book that only feedback configurations are able to achieve the robustness of a control system.

Due to the increasing complexity of physical systems under control and rising demands on system properties, most industrial control systems are no longer single-input and single-output (SISO) but multi-input and multi-output (MIMO) systems with a high interrelationship (coupling) between these channels. The number of (state) variables in a system could be very large as well. These systems are called multivariable systems.

In order to analyze and design a control system, it is advantageous if a mathematical representation of such a relationship (a model) is available. The system

dynamics is usually governed by a set of differential equations in either open-loop or closed-loop systems. In the case of linear, time-invariant systems, which is the case this book considers, these differential equations are linear ordinary differential equations. By introducing appropriate state variables and simple manipulations, a linear, time-invariant, continuous-time control system can be described by the following model:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\tag{1.1}$$

where $x(t) \in R^n$ is the state vector, $u(t) \in R^m$ the input (control) vector, and $y(t) \in R^p$ the output (measurement) vector.

With the assumption of zero initial condition of the state variables and using Laplace transform, a transfer function matrix corresponding to the system in (1.1) can be derived as

$$G(s) := C(sI_n - A)^{-1}B + D\tag{1.2}$$

and can be further denoted in a short form by

$$G(s) =: \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]\tag{1.3}$$

It should be noted that the \mathcal{H}_∞ optimization approach is a frequency-domain method, though it utilizes the time-domain description such as (1.1) to explore the advantages in numerical computation and to deal with multivariable systems. The system given in (1.1) is assumed in this book to be minimal, i.e. completely controllable and completely observable, unless described otherwise.

In the case of discrete-time systems, similarly the model is given by

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k)\end{aligned}\tag{1.4}$$

or

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k\end{aligned}$$

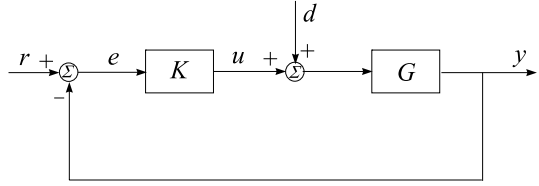
with a corresponding transfer function matrix as

$$\begin{aligned}G(z) &:= C(zI_n - A)^{-1}B + D \\ &=: \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]\end{aligned}\tag{1.5}$$

1.2 System Stabilities

An essential issue in control-systems design is the *stability*. An unstable system is of no practical value. This is because any control system is vulnerable to disturbances

Fig. 1.1 An interconnected system of G and K



and noises in a real work environment, and the effect due to these signals would adversely affect the expected, normal system output in an unstable system. Feedback control techniques may reduce the influence generated by uncertainties and achieve desirable performance. However, an inadequate feedback controller may lead to an unstable closed-loop system though the original open-loop system is stable. In this section, control-system stabilities and stabilizing controllers for a given control system will be discussed.

When a dynamic system is just described by its input/output relationship such as a transfer function (matrix), the system is stable if it generates bounded outputs for any bounded inputs. This is called the bounded-input-bounded-output (BIBO) stability. For a linear, time-invariant system modeled by a transfer function matrix ($G(s)$ in (1.2)), the BIBO stability is guaranteed if and only if all the poles of $G(s)$ are in the open-left-half complex plane, i.e. with negative real parts.

When a system is governed by a state-space model such as (1.1), a stability concept called *asymptotic stability* can be defined. A system is asymptotically stable if, for an identically zero input, the system state will converge to zero from any initial states. For a linear, time-invariant system described by a model of (1.1), it is asymptotically stable if and only if all the eigenvalues of the state matrix A are in the open-left-half complex plane, i.e. with negative real parts.

In general, the asymptotic stability of a system implies that the system is also BIBO stable, but not vice versa. However, for a system in (1.1), if $[A, B, C, D]$ is of minimal realization, the BIBO stability of the system implies that the system is asymptotically stable.

The above stabilities are defined for open-loop systems as well as closed-loop systems. For a closed-loop system (interconnected, feedback system), it is more interesting and intuitive to look at the asymptotic stability from another point of view and this is called the *internal stability* [27]. An interconnected system is internally stable if the subsystems of all input–output pairs are asymptotically stable (or the corresponding transfer function matrices are BIBO stable when the state space models are minimal, which is assumed in this chapter). Internal stability is equivalent to asymptotic stability in an interconnected, feedback system but may reveal explicitly the relationship between the original, open-loop system and the controller that influences the stability of the whole system. For the system given in Fig. 1.1, there are two inputs r and d (the disturbance at the output) and two outputs y and u (the output of the controller K).

The transfer functions from the inputs to the outputs, respectively, are

$$\begin{aligned} T_{yr} &= GK(I + GK)^{-1} \\ T_{yd} &= G(I + KG)^{-1} \\ T_{ur} &= K(I + GK)^{-1} \\ T_{ud} &= -KG(I + KG)^{-1} \end{aligned} \quad (1.6)$$

Hence, the system is internally stable if and only if all the transfer functions in (1.6) are BIBO stable, or the transfer function matrix M from $\begin{bmatrix} r \\ d \end{bmatrix}$ to $\begin{bmatrix} y \\ u \end{bmatrix}$ is BIBO stable, where

$$M := \begin{bmatrix} GK(I + GK)^{-1} & G(I + KG)^{-1} \\ K(I + GK)^{-1} & -KG(I + KG)^{-1} \end{bmatrix} \quad (1.7)$$

The stability of (1.7) is equivalent to the stability of

$$\hat{M} := \begin{bmatrix} I - GK(I + GK)^{-1} & -G(I + KG)^{-1} \\ K(I + GK)^{-1} & I - KG(I + KG)^{-1} \end{bmatrix} \quad (1.8)$$

By simple matrix manipulations, we have

$$\begin{aligned} \hat{M} &= \begin{bmatrix} (I + GK)^{-1} & -G(I + KG)^{-1} \\ K(I + GK)^{-1} & (I + KG)^{-1} \end{bmatrix} \\ &= \begin{bmatrix} I & G \\ -K & I \end{bmatrix}^{-1} \end{aligned} \quad (1.9)$$

Hence, the feedback system in Fig. 1.1 is internally stable if (1.9) is stable.

It can be shown [27] that if there is no unstable pole/zero cancellation between G and K , then any one of the four transfer functions being BIBO stable would be enough to guarantee that the whole system is internally stable.

1.3 Coprime Factorization and Stabilizing Controllers

Consider a system given in the form of (1.2) with $[A, B, C, D]$ assumed to be minimal. Matrices $(\tilde{M}(s), \tilde{N}(s)) \in \mathcal{H}_\infty$ ($(M(s), N(s)) \in \mathcal{H}_\infty$), where \mathcal{H}_∞ denotes the space of functions with no poles in the closed right-half complex plane, constitute a left (right) coprime factorization of $G(s)$ if and only if

- (i) $\tilde{M}^*(M)$ is square, and $\det(\tilde{M})(\det(M)) \neq 0$
- (ii) the plant model is given by

$$G = \tilde{M}^{-1} \tilde{N} \quad (= NM^{-1}) \quad (1.10)$$

- (iii) There exists $(\tilde{V}, \tilde{U})((V, U)) \in \mathcal{H}_\infty$ such that

$$\begin{aligned} \tilde{M}\tilde{V} + \tilde{N}\tilde{U} &= I \\ (UN + VM) &= I \end{aligned} \quad (1.11)$$

Transfer functions (or rational, fractional) matrices are *coprime* if they share no common zeros in the right-half complex plane, including at the infinity. The two equations in (iii) above are called Bezout identities [108] and are necessary and sufficient conditions for (\tilde{M}, \tilde{N}) ((M, N)) being left coprime (right coprime), respectively. The left and right coprime factorizations of $G(s)$ can be grouped together to form a Bezout double identity as the following:

$$\begin{bmatrix} V & U \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} M & -\tilde{U} \\ N & \tilde{V} \end{bmatrix} = I \quad (1.12)$$

For $G(s)$ of minimal realization (1.2) (actually G is required to be stabilizable and detectable only), the formulas for the coprime factors can be readily derived [109] as in the following theorem.

Theorem 1.1 *Let constant matrices F and H be such that $A + BF$ and $A + HC$ are both stable. Then the transfer function matrices \tilde{M} and \tilde{N} (M and N) defined in the following constitute a left (right) coprime factorization of $G(s)$:*

$$\begin{bmatrix} \tilde{N}(s) & \tilde{M}(s) \end{bmatrix} = \left[\begin{array}{c|cc} A + HC & B + HD & -H \\ \hline C & D & I \end{array} \right] \quad (1.13)$$

$$\begin{bmatrix} N(s) \\ M(s) \end{bmatrix} = \left[\begin{array}{c|c} A + BF & B \\ \hline C + DF & D \\ F & I \end{array} \right] \quad (1.14)$$

Furthermore, the following $\tilde{U}(s)$, $\tilde{V}(s)$, $U(s)$, and $V(s)$ satisfy the Bezout double identity (1.12):

$$\begin{bmatrix} \tilde{U}(s) & \tilde{V}(s) \end{bmatrix} = \left[\begin{array}{c|cc} A + HC & H & B + HD \\ \hline F & 0 & I \end{array} \right] \quad (1.15)$$

$$\begin{bmatrix} U(s) \\ V(s) \end{bmatrix} = \left[\begin{array}{c|c} A + BF & H \\ \hline F & 0 \\ C + DF & I \end{array} \right] \quad (1.16)$$

It can be easily shown that the pairs (\tilde{U}, \tilde{V}) and (U, V) are stable and coprime. Using (1.9), it is straightforward to show the following lemma.

Lemma 1.2

$$K := \tilde{V}^{-1} \tilde{U} = UV^{-1} \quad (1.17)$$

is a stabilizing controller, i.e. the closed-loop system in Fig. 1.6 is internally stable.

Further, the set of all stabilizing controllers for $G = \tilde{M}^{-1} \tilde{N} = NM^{-1}$ can be obtained in the following Youla Parameterization Theorem [109, 189, 190].

Theorem 1.3 *The set of all stabilizing controllers for G is*

$$\{(\tilde{V} + Q\tilde{N})^{-1}(\tilde{U} + Q\tilde{M}) : Q \in \mathcal{H}_\infty\} \quad (1.18)$$

The set can also be expressed as

$$\{(U + MQ)(V + NQ)^{-1} : Q \in \mathcal{H}_\infty\} \quad (1.19)$$

1.4 Signals and System Norms

In this section the basic concepts concerning signals and systems will be reviewed in brief. A control system interacts with its environment through command signals, disturbance signals and noise signals, etc. Tracking error signals and actuator driving signals are also important in control systems design. For the purpose of analysis and design, appropriate measures, the norms, must be defined for describing the “size” of these signals. From the signal norms, we can then define induced norms to measure the “gain” of the operator that represents the control system.

1.4.1 Vector Norms and Signal Norms

Let the linear space X be \mathcal{F}^m , where $\mathcal{F} = \mathcal{R}$ for the field of real numbers, or $\mathcal{F} = \mathcal{C}$ for complex numbers. For $x = [x_1, x_2, \dots, x_m]^T \in X$, the p -norm of the vector x is defined by

$$\begin{aligned} \text{1-norm} \quad \|x\|_1 &:= \sum_{i=1}^m |x_i|, & \text{for } p = 1 \\ \text{\textit{p-norm}} \quad \|x\|_p &:= \left(\sum_{i=1}^m |x_i|^p \right)^{1/p}, & \text{for } 1 < p < \infty \\ \text{\textit{\infty-norm}} \quad \|x\|_\infty &:= \max_{1 \leq i \leq m} |x_i|, & \text{for } p = \infty \end{aligned}$$

When $p = 2$, $\|x\|_2$ is the familiar Euclidean norm.

When X is a linear space of continuous or piecewise continuous time scalar-valued signals $x(t)$, $t \in \mathcal{R}$, the p -norm of a signal $x(t)$ is defined by

$$\begin{aligned} \text{1-norm} \quad \|x\|_1 &:= \int_{-\infty}^{\infty} |x(t)| dt, & \text{for } p = 1 \\ \text{\textit{p-norm}} \quad \|x\|_p &:= \left(\int_{-\infty}^{\infty} |x(t)|^p dt \right)^{1/p}, & \text{for } 1 < p < \infty \\ \text{\textit{\infty-norm}} \quad \|x\|_\infty &:= \sup_{t \in \mathcal{R}} |x(t)|, & \text{for } p = \infty \end{aligned}$$

The normed spaces, consisting of signals with finite norm as defined correspondingly, are called $L^1(\mathcal{R})$, $L^p(\mathcal{R})$, and $L^\infty(\mathcal{R})$, respectively. From a signal point of view, the 1-norm, $\|x\|_1$ of the signal $x(t)$ is the integral of its absolute value. The square of the 2-norm, $\|x\|_2^2$, is often called the *energy* of the signal $x(t)$ since that

is what it is when $x(t)$ is the current through a $1\ \Omega$ resistor. The ∞ -norm, $\|x\|_\infty$, is the amplitude or peak value of the signal, and the signal is bounded in magnitude if $x(t) \in L^\infty(\mathcal{R})$.

When X is a linear space of continuous or piecewise continuous *vector-valued* functions of the form $x(t) = [x_1(t), x_2(t), \dots, x_m(t)]^T$, $t \in \mathcal{R}$, we may have

$$L_m^p(\mathcal{R}) := \left\{ x(t) : \|x\|_p = \left(\int_{-\infty}^{\infty} \sum_{i=1}^m |x_i(t)|^p dt \right)^{1/p} < \infty, \text{ for } 1 \leq p < \infty \right\}$$

$$L_m^\infty(\mathcal{R}) := \left\{ x(t) : \|x\|_\infty = \sup_{t \in \mathcal{R}} \|x(t)\|_\infty < \infty \right\}$$

Some signals are useful for control systems analysis and design, for example, the sinusoidal signal, $x(t) = A \sin(\omega t + \phi)$, $t \in \mathcal{R}$. It is unfortunately not a 2-norm signal because of the infinite energy contained. However, the average power of $x(t)$

$$\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x^2(t) dt$$

exists. The signal $x(t)$ will be called a *power signal* if the above limit exists. The square root of the limit is the well-known r.m.s. (*root-mean-square*) value of $x(t)$. It should be noticed that the average power does not introduce a norm, since a nonzero signal may have zero average power.

1.4.2 System Norms

System norms are actually the input–output gains of the system. Suppose that \mathcal{G} is a linear and bounded system that maps the input signal $u(t)$ into the output signal $y(t)$, where $u \in (U, \|\cdot\|_U)$, $y \in (Y, \|\cdot\|_Y)$. U and Y are the signal spaces, endowed with the norms $\|\cdot\|_U$ and $\|\cdot\|_Y$, respectively. Then the norm, maximum system gain, of \mathcal{G} is defined as

$$\|\mathcal{G}\| := \sup_{u \neq 0} \frac{\|\mathcal{G}u\|_Y}{\|u\|_U} \quad (1.20)$$

or

$$\|\mathcal{G}\| = \sup_{\|u\|_U=1} \|\mathcal{G}u\|_Y = \sup_{\|u\|_U \leq 1} \|\mathcal{G}u\|_Y$$

Obviously, we have

$$\|\mathcal{G}u\|_Y \leq \|\mathcal{G}\| \cdot \|u\|_U$$

If \mathcal{G}_1 and \mathcal{G}_2 are two linear, bounded and compatible systems, then

$$\|\mathcal{G}_1 \mathcal{G}_2\| \leq \|\mathcal{G}_1\| \cdot \|\mathcal{G}_2\|$$

$\|\mathcal{G}\|$ is called the *induced norm* of \mathcal{G} with regard to the signal norms $\|\cdot\|_U$ and $\|\cdot\|_Y$. In this book, we are particularly interested in the so-called ∞ -norm

of a system. For a linear, time-invariant, stable system $\mathcal{G}: L_m^2(\mathcal{R}) \rightarrow L_p^2(\mathcal{R})$, the ∞ -norm, or the induced 2-norm, of \mathcal{G} is given by

$$\|\mathcal{G}\|_\infty = \sup_{\omega \in \mathcal{R}} \|G(j\omega)\|_2 \quad (1.21)$$

where $\|G(j\omega)\|_2$ is the spectral norm of the $p \times m$ matrix $G(j\omega)$ and $G(s)$ is the transfer function matrix of \mathcal{G} . Hence, the ∞ -norm of a system describes the maximum energy gain of the system and is decided by the peak value of the largest singular value of the frequency response matrix over the whole frequency axis. This norm is called the \mathcal{H}_∞ -norm, since we denote by \mathcal{H}_∞ the linear space of all stable linear systems.

This page intentionally left blank

Chapter 2

Modeling of Uncertain Systems

As discussed in Chap. 1, it is well understood that uncertainties are unavoidable in a real control system. The uncertainty can be classified into two categories: disturbance signals and dynamic perturbations. The former includes input and output disturbance (such as a gust on an aircraft), sensor noise and actuator noise, etc. The latter represents the discrepancy between the mathematical model and the actual dynamics of the system in operation. A mathematical model of any real system is always just an approximation of the true, physical reality of the system dynamics. Typical sources of the discrepancy include unmodeled (usually high-frequency) dynamics, neglected nonlinearities in the modeling, effects of deliberate reduced-order models, and system-parameter variations due to environmental changes and torn-and-worn factors. These modeling errors may adversely affect the stability and performance of a control system. In this chapter, we will discuss in detail how dynamic perturbations are usually described so that they can be accounted for in system robustness analysis and design.

2.1 Unstructured Uncertainties

Many dynamic perturbations that may occur in different parts of a system can, however, be lumped into one single perturbation block Δ , for instance, some unmodeled, high-frequency dynamics. This uncertainty representation is referred to as “unstructured” uncertainty. In the case of linear, time-invariant systems, the block Δ may be represented by an unknown transfer function matrix. The unstructured dynamics uncertainty in a control system can be described in different ways, such as is listed in the following, where $G_p(s)$ denotes the actual, perturbed system dynamics and $G_o(s)$ a nominal model description of the physical system.

1. Additive perturbation (see Fig. 2.1):

$$G_p(s) = G_o(s) + \Delta(s) \quad (2.1)$$

Fig. 2.1 Additive perturbation configuration

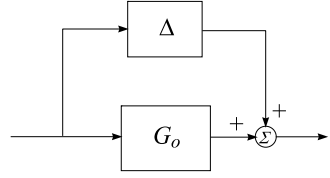


Fig. 2.2 Inverse additive perturbation configuration

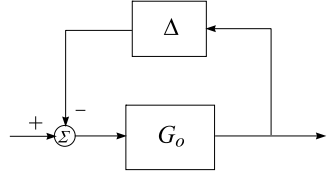


Fig. 2.3 Input multiplicative perturbation configuration

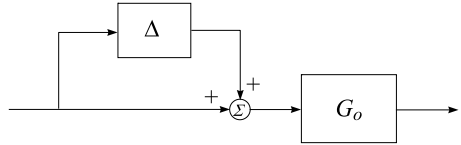


Fig. 2.4 Output multiplicative perturbation configuration

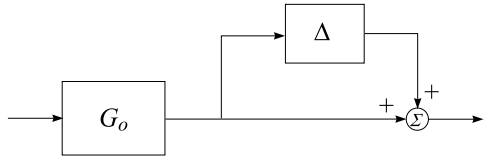
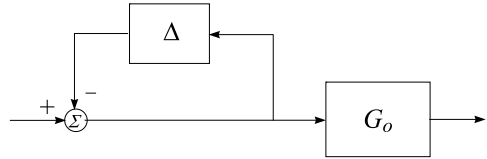


Fig. 2.5 Inverse input multiplicative perturbation configuration



2. Inverse additive perturbation (see Fig. 2.2):

$$(G_p(s))^{-1} = (G_o(s))^{-1} + \Delta(s) \quad (2.2)$$

3. Input multiplicative perturbation (see Fig. 2.3):

$$G_p(s) = G_o(s)[I + \Delta(s)] \quad (2.3)$$

4. Output multiplicative perturbation (see Fig. 2.4):

$$G_p(s) = [I + \Delta(s)]G_o(s) \quad (2.4)$$

5. Inverse input multiplicative perturbation (see Fig. 2.5):

$$(G_p(s))^{-1} = [I + \Delta(s)](G_o(s))^{-1} \quad (2.5)$$

Fig. 2.6 Inverse output multiplicative perturbation configuration

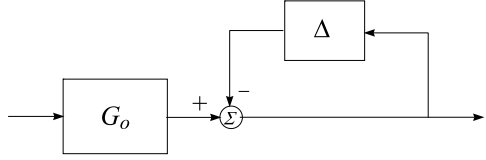


Fig. 2.7 Left coprime factor perturbations configuration

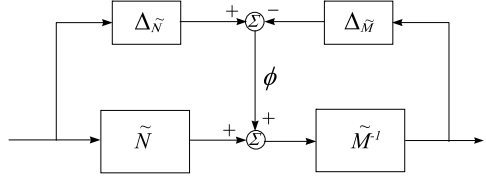
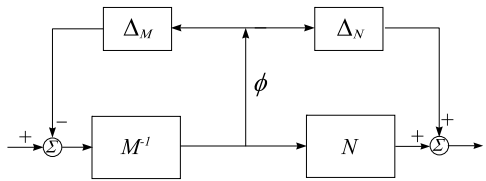


Fig. 2.8 Right coprime factor perturbations configuration



6. Inverse output multiplicative perturbation (see Fig. 2.6):

$$(G_p(s))^{-1} = (G_o(s))^{-1} [I + \Delta(s)] \quad (2.6)$$

7. Left coprime factor perturbations (see Fig. 2.7):

$$G_p(s) = (\tilde{M} + \Delta_{\tilde{M}})^{-1} (\tilde{N} + \Delta_{\tilde{N}}) \quad (2.7)$$

8. Right coprime factor perturbations (see Fig. 2.8):

$$G_p(s) = (N + \Delta_N)(M + \Delta_M)^{-1} \quad (2.8)$$

The additive uncertainty representations give an account of absolute error between the actual dynamics and the nominal model, while the multiplicative representations show relative errors.

In the last two representations, $(\tilde{M}, \tilde{N})/(M, N)$ are left/right coprime factorizations of the nominal system model $G_o(s)$, respectively; and $(\Delta_{\tilde{M}}, \Delta_{\tilde{N}})/(\Delta_M, \Delta_N)$ are the perturbations on the corresponding factors [111].

The block Δ (or, $(\Delta_{\tilde{M}}, \Delta_{\tilde{N}})/(\Delta_M, \Delta_N)$ in the coprime factor perturbations cases) is uncertain, but usually is norm-bounded. It may be bounded by a known transfer function, say $\overline{\sigma}[\Delta(j\omega)] \leq \delta(j\omega)$, for all frequencies ω , where δ is a known scalar function and $\overline{\sigma}[\cdot]$ denotes the largest singular value of a matrix. The uncertainty can thus be represented by a unit, norm-bounded block Δ cascaded with a scalar transfer function $\delta(s)$.

It should be noted that a successful robust control-system design would depend on, to a certain extent, an appropriate description of the perturbation considered, though theoretically most representations are interchangeable.

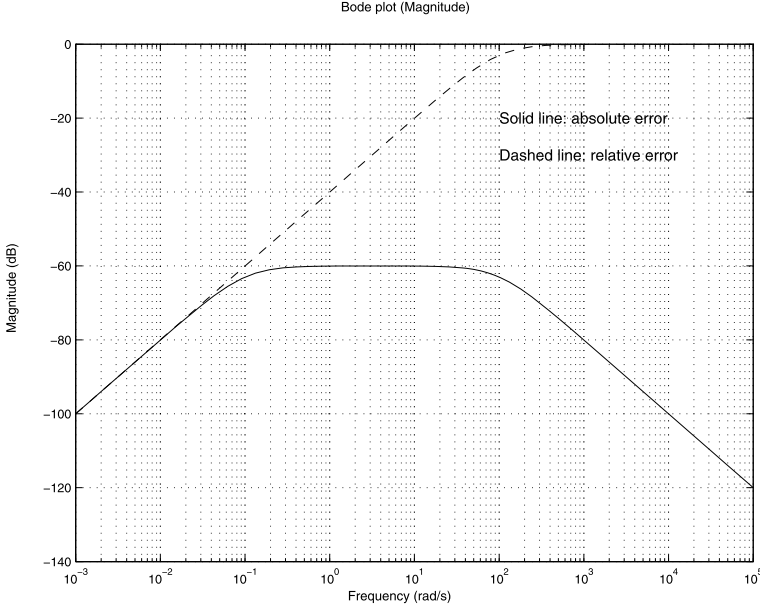


Fig. 2.9 Absolute and relative errors in Example 2.1

Example 2.1 The dynamics of many control systems may include a “slow” part and a “fast” part, for instance in a dc motor. The actual dynamics of a scalar plant may be

$$G_p(s) = g_{\text{gain}} G_{\text{slow}}(s) G_{\text{fast}}(s)$$

where g_{gain} is constant, and

$$G_{\text{slow}}(s) = \frac{1}{1 + sT}; \quad G_{\text{fast}}(s) = \frac{1}{1 + \alpha sT}, \quad \alpha \ll 1$$

In the design, it may be reasonable to concentrate on the slow response part while treating the fast response dynamics as a perturbation. Let Δ_a and Δ_m denote the additive and multiplicative perturbations, respectively. It can easily be worked out that

$$\begin{aligned} \Delta_a(s) &= G_p - g_{\text{gain}} G_{\text{slow}} = g_{\text{gain}} G_{\text{slow}} (G_{\text{fast}} - 1) \\ &= g_{\text{gain}} \frac{-\alpha sT}{(1 + sT)(1 + \alpha sT)} \\ \Delta_m(s) &= \frac{G_p - g_{\text{gain}} G_{\text{slow}}}{g_{\text{gain}} G_{\text{slow}}} = G_{\text{fast}} - 1 = \frac{-\alpha sT}{1 + \alpha sT} \end{aligned}$$

The magnitude Bode plots of Δ_a and Δ_m can be seen in Fig. 2.9, where g_{gain} is assumed to be 1. The difference between the two perturbation representations is obvious: though the magnitude of the absolute error may be small, the relative error can be large in the high-frequency range in comparison to that of the nominal plant.

2.2 Parametric Uncertainty

The unstructured uncertainty representations discussed in Sect. 2.1 are useful in describing unmodeled or neglected system dynamics. These complex uncertainties usually occur in the high-frequency range and may include unmodeled lags (time delay), parasitic coupling, hysteresis, and other nonlinearities. However, dynamic perturbations in many industrial control systems may also be caused by inaccurate description of component characteristics, torn-and-worn effects on plant components, or shifting of operating points, etc. Such perturbations may be represented by variations of certain system parameters over some possible value ranges (complex or real). They affect the low-frequency range performance and are called “parametric uncertainties”.

Example 2.2 A mass–spring–damper system can be described by the following second order, ordinary differential equation:

$$m \frac{d^2x(t)}{dt^2} + c \frac{dx(t)}{dt} + kx(t) = f(t)$$

where m is the mass, c the damping constant, k the spring stiffness, $x(t)$ the displacement and $f(t)$ the external force. For imprecisely known parameter values, the dynamic behavior of such a system is actually described by

$$(m_o + \delta_m) \frac{d^2x(t)}{dt^2} + (c_o + \delta_c) \frac{dx(t)}{dt} + (k_o + \delta_k)x(t) = f(t) \quad (2.9)$$

where m_o , c_o , and k_o denote the nominal parameter values and δ_m , δ_c and δ_k possible variations over certain ranges.

By defining the state variables x_1 and x_2 as the displacement variable and its first order derivative (velocity), the second order differential equation (2.9) may be rewritten into a standard state-space form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{m_o + \delta_m} [-(k_o + \delta_k)x_1 - (c_o + \delta_c)x_2 + f] \\ y &= x_1 \end{aligned}$$

Further, the system can be represented by an analog block diagram as in Fig. 2.10.

Notice that $\frac{1}{m_o + \delta_m}$ can be rearranged as a feedback in terms of $\frac{1}{m_o}$ and δ_m . Figure 2.10 can be redrawn as in Fig. 2.11, by pulling out all the uncertain variations.

Let z_1 , z_2 , and z_3 be \dot{x}_2 , x_2 , and x_1 , respectively, considered as another, fictitious output vector; and, d_1 , d_2 , and d_3 be the signals coming out from the perturbation blocks δ_m , δ_c , and δ_k , as shown in Fig. 2.11. The perturbed system can be arranged in the following state-space model and represented as in Fig. 2.12:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_o}{m_o} & -\frac{c_o}{m_o} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m_o} \end{bmatrix} f$$

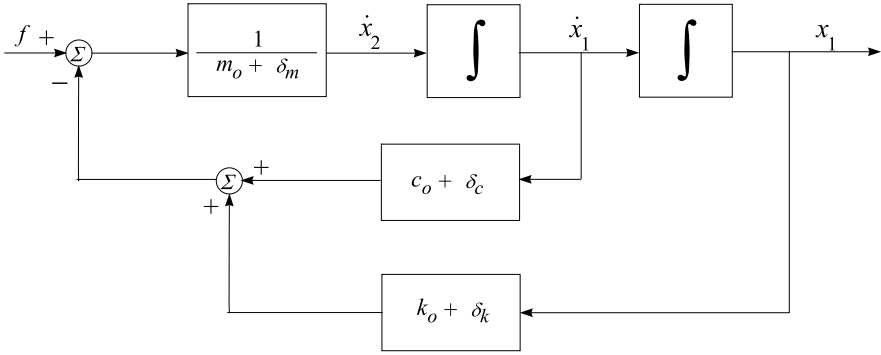


Fig. 2.10 Analog block diagram of Example 2.2

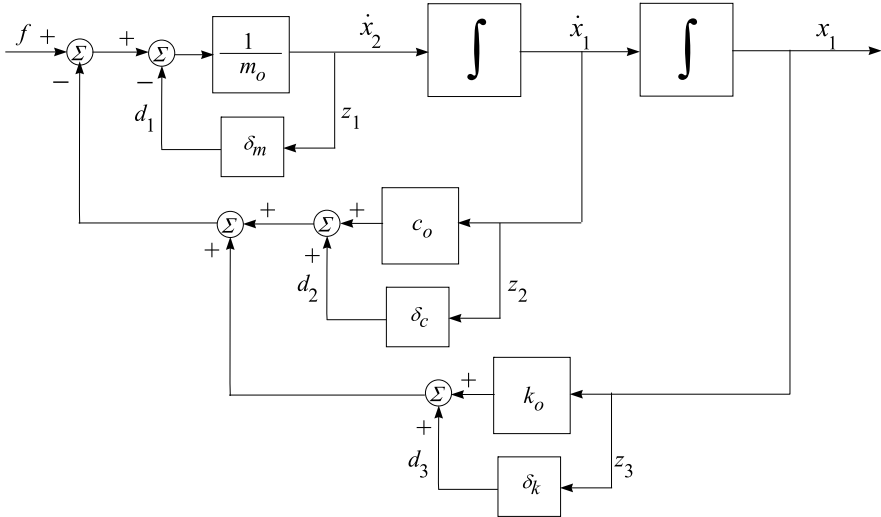


Fig. 2.11 Structured uncertainties block diagram of Example 2.2

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} -\frac{k_o}{m_o} & -\frac{c_o}{m_o} \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{m_o} \\ 0 \\ 0 \end{bmatrix} f \quad (2.10)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The state-space model of (2.10) describes the augmented, interconnection system M of Fig. 2.12. The perturbation block Δ in Fig. 2.12 corresponds to parameter variations and is called “parametric uncertainty”. The uncertain block Δ is not a full matrix but a diagonal one. It has certain structure, hence the terminology of “structured uncertainty”. More general cases will be discussed shortly in Sect. 2.4.

Fig. 2.12 Standard configuration of Example 2.2

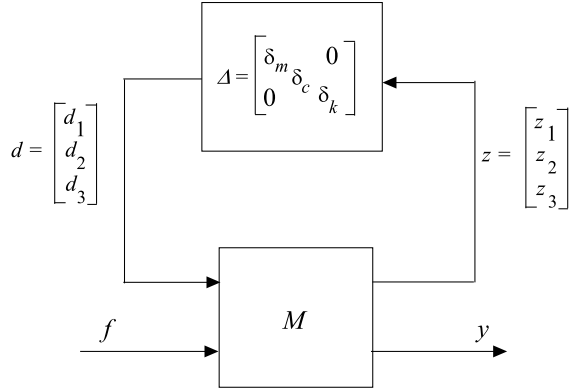
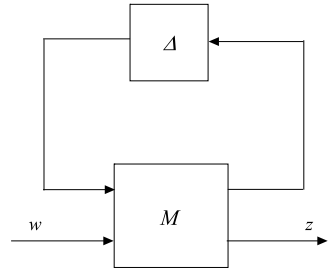


Fig. 2.13 Standard M - Δ configuration



2.3 Linear Fractional Transformations

The block diagram in Fig. 2.12 can be generalized to be a standard configuration to represent how the uncertainty affects the input/output relationship of the control system under study. This kind of representation first appeared in the circuit analysis back in the 1950s [140, 141]. It was later adopted in the robust control study [145] for uncertainty modeling. The general framework is depicted in Fig. 2.13.

The interconnection transfer function matrix M in Fig. 2.13 is partitioned as

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

where the dimensions of M_{11} conform with those of Δ . By routine manipulations, it can be derived that

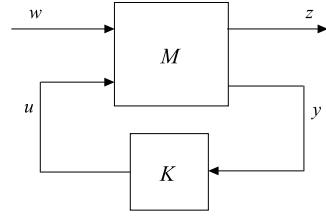
$$z = [M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12}]$$

if $(I - M_{11}\Delta)$ is invertible. When the inverse exists, we may define

$$F(M, \Delta) = M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12}$$

$F(M, \Delta)$ is called a *linear fractional transformation* (LFT) of M and Δ . Because the “upper” loop of M is closed by the block Δ , this kind of linear fractional transformation is also called an *upper linear fractional transformation* (ULFT), and denoted with a subscript u , i.e. $F_u(M, \Delta)$, to show the way of connection. Similarly,

Fig. 2.14 Lower LFT configuration



there are also *lower linear fractional transformations* (LLFT) that are usually used to indicate the incorporation of a controller K into a system. Such a lower LFT can be depicted as in Fig. 2.14 and defined by

$$F_l(M, K) = M_{11} + M_{12}K(I - M_{22}K)^{-1}M_{21}$$

With the introduction of linear fractional transformations, the unstructured uncertainty representations discussed in Sect. 2.1 may be uniformly described by Fig. 2.13, with appropriately defined interconnection matrices M s as listed below.

1. Additive perturbation:

$$M = \begin{bmatrix} 0 & I \\ I & G_o \end{bmatrix} \quad (2.11)$$

2. Inverse additive perturbation:

$$M = \begin{bmatrix} -G_o & G_o \\ -G_o & G_o \end{bmatrix} \quad (2.12)$$

3. Input multiplicative perturbation:

$$M = \begin{bmatrix} 0 & I \\ G_o & G_o \end{bmatrix} \quad (2.13)$$

4. Output multiplicative perturbation:

$$M = \begin{bmatrix} 0 & G_o \\ I & G_o \end{bmatrix} \quad (2.14)$$

5. Inverse input multiplicative perturbation:

$$M = \begin{bmatrix} -I & I \\ -G_o & G_o \end{bmatrix} \quad (2.15)$$

6. Inverse output multiplicative perturbation:

$$M = \begin{bmatrix} -I & G_o \\ -I & G_o \end{bmatrix} \quad (2.16)$$

7. Left coprime factor perturbations:

$$M = \begin{bmatrix} \begin{bmatrix} -\tilde{M}_G^{-1} \\ 0 \end{bmatrix} & \begin{bmatrix} -G_o \\ I \end{bmatrix} \\ \tilde{M}_G^{-1} & G_o \end{bmatrix} \quad (2.17)$$

where $G_o = \tilde{M}_G^{-1} \tilde{N}_G$, a left coprime factorization of the nominal plant; and, the perturbed plant is $G_p = (\tilde{M}_G + \Delta_{\tilde{M}})^{-1} (\tilde{N}_G + \Delta_{\tilde{N}})$.

8. Right coprime factor perturbations:

$$M = \begin{bmatrix} \begin{bmatrix} -M_G^{-1} & 0 \\ -G_o & I \end{bmatrix} & M_G^{-1} \\ & G_o \end{bmatrix} \quad (2.18)$$

where $G_o = N_G M_G^{-1}$, a right coprime factorization of the nominal plant; and, the perturbed plant is $G_p = (N_G + \Delta_N)(M_G + \Delta_M)^{-1}$.

In the above, it is assumed that $[I - M_{11}\Delta]$ is invertible. The perturbed system is thus

$$G_p(s) = F_u(M, \Delta)$$

In the coprime factor perturbation representations, (2.17) and (2.18), $\Delta = [\Delta_{\tilde{M}} \ \Delta_{\tilde{N}}]$ and $\Delta = \begin{bmatrix} \Delta_M \\ \Delta_N \end{bmatrix}$, respectively. The block Δ in (2.11)–(2.18) is supposed to be a “full” matrix, i.e. it has no specific *structure*.

2.4 Structured Uncertainties

In many robust design problems, it is more likely that the uncertainty scenario is a mixed case of those described in Sects. 2.1 and 2.2. The uncertainties under consideration would include unstructured uncertainties, such as unmodeled dynamics, as well as parameter variations. All these uncertain parts still can be taken out from the dynamics and the whole system can be rearranged in a standard configuration of (upper) linear fractional transformation $F(M, \Delta)$. The uncertain block Δ would then have the following general form:

$$\Delta = \text{diag}[\delta_1 I_{r_1}, \dots, \delta_s I_{r_s}, \Delta_1, \dots, \Delta_f], \quad \delta_i \in \mathcal{C}, \Delta_j \in \mathcal{C}^{m_j \times m_j} \quad (2.19)$$

where $\sum_{i=1}^s r_i + \sum_{j=1}^f m_j = n$ with n is the dimension of the block Δ . We may define the set of such Δ as $\mathbf{\Delta}$. The total block Δ thus has two types of uncertain block: s repeated *scalar* blocks and f *full* blocks. The parameters δ_i of the repeated scalar blocks can be real numbers only, if further information of the uncertainties is available. However, in the case of real numbers, the analysis and design would be even harder. The full blocks in (2.19) need not be square, but by restricting them as such makes the notation much simpler.

When a perturbed system is described by an LFT with the uncertain block of (2.19), the Δ considered has a certain structure. It is thus called “structured uncertainty”. Apparently, using a lumped, full block to model the uncertainty in such cases, for instance in Example 2.2, would lead to pessimistic analysis of the system behavior and produce conservative designs.

This page intentionally left blank

Chapter 3

Robust Design Specifications

A control system is *robust* if it remains stable and obeys certain performance criteria in the presence of possible uncertainties as discussed in Chap. 2. The robust design is to find a controller, for a given system, such that the closed-loop system is robust. The \mathcal{H}_∞ optimization approach and its related approaches, being developed in the last two decades and still an active research area, have been shown to be effective and efficient robust design methods for linear, time-invariant control systems. We will first introduce in this chapter the *Small-Gain Theorem*, which plays an important role in the \mathcal{H}_∞ optimization methods, and then discuss the stabilization and performance requirements in robust designs using the \mathcal{H}_∞ optimization and related ideas.

3.1 Small-Gain Theorem and Robust Stabilization

The Small-Gain Theorem is of central importance in the derivation of many stability tests. In general, it provides only a sufficient condition for stability and is therefore potentially conservative. The Small-Gain Theorem is applicable to general operators. What will be included here is, however, a version that is suitable for the \mathcal{H}_∞ optimization designs, and in this case, it is a sufficient *and* necessary result.

Consider the feedback configuration in Fig. 3.1, where $G_1(s)$ and $G_2(s)$ are the transfer function matrices of corresponding linear, time-invariant systems. We then have the following theorem.

Theorem 3.1 [28] *If $G_1(s)$ and $G_2(s)$ are stable, i.e. $G_1 \in \mathcal{H}_\infty$, $G_2 \in \mathcal{H}_\infty$, then the closed-loop system is internally stable if and only if*

$$\|G_1 G_2\|_\infty < 1 \quad \text{and} \quad \|G_2 G_1\|_\infty < 1$$

A closed-loop system of the plant G and controller K is robustly stable if it remains stable for all possible, under certain definition, perturbations on the plant. This implies, of course, that K is a stabilizing controller for the nominal plant G ,

Fig. 3.1 A feedback configuration

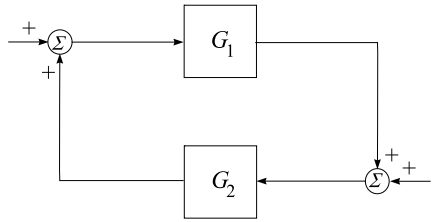
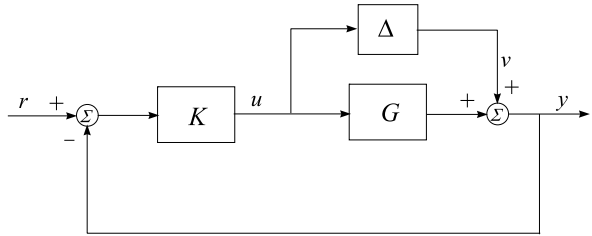


Fig. 3.2 Additive perturbation configuration



since we always assume that the perturbation set includes zero (no perturbation). Let us consider the case of additive perturbation as depicted in Fig. 3.2, where $\Delta(s)$ is the perturbation, a “full” matrix unknown but stable.

It is easy to work out that the transfer function from the signal v to u is $T_{uv} = -K(I + GK)^{-1}$. As mentioned earlier, the controller K should stabilize the nominal plant G . Hence, from the Small-Gain Theorem, we have the following theorem.

Theorem 3.2 [17, 121] *For stable $\Delta(s)$, the closed-loop system is robustly stable if $K(s)$ stabilizes the nominal plant and the following holds:*

$$\|\Delta K(I + GK)^{-1}\|_{\infty} < 1$$

and

$$\|K(I + GK)^{-1}\Delta\|_{\infty} < 1$$

or, in a strengthened form,

$$\|K(I + GK)^{-1}\|_{\infty} < \frac{1}{\|\Delta\|_{\infty}} \quad (3.1)$$

The second condition becomes necessary, when the unknown Δ may have all phases.

If required to find a controller to robustly stabilize the largest possible set of perturbations, in the sense of ∞ -norm, it is then clear that we need to solve the following minimization problem:

$$\min_{K \text{ stabilizing}} \|K(I + GK)^{-1}\|_{\infty} \quad (3.2)$$

In many cases, we may have a priori knowledge of the perturbation, say,

$$\bar{\sigma}(\Delta(j\omega)) \leq \bar{\sigma}(W_2(j\omega)) \quad \text{for all } \omega \in \mathcal{R}$$

Then, we may rewrite the perturbation block as

$$\Delta(s) = \tilde{\Delta}(s)W_2(s)$$

where $\tilde{\Delta}(s)$ is the unit norm perturbation set. Correspondingly, the robust stabilization condition becomes

$$\|W_2K(I + GK)^{-1}\|_\infty < 1$$

and the optimization problem

$$\min_{K \text{ stabilizing}} \|W_2K(I + GK)^{-1}\|_\infty \quad (3.3)$$

Robust stabilization conditions can be derived similarly for other perturbation representations discussed in Chap. 2 and are listed below (G_o is replaced by G for the sake of simplicity).

1. Inverse additive perturbation:

$$\|G(I + KG)^{-1}\|_\infty < \frac{1}{\|\Delta\|_\infty} \quad (3.4)$$

2. Input multiplicative perturbation:

$$\|KG(I + KG)^{-1}\|_\infty < \frac{1}{\|\Delta\|_\infty} \quad (3.5)$$

3. Output multiplicative perturbation:

$$\|GK(I + GK)^{-1}\|_\infty < \frac{1}{\|\Delta\|_\infty} \quad (3.6)$$

4. Inverse input multiplicative perturbation:

$$\|(I + KG)^{-1}\|_\infty < \frac{1}{\|\Delta\|_\infty} \quad (3.7)$$

5. Inverse output multiplicative perturbation:

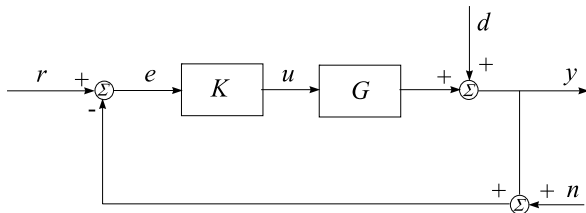
$$\|(I + GK)^{-1}\|_\infty < \frac{1}{\|\Delta\|_\infty} \quad (3.8)$$

The cases of perturbation on coprime factors will be discussed in Chap. 5.

Remark In the above discussion, the stability of the perturbation block has been assumed. Actually, the conclusions are also true if the perturbed systems have the same number of closed right-half plane poles as the nominal system does (see [107]). If even this is not satisfied, then we will have to use the coprime factor perturbation models, as will be discussed in Chap. 5.

Robust stabilization is an important issue not just as a design requirement. As a matter of fact, the \mathcal{H}_∞ design and related approaches first formulate the stability as well as performance design specifications as a robust stabilization problem and then solve the robust stabilization problem to find a controller.

Fig. 3.3 A closed-loop configuration of G and K



3.2 Performance Considerations

Figure 3.3 depicts a typical closed-loop system configuration, where G is the plant and K the controller to be designed. r , y , u , e , d , n are, respectively, the reference input, output, control signal, error signal, disturbance, and measurement noise. With a little abuse of notations, we do not distinguish the notations of signals in the time or frequency domains. The following relationships are immediately available:

$$\begin{aligned} y &= (I + GK)^{-1}GKr + (I + GK)^{-1}d - (I + GK)^{-1}GKn \\ u &= K(I + GK)^{-1}r - K(I + GK)^{-1}d - K(I + GK)^{-1}n \\ e &= (I + GK)^{-1}r - (I + GK)^{-1}d - (I + GK)^{-1}n \end{aligned}$$

Assume that the signals r , d , n are energy bounded and have been normalized, i.e. lying in the unit ball of \mathcal{L}_2 space. We, however, do not know what exactly these signals are. It is required that the usual performance specifications, such as tracking, disturbance attenuation and noise rejection, should be as good as possible for any r , d or n whose energy does not exceed 1. From the discussions in Chap. 1 on signal and system norms, it is clear that we should minimize the ∞ -norm, the gain, of corresponding transfer function matrices. Hence, the design problem is that over the set of all stabilizing controller K s, (i.e. those K s make the closed-loop system internally stable), find the optimal one that minimizes

- for good tracking,
 $\|(I + GK)^{-1}\|_\infty$
- for good disturbance attenuation,
 $\|(I + GK)^{-1}\|_\infty$
- for good noise rejection,
 $\|-(I + GK)^{-1}GK\|_\infty$
- for less control energy,
 $\|K(I + GK)^{-1}\|_\infty$

It is conventional to denote $\mathcal{S} := (I + GK)^{-1}$, the *sensitivity* function, and $\mathcal{T} := (I + GK)^{-1}GK$, the *complementary sensitivity* function.

In general, weighting functions would be used in the above minimization to meet the design specifications. For instance, instead of minimizing the sensitivity function alone, we would aim at solving

$$\min_{K \text{ stabilizing}} \|W_1 \mathcal{S} W_d\|_\infty$$

where W_1 is chosen to tailor the tracking requirement and is usually a high-gain low-pass filter type, W_d can be regarded as a generator that characterizes all relevant disturbances in the case considered. Usually, the weighting functions are stable and of minimum phase.

3.3 Structured Singular Values

Systems with uncertain dynamics can all be put in the standard $M-\Delta$ configuration of Fig. 2.13. The robust stabilization conditions derived in Sect. 3.1 are sufficient and necessary conditions for unstructured uncertainties, i.e. Δ is a full block and will have all phases. In the case of structured uncertainty (Sect. 2.4), these robust stabilization results could be very conservative. To deal with structured uncertainties, we need to introduce the so-called *structured singular values* (SSV).

In fact, these robust stabilization results, such as that in Theorem 3.2, can be equivalently written as [17, 121]

$$\det[I - M(j\omega)\Delta(j\omega)] \neq 0, \quad \forall \omega \in \mathcal{R}, \forall \Delta \quad (3.9)$$

where the nominal (closed-loop) system $M(s)$ is assumed to be stable as usual.

This condition for robust stability is sufficient and necessary even for structured uncertainty Δ . Roughly speaking, in order to have the closed-loop system robustly stable, all the uncertainties of a known structure of (2.19) should be small enough not to violate the condition (i.e. not make $I - M(j\omega)\Delta(j\omega)$ singular at any frequency ω). On the other hand, for a given M with a fixed controller K and a known structure of the uncertainties, the smallest “size” of the uncertainty that makes $I - M(j\omega)\Delta(j\omega)$ singular at some frequency ω describes how robustly stable the controller K is in dealing with such structured uncertainties. This measurement is the so-called structured singular values (SSV) introduced below.

First, as in Sect. 2.4, we define the structure of uncertainty of (2.19) and repeat here for convenience,

$$\Delta = \{\text{diag}[\delta_1 I_{r_1}, \dots, \delta_s I_{r_s}, \Delta_1, \dots, \Delta_f] : \delta_i \in \mathcal{C}, \Delta_j \in \mathcal{C}^{m_j \times m_j}\} \quad (3.10)$$

where $\sum_{i=1}^s r_i + \sum_{j=1}^f m_j = n$ with n is the dimension of the block Δ . We also assume the set of Δ is bounded. And, we may thus define a normalized set of structured uncertainty by

$$\mathbf{\Delta} := \{\Delta : \bar{\sigma}(\Delta) < 1, \Delta \in \Delta\} \quad (3.11)$$

Definition 3.3 For $M \in \mathcal{C}^{n \times n}$, the structured singular value $\mu_{\Delta}(M)$ of M with respect to Δ is the number defined such that $\mu_{\Delta}^{-1}(M)$ is equal to the smallest $\bar{\sigma}(\Delta)$ needed to make $(I - M\Delta)$ singular (rank deficiency). That is

$$\mu_{\Delta}^{-1}(M) := \min_{\Delta \in \Delta} \{\bar{\sigma}(\Delta) : \det(I - M\Delta) = 0\} \quad (3.12)$$

If there is no $\Delta \in \Delta$ such that $\det(I - M\Delta) = 0$, then $\mu_{\Delta}(M) := 0$.

When M is an interconnected transfer matrix as in Fig. 2.13, the structured singular value, with respect to Δ , is defined by

$$\mu_{\Delta}(M(s)) := \sup_{\omega \in \mathcal{R}} \mu_{\Delta}(M(j\omega)) \quad (3.13)$$

Correspondingly, the uncertainty set may be defined as

$$\mathcal{M}(\Delta) := \{\Delta(\cdot) \in \mathcal{RH}_{\infty} : \Delta(j\omega) \in \Delta \text{ for all } \omega \in \mathcal{R}\} \quad (3.14)$$

When the uncertainty structure is fixed, we may omit the subscript Δ of $\mu_{\Delta}(M)$ for brevity.

The reciprocal of the structured singular value denotes a frequency-dependent stability margin [24, 145]. The robust stability result with regard to structured uncertainty is now given in the following theorem.

Theorem 3.4 [31, 161] *Let the nominal feedback system $(M(s))$ be stable and let $\beta > 0$ be an uncertainty bound, i.e. $\|\Delta\|_{\infty} < \beta, \forall \Delta(\cdot) \in \mathcal{M}(\Delta)$. The perturbed system of Fig. 2.13 is robustly stable, with respect to Δ , if and only if $\mu_{\Delta}(M(s)) \leq \frac{1}{\beta}$.*

It is obvious that if the uncertainty lies in the unit ball $\mathbf{B}\Delta$, the robust stability condition is then $\mu_{\Delta}(M(s)) \leq 1$.

$\mu_{\Delta}(M(s))$ is frequency dependent and is calculated at “each” frequency over a reasonable range in practical applications.

In the literature, $\mu_{\Delta}(M(s))$ is sometimes redefined as $\|M\|_{\mu}$ for an interconnected transfer function matrix $M(s)$. This notation is convenient; however, it should be clear that it is not a norm. It does not satisfy the three basic properties of a norm. Also, it depends on $M(s)$ as well as the uncertainty structure of Δ .

The structured singular value plays an important role in robust design. As became clear in the early parts of this chapter, the \mathcal{H}_{∞} optimization approach can deal with robust stabilization problems with regard to unstructured uncertainties and can achieve nominal performance requirements. In the case of structured uncertainty, Theorem 3.4 gives a sufficient and necessary condition for robust stabilization. Furthermore, the robust performance design can also be transformed into a robust stabilization problem with regard to structured uncertainties, as will be described in Chap. 6. However, the computation of the structured singular value is not an easy task. It is still an open topic and requires further research.

A few properties of the structured singular value and its computation are listed below. Interested readers are referred to [31, 41, 42, 132–134, 196] for details. Coded routines for the μ computation are available in software packages MATLAB[®] [11] and Slicot [78].

Let $\rho(\cdot)$ denote the spectral radius of a square matrix. By direct manipulations, we have the following lemma.

Lemma 3.5 *For a square, constant matrix M ,*

$$\mu(M) = \max_{\Delta \in \mathbf{B}\Delta} \rho(M\Delta)$$

The following properties of μ can also easily be derived:

- $\mu(\alpha M) = |\alpha| \cdot \mu(M), \forall \alpha \in \mathcal{C}$
- $\det(I - M\Delta) \neq 0, \forall \Delta \in \mathbf{B}\Delta \iff \mu(M) \leq 1$
- if $\Delta = \{\delta I_n : \delta \in \mathcal{C}\}$ ($s = 1, f = 0; r_1 = n$) $\implies \mu(M) = \rho(M)$
- if $\Delta = \mathcal{C}^{n \times n}$ ($s = 0, f = 1; m_1 = n$) $\implies \mu(M) = \overline{\sigma}(M)$

In the above, s , f , and n are dimensions of the uncertainty set Δ as defined in (3.10).

Further, we have the following lemma.

Lemma 3.6

$$\rho(M) \leq \mu(M) \leq \overline{\sigma}(M)$$

Lemma 3.6 gives upper and lower bounds for $\mu(M)$. They are, however, not particularly useful for the computation of $\mu(M)$, because the gap between $\rho(M)$ and $\overline{\sigma}(M)$ could be arbitrarily large. More accurate bounds are needed and these can be obtained by using some transformations on M that may change the values of $\rho(M)$ and $\overline{\sigma}(M)$ but not change $\mu(M)$. The following two constant matrix sets are introduced for this purpose.

Define

$$\mathbf{U} = \{U \in \Delta : UU^* = I_n\}$$

and

$$\begin{aligned} \mathbf{D} = \{D = \text{diag}[D_1, \dots, D_s, d_1 I_{m_1}, \dots, d_f I_{m_f}] : \\ D_i \in \mathcal{C}^{r_i \times r_i}, D_i = D_i^* > 0, d_j > 0\} \end{aligned}$$

The matrix sets \mathbf{U} and \mathbf{D} match the structure of Δ . \mathbf{U} is of a (block-) diagonal structure of unitary matrices and for any $D \in \mathbf{D}$ and $\Delta \in \Delta$, D (D^{-1}) commutes with Δ . Furthermore, for any $\Delta \in \Delta$, $U \in \mathbf{U}$, and $D \in \mathbf{D}$, we have

- $U^* \in \mathbf{U}$, $U\Delta \in \Delta$, $\Delta U \in \Delta$, and $\overline{\sigma}(U\Delta) = \overline{\sigma}(\Delta U) = \overline{\sigma}(\Delta)$
- $D\Delta D^{-1} = \Delta$, $D\Delta D^{-1} \in \Delta$, and $\overline{\sigma}(D\Delta D^{-1}) = \overline{\sigma}(\Delta)$

More importantly, we have

$$\rho(MU) \leq \mu(MU) = \mu(M) = \mu(DMD^{-1}) \leq \overline{\sigma}(DMD^{-1}) \quad (3.15)$$

In the above, $\mu(MU) = \mu(M)$ is derived from $\det(I - M\Delta) = \det(I - MUU^*\Delta)$ and $U^*\Delta \in \Delta$, $\overline{\sigma}(U^*\Delta) = \overline{\sigma}(\Delta)$. Also, $\mu(M) = \mu(DMD^{-1})$ can be seen from $\det(I - DMD^{-1}\Delta) = \det(I - DM\Delta D^{-1}) = \det(I - M\Delta)$.

The relations in (3.15) directly lead to the following theorem.

Theorem 3.7

$$\max_{U \in \mathbf{U}} \rho(MU) \leq \mu(M) \leq \inf_{D \in \mathbf{D}} \overline{\sigma}(DMD^{-1})$$

Theorem 3.7 provides tighter upper and lower bounds on $\mu(M)$. In [31], it was shown that the above lower bound is actually an equality,

$$\max_{U \in \mathbf{U}} \rho(MU) = \mu(M)$$

Unfortunately, this optimization problem is not convex. $\rho(MU)$ may have multiple local maxima. Direct computation of $\max_{U \in \mathbf{U}} \rho(MU)$ may not find a global maximum. On the other hand, the upper bound of $\mu(M)$ in Theorem 3.7 is easier to find, since $\bar{\sigma}(DM D^{-1})$ is convex in $\ln D$ [33, 153]. However, this upper bound is not always equal to $\mu(M)$. For the cases of $2s + f \leq 3$, it can be shown that

$$\mu(M) = \inf_{D \in \mathbf{D}} \bar{\sigma}(DM D^{-1})$$

The problem of calculating $\mu(M)$ is therefore reduced to an optimal diagonal scaling problem. Most algorithms proposed so far for the structured singular values compute this upper bound.

Chapter 4

\mathcal{H}_∞ Design

A control system is *robust* if it remains stable and obeys certain performance criteria in the presence of possible uncertainties as discussed in Chap. 2. The robust design is to find a controller, for a given system, such that the closed-loop system is robust. The \mathcal{H}_∞ optimization approach, having been developed in the last two decades and still an active research area, has been shown to be an effective and efficient robust design method for linear, time-invariant control systems. In the previous chapter, various robust stability considerations and nominal performance requirements were formulated as a minimization problem of the infinitive norm of a closed-loop transfer function matrix. Hence, in this chapter, we shall discuss how to formulate a robust design problem into such a minimization problem and how to find the solutions. The \mathcal{H}_∞ optimization approach solves, in general, the robust stabilization problems and nominal performance designs.

4.1 Mixed Sensitivity \mathcal{H}_∞ Optimization

Every practicing control engineer knows very well that it will never be appropriate in any industrial design to use just a single cost function, such as those formulated in Chap. 3. A reasonable design would use a combination of these functions. For instance, it makes sense to require a good tracking as well as to limit the control signal energy, as depicted in Fig. 4.1. We may then want to solve the following *mixed sensitivity* (or so-called *S over KS*) problem:

$$\min_{K \text{ stabilizing}} \left\| \begin{array}{c} (I + GK)^{-1} \\ K(I + GK)^{-1} \end{array} \right\|_\infty \quad (4.1)$$

This cost function can also be interpreted as the design objectives of nominal performance, good tracking or disturbance attenuation, and robust stabilization, with regard to additive perturbation.

In order to adopt a unified solution procedure, the above cost function (4.1) can be recast into a standard configuration as in Fig. 4.2. This can be obtained by using the

Fig. 4.1 A mixed sensitivity consideration

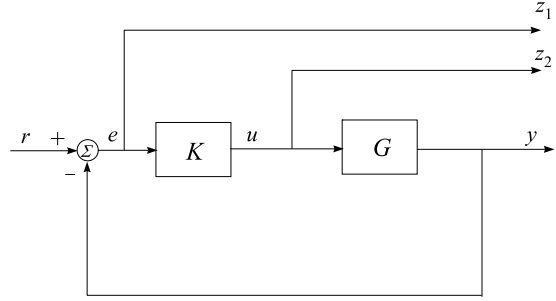
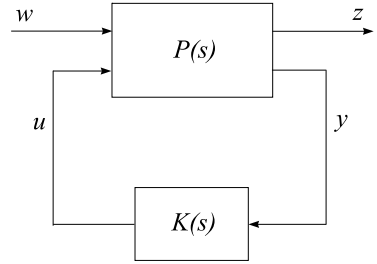


Fig. 4.2 The standard \mathcal{H}_∞ configuration



LFT technique introduced in Chap. 2 and by specifying/grouping signals into sets of external inputs, outputs, input to the controller and output from the controller, which of course is the control signal. Note that in Fig. 4.2 all the external inputs are denoted by w ; z denotes the output signals to be minimized/penalized that includes both performance and robustness measures, y is the vector of measurements available to the controller K , and u the vector of control signals. $P(s)$ is called the *generalized plant* or *interconnected system*. The objective is to find a stabilizing controller K to minimize the output z , in the sense of energy, over all w with energy less than or equal to 1. Thus, it is equivalent to minimizing the \mathcal{H}_∞ -norm of the transfer function from w to z .

Partitioning the interconnected system P as

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix}$$

we directly find

$$\begin{aligned} z &= [P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}]w \\ &=: \mathcal{F}_l(P, K)w \end{aligned}$$

where $\mathcal{F}_l(P, K)$ is the *lower linear fractional transformation* of P and K . The design objective now becomes

$$\min_{K \text{ stabilizing}} \|\mathcal{F}_l(P, K)\|_\infty \quad (4.2)$$

and is referred to as the \mathcal{H}_∞ optimization problem.

Referring to the problem in (4.1), it is easy to derive its standard form by defining $w = r$, $z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} e \\ u \end{bmatrix}$, $y = e$ and $u = u$. Consequently, the interconnected system is

$$P = \begin{bmatrix} I & -G \\ 0 & I \\ I & -G \end{bmatrix} \quad (4.3)$$

where we may set

$$\begin{aligned} P_{11} &= \begin{bmatrix} I \\ 0 \end{bmatrix}, & P_{12} &= \begin{bmatrix} -G \\ I \end{bmatrix} \\ P_{21} &= I, & P_{22} &= -G \end{aligned}$$

Other mixed cases of cost transfer function matrices such as S over T , S over T over KS , etc., can be dealt with similarly to formulate into the standard configuration. In practical designs, it is often necessary to include (closed-loop) weights with these cost functions. For instance, instead of (4.1), we may have to consider with $z_1 = W_1 e$ and $z_2 = W_2 u$,

$$\min_{K \text{ stabilizing}} \left\| \begin{bmatrix} W_1(I + GK)^{-1} \\ W_2K(I + GK)^{-1} \end{bmatrix} \right\|_\infty \quad (4.4)$$

These weighting functions can be easily absorbed into the interconnected system $P(s)$, as in this case,

$$P = \begin{bmatrix} W_1 & -W_1G \\ 0 & W_2 \\ I & -G \end{bmatrix} \quad (4.5)$$

4.2 2-Degree-of-Freedom \mathcal{H}_∞ Design

Among control-system design specifications, reference-signal tracking is often a requirement. The output of a designed control system is required to follow a pre-selected signal, or in a more general sense, the system is forced to track, for instance, the step response of a specified model (infinite-time model following task). A *2-degree-of-freedom* (2DOF) control scheme naturally suits this situation. The idea of a 2DOF scheme is to use a feedback controller (K_2) to achieve the internal and robust stability, disturbance rejection, etc., and to design another controller (K_1) on the feedforward path to meet the tracking requirement, which minimizes the difference between the output of the overall system and that of the reference model. 2DOF control schemes were successfully applied in some practical designs [75, 98, 125].

Figure 4.3 shows one structure of these 2DOF control schemes. In this configuration, in addition to the internal stability requirement, two signals e and u are to be minimized. The signal e shows the difference between the system output and the reference model output. u is the control signal and also is related to robust stability

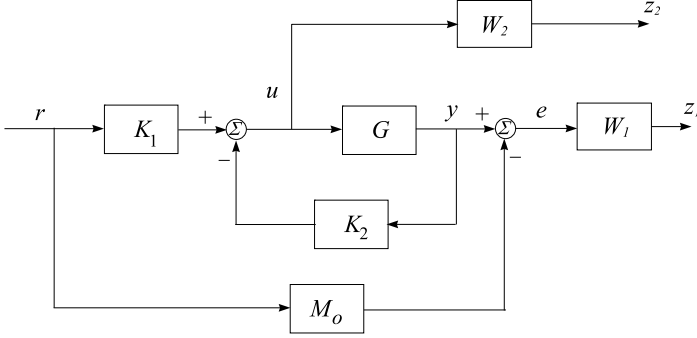


Fig. 4.3 A 2DOF design configuration

in the additive perturbation case. In Fig. 4.3, two weighting functions are included to reflect the trade-off between and/or characteristics of these two penalized signals.

The configuration of Fig. 4.3 can be rearranged as the standard configuration of Fig. 4.2 by defining $w = r$, $z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} W_1 e \\ W_2 u \end{bmatrix}$, $y = \begin{bmatrix} r \\ y \end{bmatrix}$ and $u = u$. Note that in this configuration, $K = [K_1 - K_2]$. Consequently, the interconnected system is

$$P = \begin{bmatrix} -W_1 M_o & W_1 G \\ 0 & W_2 \\ I & 0 \\ 0 & G \end{bmatrix}$$

where we may set

$$\begin{aligned} P_{11} &= \begin{bmatrix} -W_1 M_o \\ 0 \end{bmatrix}, & P_{12} &= \begin{bmatrix} W_1 G \\ W_2 \end{bmatrix} \\ P_{21} &= \begin{bmatrix} I \\ 0 \end{bmatrix}, & P_{22} &= \begin{bmatrix} 0 \\ G \end{bmatrix} \end{aligned}$$

4.3 \mathcal{H}_∞ Suboptimal Solutions

The solution to the optimization problem (4.2) is not unique except in the scalar case [64, 196]. Generally speaking, there are no analytic formulas for the solutions. In practical design, it is usually sufficient to find a stabilizing controller K such that the \mathcal{H}_∞ -norm of the closed-loop transfer function is less than a given positive number, i.e.,

$$\|\mathcal{F}_l(P, K)\|_\infty < \gamma \quad (4.6)$$

where $\gamma > \gamma_o := \min_{K \text{ stabilizing}} \|\mathcal{F}_l(P, K)\|_\infty$. This is called the \mathcal{H}_∞ *suboptimal problem*. When certain conditions are satisfied, there are formulas to construct a set of controllers that solve the problem (4.6). The solution set is characterized by a free parameter $Q(s)$, which is stable and of ∞ -norm less than γ .

It is imaginable that if we successively reduce the value of γ , starting from a relatively large number to ensure the existence of a suboptimal solution, we may obtain an optimal solution. It should, however, be pointed out here that when γ is approaching its minimum value γ_o the problem would become more and more ill-conditioned numerically. Hence, the “solution” thus obtained might be very unreliable.

4.3.1 Solution Formulas for Normalized Systems

Let the state-space description of the generalized (interconnected) system P in Fig. 4.2 be given by

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B_1w(t) + B_2u(t) \\ z(t) &= C_1x(t) + D_{11}w(t) + D_{12}u(t) \\ y(t) &= C_2x(t) + D_{21}w(t)\end{aligned}$$

where $x(t) \in R^n$ is the state vector, $w(t) \in R^{m_1}$ the exogenous input vector, $u(t) \in R^{m_2}$ the control input vector, $z(t) \in R^{p_1}$ the error (output) vector, and $y(t) \in R^{p_2}$ the measurement vector, with $p_1 \geq m_2$ and $p_2 \leq m_1$. $P(s)$ may be further denoted as

$$\begin{aligned}P(s) &= \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \\ &= \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{array} \right] \\ &=: \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]\end{aligned}\tag{4.7}$$

Note that in the above definition it is assumed that there is no direct link between the control input and the measurement output, i.e. $D_{22} = 0$. This assumption is reasonable because most industrial control systems are strictly proper and the corresponding $P(s)$ would have a zero D_{22} in a sensible design configuration. The case of a nonzero direct term between $u(t)$ and $y(t)$ will be, however, considered in the next subsection for the sake of completeness.

The \mathcal{H}_∞ solution formulas use solutions of two *algebraic Riccati equations* (ARE). An algebraic Riccati equation

$$E^T X + X E - X W X + Q = 0$$

where $W = W^T$ and $Q = Q^T$ uniquely corresponds to a Hamiltonian matrix

$$\begin{bmatrix} E & -W \\ -Q & -E^T \end{bmatrix}$$

The stabilizing solution X , if it exists, is a symmetric matrix that solves the ARE and is such that $E - WX$ is a stable matrix. The stabilizing solution is denoted as

$$X := \mathbf{Ric} \begin{bmatrix} E & -W \\ -Q & -E^T \end{bmatrix}$$

Define

$$R_n := D_{1*}^T D_{1*} - \begin{bmatrix} \gamma^2 I_{m_1} & 0 \\ 0 & 0 \end{bmatrix}$$

and

$$\tilde{R}_n := D_{*1} D_{*1}^T - \begin{bmatrix} \gamma^2 I_{p_1} & 0 \\ 0 & 0 \end{bmatrix}$$

where

$$D_{1*} = \begin{bmatrix} D_{11} & D_{12} \end{bmatrix} \quad \text{and} \quad D_{*1} = \begin{bmatrix} D_{11} \\ D_{21} \end{bmatrix}$$

Assume that R_n and \tilde{R}_n are nonsingular. We define two Hamiltonian matrices \mathbf{H} and \mathbf{J} as

$$\begin{aligned} \mathbf{H} &:= \begin{bmatrix} A & 0 \\ -C_1^T C_1 & -A^T \end{bmatrix} - \begin{bmatrix} B \\ -C_1^T D_{1*} \end{bmatrix} R_n^{-1} \begin{bmatrix} D_{1*}^T C_1 & B^T \end{bmatrix} \\ \mathbf{J} &:= \begin{bmatrix} A^T & 0 \\ -B_1 B_1^T & -A \end{bmatrix} - \begin{bmatrix} C^T \\ -B_1 D_{*1}^T \end{bmatrix} \tilde{R}_n^{-1} \begin{bmatrix} D_{*1} B_1^T & C \end{bmatrix} \end{aligned}$$

Let

$$X := \mathbf{Ric}(\mathbf{H})$$

$$Y := \mathbf{Ric}(\mathbf{J})$$

Based on X and Y , a state feedback matrix F and an observer gain matrix L can be constructed, which will be used in the solution formulas,

$$\begin{aligned} F &:= -R_n^{-1} (D_{1*}^T C_1 + B^T X) =: \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} =: \begin{bmatrix} F_{11} \\ F_{12} \\ F_2 \end{bmatrix} \\ L &:= -(B_1 D_{*1}^T + Y C^T) \tilde{R}_n^{-1} =: \begin{bmatrix} L_1 & L_2 \end{bmatrix} =: \begin{bmatrix} L_{11} & L_{12} & L_2 \end{bmatrix} \end{aligned}$$

where F_1 , F_2 , F_{11} , and F_{12} have $m_1, m_2, m_1 - p_2$, and p_2 rows, respectively, and L_1, L_2, L_{11} and L_{12} have $p_1, p_2, p_1 - m_2$, and m_2 columns, respectively.

Glover and Doyle [57] derived necessary and sufficient conditions for the existence of an \mathcal{H}_∞ suboptimal solution and further parameterized all such controllers. The results are obtained under the following assumptions:

- (A1) (A, B_2) is stabilizable and (C_2, A) detectable;
- (A2) $D_{12} = \begin{bmatrix} 0 \\ I_{m_2} \end{bmatrix}$ and $D_{21} = \begin{bmatrix} 0 & I_{p_2} \end{bmatrix}$;
- (A3) $\begin{bmatrix} A - j\omega I & B_2 \\ C_1 & D_{12} \end{bmatrix}$ has full column rank for all ω ;
- (A4) $\begin{bmatrix} A - j\omega I & B_1 \\ C_2 & D_{21} \end{bmatrix}$ has full row rank for all ω .

Together with appropriate partition of

$$D_{11} = \begin{bmatrix} D_{1111} & D_{1112} \\ D_{1121} & D_{1122} \end{bmatrix},$$

where D_{1122} has m_2 rows and p_2 columns, the solution formulas are given in the following theorem.

Theorem 4.1 [196] *Suppose $P(s)$ satisfies the assumptions (A1)–(A4).*

- (a) *There exists an internally stabilizing controller $K(s)$ such that $\|\mathcal{F}_l(P, K)\|_\infty < \gamma$ if and only if*

(i)

$$\gamma > \max(\overline{\sigma}[D_{1111}, D_{1112}], \overline{\sigma}[D_{1111}^T, D_{1121}^T])$$

and

- (ii) *there exist stabilizing solutions $X \geq 0$ and $Y \geq 0$ satisfying the two AREs corresponding to the Hamiltonian matrices \mathbf{H} and \mathbf{J} , respectively, and such that*

$$\rho(XY) < \gamma^2$$

where $\rho(\cdot)$ denotes the spectral radius.

- (b) *Given that the conditions of part (a) are satisfied, then all rational, internally stabilizing controllers, $K(s)$, satisfying $\|\mathcal{F}_l(P, K)\|_\infty < \gamma$ are given by*

$$K(s) = \mathcal{F}_l(M, \Phi)$$

for any rational $\Phi(s) \in \mathcal{H}_\infty$ such that $\|\Phi(s)\|_\infty < \gamma$, where $M(s)$ has the realization

$$M(s) = \left[\begin{array}{c|cc} \hat{A} & \hat{B}_1 & \hat{B}_2 \\ \hline \hat{C}_1 & \hat{D}_{11} & \hat{D}_{12} \\ \hat{C}_2 & \hat{D}_{21} & 0 \end{array} \right]$$

and

$$\hat{D}_{11} = -D_{1121}D_{1111}^T(\gamma^2I - D_{1111}D_{1111}^T)^{-1}D_{1112} - D_{1122}$$

$\hat{D}_{12} \in R^{m_2 \times m_2}$ and $\hat{D}_{21} \in R^{p_2 \times p_2}$ are any matrices (e.g. Cholesky factors) satisfying

$$\hat{D}_{12}\hat{D}_{12}^T = I - D_{1121}(\gamma^2I - D_{1111}^TD_{1111})^{-1}D_{1121}^T$$

$$\hat{D}_{21}^T\hat{D}_{21} = I - D_{1112}^T(\gamma^2I - D_{1111}D_{1111}^T)^{-1}D_{1112}$$

and

$$\begin{aligned}
\hat{B}_2 &= Z(B_2 + L_{12})\hat{D}_{12} \\
\hat{C}_2 &= -\hat{D}_{21}(C_2 + F_{12}) \\
\hat{B}_1 &= -ZL_2 + \hat{B}_2\hat{D}_{12}^{-1}\hat{D}_{11} \\
&= -ZL_2 + Z(B_2 + L_{12})\hat{D}_{11} \\
\hat{C}_1 &= F_2 + \hat{D}_{11}\hat{D}_{21}^{-1}\hat{C}_2 \\
&= F_2 - \hat{D}_{11}(C_2 + F_{12}) \\
\hat{A} &= A + BF + \hat{B}_1\hat{D}_{21}^{-1}\hat{C}_2 \\
&= A + BF - \hat{B}_1(C_2 + F_{12})
\end{aligned}$$

where

$$Z = (I - \gamma^{-2}YX)^{-1}$$

When $\Phi(s) = 0$ is chosen, the corresponding suboptimal controller is called the *central* controller that is widely used in the \mathcal{H}_∞ optimal design and has the state-space form

$$K_o(s) = \left[\begin{array}{c|c} \hat{A} & \hat{B}_1 \\ \hline \hat{C}_1 & \hat{D}_{11} \end{array} \right]$$

In the assumptions made earlier, (A2) assumes that the matrices D_{12} and D_{21} are in normalized forms and the system $P(s)$ is thus a so-called normalized system. When these two matrices are of full rank but not necessarily in the normalized forms will be discussed later.

4.3.2 Solution to S -over- KS Design

The mixed sensitivity optimization S -over- KS is a common design problem in practice. The problem is formulated as in (4.1) or (4.4), and the interconnected system is in (4.3) or (4.5). In this case, the second algebraic Riccati equation is of a special form, i.e. the constant term is zero. We list the solution formulas for this special design problem here.

For simplicity, we consider the suboptimal design of (4.1). That is, for a given γ , we want to find a stabilizing controller $K(s)$ such that

$$\left\| \begin{array}{c} (I + GK)^{-1} \\ K(I + GK)^{-1} \end{array} \right\|_\infty < \gamma$$

The interconnected system in the standard configuration is

$$P = \left[\begin{array}{cc|cc} I_p & -G & A & 0 \\ 0 & I_m & C & -B \\ I_p & -G & C & 0 \end{array} \right] = \left[\begin{array}{c|cc} A & 0 & -B \\ \hline \begin{bmatrix} C \\ 0 \end{bmatrix} & \begin{bmatrix} I_p \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ I_m \end{bmatrix} \end{array} \right]$$

where we assume that $G(s)$ has m inputs and p outputs, and

$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array} \right]$$

It is clear that γ must be larger than 1 from the assumption (a)(i) of Theorem 4.1, because the 2-norm of the constant matrix D_{11} of the above $P(s)$ is 1. In this case, the two algebraic Riccati equations are

$$\begin{aligned} A^T X + XA - XBB^T X + (1 - \gamma^{-2})^{-1} C^T C &= 0 \\ AY + YA^T - YC^T CY &= 0 \end{aligned}$$

The formula of the central controller is

$$K_o = \left[\begin{array}{c|c} \frac{A - BB^T X - (1 - \gamma^{-2})^{-1} ZYC^T C}{B^T X} & ZYC^T \\ \hline & 0 \end{array} \right]$$

where $Z = (I - \gamma^{-2} YX)^{-1}$.

The formulas for all suboptimal controllers are

$$K(s) = \mathcal{F}_l(M, \Phi)$$

with $M(s)$ having the state-space realization

$$M(s) = \left[\begin{array}{cc|cc} \frac{A - BB^T X - (1 - \gamma^{-2})^{-1} ZYC^T C}{B^T X} & ZYC^T & -ZB & \\ \hline & 0 & I_m & \\ & -(1 - \gamma^{-2})^{-1} C & I_p & 0 \end{array} \right]$$

and $\Phi(s) \in \mathcal{H}_\infty$, such that $\|\Phi\|_\infty < \gamma$.

4.3.3 The Case of $D_{22} \neq 0$

When there is a direct link between the control input and the measurement output, the matrix D_{22} will not disappear in (4.7). The controller formulas for the case $D_{22} \neq 0$ are discussed here.

As a matter of fact, the D_{22} term can be easily separated from the rest of the system as depicted in Fig. 4.4. A controller $\tilde{K}(s)$ for the system with zero D_{22} will be synthesized first, and then the controller $\tilde{K}(s)$ for the original system can be recovered from $K(s)$ and D_{22} by

$$\tilde{K}(s) = K(s)(I + D_{22}K(s))^{-1}$$

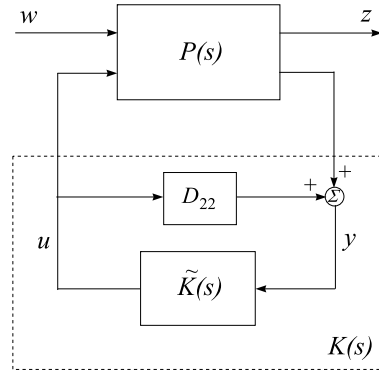
The state-space model of $\tilde{K}(s)$ can be derived as

$$\tilde{K}(s) = \left[\begin{array}{c|c} \frac{A_K - B_K D_{22}(I + D_K D_{22})^{-1} C_K}{(I + D_K D_{22})^{-1} C_K} & B_K (I + D_{22} D_K)^{-1} \\ \hline & D_K (I + D_{22} D_K)^{-1} \end{array} \right]$$

where we assume that

$$K(s) = \left[\begin{array}{c|c} \frac{A_K}{C_K} & \frac{B_K}{D_K} \end{array} \right]$$

Fig. 4.4 The case of nonzero D_{22}



4.3.4 Normalization Transformations

In general, the system data given would not be in the normalization form as discussed in Sect. 4.3.1, though D_{12} will be of full column rank and D_{21} of full row rank, respectively, for any realistic control systems. In order to apply the results in Theorem 4.1, certain transformations must be used first. Using the *singular value decomposition* (SVD) or otherwise, we may find orthonormal matrices U_{12} , V_{12} , U_{21} , and V_{21} , such that

$$U_{12}D_{12}V_{12}^T = \begin{bmatrix} 0 \\ \Sigma_{12} \end{bmatrix} \quad (4.8)$$

$$U_{21}D_{21}V_{21}^T = \begin{bmatrix} 0 & \Sigma_{21} \end{bmatrix} \quad (4.9)$$

where $\Sigma_{12}: m_2 \times m_2$ and $\Sigma_{21}: p_2 \times p_2$ are nonsingular. Furthermore, we have

$$U_{12}D_{12}V_{12}^T\Sigma_{12}^{-1} = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (4.10)$$

$$\Sigma_{21}^{-1}U_{21}D_{21}V_{21}^T = \begin{bmatrix} 0 & I \end{bmatrix} \quad (4.11)$$

The right-hand sides of the above equations are now in the normalized form.

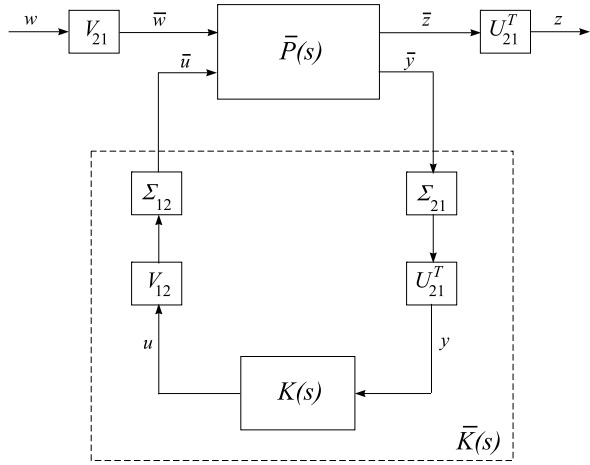
When $p_1 > m_2$ and $p_2 < m_1$, the matrices U_{12} and V_{21} can be partitioned as

$$U_{12} = \begin{bmatrix} U_{121} \\ U_{122} \end{bmatrix} \quad (4.12)$$

$$V_{21} = \begin{bmatrix} V_{211} \\ V_{212} \end{bmatrix} \quad (4.13)$$

with $U_{121}: (p_1 - m_2) \times p_1$, $U_{122}: m_2 \times p_1$, $V_{211}: (m_1 - p_2) \times m_1$ and $V_{212}: p_2 \times m_1$.

The normalization of $P(s)$ into $\bar{P}(s)$ is based on the above transformations and shown in Fig. 4.5.

Fig. 4.5 Normalization configuration

Given $P(s)$, the state-space form of $\bar{P}(s)$ is obtained as follows:

$$\begin{aligned}
 \bar{B}_1 &= B_1 V_{21}^T \\
 \bar{B}_2 &= B_2 V_{12}^T \Sigma_{12}^{-1} \\
 \bar{C}_1 &= U_{12} C_1 \\
 \bar{C}_2 &= \Sigma_{21}^{-1} U_{21} C_2 \\
 \bar{D}_{11} &= U_{12} D_{11} V_{21}^T = \begin{bmatrix} U_{121} D_{11} V_{211}^T & U_{121} D_{11} V_{212}^T \\ U_{122} D_{11} V_{211}^T & U_{122} D_{11} V_{212}^T \end{bmatrix} \\
 \bar{D}_{12} &= \begin{bmatrix} 0 \\ I \end{bmatrix} = U_{12} D_{12} V_{12}^T \Sigma_{12}^{-1} \\
 \bar{D}_{21} &= \begin{bmatrix} 0 & I \end{bmatrix} = \Sigma_{21}^{-1} U_{21} D_{21} V_{21}^T
 \end{aligned}$$

Since V_{21} and U_{12} are orthonormal, the infinitive norms of the transfer function matrices from w to z and from \bar{w} to \bar{z} are the same, i.e. $\|T_{zw}\|_\infty = \|T_{\bar{z}\bar{w}}\|_\infty$, with obviously $K(s) = V_{12}^T \Sigma_{12}^{-1} \bar{K}(s) \Sigma_{21}^{-1} U_{21}$, where $\bar{K}(s)$ is a suboptimal solution with regard to $\bar{P}(s)$.

4.3.5 Direct Formulas for \mathcal{H}_∞ Suboptimal Central Controller

Based on the transformations discussed in the last subsection and by further manipulations, direct solution formulas can be derived for the general, non-normalized system data (4.7) [67].

We first discuss the two algebraic Riccati equations that are essential in the computations of \mathcal{H}_∞ suboptimal controllers. The conclusion is a little surprising: the two AREs remain the same as the matrices in the normalized system data (matrices

with bars) replaced by corresponding matrices in the general form (matrices without bars) of $P(s)$. This is obtained by the following routine matrix manipulations:

$$\begin{aligned}
 \bar{C}_1^T \bar{C}_1 &= C_1^T U_{12}^T U_{12} C_1 = C_1^T C_1 \\
 \begin{bmatrix} \bar{B} \\ -\bar{C}_1^T \bar{D}_{1*} \end{bmatrix} &= \begin{bmatrix} B_1 V_{21}^T & B_2 V_{12}^T \Sigma_{12}^{-1} \\ -C_1^T U_{12}^T U_{12} D_{11} V_{21}^T & -C_1^T U_{12}^T U_{12} D_{12} V_{12}^T \Sigma_{12}^{-1} \end{bmatrix} \\
 &= \begin{bmatrix} B \\ -C_1^T D_{1*} \end{bmatrix} \begin{bmatrix} V_{21}^T & 0 \\ 0 & V_{12}^T \Sigma_{12}^{-1} \end{bmatrix} \\
 \bar{D}_{1*}^T \bar{D}_{1*} &= \begin{bmatrix} V_{21} D_{11}^T U_{12}^T \\ \Sigma_{12}^{-T} V_{12} D_{12}^T U_{12}^T \end{bmatrix} \begin{bmatrix} U_{12} D_{11} V_{21}^T & U_{12} D_{12} V_{12}^T \Sigma_{12}^{-1} \end{bmatrix} \\
 &= \begin{bmatrix} V_{21} & 0 \\ 0 & \Sigma_{12}^{-T} V_{12} \end{bmatrix} D_{1*}^T D_{1*} \begin{bmatrix} V_{21}^T & 0 \\ 0 & V_{12}^T \Sigma_{12}^{-1} \end{bmatrix}
 \end{aligned}$$

where

$$D_{1*} := \begin{bmatrix} D_{11} & D_{12} \end{bmatrix} \quad (4.14)$$

Thus,

$$R_n = \begin{bmatrix} V_{21} & 0 \\ 0 & \Sigma_{12}^{-T} V_{12} \end{bmatrix} R \begin{bmatrix} V_{21}^T & 0 \\ 0 & V_{12}^T \Sigma_{12}^{-1} \end{bmatrix}$$

with

$$R := D_{1*}^T D_{1*} - \begin{bmatrix} \gamma^2 I_{m_1} & 0 \\ 0 & 0 \end{bmatrix} \quad (4.15)$$

Consequently, the Hamiltonian \mathbf{H} defined for normalized system data is given by

$$\mathbf{H} = \begin{bmatrix} A & 0 \\ -C_1^T C_1 & -A^T \end{bmatrix} - \begin{bmatrix} B \\ -C_1^T D_{1*} \end{bmatrix} R^{-1} \begin{bmatrix} D_{1*}^T C_1 & B^T \end{bmatrix} \quad (4.16)$$

Similarly, the Hamiltonian \mathbf{J} is given by

$$\mathbf{J} = \begin{bmatrix} A^T & 0 \\ -B_1 B_1^T & -A \end{bmatrix} - \begin{bmatrix} C^T \\ -B_1 D_{*1}^T \end{bmatrix} \tilde{R}^{-1} \begin{bmatrix} D_{*1} B_1^T & C \end{bmatrix} \quad (4.17)$$

where

$$D_{*1} := \begin{bmatrix} D_{11} \\ D_{21} \end{bmatrix} \quad (4.18)$$

and

$$\tilde{R} := D_{*1} D_{*1}^T - \begin{bmatrix} \gamma^2 I_{p_1} & 0 \\ 0 & 0 \end{bmatrix} \quad (4.19)$$

with

$$\tilde{R}_n = \begin{bmatrix} U_{12} & 0 \\ 0 & \Sigma_{21}^{-1} U_{21} \end{bmatrix} \tilde{R} \begin{bmatrix} U_{12}^T & 0 \\ 0 & U_{21}^T \Sigma_{21}^{-T} \end{bmatrix}$$

Hence, we have shown that the two AREs are exactly the same and so we may, in the following, let X and Y be the stabilizing solutions of the two AREs corresponding to the general system data.

Another pair of matrices used in Theorem 4.1 may also be defined from the solutions X and Y ,

$$F := -R^{-1}(D_{1*}^T C_1 + B^T X) =: \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \quad (4.20)$$

$$L := -(B_1 D_{*1}^T + Y C^T) \tilde{R}^{-1} =: [L_1 \quad L_2] \quad (4.21)$$

with $F_1: m_1 \times n$, $F_2: m_2 \times n$, $L_1: n \times p_1$ and $L_2: n \times p_2$. It is easy to check that

$$\bar{F} = \begin{bmatrix} V_{21} & 0 \\ 0 & \Sigma_{12} V_{12} \end{bmatrix} F$$

and

$$\begin{aligned} \bar{F}_{11} &= V_{211} F_1 \\ \bar{F}_{12} &= V_{212} F_1 \\ \bar{F}_2 &= \Sigma_{12} V_{12} F_2 \end{aligned}$$

Similarly, we have

$$\bar{L} = L \begin{bmatrix} U_{12}^T & 0 \\ 0 & U_{21}^T \Sigma_{21} \end{bmatrix}$$

and

$$\begin{aligned} \bar{L}_{11} &= L_1 U_{121}^T \\ \bar{L}_{12} &= L_1 U_{122}^T \\ \bar{L}_2 &= L_2 U_{21}^T \Sigma_{21} \end{aligned}$$

Using the above results and Theorem 4.1, with further manipulations, we can get the following corollary. Note that the assumption (A2) is relaxed to the general case:

(A2) D_{12} is of full column rank and D_{21} of full row rank

Also, note that for the sake of simplicity, only the formulas for the central suboptimal controller is given. Formulas for all suboptimal controllers can be derived accordingly.

Corollary 4.2 Suppose $P(s)$ satisfies the assumptions (A1)–(A4).

- (a) There exists an internally stabilizing controller $K(s)$ such that $\|F_l(P, K)\|_\infty < \gamma$ if and only if
- (i)

$$\gamma > \max\{\bar{\sigma}(U_{121} D_{11}), \bar{\sigma}(D_{11} V_{211}^T)\} \quad (4.22)$$

where U_{121} and V_{211} are as defined in (4.12) and (4.13), and

- (ii) *there exist solutions $X \geq 0$ and $Y \geq 0$ satisfying the two AREs corresponding to (4.16) and (4.17), respectively, and such that*

$$\rho(XY) < \gamma^2 \quad (4.23)$$

where $\rho(\bullet)$ denotes the spectral radius.

- (b) *Given that the conditions of part (a) are satisfied, then the central \mathcal{H}_∞ suboptimal controller for $P(s)$ is given by*

$$K(s) = \left[\frac{A_K}{C_K} \middle| \frac{B_K}{D_K} \right]$$

where

$$\begin{aligned} D_K &= -\gamma^2 (U_{122} D_{12})^{-1} U_{122} (\gamma^2 I - D_{11} V_{211}^T V_{211} D_{11}^T U_{121}^T U_{121})^{-1} \\ &\quad \times D_{11} V_{212}^T (D_{21} V_{212}^T)^{-1} \end{aligned} \quad (4.24)$$

$$B_K = -Z [L_2 - (B_2 + L_1 D_{12}) D_K] \quad (4.25)$$

$$C_K = F_2 - D_K (C_2 + D_{21} F_1) \quad (4.26)$$

$$A_K = A + B F - B_K (C_2 + D_{21} F_1) \quad (4.27)$$

where

$$Z = (I - \gamma^{-2} Y X)^{-1} \quad (4.28)$$

and U_{12} , V_{21} , F , and L are as defined in (4.8) and (4.12), (4.9) and (4.13), (4.20), and (4.21), respectively.

Remark The above corollary addresses the general case of $p_1 > m_2$ and $p_2 < m_1$. There are three special cases of the dimensions in which the solution formulas would be simpler.

Case 1: $p_1 = m_2$ and $p_2 < m_1$.

In this case, the orthogonal transformation on D_{12} is not needed. The condition (a)(i) in Corollary 4.2 is reduced to $\gamma > \overline{\sigma}(D_{11} V_{211}^T)$, and

$$D_K = -D_{12}^{-1} D_{11} V_{212}^T (D_{21} V_{212}^T)^{-1}$$

Case 2: $p_1 > m_2$ and $p_2 = m_1$.

In this case, the orthogonal transformation on D_{21} is not needed. The condition (a)(i) in Corollary 4.2 is reduced to $\gamma > \overline{\sigma}(U_{121} D_{11})$, and

$$D_K = -(U_{122} D_{12})^{-1} U_{122} D_{11} D_{21}^{-1}$$

Case 3: $p_1 = m_2$ and $p_2 = m_1$.

In this case, both orthogonal transformations are not needed. The condition (a)(i) in Corollary 4.2 is reduced to any positive γ , and

$$D_K = -D_{12}^{-1} D_{11} D_{21}^{-1}$$

Another special case is when $D_{11} = 0$, in which the central controller is simply given by

$$K(s) = \left[\frac{A + BF + ZL_2(C_2 + D_{21}F_1)}{F_2} \mid \frac{-ZL_2}{0} \right]$$

4.4 Formulas for Discrete-Time Cases

In this section, formulas for \mathcal{H}_∞ solutions in discrete-time cases [62, 63] will be given. Consider a generalized, linear, discrete-time system, described by the equations

$$\begin{aligned} x_{k+1} &= Ax_k + B_1 w_k + B_2 u_k \\ z_k &= C_1 x_k + D_{11} w_k + D_{12} u_k \\ y_k &= C_2 x_k + D_{21} w_k + D_{22} u_k \end{aligned} \quad (4.29)$$

where $x_k \in \mathcal{R}^n$ is the state vector, $w_k \in \mathcal{R}^{m_1}$ is the exogenous input vector (the disturbance), $u_k \in \mathcal{R}^{m_2}$ is the control input vector, $z_k \in \mathcal{R}^{p_1}$ is the error vector, and $y_k \in \mathcal{R}^{p_2}$ is the measurement vector, with $p_1 \geq m_2$ and $p_2 \leq m_1$. The transfer function matrix of the system will be denoted by

$$\begin{aligned} P(z) &= \begin{bmatrix} P_{11}(z) & P_{12}(z) \\ P_{21}(z) & P_{22}(z) \end{bmatrix} \\ &= \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array} \right] \\ &=: \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \end{aligned} \quad (4.30)$$

Similar to the continuous-time case, the \mathcal{H}_∞ suboptimal discrete-time control problem is to find an internally stabilizing controller $K(z)$ such that, for a prespecified positive value γ ,

$$\|F_\ell(P, K)\|_\infty < \gamma \quad (4.31)$$

We need the following assumptions.

- (A1) (A, B_2) is stabilizable and (C_2, A) is detectable;
- (A2) $\begin{bmatrix} A - e^{j\Theta} I_n & B_2 \\ C_1 & D_{12} \end{bmatrix}$ has full column rank for all $\Theta \in [0, 2\pi)$;
- (A3) $\begin{bmatrix} A - e^{j\Theta} I_n & B_1 \\ C_2 & D_{21} \end{bmatrix}$ has full row rank for all $\Theta \in [0, 2\pi)$.

We shall also assume that a loop-shifting transformation that enables us to set $D_{22} = 0$ has been carried out. The general case ($D_{22} \neq 0$) will be dealt with at the end of this section.

Note that the method under consideration does not involve reduction of the matrices D_{12} and D_{21} to some special form, as is usually required in the design of continuous-time \mathcal{H}_∞ controllers.

Let

$$\bar{C} = \begin{bmatrix} C_1 \\ 0 \end{bmatrix}, \quad \bar{D} = \begin{bmatrix} D_{11} & D_{12} \\ I_{m_1} & 0 \end{bmatrix}$$

and define

$$J = \begin{bmatrix} I_{p_1} & 0 \\ 0 & -\gamma^2 I_{m_1} \end{bmatrix}, \quad \hat{J} = \begin{bmatrix} I_{m_1} & 0 \\ 0 & -\gamma^2 I_{m_2} \end{bmatrix}, \quad \tilde{J} = \begin{bmatrix} I_{m_1} & 0 \\ 0 & -\gamma^2 I_{p_1} \end{bmatrix}$$

Let X_∞ be the solution to the discrete-time Riccati equation

$$X_\infty = \bar{C}^T J \bar{C} + A^T X_\infty A - L^T R^{-1} L \quad (4.32)$$

where

$$R = \bar{D}^T J \bar{D} + B^T X_\infty B =: \begin{bmatrix} R_1 & R_2^T \\ R_2 & R_3 \end{bmatrix}$$

$$L = \bar{D}^T J \bar{C} + B^T X_\infty A =: \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}$$

Assume that there exists an $m_2 \times m_2$ matrix V_{12} such that

$$V_{12}^T V_{12} = R_3$$

and an $m_1 \times m_1$ matrix V_{21} such that

$$V_{21}^T V_{21} = -\gamma^{-2} \nabla, \quad \nabla = R_1 - R_2^T R_3^{-1} R_2 < 0$$

Define the matrices

$$\begin{bmatrix} A_t & \tilde{B}_t \\ C_t & \tilde{D}_t \end{bmatrix} =: \left[\begin{array}{c|cc} A_t & \tilde{B}_{t1} & \tilde{B}_{t2} \\ \hline C_{t1} & \tilde{D}_{t11} & \tilde{D}_{t12} \\ C_{t2} & \tilde{D}_{t21} & \tilde{D}_{t22} \end{array} \right]$$

$$= \left[\begin{array}{c|cc} A - B_1 \nabla^{-1} L_\nabla & B_1 V_{21}^{-1} & 0 \\ \hline V_{12} R_3^{-1} (L_2 - R_2 \nabla^{-1} L_\nabla) & V_{12} R_3^{-1} R_2 V_{21}^{-1} & I \\ \hline C_2 - D_{21} \nabla^{-1} L_\nabla & D_{21} V_{21}^{-1} & 0 \end{array} \right]$$

where

$$L_\nabla = L_1 - R_2^T R_3^{-1} L_2$$

Let Z_∞ be the solution to the discrete-time Riccati equation

$$Z_\infty = \tilde{B}_t \hat{J} \tilde{B}_t^T + A_t Z_\infty A_t^T - M_t S_t^{-1} M_t^T \quad (4.33)$$

in which

$$S_t = \tilde{D}_t \hat{J} \tilde{D}_t^T + C_t Z_\infty C_t^T =: \begin{bmatrix} S_{t1} & S_{t2} \\ S_{t2}^T & S_{t3} \end{bmatrix}$$

$$M_t = \tilde{B}_t \hat{J} \tilde{D}_t^T + A_t Z_\infty C_t^T =: [M_{t1} \quad M_{t2}]$$

Equations (4.32) and (4.33) are referred to as the *X-Riccati equation* and *Z-Riccati equation*, respectively.

A stabilizing controller that satisfies

$$\|F_\ell(P, K)\|_\infty < \gamma$$

exists if and only if [62]

- (1) there exists a solution to the Riccati equation (4.32) satisfying

$$X_\infty \geq 0$$

$$\nabla < 0$$

such that $A - BR^{-1}L$ is asymptotically stable;

- (2) there exists a solution to the Riccati equation (4.33) such that

$$Z_\infty \geq 0$$

$$S_{t_1} - S_{t_2} S_{t_3}^{-1} S_{t_2}^T < 0$$

with $A_t - M_t S_t^{-1} C_t$ asymptotically stable.

In this case, a controller that achieves the objective may be given by [62]

$$\begin{aligned}\hat{x}_{k+1} &= A_t \hat{x}_k + B_2 u_k + M_{t_2} S_{t_3}^{-1} (y_k - C_{t_2} \hat{x}_k) \\ V_{12} u_k &= -C_{t_1} \hat{x}_k - S_{t_2} S_{t_3}^{-1} (y_k - C_{t_2} \hat{x}_k)\end{aligned}$$

which yields

$$K_0 = \left[\begin{array}{c|c} \frac{A_t - B_2 V_{12}^{-1} (C_{t_1} - S_{t_2} S_{t_3}^{-1} C_{t_2}) - M_{t_2} S_{t_3}^{-1} C_{t_2}}{-V_{12}^{-1} (C_{t_1} - S_{t_2} S_{t_3}^{-1} C_{t_2})} & \frac{-B_2 V_{12}^{-1} S_{t_2} S_{t_3}^{-1} + M_{t_2} S_{t_3}^{-1}}{-V_{12}^{-1} S_{t_2} S_{t_3}^{-1}} \end{array} \right] \quad (4.34)$$

This is the so-called *central controller*, which is widely used in practice.

Consider now the general case of $D_{22} \neq 0$. Suppose

$$\hat{K} = \left[\begin{array}{c|c} \hat{A}_k & \hat{B}_k \\ \hline \hat{C}_k & \hat{D}_k \end{array} \right]$$

is a stabilizing controller for D_{22} set to zero, and satisfies

$$\left\| F_\ell \left(P - \begin{bmatrix} 0 & 0 \\ 0 & D_{22} \end{bmatrix}, \hat{K} \right) \right\|_\infty < \gamma$$

Then

$$\begin{aligned}F_\ell(P, \hat{K}(I + D_{22}\hat{K})^{-1}) &= P_{11} + P_{12}\hat{K}(I + D_{22}\hat{K} - P_{22}\hat{K})^{-1}P_{21} \\ &= F_\ell \left(P - \begin{bmatrix} 0 & 0 \\ 0 & D_{22} \end{bmatrix}, \hat{K} \right)\end{aligned}$$

In this way, a controller \hat{K} for

$$P - \begin{bmatrix} 0 & 0 \\ 0 & D_{22} \end{bmatrix}$$

yields a controller $K = \hat{K}(I + D_{22}\hat{K})^{-1}$ for P . It may be shown that

$$K = \left[\frac{\hat{A}_k - \hat{B}_k D_{22} (I_{m_2} + \hat{D}_k D_{22})^{-1} \hat{C}_k}{(I_{m_2} + \hat{D}_k D_{22})^{-1} \hat{C}_k} \mid \frac{\hat{B}_k - \hat{B}_k D_{22} (I_{m_2} + \hat{D}_k D_{22})^{-1} \hat{D}_k}{(I_{m_2} + \hat{D}_k D_{22})^{-1} \hat{D}_k} \right]$$

In order to find K from \hat{K} we must exclude the possibility of the feedback system becoming ill-posed, i.e. $\det(I + \hat{K}(\infty)D_{22}) = 0$.

Chapter 5

\mathcal{H}_∞ Loop-Shaping Design Procedures

In the previous chapter the \mathcal{H}_∞ optimization approach was introduced, which formulated robust stabilization and nominal performance requirements as (sub)optimization problems of the \mathcal{H}_∞ norm of certain cost functions. Several formulations of cost function are applicable in the robust controller design, for instance, the weighted S/KS and S/T design methods. The optimization of S/KS , where S is the sensitivity function and K the controller to be designed, could achieve the nominal performance in terms of tracking or output disturbance rejection and robustly stabilize the system against additive model perturbations. On the other hand, the mixed sensitivity optimization of S/T , where T is the complementary sensitivity function, could achieve robust stability against multiplicative model perturbations in addition to the nominal performance. Both of them are useful robust controller design methods, but the model perturbation representations are limited by the condition on the number of right-half complex plane poles. Also, there may exist undesirable pole-zero cancellations between the nominal model and the \mathcal{H}_∞ controllers [151]. In this chapter, an alternative way to represent the model uncertainty is introduced. The uncertainty is described by the perturbations directly on the coprime factors of the nominal model [171, 172]. The \mathcal{H}_∞ robust stabilization against such perturbations and the consequently developed design method, the \mathcal{H}_∞ loop-shaping design procedure (LSDP) [111, 112], could relax the restrictions on the number of right-half plane poles and produce no pole-zero cancellations between the nominal model and controller designed. This method does not require an iterative procedure to obtain an optimal solution and thus raises the computational efficiency. Furthermore, the \mathcal{H}_∞ LSDP inherits classical loop-shaping design ideas so that practicing control engineers would feel more comfortable to use it.

5.1 Robust Stabilization Against Normalized Coprime Factor Perturbations

Matrices $(\tilde{M}, \tilde{N}) \in \mathcal{H}_\infty^+$, where \mathcal{H}_∞^+ denotes the space of functions with no poles in the closed right-half complex plane, constitute a left coprime factorization of a given plant model G if and only if

- (i) \tilde{M} is square, and $\det(\tilde{M}) \neq 0$
- (ii) the plant model is given by

$$G = \tilde{M}^{-1} \tilde{N} \quad (5.1)$$

- (iii) there exists $(\tilde{V}, \tilde{U}) \in \mathcal{H}_\infty^+$ such that

$$\tilde{M} \tilde{V} + \tilde{N} \tilde{U} = I \quad (5.2)$$

A left coprime factorization of a plant model G as defined in (5.1) is *normalized* if and only if

$$\tilde{N} \tilde{N}^- + \tilde{M} \tilde{M}^- = I, \quad \forall s \quad (5.3)$$

where $\tilde{N}^-(s) = \tilde{N}^T(-s)$, etc.

For a minimal realization of $G(s)$,

$$\begin{aligned} G(s) &= D + C(sI - A)^{-1} B \\ &=: \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \end{aligned} \quad (5.4)$$

A state-space construction for the normalized left coprime factorization can be obtained in terms of the solution to the generalized filter algebraic Riccati equation

$$\begin{aligned} (A - BD^T R^{-1} C)Z + Z(A - BD^T R^{-1} C)^T - ZC^T R^{-1} CZ \\ + B(I - D^T R^{-1} D)B^T = 0 \end{aligned} \quad (5.5)$$

where $R := I + DD^T$ and $Z \geq 0$ is the unique stabilizing solution. If $H = -(ZC^T + BD^T)R^{-1}$, then

$$\begin{bmatrix} \tilde{N} & \tilde{M} \end{bmatrix} := \left[\begin{array}{c|cc} A + HC & B + HD & H \\ \hline R^{-1/2}C & R^{-1/2}D & R^{-1/2} \end{array} \right] \quad (5.6)$$

is a normalized left coprime factorization of G such that $G = \tilde{M}^{-1} \tilde{N}$.

The normalized right coprime factorization can be defined similarly and a state-space representation can be similarly obtained in terms of the solution to the generalized control algebraic Riccati equation [111],

$$\begin{aligned} (A - BS^{-1} D^T C)^T X + X(A - BS^{-1} D^T C) - XBS^{-1} B^T X \\ + C^T (I - DS^{-1} D^T) C = 0 \end{aligned} \quad (5.7)$$

where $S := I + D^T D$ and $X \geq 0$ is the unique stabilizing solution.

A perturbed plant transfer function can be described by

$$G_\Delta = (\tilde{M} + \Delta_{\tilde{M}})^{-1} (\tilde{N} + \Delta_{\tilde{N}})$$

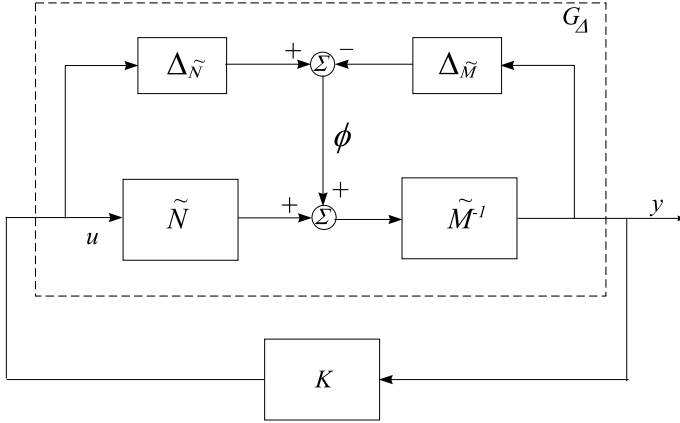


Fig. 5.1 Robust stabilization with regard to coprime factor uncertainty

where $(\Delta_{\tilde{M}}, \Delta_{\tilde{N}})$ are unknown but stable transfer functions that represent the uncertainty (perturbation) in the nominal plant model. The design objective of robust stabilization is to stabilize not only the nominal model G , but the family of perturbed plants defined by

$$\mathcal{G}_\epsilon = \{(\tilde{M} + \Delta_{\tilde{M}})^{-1}(\tilde{N} + \Delta_{\tilde{N}}) : \|\Delta_{\tilde{M}}, \Delta_{\tilde{N}}\|_\infty < \epsilon\}$$

where $\epsilon > 0$ is the stability margin. Using a feedback controller K as shown schematically in Fig. 5.1 and the Small-Gain Theorem, the feedback system $(\tilde{M}, \tilde{N}, K, \epsilon)$ is robustly stable if and only if (G, K) is internally stable and

$$\left\| \begin{bmatrix} K(I - GK)^{-1}\tilde{M}^{-1} \\ (I - GK)^{-1}\tilde{M}^{-1} \end{bmatrix} \right\|_\infty \leq \epsilon^{-1}$$

To maximize the robust stability of the closed-loop system given in Fig. 5.1, one must minimize

$$\gamma := \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} \tilde{M}^{-1} \right\|_\infty$$

The lowest achievable value of γ for all stabilizing controllers K is

$$\gamma_o = \inf_{K \text{ stabilizing}} \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} \tilde{M}^{-1} \right\|_\infty \quad (5.8)$$

and is given in [111] by

$$\gamma_o = (1 - \|\begin{bmatrix} \tilde{N} & \tilde{M} \end{bmatrix}\|_H^2)^{-1/2} \quad (5.9)$$

where $\|\cdot\|_H$ denotes the Hankel norm. From [111]

$$\|\begin{bmatrix} \tilde{N} & \tilde{M} \end{bmatrix}\|_H^2 = \lambda_{\max}(ZX(I + ZX)^{-1}) \quad (5.10)$$

where $\lambda_{\max}(\cdot)$ represents the maximum eigenvalue, hence from (5.9),

$$\gamma_o = (1 + \lambda_{\max}(ZX))^{1/2} \quad (5.11)$$

From [111], all controllers optimizing γ are given by $K = UV^{-1}$, where U and V are stable and are right coprime factorizations of K , and where

$$\left\| \begin{bmatrix} -\tilde{N}^* \\ \tilde{M}^* \end{bmatrix} + \begin{bmatrix} U \\ V \end{bmatrix} \right\|_\infty = \|\begin{bmatrix} \tilde{N} & \tilde{M} \end{bmatrix}\|_H$$

This is a Hankel approximation problem and can be solved using an algorithm developed by Glover [55].

A controller that achieves a $\gamma > \gamma_o$ is given in [111] by

$$K := \left[\begin{array}{c|c} A + BF + \gamma^2(L^T)^{-1}ZC^T(C + DF) & \gamma^2(L^T)^{-1}ZC^T \\ \hline B^T X & -D^T \end{array} \right] \quad (5.12)$$

where $F = -S^{-1}(D^T C + B^T X)$ and $L = (1 - \gamma^2)I + XZ$.

However, if $\gamma = \gamma_o$, then $L = XZ - \lambda_{\max}(XZ)I$, which is singular, and thus (5.12) cannot be implemented. This problem can be resolved using the descriptor system [147, 148]. A controller that achieves $\gamma \geq \gamma_o$ can be given in the descriptor form by

$$K := \left[\begin{array}{c|c} -L^T s + L^T(A + BF) + \gamma^2 ZC^T(C + DF) & \gamma^2 ZC^T \\ \hline B^T X & -D^T \end{array} \right] \quad (5.13)$$

5.2 Loop-Shaping Design Procedures

In the classical control systems design with single-input–single-output (SISO) systems, it is a well-known and practically effective technique to use a compensator to alter the frequency response (the Bode diagram) of the open-loop transfer function so that the unity feedback system will achieve stability, good performance and certain robustness. Indeed, these performance requirements discussed in Chap. 3 can be converted into corresponding frequency requirements on the open-loop system. For instance, in order to achieve good tracking, it is required that $\|(I + GK)^{-1}\|_\infty$ should be small. That is $\overline{\sigma}((I + GK)^{-1}(j\omega)) \ll 1$, for ω over a low-frequency range, since the signals to be tracked are usually of low frequency. This in turn implies that $\underline{\sigma}(GK(j\omega)) \gg 1$, for ω over that frequency range. Similar deductions can be applied to other design-performance specifications.

However, a direct extension of the above method into multivariable systems is difficult not only because multi-input and multi-output (MIMO) are involved but also due to the lack of phase information in MIMO cases that makes it impossible to predict the stability of the closed-loop system formed by the unity feedback. However, based on the robust stabilization against perturbations on normalized coprime

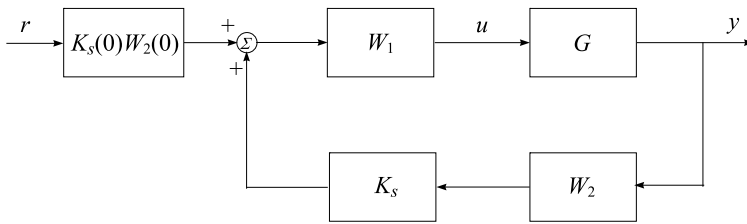


Fig. 5.2 One-degree-of-freedom LSDP configuration

factorizations, a design method, known as the \mathcal{H}_∞ loop-shaping design procedure (LSDP), has been developed [112]. The LSDP method augments the plant with appropriately chosen weights so that the frequency response of the open-loop system (the weighted plant) is reshaped in order to meet the closed-loop performance requirements. Then a robust controller is synthesized to meet the stability.

This loop-shaping design procedure can be carried out in the following steps.

- (i) Using a precompensator, W_1 , and/or a postcompensator, W_2 , as depicted in Fig. 5.2, the singular values of the nominal system G are modified to give a desired loop shape. Usually, the least singular value of the weighted system should be made large over the low-frequency range to achieve good performance such as the tracking, the largest singular value is small over the high-frequency range to deal with unmodeled dynamics, and the bandwidth affects the system response speed, while the slope of the singular values near the bandwidth frequency should not be too steep. The nominal system and weighting functions W_1 and W_2 are combined to form the shaped system, G_s , where $G_s = W_2 G W_1$. It is assumed that W_1 and W_2 are such that G_s contains no hidden unstable modes.
- (ii) A feedback controller, K_s , is synthesized that robustly stabilizes the normalized left coprime factorization of G_s , with a stability margin ϵ . It can be shown [112] that if ϵ is not less than 0.2, the frequency response of $K_s W_2 G W_1$ will be similar to that of $W_2 G W_1$. On the other hand, if the achievable ϵ is too large, this would probably indicate an overdesigned case with respect to the robustness, which means that the performance of the system may possibly be improved by using a larger γ in computing K_s .
- (iii) The final feedback controller, K_{final} , is then constructed by combining the \mathcal{H}_∞ controller K_s , with the weighting functions W_1 and W_2 such that

$$K_{\text{final}} = W_1 K_s W_2$$

For a tracking problem, the reference signal generally comes between K_s and W_1 . In order to reduce the steady-state tracking error, a constant gain $K_s(0)W_2(0)$ is placed on the feedforward path, where

$$K_s(0)W_2(0) = \lim_{s \rightarrow 0} K_s(s)W_2(s)$$

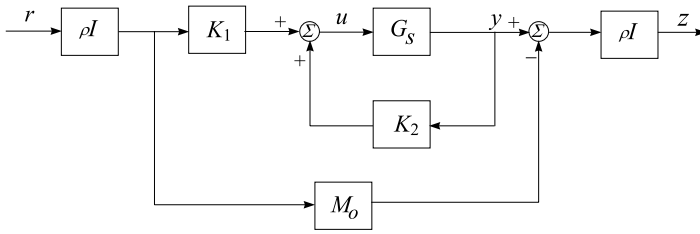


Fig. 5.3 Two-degree-of-freedom LSDP

The closed-loop transfer function between the reference r and the plant output y then becomes

$$Y(s) = [I - G(s)K_{\text{final}}(s)]^{-1} G(s)W_1(s)K_s(0)W_2(0)R(s)$$

The above design procedure can be developed further into a two-degree-of-freedom (2DOF) scheme as shown in Fig. 5.3.

The philosophy of the 2DOF scheme is to use the feedback controller $K_2(s)$ to meet the requirements of internal and robust stability, disturbance rejection, measurement noise attenuation, and sensitivity minimization. The precompensator K_1 is applied to the reference signal, which optimizes the response of the overall system to the command input such that the output of the system would be “near” to that of a chosen ideal system M_o . The feedforward compensator K_1 depends on design objectives and can be synthesized together with the feedback controller in a single step via the \mathcal{H}_∞ LSDP [75]. In the 2DOF LSDP, the postcompensator W_2 is usually chosen as an identity matrix. The constant ρ is a scaling parameter used to emphasize the model-matching part in the design.

5.3 Formulas for the Discrete-Time Case

In this section, the formulas for the (1-block) \mathcal{H}_∞ LSDP controller in the discrete-time case will be introduced [68].

5.3.1 Normalized Coprime Factorization of Discrete-Time Plant

Let $G(z)$ be a minimal realization, discrete-time model of a plant,

$$\begin{aligned} G(z) &= D + C(zI - A)^{-1}B \\ &:= \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \end{aligned}$$

with $A: n \times n$, $B: n \times m$, $C: p \times n$, and $D: p \times m$.

Matrices $(\tilde{M}(z), \tilde{N}(z)) \in \mathcal{H}_\infty^+$, where \mathcal{H}_∞^+ denotes the space of functions with all poles in the open unit disc of the complex plane, constitute a left coprime factorization of $G(z)$ if and only if

- (i) \tilde{M} is square, and $\det(\tilde{M}) \neq 0$
- (ii) the plant model is given by

$$G = \tilde{M}^{-1} \tilde{N} \quad (5.14)$$

- (iii) There exists $(\tilde{V}, \tilde{U}) \in \mathcal{H}_\infty^+$ such that

$$\tilde{M} \tilde{V} + \tilde{N} \tilde{U} = I_p \quad (5.15)$$

A left coprime factorization of G as defined in (5.14) is *normalized* if and only if

$$\tilde{N}(z) \tilde{N}^T \left(\frac{1}{z} \right) + \tilde{M}(z) \tilde{M}^T \left(\frac{1}{z} \right) = I_p \quad (5.16)$$

The concept of right coprime factorization and normalized right coprime factorization can be introduced similarly. However, the work presented here will follow the (normalized) left coprime factorization, although all results concerning the (normalized) right coprime factorization can be derived similarly.

State-space constructions for the normalized coprime factorizations can be obtained in terms of the solutions to the following two *discrete algebraic Riccati equations* (DARE):

$$\Phi^T P \Phi - P - \Phi^T P B Z_1 Z_1^T B^T P \Phi + C^T R_1^{-1} C = 0 \quad (5.17)$$

and

$$\Phi Q \Phi^T - Q - \Phi Q C^T Z_2^T Z_2 C Q \Phi^T + B R_2^{-1} B^T = 0 \quad (5.18)$$

where $R_1 = I_p + D D^T$, $R_2 = I_m + D^T D$, $\Phi = A - B R_2^{-1} D^T C$, $Z_1 Z_1^T = (R_2 + B^T P B)^{-1}$, $Z_2^T Z_2 = (R_1 + C Q C^T)^{-1}$. And, $P \geq 0$ and $Q \geq 0$ are the unique stabilizing solutions, respectively.

Without loss of generality, we may assume that both Z_1 and Z_2 are square matrices, and $Z_1 = Z_1^T$, $Z_2 = Z_2^T$.

Further, by defining $H = -(A Q C^T + B D^T) Z_2^T Z_2$ and $F = -Z_1 Z_1^T (B^T P A + D^T C)$, then

$$\begin{bmatrix} \tilde{N} & \tilde{M} \end{bmatrix} := \left[\begin{array}{c|c|c} A + H C & B + H D & H \\ \hline Z_2 C & Z_2 D & Z_2 \end{array} \right] \quad (5.19)$$

and

$$\begin{bmatrix} N \\ M \end{bmatrix} := \left[\begin{array}{c|c} A + B F & B Z_1 \\ \hline C + D F & D Z_1 \\ F & Z_1 \end{array} \right] \quad (5.20)$$

are the normalized left and right coprime factorizations of G , respectively.

5.3.2 Robust Controller Formulas

As in the continuous-time case, the discrete-time \mathcal{H}_∞ loop-shaping design procedure is based on the construction of a robust stabilizing controller against the perturbations on the coprime factors. The same form of cost function for robust stabilization will be derived, and the optimal achievable γ is given by

$$\gamma_o = (1 + \lambda_{\max}(QP))^{1/2} \quad (5.21)$$

where Q and P are the solutions to (5.17) and (5.18), respectively.

For a given $\gamma > \gamma_o$, a suboptimal \mathcal{H}_∞ LSDP controller K in the discrete-time case can be recast as a standard \mathcal{H}_∞ suboptimal control problem as was discussed in Sect. 4.4. The generalized (interconnected) system in this case is

$$\begin{aligned} P(z) &= \begin{bmatrix} P_{11}(z) & P_{12}(z) \\ P_{21}(z) & P_{22}(z) \end{bmatrix} \\ &= \left[\begin{array}{c|c} 0 & I_m \\ \hline \tilde{M}^{-1} & G \end{array} \right] \\ &= \left[\begin{array}{c|cc} A & -HZ_2^{-1} & B \\ \hline 0 & 0 & I_m \\ C & Z_2^{-1} & D \\ C & Z_2^{-1} & D \end{array} \right] \end{aligned} \quad (5.22)$$

Following the solution procedure given in Sect. 4.4, one more DARE needs to be solved in order to compute the required controller. In general \mathcal{H}_∞ suboptimal problems, two more algebraic Riccati equations are to be solved. Here, however, due to the structure of $P(z)$ in (5.22), it can be shown that the solution to one of the DARE is always zero. The third DARE is the following:

$$\begin{aligned} A^T X_\infty A - X_\infty - \tilde{F}^T \left(R + \begin{bmatrix} -Z_2^{-1} H^T \\ R_2^{-1/2} B^T \end{bmatrix} X_\infty \begin{bmatrix} -H Z_2^{-1} & B R_2^{-1/2} \end{bmatrix} \right) \tilde{F} \\ + C^T C = 0 \end{aligned} \quad (5.23)$$

where

$$\begin{aligned} \tilde{F} &= - \left(R + \begin{bmatrix} -Z_2^{-1} H^T \\ R_2^{-1/2} B^T \end{bmatrix} X_\infty \begin{bmatrix} -H Z_2^{-1} & B R_2^{-1/2} \end{bmatrix} \right)^{-1} \\ &\quad \times \left(\begin{bmatrix} -Z_2^{-1} C \\ D^T R_1^{-1/2} C \end{bmatrix} + \begin{bmatrix} -Z_2^{-1} H^T \\ R_2^{-1/2} B^T \end{bmatrix} X_\infty A \right) \end{aligned}$$

and

$$R = \begin{bmatrix} Z_2^{-2} - \gamma^2 I_p & Z_2^{-1} R_1^{-1/2} D \\ D^T R_1^{-1/2} Z_2^{-1} & I_m \end{bmatrix}$$

Further, by defining $\tilde{F} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$, where $F_1: p \times n$ and $F_2: m \times n$, a suboptimal \mathcal{H}_∞ discrete-time LSDP (DLSDP) controller K can be constructed as

$$K(z) = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array} \right]$$

where

$$\begin{aligned} A_K &= \hat{A}_K - \hat{B}_K D(I + \hat{D}_K D)^{-1} \hat{C}_K \\ B_K &= \hat{B}_K (I + D \hat{D}_K)^{-1} \\ C_K &= (I + \hat{D}_K D)^{-1} \hat{C}_K \\ D_K &= \hat{D}_K (I + D \hat{D}_K)^{-1} \end{aligned} \quad (5.24)$$

with

$$\begin{aligned} \hat{D}_K &= -(R_2 + B^T X_\infty B)^{-1} (D^T - B^T X_\infty H) \\ \hat{B}_K &= -H + B \hat{D}_K \\ \hat{C}_K &= R_2^{-1/2} F_2 - \hat{D}_K (C + Z_2^{-1} F_1) \\ \hat{A}_K &= A + HC + B \hat{C}_K \end{aligned} \quad (5.25)$$

5.3.3 The Strictly Proper Case

It may be appropriate to say that most (formulated) plants considered in the practical, discrete-time control-systems design are, in using the \mathcal{H}_∞ optimization approach in particular, strictly proper, i.e. $D = 0$. This is not only because most physical plants in industrial studies are strictly proper, as in the continuous-time case, but also because the \mathcal{H}_∞ controllers designed tend to be proper due to the “flatness” of the optimality sought in the synthesis. Hence, when the (formulated) plant is just proper, it is possible to encounter the problem of an algebraic loop in the implementation of the resultant controller.

When the plant under consideration is strictly proper, all the computations and formulas described in Sect. 5.3.2 will be significantly simpler. The two DAREs (5.17) and (5.18) become

$$A^T P A - P - A^T P B Z_1 Z_1^T B^T P A + C^T C = 0 \quad (5.26)$$

and

$$A Q A^T - Q - A Q C^T Z_2^T Z_2 C Q A^T + B B^T = 0 \quad (5.27)$$

where $Z_1 Z_1^T = (I_m + B^T P B)^{-1}$, $Z_2^T Z_2 = (I_p + C Q C^T)^{-1}$.

The third DARE (5.23) is now the following:

$$\begin{aligned} A^T X_\infty A - X_\infty - \tilde{F}^T \left(R + \begin{bmatrix} -Z_2^{-1} H^T \\ B^T \end{bmatrix} X_\infty \begin{bmatrix} -H Z_2^{-1} & B \end{bmatrix} \right) \tilde{F} \\ + C^T C = 0 \end{aligned} \quad (5.28)$$

where

$$\begin{aligned} \tilde{F} = & - \left(R + \begin{bmatrix} -Z_2^{-1} H^T \\ B^T \end{bmatrix} X_\infty \begin{bmatrix} -H Z_2^{-1} & B \end{bmatrix} \right)^{-1} \\ & \times \left(\begin{bmatrix} -Z_2^{-1} C \\ 0 \end{bmatrix} + \begin{bmatrix} -Z_2^{-1} H^T \\ B^T \end{bmatrix} X_\infty A \right) \end{aligned}$$

and

$$\begin{aligned} R &= \begin{bmatrix} Z_2^{-2} - \gamma^2 I_p & 0 \\ 0 & I_m \end{bmatrix} \\ H &= -A Q C^T Z_2^T Z_2 \end{aligned}$$

Further, by defining $\tilde{F} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$, where $F_1: p \times n$ and $F_2: m \times n$, the suboptimal \mathcal{H}_∞ DLSDP controller K in the case of a strictly proper G can be constructed as

$$K(z) = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array} \right]$$

where

$$\begin{aligned} D_K &= (I_m + B^T X_\infty B)^{-1} B^T X_\infty H \\ B_K &= -H + B D_K \\ C_K &= F_2 - D_K (C + Z_2^{-1} F_1) \\ A_K &= A + H C + B C_K \end{aligned} \quad (5.29)$$

5.3.4 On the Three DARE Solutions

As discussed above, the discrete-time \mathcal{H}_∞ LSPD suboptimal controller formulae require the solutions to the three discrete-time algebraic Riccati equations, (5.17), (5.18), and (5.23), or (5.26), (5.27), and (5.28) in the strictly proper case. In this subsection, we will show that there is a relation between these three solutions, namely the solution X_∞ to the third DARE can be calculated directly from the first two solutions P and Q . This fact is important and useful, especially in the numerical implementation of the DLSDP routines.

We start with a general DARE, hence the notations are not related to those defined earlier in the chapter. We have

$$F^T X F - X - F^T X G_1 (G_2 + G_1^T X G_1)^{-1} G_1^T X F + C^T C = 0 \quad (5.30)$$

where $F, H, X \in \mathcal{R}^{n \times n}$, $G_1 \in \mathcal{R}^{n \times m}$, $G_2 \in \mathcal{R}^{m \times m}$, and $G_2 = G_2^T > 0$. We assume that (F, G_1) is a stabilizable pair and that (F, C) is a detectable pair. We also define $G = G_1 G_2^{-1} G_1^T$.

It is well known [136] that solutions to DARE (5.30) are closely linked with a matrix pencil pair (M, L) , where

$$M = \begin{bmatrix} F & 0 \\ -H & I \end{bmatrix}, \quad L = \begin{bmatrix} I & G \\ 0 & F^T \end{bmatrix} \quad (5.31)$$

It also can be shown that if there exist $n \times n$ matrices S , U_1 and U_2 , with U_1 invertible, such that

$$M \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = L \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} S \quad (5.32)$$

then, $X = U_2 U_1^{-1}$ is a solution to (5.30). Further, the matrix $F - G_1 (G_2 + G_1^T X G_1)^{-1} G_1^T X F$ shares the same spectrum as S . Hence, if S is stable, i.e. all the eigenvalues are within the open unit disc, $F - G_1 (G_2 + G_1^T X G_1)^{-1} G_1^T X F$ is also stable. Such an X is non-negative definite and unique, and is called the stabilizing solution to (5.30).

Under the above assumptions on (5.30), it was shown in [136] that none of the generalized eigenvalues of the pair (M, L) lies on the unit circle, and if $\lambda \neq 0$ is a generalized eigenvalue of the pair, then $1/\lambda$ is also a generalized eigenvalue of the same multiplicity. In other words, the stable spectrum, consisting of n generalized eigenvalues lying in the open unit disc, is unique. Therefore, if there exists another triple (V_1, V_2, T) satisfying (5.32), with V_1 being invertible and T stable, then there must exist an invertible R such that $T = R^{-1} S R$. Consequently,

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} R^{-1} \quad (5.33)$$

The solution, of course, remains the same, since $X = V_2 V_1^{-1} = (U_2 R)(U_1 R)^{-1} = U_2 U_1^{-1}$.

In the present case, we can accordingly define the three matrix pencils as

$$\begin{aligned} M_P &= \begin{bmatrix} \Phi & 0 \\ -C^T R_1^{-1} C & I \end{bmatrix} \\ L_P &= \begin{bmatrix} I & B R_2^{-1} B^T \\ 0 & \Phi^T \end{bmatrix} \end{aligned} \quad (5.34)$$

$$\begin{aligned}
M_Q &= \begin{bmatrix} \Phi^T & 0 \\ -BR_2^{-1}B^T & I \end{bmatrix} \\
L_Q &= \begin{bmatrix} I & C^T R_1^{-1}C \\ 0 & \Phi \end{bmatrix}
\end{aligned} \tag{5.35}$$

$$\begin{aligned}
M_X &= \begin{bmatrix} A - [-HZ_2^{-1} & BR_2^{-1/2}]R^{-1} \begin{bmatrix} Z_2^{-1}C \\ D^T R_1^{-1/2}C \end{bmatrix} & 0 \\ -C^T C + [C^T Z_2^{-1} & C^T R_1^{-1/2}D]R^{-1} \begin{bmatrix} Z_2^{-1}C \\ D^T R_1^{-1/2}C \end{bmatrix} & I \end{bmatrix} \\
L_X &= \begin{bmatrix} I & [-HZ_2^{-1} & BR_2^{-1/2}]R^{-1} \begin{bmatrix} -Z_2^{-1}H^T \\ R_2^{-1/2}B^T \end{bmatrix} \\ 0 & A^T - [C^T Z_2^{-1} & C^T R_1^{-1/2}D]R^{-1} \begin{bmatrix} -Z_2^{-1}H^T \\ R_2^{-1/2}B^T \end{bmatrix} \end{bmatrix}
\end{aligned} \tag{5.36}$$

With all the above properties of the DAREs and the notations, the following theorem may be proved [68].

Theorem 5.1 *Let P , Q and X_∞ be the stabilizing solutions to the DAREs (5.17), (5.18) and (5.23), (or, (5.26), (5.27) and (5.28) when G is strictly proper), respectively, the following identity holds:*

$$\begin{aligned}
X_\infty &= P[(1 - \gamma^{-2})I_n - \gamma^{-2}QP]^{-1} \\
&= \gamma^2 P[\gamma^2 I_n - (I_n + QP)]^{-1}
\end{aligned} \tag{5.37}$$

A similar result can be found in [173] for the relationship between three discrete-time algebraic Riccati equations arising in the general \mathcal{H}_∞ suboptimal design. The results concerning the discrete-time loop-shaping design procedure are with three different DAREs and the conclusion is of a slightly different form.

5.4 A Mixed Optimization Design Method with LSDP

It is well known that control-system design problems can be naturally formulated as constrained optimization problems, the solutions of which will characterize acceptable designs. The numerical optimization approach to controller design can directly tackle design specifications in both the frequency domain and the time domain. The optimization problems derived, however, are usually very complicated with many unknowns, many nonlinearities, many constraints, and in most cases they are multiobjective with several conflicting design aims that need to be simultaneously achieved. It is also known that a direct parameterization of the controller will increase the complexity of the optimization problem.

On the other hand, the \mathcal{H}_∞ loop-shaping design procedure (LSDP) discussed in this chapter is an efficient, robust design method that just depends on appropriate choice of weighting functions to shape the open-loop system. In general, the weighting functions are of low orders. This indicates if we take the weighting functions as design parameters it may well combine the advantages of both the numerical optimization approach and the \mathcal{H}_∞ LSDP in terms of flexible formulation of design objectives, mature numerical algorithms, tractable optimization problems, guaranteed stability and a certain degree of robustness. Such a design method is proposed in [178]. The numerical optimization component of the method is based on the *method of inequalities* (MOI).

Performance specifications for control-systems design are frequently, and more naturally, given in terms of algebraic or functional inequalities, rather than in the minimization of some objective function. For example, the system may be required to have a rise time of less than one second, a settling time of less than five seconds and an overshoot of less than 10 %. In such cases, it is obviously more logical and convenient if the design problem is expressed explicitly in terms of such inequalities. The method of inequalities [191] is a computer-aided multiobjective design approach, where the desired performance is represented by such a set of algebraic inequalities and where the aim of the design is to simultaneously satisfy these inequalities. The design problem is expressed as

$$\phi_i(p) \leq \epsilon_i, \quad \text{for } i = 1, \dots, n \quad (5.38)$$

where ϵ_i are real numbers, $p \in P$ is a real vector (p_1, p_2, \dots, p_q) chosen from a given set P , and ϕ_i are real functions of p . The functions ϕ_i are performance indices, the components of p represent the design parameters, and ϵ_i are chosen by the designer and represent the largest tolerable values of ϕ_i . The aim is the satisfaction of the set of inequalities so that an acceptable design p is reached.

For control-systems design, the functions $\phi_i(p)$ may be functionals of the system step response, for example, the rise-time, overshoot or the integral absolute tracking error, or functionals of the frequency responses, such as the bandwidth. They can also represent measures of the system stability, such as the maximum real part of the closed-loop poles. Additional inequalities that arise from the physical constraints of the system can also be included, to restrict, for example, the maximum control signal. In practice, the constraints on the design parameters p that define the set P are also included in the inequality set, e.g. to constrain the possible values of some of the design parameters or to limit the search to stable controllers only.

Each inequality $\phi_i(p) \leq \epsilon_i$ of the set of inequalities (5.38) defines a set \mathcal{S}_i of points in the q -dimensional space \mathcal{R}^q and the coordinates of this space are p_1, p_2, \dots, p_q , so

$$\mathcal{S}_i = \{p : \phi_i(p) \leq \epsilon_i\} \quad (5.39)$$

The boundary of this set is defined by $\phi_i(p) = \epsilon_i$. A point $p \in \mathcal{R}^q$ is a solution to the set of inequalities (5.38) if and only if it lies inside every set $\mathcal{S}_i, i = 1, 2, \dots, n$,

and hence inside the set \mathcal{S} that denotes the intersection of all the sets \mathcal{S}_i :

$$\mathcal{S} = \bigcap_{i=1}^n \mathcal{S}_i \quad (5.40)$$

\mathcal{S} is called the admissible set and any point p in \mathcal{S} is called an admissible point denoted p_s .

The objective is thus to find a point p such that $p \in \mathcal{S}$. Such a point satisfies the set of inequalities (5.38) and is said to be a solution. In general, a point p_s is not unique unless the set \mathcal{S} is a singleton in the space \mathcal{R}^d . In some cases, there is no solution to the problem, i.e. \mathcal{S} is an empty set. It is then necessary to relax the boundaries of some of the inequalities, i.e. increase some of the numbers ϵ_i , until an admissible point p_s is found.

The actual solution to the set of inequalities (5.38) may be obtained by means of numerical search algorithms, such as the moving-boundaries process (MBP); details of the MBP may be found in [176, 191]. The procedure for obtaining a solution is interactive, in that it requires supervision and intervention from the designer. The designer needs to choose the configuration of the design, which determines the dimension of the design parameter vector p , and initial values for the design parameters. The progress of the search algorithm should be monitored and, if a solution is not found, the designer may either change the starting point, relax some of the desired bounds ϵ_i , or change the design configuration. Alternatively, if a solution is found easily, to improve the quality of the design, the bounds could be tightened or additional design objectives could be included in (5.38). The design process is thus a two-way process, with the MOI providing information to the designer about conflicting requirements, and the designer making decisions about the “trade-offs” between design requirements based on this information as well as on the designer’s knowledge, experience, and intuition about the particular problem. The designer can be supported in this role by various graphical displays [131] or expert systems [66].

A difficult problem in control-systems design using the MOI method is how to define the design parameter vector p . A straightforward idea is to directly parametrize the controller, i.e. let p be the system matrices of the controller, or the coefficients of the numerator and denominator polynomials of the controller. In doing so, the designer has to choose the order of the controller first. In general, the lower the dimension of the design vector p , the easier it is for numerical search algorithms to find a solution, if one exists. Choosing a low-order controller, say a PI controller $p_1 + \frac{p_2}{s}$, may reduce the dimension of p . However, it may not yield a solution and, in that case, a solution may exist with higher order controllers. A further limitation of using the MOI alone in the design is that an initial point that gives the stability of the closed-loop system must be located as a prerequisite to searching the parameter space in order to improve the index set of (5.38).

Two aspects of the \mathcal{H}_∞ LSDP make it amenable to combine this approach with the MOI. First, unlike the standard \mathcal{H}_∞ -optimization approaches, the \mathcal{H}_∞ -optimal controller for the weighted plant can be synthesized from the solutions of two algebraic Riccati equations (5.5) and (5.7) and does not require time-consuming γ -

iteration. Second, in the LSDP described earlier in this chapter, the weighting functions are chosen by considering the open-loop response of the weighted plant, so effectively the weights W_1 and W_2 are the design parameters. This means that the design problem can be formulated as in the method of inequalities, with the parameters of the weighting functions used as the design parameters p to satisfy the set of closed-loop performance inequalities. Such an approach to the MOI overcomes the limitations of the MOI. The designer does not have to choose the order of the controller, but instead chooses the order of the weighting functions. With low-order weighting functions, high-order controllers can be synthesized that often lead to significantly better performance or robustness than if simple low-order controllers were used. Additionally, the problem of finding an initial point for system stability does not exist, because the stability of a closed-loop system is guaranteed through the solution to the robust stabilization problem, provided that the weighting functions do not cause undesirable pole/zero cancellations.

The design problem is now stated as follows.

Problem For the system of Fig. 5.2, find a (W_1, W_2) such that

$$\gamma_o(W_1, W_2) \leq \epsilon_\gamma \quad (5.41)$$

and

$$\phi_i(W_1, W_2) \leq \epsilon_i \quad \text{for } i = 1, \dots, n \quad (5.42)$$

where

$$\gamma_o = [1 + \lambda_{\max}(ZX)]^{1/2}$$

with Z and X the solutions of the two AREs, (5.5) and (5.7), of the weighted plant.

In the above formulation, ϕ_i are functionals representing design objectives, ϵ_γ and ϵ_i are real numbers representing desired bounds on γ_o and ϕ_i , respectively, and (W_1, W_2) a pair of fixed order weighting functions with real parameters $w = (w_1, w_2, \dots, w_q)$.

Consequently, a design procedure can be stated as follows.

Design Procedure A design procedure to solve the above problem is

1. Define the plant G , and define the functionals ϕ_i .
2. Define the values of ϵ_γ and ϵ_i .
3. Define the form and order of the weighting functions W_1 and W_2 . Bounds should be placed on the values of w_j to ensure that W_1 and W_2 are stable and minimum phase to prevent undesirable pole/zero cancellations. The order of the weighting functions, and hence the value of q , should be small initially.
4. Define initial values of w_j based on the open-loop frequency response of the plant.

5. Implement the MBP, or other appropriate algorithms, in conjunction with (5.12) or (5.13) to find a (W_1, W_2) that satisfies inequalities (5.41) and (5.42). If a solution is found, the design is satisfactory; otherwise, either increase the order of the weighting functions, or relax one or more of the desired bounds ϵ , or try again with different initial values of w .
6. With satisfactory weighting functions W_1 and W_2 , a satisfactory feedback controller is obtained from (5.12) or (5.13).

This mixed optimization design approach has been applied in some design cases. Examples of application and some extensions can be found in [124, 160, 177–181].

Chapter 6

μ -Analysis and Synthesis

As discussed earlier in the book, the \mathcal{H}_∞ optimization approach may achieve robust stabilization against unstructured system perturbations and nominal performance requirements. It is though possible that by applying appropriate weighting functions some robust performance requirements can be obtained. Satisfactory designs have been reported, in particular when using the \mathcal{H}_∞ loop-shaping design methods. In order to achieve robust stability **and** robust performance, design methods based on the structured singular value μ can be used. In this chapter, we first show how a robust performance design specification for a control system with unstructured/structured perturbations can be transformed into a robust stabilization problem with regard to structured perturbation. We then discuss how these design specifications are related to some indications of this new setting, followed by introduction of two iterative μ -synthesis methods, the D - K iteration and μ - K iteration methods.

We recall, as defined in Chap. 3, for $M \in \mathcal{C}^{n \times n}$ and a known structure of Δ (usually representing uncertainties),

$$\Delta = \left\{ \text{diag}[\delta_1 I_{r_1}, \dots, \delta_s I_{r_s}, \Delta_1, \dots, \Delta_f] : \delta_i \in \mathcal{C}, \Delta_j \in \mathcal{C}^{m_j \times m_j} \right\} \quad (6.1)$$

where $\sum_{i=1}^s r_i + \sum_{j=1}^f m_j = n$ with n is the dimension of the block Δ , the structure singular value μ is defined by

$$\mu_{\Delta}^{-1}(M) := \min_{\Delta \in \Delta} \left\{ \overline{\sigma}(\Delta) : \det(I - M\Delta) = 0 \right\} \quad (6.2)$$

If there is no $\Delta \in \Delta$ such that $\det(I - M\Delta) = 0$, then $\mu_{\Delta}(M) := 0$.

Some important properties of μ have been listed in Chap. 3.

6.1 Consideration of Robust Performance

When M is an interconnected transfer function matrix $M(s)$ formed with respect to the uncertainty set Δ , the structured singular value of $M(s)$

Fig. 6.1 Standard M – Δ configuration

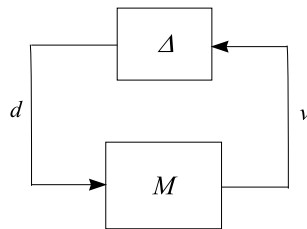
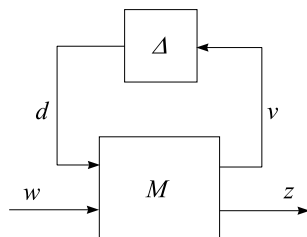


Fig. 6.2 Standard M – Δ configuration for RP analysis



$$\mu_{\Delta}(M(s)) := \sup_{\omega \in \mathcal{R}} \mu_{\Delta}(M(j\omega)) \quad (6.3)$$

indicates the robust stability of the perturbed system. Without loss of generality, we assume that the uncertainties have been normalized in the rest of this chapter, i.e. $\Delta \in \mathbf{B}\mathbf{A}$. From Theorem 3.4, the standard configuration Fig. 6.1 is robustly stable if $M(s)$ is stable and $\mu_{\Delta}(M(s)) < 1$ (or $\|M\|_{\mu} < 1$).

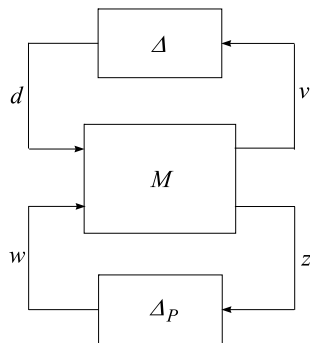
As discussed in Chap. 3 in design of control systems, in addition to the stability the performance of the system must be taken into account. The designed system should perform well (for instance, good tracking) against exogenous disturbances, which requires a closed-loop structure. The feedback controller is usually designed based on a nominal plant model. In the case of the existence of plant perturbations, the closed-loop system may well possibly perform badly, even degrade to an unsatisfactory level. The *robust performance* requires that a designed control system maintains a satisfactory performance level even in the presence of plant dynamic uncertainties.

We now expand Fig. 6.1 to include input and output signals, as depicted in Fig. 6.2.

In Fig. 6.2, w , z , v and d are usually vector signals. w denotes the exogenous input typically including command signals, disturbances, noises, etc.; z denotes the error output usually consisting of regulator output, tracking errors, filtered actuator signals, etc.; v and d are the input and output signals of the dynamic uncertainties. System-performance specifications can usually be interpreted as reduction of z with respect of w . With the assumption that w and z are both energy-bounded signals, the performance requirement is equivalent to the minimization of the \mathcal{H}_{∞} norm of the transfer function from w to z . Let M be partitioned accordingly as

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

Fig. 6.3 Standard $M-\Delta$ configuration with Δ_p analysis



it can easily be derived that

$$\begin{aligned} z &= [M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12}]w \\ &= F_u(M, \Delta)w \end{aligned} \quad (6.4)$$

Using normalization, a satisfactory level of performance requirement can be set as

$$\|F_u(M, \Delta)\|_\infty < 1 \quad (6.5)$$

Equation (6.5) implies the stability of $F_u(M, \Delta)$, which means robust stability with respect to the plant perturbations Δ .

Condition (6.5) is equivalent to the system loop in Fig. 6.3 to be robustly stable with regard to a fictitious uncertainty block Δ_p . Δ_p is unstructured with appropriate dimensions, satisfies $\|\Delta_p\|_\infty \leq 1$ and is usually called the performance uncertainty block. The part enclosed in the dotted line is, of course, $F_u(M, \Delta)$.

For *robust performance*, (6.5) should hold for all $\Delta \in \mathbf{B}\mathbf{\Delta}$. Based on Fig. 6.3, the robust performance design as well as robust stabilization against Δ , can be equivalently considered as a robust stabilization problem in Fig. 6.1 with the uncertainty block to be replaced by $\tilde{\Delta}$ where

$$\tilde{\Delta} \in \tilde{\mathbf{\Delta}} := \{\text{diag}\{\Delta, \Delta_p\} : \Delta \in \mathbf{B}\mathbf{\Delta}, \|\Delta_p\|_\infty \leq 1\}$$

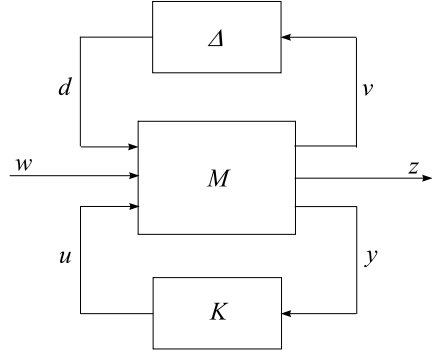
This is thus a robust stabilization problem with respect to a structured uncertainty $\tilde{\Delta}$. And, we have the following indications.

- *Robust Performance (RP)* $\iff \|M\|_\mu < 1$, for structured uncertainty $\tilde{\mathbf{\Delta}}$
- *Robust Stability (RS)* $\iff \|M_{11}\|_\mu < 1$, for structured uncertainty $\mathbf{B}\mathbf{\Delta}$
- *Nominal Performance (NP)* $\iff \|M_{22}\|_\infty < 1$
- *Nominal Stability (NS)* $\iff M$ is internally stable

Of course, if the uncertainty Δ is unstructured, then the robust stability requirement corresponds to $\|M_{11}\|_\infty < 1$.

It is easy to see that, by setting $\Delta_p \equiv 0$, $\|M\|_\mu < 1$ implies $\|M_{11}\|_\mu < 1$. Hence, the former condition yields a *robust stability and robust performance* (RSRP) design.

Fig. 6.4 Standard M – Δ configuration with K



6.2 μ -Synthesis: D – K Iteration Method

The transfer function matrix $M(s)$ in Fig. 6.2 usually contains the controller K , hence a closed-loop interconnected matrix. For the purpose of controller design, we may rearrange the configuration and explicitly show the dependence of the closed-loop system on the controller K . That is, Fig. 6.2 is again shown in Fig. 6.4.

The nominal, open-loop interconnected transfer function matrix $P(s)$ in Fig. 6.4 does not include the controller $K(s)$ nor any perturbations that are under consideration. $P(s)$ may be partitioned as

$$P(s) = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix}$$

The signals y and u represent the feedback signals (measured output, tracking error, etc., the input to the controller) and the control signal (the output from the controller), respectively.

The relationship between M in Fig. 6.2 and P can be obtained by straightforward calculations as

$$\begin{aligned} M(P, K) &= F_l(P, K) \\ &= \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} + \begin{bmatrix} P_{13} \\ P_{23} \end{bmatrix} K (I - P_{33}K)^{-1} \begin{bmatrix} P_{31} & P_{32} \end{bmatrix} \end{aligned} \quad (6.6)$$

where $M(s)$ is explicitly written as $M(P, K)$ to show that M is formed by P and K .

For the RSRP design, it is required to find a stabilizing controller K such that

$$\sup_{\omega \in \mathcal{R}} \mu[M(P, K)(j\omega)] < 1 \quad (6.7)$$

where the subscript $\tilde{\Delta}$ has been suppressed for the sake of simplicity.

Or, for the “optimal” RSRP design, the objective is to solve for K

$$\inf_{K(s)} \sup_{\omega \in \mathcal{R}} \mu[M(P, K)(j\omega)] \quad (6.8)$$

An iterative method was proposed to solve (6.8) in [32]. The method is called the D - K iteration μ -synthesis method, and is based on solving the following optimization problem, for a stabilizing controller K and a diagonal constant scaling matrix D :

$$\inf_{K(s)} \sup_{\omega \in \mathcal{R}} \inf_{D \in \mathbf{D}} \bar{\sigma}[DM(P, K)D^{-1}(j\omega)] \quad (6.9)$$

where the scaling matrix set \mathbf{D} is defined in Sect. 3.3. The justification for using (6.9) is obvious, from Theorem 3.7 and the discussion afterwards, with $\mu[M(P, K)]$ replaced by its upper bound $\inf_{D \in \mathbf{D}} \bar{\sigma}[DM(P, K)D^{-1}]$.

Corresponding to the case of (6.7), a stabilizing controller is to be found such that

$$\sup_{\omega \in \mathcal{R}} \inf_{D \in \mathbf{D}} \bar{\sigma}[DM(P, K)D^{-1}(j\omega)] < 1 \quad (6.10)$$

The D - K iteration method is to alternately minimize (6.9), or to reduce the left-hand-side value of (6.10), for K and D in turn while keeping the other one fixed.

For a given matrix D , either constant or transfer function, (6.9) is a standard \mathcal{H}_∞ optimization problem

$$\inf_{K(s)} \|DM(P, K)D^{-1}\|_\infty \quad (6.11)$$

that can be further written as

$$\inf_K \|DF_l(P, K)D^{-1}\|_\infty = \inf_K \|F_l(\tilde{P}, K)\|_\infty \quad (6.12)$$

with $\tilde{P} = \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} D^{-1} & 0 \\ 0 & I \end{bmatrix}$ compatible with the partition of P .

On the other hand, for a fixed $K(s)$, $\inf_{D \in \mathbf{D}} \bar{\sigma}[DM(P, K)D^{-1}(j\omega)]$ is a convex optimization problem at each frequency ω . After the minimization over a range of frequency of interest, the resultant diagonal (constant) matrices D s can be approximated, via curve fitting, by a stable, minimum phase, rational transfer function matrix $D(s)$, which will be used in the next iteration for K .

The D - K iterative μ -synthesis algorithm is thus:

Step 1: Start with an initial guess for D , usually set $D = I$.

Step 2: Fix D and solve the \mathcal{H}_∞ -optimization for K ,

$$K = \arg \inf_K \|F_l(\tilde{P}, K)\|_\infty$$

Step 3: Fix K and solve the following convex optimization problem for D at each frequency over a selected frequency range:

$$D(j\omega) = \arg \inf_{D \in \mathbf{D}} \bar{\sigma}[DF_l(P, K)D^{-1}(j\omega)]$$

Step 4: Curve fit $D(j\omega)$ to get a stable, minimum-phase $D(s)$; go to Step 2 and repeat, until a prespecified convergence tolerance or (6.10) is achieved, or a prespecified maximum iteration number is reached.

Successful applications of the D – K iteration method have been reported. This method has also been applied in the case studies in Part II of this book. It should be noticed, however, that the algorithm may not converge in some cases. The resultant controller may be very conservative, particularly in the case of real and/or mixed perturbations, due to the possible gap between $\mu(M)$ and $\inf_{D \in \mathbf{D}} \bar{\sigma}[DM D^{-1}]$ in general cases, and due to the lack of powerful algorithms to compute the real and mixed μ values and to solve the optimization problem for D . Another adverse effect in practical designs is that the order of resultant μ -controllers is usually very high, and hence controller-order reduction must be applied.

6.3 μ -Synthesis: μ – K Iteration Method

As discussed above, the structured singular value μ plays an important role in the robust stability and robust performance designs. The μ -design is to seek a stabilizing controller that either minimizes or reduces the μ value over a frequency range. There is another design method proposed in the literature that can be applied to find a μ -controller. That is the so-called μ – K iteration method [99, 101, 139].

It is shown in [72] that in many optimization-based control-system designs, the resultant, optimal controller will make a “well-posed” cost function (i.e. satisfying certain assumptions [72], Theorem 4.1) *constant* in the frequency ω almost everywhere. This feature is obvious in the \mathcal{H}_∞ optimizations and can also be observed in the μ -designs using the D – K iterations where the controller $K(s)$ obtained in the iteration gradually flattens the curve of $\mu(M)$. The μ – K iteration method is motivated by the above. In the μ – K iterations, the obtained μ curves are used as weighting functions on the \mathcal{H}_∞ cost function $\|F_l(\tilde{P}, K)\|_\infty$ with the aim to suppress the peak values of the μ curves in the successive iterations. An algorithm to implement the μ – K iteration method is described below.

Step 1: Solve the \mathcal{H}_∞ -optimization problem for K_0 ,

$$K_0 := \arg \inf_K \|F_l(\tilde{P}, K)\|_\infty$$

Step 2: Compute the μ curve corresponding to K_0 over a chosen frequency range,

$$\mu_0(j\omega) := \mu[F_l(P, K_0)(j\omega)]$$

Step 3: Normalize μ_0 by its maximum value, i.e.

$$\tilde{\mu}_0 := \frac{\mu_0}{\max_\omega \mu_0}$$

Step 4: Curve fit $\tilde{\mu}_0(j\omega)$ to get a stable, minimum-phase, rational $\tilde{\mu}_0(s)$.

Step 5: Solve for the \mathcal{H}_∞ optimal controller $K_1(s)$,

$$K_1 := \arg \inf_K \left\| \tilde{\mu}_0 F_l(\tilde{P}, K) \right\|_\infty \quad (6.13)$$

go to Step 2, multiply the newly obtained μ curve function onto the previous cost function in (6.13) (e.g. $\|\tilde{\mu}_1 \tilde{\mu}_0 F_l(\tilde{P}, K)\|_\infty$); repeat until the μ curve is sufficiently flat or until a desired level of performance (measured by the peak value of μ) has been achieved.

This page intentionally left blank

Chapter 7

Lower-Order Controllers

There is a dilemma concerning design of control systems. Due to increasing demands on quality and productivity of industrial systems and with deeper understanding of these systems, mathematical models derived to represent the system dynamics are more complete, usually of multi-input-multi-output form, and are of high orders. Consequently, the controllers designed are complex. The order of such controllers designed using, for instance, the \mathcal{H}_∞ optimization approach or the μ -method, is higher than, or at least similar to, that of the plant. On the other hand, in the implementation of controllers, high-order controllers will lead to high cost, difficult commissioning, poor reliability and potential problems in maintenance. Lower-order controllers are always welcomed by practicing control engineers. Hence, how to obtain a low-order controller for a high-order plant is an important and interesting task, and is the subject of the present chapter.

In general, there are three directions to obtain a lower-order controller for a relatively high-order plant, as depicted in Fig. 7.1,

- (1) plant model reduction followed by controller design;
- (2) controller design followed by controller-order reduction; and,
- (3) direct design of low-order controllers.

Approaches (1) and (2) are widely used and can be used together. When a controller is designed using a robust design method, Approach (1) would usually produce a stable closed loop, though the reduction of the plant order is likely to be limited. In Approach (2), there is freedom in choosing the final order of the controller, but the stability of the closed-loop system should always be verified.

The third approach usually would heavily depend on some properties of the plant, and require numerous computations. This approach will not be introduced in this book. Interested readers are referred to [14, 23, 65, 76, 80].

From the viewpoint of being a control system, model reductions (usually referred to the reduction of orders of the original plant models) and controller reductions are similar. Indeed, most methods introduced in this chapter are applicable in both cases, though some methods may be more suitable for one of them. This will be discussed along with the introduction of methods.

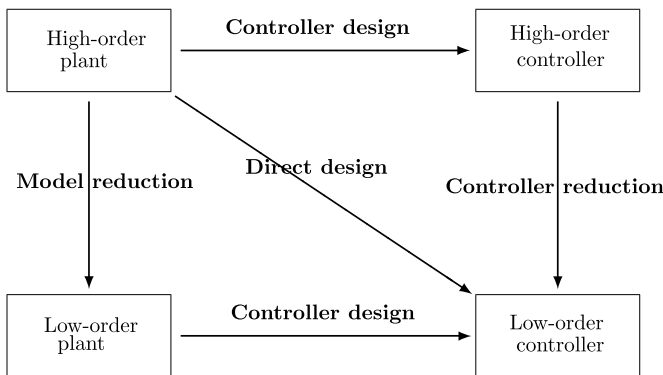


Fig. 7.1 Diagram for design of low-order controllers

In this book, only details of continuous-time case reduction will be discussed. Where the method is applicable in the discrete-time case as well references will be given. When we use the phrase order reductions of plants/systems, we actually refer to the order reduction of a model of the plant/system.

In the rest of the chapter, let the original system have a state-space representation

$$G(s) := C(sI - A)^{-1}B + D \quad (7.1)$$

$$=: \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \quad (7.2)$$

where $A: n \times n$, $B: n \times m$, $C: p \times n$, $D: p \times m$, and $[A, B, C]$ is assumed to be a minimal realization. A reduced-order system $G_r(s)$ is represented by

$$G_r(s) := C_r(sI - A_r)^{-1}B_r + D_r \quad (7.3)$$

$$=: \left[\begin{array}{c|c} A_r & B_r \\ \hline C_r & D_r \end{array} \right] \quad (7.4)$$

with $A_r: r \times r$, $B_r: r \times m$, $C_r: p \times r$, $D_r: p \times m$, and $r < n$.

7.1 Absolute-Error Approximation Methods

The aim of the following methods is to find a reduced-order system $G_r(s)$ such that some norm of the error system is small, i.e. to minimize

$$\|G(s) - G_r(s)\| \quad (7.5)$$

Three methods are to be introduced: Balanced Truncation, Singular Perturbation Approximation (Balanced Residualization), and Hankel-Norm Approximation. These methods are for *stable* systems only. The case of unstable systems will be discussed at the end of this section.

7.1.1 Balanced Truncation Method

The general idea of truncation methods is to neglect those parts of the original system that are less observable and/or less controllable. Hopefully, this would lead to a system that is of lower order and retains the important dynamic behavior of the original system. However, in some systems, a mode would be weakly observable but highly controllable, or *vice versa*. To delete such a mode may be inappropriate with regard to the whole characteristics of the system. Hence, in the balanced truncation method [38, 39, 120, 138], a state similarity transformation is applied first to “balance” the controllability and observability features of the system.

A stable system $G(s)$ is called *balanced* if the solutions P and Q to the following Lyapunov equations:

$$AP + PA^T + BB^T = 0 \quad (7.6)$$

$$A^T Q + QA + C^T C = 0 \quad (7.7)$$

are such that $P = Q = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n) := \Sigma$, with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$. P and Q are called the controllability Gramian and observability Gramian, respectively. When the system is balanced, both Gramians are diagonal and equal. σ_i , $i = 1, \dots, n$, is the i th Hankel singular value of the system.

For a general system not in the balanced realization form, the following algorithm [92] can be applied.

Balanced Realization Algorithm

Step 1: Calculate the Gramians P and Q from (7.6) and (7.7), respectively

Step 2: Calculate a Cholesky factor R of Q , i.e. $Q = R^T R$

Step 3: Form a positive-definite matrix RPR^T and diagonalize it,

$$RPR^T = U \Sigma^2 U^T$$

where U is an orthonormal matrix, $U^T U = I$, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$

Step 4: Let $T = \Sigma^{-\frac{1}{2}} U^T R$. $[T, T^{-1}]$ is a required state similarity transformation (balancing transformation). That is, $[TAT^{-1}, TB, CT^{-1}]$ is a balanced realization.

We now assume that the state-space model of the original system $G(s)$, $[A, B, C, D]$, is already in the balanced realization form. Assume $\Sigma = \text{diag}(\Sigma_1, \Sigma_2)$, with $\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_r)$, $\Sigma_2 = \text{diag}(\sigma_{r+1}, \dots, \sigma_n)$, where $\sigma_r > \sigma_{r+1}$. The matrices A , B , and C can be partitioned compatibly as $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, $B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$, and $C = \begin{bmatrix} C_1 & C_2 \end{bmatrix}$. Then, a reduced-order system $G_r(s)$ can be defined by

$$G_r(s) = C_1(sI - A_{11})^{-1} B_1 + D = \left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & D \end{array} \right] \quad (7.8)$$

Such a $G_r(s)$ is of r th order and is called a balanced truncation of the full order (n th) system $G(s)$. It can be shown that $G_r(s)$ is stable, in the balanced realization form, and

$$\|G(s) - G_r(s)\|_\infty \leq 2 \operatorname{tr}(\Sigma_2) \quad (7.9)$$

where $\operatorname{tr}(\Sigma_2)$ denotes the trace of the matrix Σ_2 , i.e. $\operatorname{tr}(\Sigma_2) = \sigma_{r+1} + \cdots + \sigma_n$, the sum of the last $(n - r)$ Hankel singular values [39, 55].

In most applications, to reduce the original system into an r th-order system there should be a large gap between σ_r and σ_{r+1} , i.e. $\sigma_r \gg \sigma_{r+1}$.

7.1.2 Singular Perturbation Approximation

In many engineering systems, the steady-state gain of a system, usually called *dc-gain* (the system gain at infinitive time, equivalent to $G(0)$), plays an important role in assessing system performances. It is thus better to maintain the dc gain in a reduced-order model, i.e. $G_r(0) = G(0)$. The Balanced Truncation method introduced in the last subsection does not keep the dc gain unchanged in the reduced-order system. The singular perturbation approximation method [103] (or, balanced residualization method [149]) presented below does retain the dc gain.

Assume that $[A, B, C, D]$ is a minimal and balanced realization of a stable system $G(s)$, and partitioned compatibly as in the previous subsection. It can be shown that A_{22} is stable (e.g. see Theorem 4.2 of [55]), and thus invertible. Define

$$A_r = A_{11} - A_{12}A_{22}^{-1}A_{21} \quad (7.10)$$

$$B_r = B_1 - A_{12}A_{22}^{-1}B_2 \quad (7.11)$$

$$C_r = C_1 - C_2A_{22}^{-1}A_{21} \quad (7.12)$$

$$D_r = D - C_2A_{22}^{-1}B_2 \quad (7.13)$$

A reduced-order system $G_r(s)$ defined by

$$G_r(s) = C_r(sI - A_r)^{-1}B_r + D_r \quad (7.14)$$

is called a singular perturbation approximation (or, balanced residualization) of $G(s)$. It is a straightforward computation to show that the dc-gain remains unchanged, i.e.

$$-CA^{-1}B + D = -C_rA_r^{-1}B_r + D_r \quad (7.15)$$

by noting that

$$\begin{bmatrix} I & 0 \\ -A_{22}^{-1}A_{21} & I \end{bmatrix} \begin{bmatrix} I & 0 \\ A_{22}^{-1}A_{21} & I \end{bmatrix} = I \quad (7.16)$$

$$\begin{bmatrix} I & A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix} = I \quad (7.17)$$

and

$$\left(\begin{bmatrix} I & -A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{22}^{-1}A_{21} & I \end{bmatrix} \right)^{-1} = \begin{bmatrix} A_r^{-1} & 0 \\ 0 & A_{22}^{-1} \end{bmatrix} \quad (7.18)$$

It can also be shown that such a reduction $G_r(s)$ is a stable and balanced realization [149] and enjoys the same error bound as the balanced truncation method, i.e.

$$\|G(s) - G_r(s)\|_{\infty} \leq 2(\sigma_{r+1} + \cdots + \sigma_n)$$

It can be seen that instead of discarding the “less important” part totally as in the balanced truncation method, the derivative of x_2 in the following equation is set to zero, in the singular perturbation approximation (balanced residualization) method,

$$\dot{x}_2 = A_{21}X_1 + A_{22}x_2 + B_2u \quad (7.19)$$

x_2 is then solved in (7.19) in terms of x_1 and u , and is substituted as residual into the state equation of x_1 and output equation to obtain the reduced-order system $G_r(s)$ as given above.

This idea resembles what happens in analysis of singular perturbation systems with

$$\epsilon \dot{x}_2 = A_{21}X_1 + A_{22}x_2 + B_2u$$

where $0 < \epsilon \ll 1$, and hence the term of singular perturbation approximation.

7.1.3 Hankel-Norm Approximation

For a stable system $G(s)$ with Hankel singular values defined in Sect. 7.1.1, the largest Hankel singular value σ_1 is defined as the Hankel-norm of $G(s)$ [55]. The Hankel-norm denotes the largest possible \mathcal{L}_2 -gain from past inputs to future outputs. In some order-reduction cases, minimization of the Hankel-norm of the error system is more appropriate and thus required.

Approximation 1 Let $G(s)$ represent a stable and square system with a state-space model $[A, B, C, D]$ of minimal and balanced realization. Let the Gramians be $P = Q = \text{diag}(\Sigma_1, \sigma_l I_l)$, where σ is the smallest Hankel singular value with multiplicity l and every diagonal element of Σ_1 is larger than σ . Let $[A, B, C]$ be partitioned compatibly. An $(n - l)$ -th-order system $G_h(s)$ can be constructed as follows. Define

$$\hat{A} = \Gamma^{-1}(\sigma^2 A_{11}^T + \Sigma_1 A_{11} \Sigma_1 - \sigma C_1^T U B_1^T) \quad (7.20)$$

$$\hat{B} = \Gamma^{-1}(\Sigma_1 B_1 + \sigma C_1^T U) \quad (7.21)$$

$$\hat{C} = C_1 \Sigma_1 + \sigma U B_1^T \quad (7.22)$$

$$\hat{D} = D - \sigma U \quad (7.23)$$

where U is an orthonormal matrix satisfying

$$B_2 = -C_2^T U \quad (7.24)$$

and

$$\Gamma = \Sigma_1^2 - \sigma^2 I \quad (7.25)$$

The reduced-order system $G_h(s)$ is defined as

$$G_h(s) = \hat{C}(sI - \hat{A})^{-1} \hat{B} + \hat{D} \quad (7.26)$$

It is shown in [55] that the $(n - l)$ th-order $G_h(s)$ is stable and is an optimal approximation of $G(s)$ satisfying

$$\|G(s) - G_h(s)\|_H = \sigma \quad (7.27)$$

And, it is also true that $G(s) - G_h(s)$ is *all-pass* with the inf-norm

$$\|G(s) - G_h(s)\|_\infty = \sigma \quad (7.28)$$

Approximation 2 It can be shown that the Hankel singular values of $G_h(s)$ defined in (7.26) are correspondingly equal to those first $(n - l)$ Hankel singular values of $G(s)$. Hence, the above reduction formula can be repeatedly applied to get further reduced-order systems with known error bounds.

Let the Hankel singular values of $G(s)$ be $\sigma_1 > \sigma_2 > \dots > \sigma_r$ with multiplicities m_i , $i = 1, \dots, r$, i.e. $m_1 + m_2 + \dots + m_r = n$. By repeatedly applying the formulas (7.20)–(7.26), we may have

$$G(s) = D_0 + \sigma_1 E_1(s) + \sigma_2 E_2(s) + \dots + \sigma_r E_r(s) \quad (7.29)$$

where D_0 is a constant matrix and $E_i(s)$, $i = 1, \dots, r$, are stable, norm-1, all-pass transfer function matrices. $E_i(s)$ s are the differences at each approximation. Consequently, we may define reduced-order models, for $k = 1, \dots, r - 1$,

$$\hat{G}_k(s) = D_0 + \sum_{i=1}^k \sigma_i E_i(s) \quad (7.30)$$

Such a $\hat{G}_k(s)$ is stable, with the order $m_1 + \dots + m_k$, and satisfies

$$\|G(s) - \hat{G}_k(s)\|_\infty \leq (\sigma_{k+1} + \dots + \sigma_r) \quad (7.31)$$

However, $\hat{G}_k(s)$ is not an optimal Hankel approximation, for $k < r - 1$. The method to obtain an optimal Hankel approximation with “general” order is given below.

Approximation 3 Let the Hankel singular values of $G(s)$ be $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k > \sigma_{k+1} = \dots = \sigma_{k+l} > \sigma_{k+l+1} \geq \dots \geq \sigma_n$. Apply appropriate state similarity transformations to make the Gramians of $G(s)$ be arranged as

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k, \sigma_{k+l+1}, \dots, \sigma_n, \sigma_{k+1}, \dots, \sigma_{k+l})$$

Define the last l Hankel singular values to be σ . Following the formulas (7.20)–(7.26), define an $(n - l)$ th-order $\hat{G}(s)$. This $\hat{G}(s)$ is not stable but has exactly k stable poles. The k th-order stable part $G_{h,k}(s)$ of $\hat{G}(s)$, obtained by using modal decompositions say, is an k th-order Hankel optimal approximation of $G(s)$ and satisfies

$$\|G(s) - G_{h,k}(s)\|_H = \sigma \quad (7.32)$$

Nonsquare plants can be augmented with zero columns/rows and then be applied by the above procedures.

Remarks

1. The three methods introduced in the last three subsections can be applied to original systems (plants) as well as to controllers. However, most reported cases are on plant reductions. This may be due to robust controller design methods used subsequently that leads to better closed-loop performance even with a reduced-order plant. Examples of application on controller-size reduction can be found in [149]. In [149] it is also observed that the Balanced Truncation method and the Hankel-norm Approximation usually perform better at high frequency, while the singular perturbation approximation (balanced residualization) method performs better in the low- and medium-frequency ranges.
2. Glover shows in [55] that any stable, r th-order approximation G_r of $G(s)$ can never achieve $\|G(s) - G_r(s)\|_\infty \leq \sigma_{r+1}$. This lower error bound may serve as a yardstick to compare with the actual error obtained in practice.
3. All the three methods are applicable for *stable* systems only. If a system is unstable, modal decomposition can be applied first. That is, find a stable $G_s(s)$ and an antistable $G_{us}(s)$ (with all the poles in the closed right-half complex plane) such that

$$G(s) = G_s(s) + G_{us}(s) \quad (7.33)$$

Then, $G_s(s)$ can be reduced to $G_{sr}(s)$, by using any of the three methods, and a reduced-order system of the original $G(s)$ can be formed as

$$G_r(s) = G_{sr}(s) + G_{us}(s) \quad (7.34)$$

The routines to calculate a modal decomposition can be found in software packages such as MATLAB® or Slicot.

4. The formulas introduced here are for continuous-time systems. In the case of discrete-time systems, the Gramians are calculated from the discrete Lyapunov equations instead,

$$AP A^T - P + BB^T = 0 \quad (7.35)$$

$$A^T Q A - Q + C^T C = 0 \quad (7.36)$$

The balanced truncation method can then be applied similar to the case of continuous time. However, it should be noted that the reduced-order system is no longer in a balanced realization form [43, 138], though the same error bound still holds [7].

For using the singular perturbation approximation (balanced residualization) on a system with zero D -matrix, the reduced-order system $G_r(s) = [A_r, B_r, C_r, D_r]$ can be instead defined by

$$\begin{aligned} A_r &= A_{11} + A_{12}(I - A_{22})^{-1}A_{21} \\ B_r &= B_1 + A_{12}(I - A_{22})^{-1}B_2 \\ C_r &= C_1 + C_2(I - A_{22})^{-1}A_{21} \\ D_r &= C_2(I - A_{22})^{-1}B_2 \end{aligned} \quad (7.37)$$

Such a reduced-order system is still in a balanced realization and enjoys the same error bound [7, 44, 103, 123].

The discrete-time case of Hankel-norm approximation has been studied for a long time and is also called the Hankel-matrix approximation. Details can be found in [6, 89, 90, 129, 194].

5. Research has been conducted on numerical implementations of the above reduction methods. For instance, in balanced transformation in order to avoid numerical instability of forming products BB^T and $C^T C$, algorithms for direction calculation of the Choleski factors and improved balanced truncation scheme have been proposed [70, 93, 162]. Also, to avoid ill-conditioned computational problems, balancing-free approaches can be considered [146, 147, 168]. It is recommended that the model-reduction subroutines developed in the software package Slicot be used because of their superior numerical properties.

7.2 Reduction via Fractional Factors

The modal decomposition can be used to reduce the order of a general, unstable system as discussed in the last section. However, the order of the antistable part will not be reduced at all. Another reduction method applicable in the case of unstable systems is via reduction of normalized coprime factors of the system introduced in Chap. 5.

For a minimal realization model $G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array} \right]$, recall that a normalized left co-prime factorization is defined by $G(s) = \tilde{M}(s)^{-1} \tilde{N}(s)$, where $\tilde{M}(s)$ and $\tilde{N}(s)$ are stable and satisfy (5.2) and (5.3). Note that we assume $G(s)$ has a zero D -matrix and model reduction is conducted with regard to such a strictly proper system. In the case of a nonzero feedthrough term in $G(s)$, the nonzero D -matrix should be added to the reduced-order model. Such a treatment greatly simplifies formulas. It will keep the high-frequency gain of the original system in the *fractional balanced truncation* (FBT) method, or maintain the dc-gain in the *fractional singular perturbation approximation* (FSPA) method, both introduced below.

The factors $\tilde{M}(s)$ and $\tilde{N}(s)$ have the following state-space models:

$$\begin{bmatrix} \tilde{N} & \tilde{M} \end{bmatrix} = \left[\begin{array}{c|c} A + HC & B \quad H \\ \hline C & 0 \quad I \end{array} \right] \quad (7.38)$$

where $H = -ZC^T$ with $Z > 0$ solves the following algebraic Riccati equation:

$$AZ + ZA^T - ZC^T CZ + BB^T = 0 \quad (7.39)$$

$[\tilde{N} \ \tilde{M}]$ in (7.38) is stable and a balanced realization transformation can be found. Following the balanced realization algorithm in Sect. 7.1.1, a state similarity transformation $[T, T^{-1}]$ can be obtained such that $[T(A + HC)T^{-1}, T[B \ H], CT^{-1}]$ is a balanced realization with the Gramian $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$. To reduce the system to r th order, where $\sigma_r > \sigma_{r+1}$, let

$$T(A + HC)T^{-1} =: \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} = \begin{bmatrix} A_{11} + H_1 C_1 & A_{12} + H_1 C_2 \\ A_{21} + H_2 C_1 & A_{22} + H_2 C_2 \end{bmatrix} \quad (7.40)$$

$$T[B \ H] =: \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix} = \begin{bmatrix} B_1 & H_1 \\ B_2 & H_2 \end{bmatrix} \quad (7.41)$$

$$CT^{-1} =: [\tilde{C}_1 \ \tilde{C}_2] = [C_1 \ C_2] \quad (7.42)$$

$$\tilde{D} := [0 \ I] \quad (7.43)$$

where

$$TAT^{-1} =: \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (7.44)$$

$$TB =: \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \quad (7.45)$$

$$CT^{-1} =: [C_1 \ C_2] \quad (7.46)$$

$$TH =: \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \quad (7.47)$$

Accordingly, the Gramian is divided as

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \quad (7.48)$$

with $\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_r)$ and $\Sigma_2 = \text{diag}(\sigma_{r+1}, \dots, \sigma_n)$.

Now, the reduction methods introduced in Sect. 7.1 can be applied to obtain an r th-order $[\tilde{N}_r \ \tilde{M}_r]$, which leads to a reduced order (r th-order) model $G_r(s) = \tilde{M}_r(s)^{-1} \tilde{N}_r(s)$.

7.2.1 Fractional Balanced Truncation (FBT) Method

In this direct truncation method, we define

$$[\tilde{N}_r \ \tilde{M}_r] := \left[\begin{array}{c|c} \tilde{A}_{11} & \tilde{B}_1 \\ \hline \tilde{C}_1 & [0 \ I] \end{array} \right] = \left[\begin{array}{c|c} A_{11} + H_1 C_1 & B_1 \ H_1 \\ \hline C_1 & 0 \ I \end{array} \right] \quad (7.49)$$

It is easy to check that the above realization of $[\tilde{N}_r \ \tilde{M}_r]$ is still a balanced realization with the Gramian Σ_1 . Define

$$G_r(s) := \tilde{M}_r(s)^{-1} \tilde{N}_r(s) \quad (7.50)$$

By direct manipulations, we have

$$G_r(s) = \left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & 0 \end{array} \right] \quad (7.51)$$

An error bound directly follows from the result in Sect. 7.1.1 and as shown in [119] it is

$$\|[\tilde{N} - \tilde{N}_r \ \tilde{M} - \tilde{M}_r]\|_\infty \leq 2 \sum_{i=r+1}^n \sigma_i \quad (7.52)$$

7.2.2 Fractional Singular Perturbation Approximation (FSPA) Method

Naturally, the singular perturbation approximation method (or, the balanced residualization method) in Sect. 7.1.2 can be used to reduce the order of $[\tilde{N} \ \tilde{M}]$.

Define

$$A_r = \tilde{A}_{11} - \tilde{A}_{12} \tilde{A}_{22}^{-1} \tilde{A}_{21} \quad (7.53)$$

$$B_r = \tilde{B}_1 - \tilde{A}_{12} \tilde{A}_{22}^{-1} \tilde{B}_2 \quad (7.54)$$

$$C_r = \tilde{C}_1 - \tilde{C}_2 \tilde{A}_{22}^{-1} \tilde{A}_{21} \quad (7.55)$$

$$D_r = \tilde{D} - \tilde{C}_2 \tilde{A}_{22}^{-1} \tilde{B}_2 \quad (7.56)$$

Furthermore, B_r and D_r can be compatibly partitioned as

$$B_r = [B_{r,1} \quad B_{r,2}] := [B_1 - \tilde{A}_{12} \tilde{A}_{22}^{-1} B_2 \quad H_1 - \tilde{A}_{12} \tilde{A}_{22}^{-1} H_2] \quad (7.57)$$

$$D_r = [D_{r,1} \quad D_{r,2}] := [-C_2 \tilde{A}_{22}^{-1} B_2 \quad I - C_2 \tilde{A}_{22}^{-1} H_2] \quad (7.58)$$

Hence, let

$$[\tilde{N}_r \quad \tilde{M}_r] := \left[\begin{array}{c|cc} A_r & B_{r,1} & B_{r,2} \\ \hline C_r & D_{r,1} & D_{r,2} \end{array} \right] \quad (7.59)$$

which is of balanced realization form with the Gramian Σ_1 . An r th-order model $G_r(s)$ is then obtained by

$$G_r(s) = \tilde{M}_r(s)^{-1} \tilde{N}_r(s) = \left[\begin{array}{c|c} A_r - B_{r,2} D_{r,2}^{-1} C_r & B_{r,1} - B_{r,2} D_{r,2}^{-1} D_{r,1} \\ \hline D_{r,2}^{-1} C_r & D_{r,2}^{-1} D_{r,1} \end{array} \right] \quad (7.60)$$

In the case that the original model $G(s)$ is not strictly proper, the nonzero feedthrough term should be added in the D -matrix in (7.60).

The error bound (7.52) obviously holds here as well, from the result in Sect. 7.1.2.

Remarks

1. Meyer [119] shows that $\|[\tilde{N} \quad \tilde{M}]\|_H < 1$, i.e. $\Sigma < I$. Also, the infinitive norm $\|[\tilde{N} \quad \tilde{M}]\|_\infty = 1$, because of the normalized coprime factorization.
2. The result can be obtained that, in either the FBT method or FSPA method, we have

$$\|G(s) - G_r(s)\|_\infty \leq \|\tilde{M}_r^{-1}\|_\infty \|[\tilde{N} - \tilde{N}_r \quad \tilde{M} - \tilde{M}_r]\|_\infty \|\tilde{M}^{-1}\|_\infty \quad (7.61)$$

by writing

$$\begin{aligned} G - G_r &= \tilde{M}^{-1} \tilde{N} - \tilde{M}_r^{-1} \tilde{N}_r \\ &= \tilde{M}_r^{-1} \left([\tilde{N} - \tilde{N}_r \quad \tilde{M} - \tilde{M}_r] \begin{bmatrix} I \\ -\tilde{M}^{-1} \tilde{N} \end{bmatrix} \right) \\ &= \tilde{M}_r^{-1} \left([\tilde{N} - \tilde{N}_r \quad \tilde{M} - \tilde{M}_r] \tilde{M}^{-1} \begin{bmatrix} \tilde{M} \\ -\tilde{N} \end{bmatrix} \right) \end{aligned} \quad (7.62)$$

and by the fact that $\|[\begin{smallmatrix} \tilde{M} \\ -\tilde{N} \end{smallmatrix}]\|_\infty = 1$.

3. Note that the methods introduced above are based on the left coprime factorization. Similarly, model reductions can be done with regard to the normalized *right* coprime factorization.
4. The model reduction in the discrete-time case using the fractional balanced truncation method is straightforward, and using the fractional singular perturbation approximation method can be found in [123, 127].

7.3 Relative-Error Approximation Methods

As discussed in Sect. 7.1.3, the balanced truncation method gives a good approximation over high-frequency ranges, while the singular perturbation approximation performs better over low- and medium-frequency ranges. If a reduced-order system is required in some practical problems to approximate equally well over the whole frequency range, then the method called the *balanced stochastic truncation* (BST) method may be considered [26, 60, 61, 102, 169]). The effect of this method may be viewed, for a stable, square and invertible $G(s)$, as a minimization of

$$\|G^{-1}(s)(G(s) - G_r(s))\|_{\infty} \quad (7.63)$$

Hence, the reduced-order system $G_r(s)$ approximates the original system in the sense of making $G^{-1}G_r$ nearer to identity. Problem (7.63) represents a minimization of a relative error and is one of several relative-error approximation methods (e.g., see [56, 58]).

The idea of the BST method is the following. First, a spectral factor $W(s)$ of $G(s)G^{-}(s)$ is to be found. That is,

$$W^{-}(s)W(s) = G(s)G^{-}(s)$$

where $W^{-}(s) := W^T(-s)$, similarly for $G^{-}(s)$; and $W(s)$ is stable, square and of minimum phase (i.e. $(W(s))^{-1} \in \mathcal{H}_{\infty}$). $W(s)$ contains the “magnitude” part of $G(s)$. Correspondingly, a “phase” matrix of $G(s)$ can be defined as $F(s) = (W^{-}(s))^{-1}G(s)$. $F(s)$ is an all-pass transfer function matrix and contains both stable and unstable modes. The BST method is then to apply a balanced truncation on the stable part of $F(s)$ (which is of the same order as $G(s)$), with the same state similarity transformation and partition on the state space model of $G(s)$ to obtain a reduced-order $G_r(s)$.

For a given n th-order, stable and square $G(s) = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ we assume this is a minimal realization and the invertibility of $G(s)$ implies the existence of D^{-1} .

The computational steps of the BST method can be described as

Step 1: Solve the Lyapunov equation

$$AP + PA^T + BB^T = 0 \quad (7.64)$$

where the solution $P > 0$.

Step 2: Let

$$B_W = PC^T + BD^T \quad (7.65)$$

$$C_W = D^{-1}(C - B_W^T Q_W) \quad (7.66)$$

where Q_W is the stabilizing solution to the following Riccati equation:

$$A^T Q_W + Q_W A + C_W^T C_W = 0 \quad (7.67)$$

Remark: (A, B_W, C_W) forms the stable part of $F(s)$.

Step 3: Decide a balanced realization transformation with regard to (P, Q_W) and apply the transformation onto $G(s)$. Let the balanced Gramian matrix be $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ in descending order.

Remarks: (1) After applying the above transformation, $G(s)$ is, in general, not in the balanced realization form, but with its controllability Gramian being Σ ; (2) $\Sigma \leq I$, due to $F(s)$ being an all-pass matrix.

Step 4: Partition Σ as

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \quad (7.68)$$

where $\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_r)$, $\Sigma_2 = \text{diag}(\sigma_{r+1}, \dots, \sigma_n)$, with $\sigma_r > \sigma_{r+1}$. Partition compatibly the matrices A , B , and C (of the transformed state-space model of $G(s)$) $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, $B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$, and $C = \begin{bmatrix} c_1 & c_2 \end{bmatrix}$. Then, a reduced-order system $G_r(s)$ can be defined by

$$G_r(s) = C_1(sI - A_{11})^{-1}B_1 + D \quad (7.69)$$

For this reduced-order system, a relative error bound can be derived [61] as

$$\|G^{-1}(G - G_r)\|_\infty \leq \prod_{i=r+1}^n \frac{1 + \sigma_i}{1 - \sigma_i} - 1 \quad (7.70)$$

The errors between the phase matrices, with the same antistable and constant parts, are bounded by

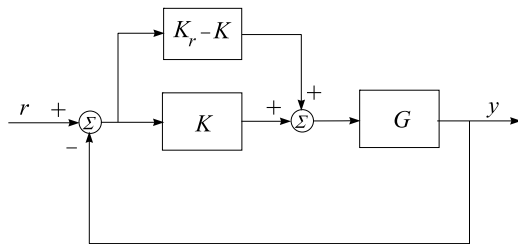
$$\|F(s) - F_r(s)\|_\infty \leq 4(\sigma_{r+1} + \dots + \sigma_n) \quad (7.71)$$

$$\|F(s) - F_r(s)\|_H \leq 2(\sigma_{r+1} + \dots + \sigma_n) \quad (7.72)$$

Remarks

1. The BST method can be applied to nonsquare $G(s)$ as well, with slight modifications. The invertibility of $G(s)$ is changed to the assumption that D is of full row rank. The constant matrix of the square spectral factor $W(s)$ would be D_W , with $D_W^T D_W = DD^T$, and the output matrix $C_W = D_W(DD^T)^{-1}(C - B_W^T Q_W)$.

Fig. 7.2 Closed-loop system with reduced-order controllers



However, in the nonsquare case, there would be no explicit explanation of the relative error format (7.63). The reduction just shows an approximation with respect to phase.

2. In the above method, instead of balanced truncation, the Hankel-norm approximation can be used [56]. That is, the balanced realization of the stable part, $F_s(s)$, of the phase matrix $F(s)$ is to be approximated by a Hankel-norm approximant, $F_{s,r}(s)$, calculated using the formula in Sect. 7.1.3. The reduced model, G_r , can then be defined as

$$G_r = G - W^-(F_s - F_{s,r}) \quad (7.73)$$

It can be shown [56, 61] that G_r is stable and satisfies the following error bound:

$$\|G^{-1}(G - G_r)\|_\infty \leq \prod_{i=r+1}^n (1 + \sigma_i) - 1 \quad (7.74)$$

7.4 Frequency-Weighted Approximation Methods

The model-order reduction approaches introduced above can be in theory applied to plants (the original system models) as well as to controllers. However, to reduce the order of a designed controller, it is necessary to take into account the plant that is being compensated and other design specifications of the closed-loop system. With such considerations, the controller-order reduction problem would be better formulated as a frequency-weighted model reduction, and suitable approaches have been suggested.

Assume that a controller $K(s)$ has been designed for a plant with a model $G(s)$, and denote a reduced-order controller by $K_r(s)$. The configuration with $K(s)$ replaced by $K_r(s)$ in the closed-loop system can be depicted by Fig. 7.2.

From the Small-Gain Theorem (Theorem 3.1), it is easy to obtain the result that the closed-loop system with the reduced-order controller $K_r(s)$ remains stable if $K(s)$ and $K_r(s)$ have the same number of unstable poles and if

$$\|[K(s) - K_r(s)]G(s)[I + K(s)G(s)]^{-1}\|_\infty < 1 \quad (7.75)$$

or

$$\| [I + G(s)K(s)]^{-1} G(s) [K(s) - K_r(s)] \|_{\infty} < 1 \quad (7.76)$$

Let $W_i(s) := G(s)[I + K(s)G(s)]^{-1}$ and $W_o(s) := [I + G(s)K(s)]^{-1} G(s)$. Then in order to reserve the stability of the closed-loop system, a reduced-order controller $K_r(s)$ is sought to minimize the frequency-weighted cost functions

$$\| [K(s) - K_r(s)] W_i(s) \|_{\infty} \quad (7.77)$$

or

$$\| W_o(s) [K(s) - K_r(s)] \|_{\infty} \quad (7.78)$$

Note in this case the input frequency weight function $W_i(s)$ equals the output frequency weight function $W_o(s)$.

Another consideration is about the performance of the closed-loop system. The performance is closely related to the transfer function of the closed-loop system. Naturally, to maintain the performance of the designed, closed-loop system, it requires the transfer function of the closed-loop system with reduced-order controller to be as near as possible to that with the original controller. The two transfer functions are, respectively,

$$G(s)K(s)[I + G(s)K(s)]^{-1} \quad \text{and} \quad G(s)K_r(s)[I + G(s)K_r(s)]^{-1}$$

The difference between these two transfer functions, by neglecting terms of second and higher orders in $K - K_r$, is

$$G(s)[I + K(s)G(s)]^{-1} [K(s) - K_r(s)] [I + G(s)K(s)]^{-1}$$

Hence, a reduced-order controller $K_r(s)$ should try to minimize

$$\| W_o(s) [K(s) - K_r(s)] W_i(s) \|_{\infty} \quad (7.79)$$

where $W_i(s) := [I + G(s)K(s)]^{-1}$ and $W_o(s) := G(s)[I + K(s)G(s)]^{-1}$.

Let us now concentrate on the general form of frequency-weighted model reduction of (7.79), but replacing $K(s)$ and $K_r(s)$ by $G(s)$ and $G_r(s)$, respectively. Assume that $G(s)$ is stable and has the minimal realization as defined in (7.2). The input weight function $W_i(s)$ and the output weight function $W_o(s)$ are also stable with minimal realizations: $W_i(s) = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}$ and $W_o(s) = \begin{bmatrix} A_o & B_o \\ C_o & D_o \end{bmatrix}$, respectively.

Remark Obviously the stability assumption of $G(s)$ would be a restriction in the case of controller-order reduction. In the case of unstable $G(s)$, the modal decomposition discussed in Sect. 7.1.3 can be considered.

The augmented systems $G(s)W_i(s)$ and $W_o(s)G(s)$ have the state-space models

$$G(s)W_i(s) = \left[\begin{array}{c|c} \hat{A}_i & \hat{B}_i \\ \hline \hat{C}_i & \hat{D}_i \end{array} \right] = \left[\begin{array}{cc|c} A & BC_i & BD_i \\ 0 & A_i & B_i \\ \hline C & DC_i & DD_i \end{array} \right] \quad (7.80)$$

$$W_oG(s) = \left[\begin{array}{c|c} \hat{A}_o & \hat{B}_o \\ \hline \hat{C}_o & \hat{D}_o \end{array} \right] = \left[\begin{array}{cc|c} A & 0 & B \\ B_oC & A_o & B_oD \\ \hline D_oC & C_o & D_oD \end{array} \right] \quad (7.81)$$

Let \hat{P} and \hat{Q} be two non-negative matrices satisfying the following two Lyapunov equations, respectively:

$$\hat{A}\hat{P} + \hat{P}\hat{A}^T + \hat{B}\hat{B}^T = 0 \quad (7.82)$$

$$\hat{A}^T\hat{Q} + \hat{Q}\hat{A} + \hat{C}^T\hat{C} = 0 \quad (7.83)$$

Furthermore, partition \hat{P} and \hat{Q} as

$$\hat{P} = \begin{bmatrix} P & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \quad (7.84)$$

$$\hat{Q} = \begin{bmatrix} Q & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \quad (7.85)$$

where P and Q are of n -dimension, and are called the input weighted Gramian and output weighted Gramian, respectively.

Several frequency-weighted model-reduction algorithms use balanced realization transformations on P and Q or are related with truncations. Three such methods are introduced below.

7.4.1 Frequency-Weighted Balanced Truncation (FWBT)

Enns [38, 39] proposes to find a balanced realization on P and Q , i.e. to find an invertible $n \times n$ matrix T (see Sect. 7.1.1) such that

$$TPT^T = (T^{-1})^TQT^{-1} = \text{diag}(\sigma_1, \dots, \sigma_r, \sigma_{r+1}, \dots, \sigma_n)$$

with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} \geq \dots \geq \sigma_n \geq 0$. Apply such a state similarity transformation (T, T^{-1}) on $G(s)$ and partition it accordingly,

$$\left[\begin{array}{c|c} TAT^{-1} & TB \\ \hline CT^{-1} & D \end{array} \right] = \left[\begin{array}{cc|c} A_{11} & A_{12} & B_1 & B_2 \\ A_{21} & A_{22} & & \\ \hline C_1 & C_2 & D & \end{array} \right] \quad (7.86)$$

where A_{11} is of $r \times r$ dimension. A reduced-order $G_r(s)$ can then be defined by

$$G_r(s) = \left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & D \end{array} \right] \quad (7.87)$$

$G_r(s)$ obtained in (7.87) is not necessarily stable, except in the cases where either $W_i = I$ or $W_o = I$ (one-side weight only). There is an error bound derived [84] for

$$\|W_o(s)[G(s) - G_r(s)]W_i(s)\|_\infty$$

However, this bound has to be computed iteratively, depending on reduced-order models of $n - 1, \dots, r + 1$, and is not practically useful.

7.4.2 Frequency-Weighted Singular Perturbation Approximation (FWSPA)

Lin and Chiu [100] introduce another truncation method to obtain a frequency-weighted reduced model. Assume that P_{22} and Q_{22} in (7.84) and (7.85), respectively, are nonsingular. This condition is guaranteed, for example, in the case that the realizations (7.80) and (7.81) are minimal, i.e. if there are no pole/zero cancellations between $G(s)$ and $W_i(s)$, nor between $W_o(s)$ and $G(s)$. Instead of applying a balanced realization transformation on P and Q as in the Enns method, a balanced realization transformation is to be found with regard to $P - P_{12}P_{22}^{-1}P_{12}^T$ and $Q - Q_{12}^T Q_{22}^{-1} Q_{12}$. This balanced realization is then applied onto the original model $G(s)$ and truncation taken in the same way as in (7.86) and (7.87).

Apparently, this method is so named because the matrices $P - P_{12}P_{22}^{-1}P_{12}^T$ and $Q - Q_{12}^T Q_{22}^{-1} Q_{12}$ are in the form of the reduced-order state matrix used in the Singular Perturbation Approximation method (Sect. 7.1.2). It is observed that, by pre/postmultiplying $\begin{bmatrix} I & -P_{12}P_{22}^{-1} \\ 0 & I \end{bmatrix}$ and $\begin{bmatrix} I & 0 \\ -P_{22}^{-1}P_{12}^T & I \end{bmatrix}$ on (7.82), and similar multiplications with regard to Q on (7.83), matrices $P - P_{12}P_{22}^{-1}P_{12}^T$ and $Q - Q_{12}^T Q_{22}^{-1} Q_{12}$ satisfy two Lyapunov equations, respectively. Hence, the diagonalized matrices of these two after the balanced realization transformations satisfy the Lyapunov equations, too. This indicates the reduced-order system is guaranteed to be stable.

There is an error bound available for this method [157]. However, it suffers the same weakness as the error bound for the Enns method. The error bound cannot be simply calculated from the original model data.

7.4.3 Frequency-Weighted Moduli Truncation Method (FWMT)

The error bounds for the above two methods are not practically useful. In [175], Wang et al. propose another truncation method with a priori computable error bound.

Quoting the upper-left blocks of (7.82) and (7.83) gives the following two matrix equations:

$$AP + PA^T + BC_i P_{12} + P_{12}^T C_i^T B^T + BD_i D_i^T B^T = 0 \quad (7.88)$$

$$A^T Q + QA + Q_{12} B_o C + C^T B_o^T Q_{12}^T + C^T D_o^T D_o C = 0 \quad (7.89)$$

Let

$$X = BC_i P_{12} + P_{12}^T C_i^T B^T + BD_i D_i^T B^T \quad (7.90)$$

$$Y = Q_{12} B_o C + C^T B_o^T Q_{12}^T + C^T D_o^T D_o C \quad (7.91)$$

Note that X and Y defined above are symmetric but not sign-definite in general. Apply congruent transformations on X and Y to obtain orthogonal matrices U and V such that

$$X = U \Theta U^T \quad (7.92)$$

$$Y = V \Gamma V^T \quad (7.93)$$

where $\Theta = \text{diag}(\theta_1, \dots, \theta_i, 0, \dots, 0)$, $\Gamma = \text{diag}(\gamma_1, \dots, \gamma_o, 0, \dots, 0)$, with $|\theta_1| \geq \dots \geq |\theta_i| > 0$ and $|\gamma_1| \geq \dots \geq |\gamma_o| > 0$. Now, define

$$\tilde{B} = U \text{diag}(|\theta_1|^{\frac{1}{2}}, \dots, |\theta_i|^{\frac{1}{2}}, 0, \dots, 0) \quad (7.94)$$

$$\tilde{C} = \text{diag}(|\gamma_1|^{\frac{1}{2}}, \dots, |\gamma_o|^{\frac{1}{2}}, 0, \dots, 0) V^T \quad (7.95)$$

Solve the following two Lyapunov equations:

$$A \tilde{P} + \tilde{P} A^T + \tilde{B} \tilde{B}^T = 0 \quad (7.96)$$

$$\tilde{Q} A + A^T \tilde{Q} + \tilde{C}^T \tilde{C} = 0 \quad (7.97)$$

It can be shown [175] that $(A, \tilde{B}, \tilde{C})$ is a minimal realization and hence the solutions \tilde{P} and \tilde{Q} to (7.96) and (7.97), respectively, are positive definite. Similar to FWBT and FWSPA methods, a balanced realization is found with regard to \tilde{P} and \tilde{Q} and the transformation is applied to the original model (A, B, C) to yield a reduced-order model $G_r(s)$. Such a $G_r(s)$ is stable, following the same reasoning as in the FWSPA method. Furthermore, the following error bound can be derived [175].

Define

$$K = \text{diag}(|\theta_1|^{\frac{1}{2}}, \dots, |\theta_i|^{\frac{1}{2}}, 0, \dots, 0) U^T B \quad (7.98)$$

$$L = C V \text{diag}(|\gamma_1|^{\frac{1}{2}}, \dots, |\gamma_o|^{\frac{1}{2}}, 0, \dots, 0) \quad (7.99)$$

Then, it is shown in [175] that

$$\|W_o(s)(G(s) - G_r(s))W_i(s)\|_{\infty} \leq k \sum_{j=r+1}^n \sigma_j \quad (7.100)$$

where

$$k = 2 \|W_o(s)L\|_\infty \|K W_i(s)\|_\infty$$

and $(\sigma_1, \cdot, \sigma_r, \sigma_{r+1}, \dots, \sigma_n)$ are the diagonal elements of the balanced form of \tilde{P} (\tilde{Q}).

This page intentionally left blank

Chapter 8

LMI Approach

In Chaps. 4, 5, and 6, we discussed how a robust controller design problem can be cast as an optimization problem, in \mathcal{H}_∞ or μ -synthesis formulations. Optimal or sub-optimal solutions can be found by following some formulas which are derived using functional analysis or operator theories. There is actually another way to consider the robust design problem, and control system design in general, as an optimization problem, solutions to which can be directly computed by convex computational procedures. That is the so called Linear Matrix Inequality (LMI) approach. In this chapter, basic concepts of LMI and a few applications of LMI in robust and other control system design problems will be introduced.

8.1 Basics About LMI

Variables involved in a linear matrix inequality problem are either scalars or sign-definite matrices. Recall that a (real) symmetric matrix M is positive (negative) definite if $x^T M x > (<) 0, \forall x \neq 0$. Also, M is called positive (negative) semi-definite if $x^T M x \geq (\leq) 0, \forall x$. The most general form of an LMI is

$$F(\mathbf{x}) = M_0 + \sum_{i=1}^l x_i M_i > 0 \quad (8.1)$$

where x_i are real, scalar variables, $\mathbf{x} = [x_1, \dots, x_l]^T$, and M_0 and M_i are constant (given), symmetric matrices of dimension $n \times n$. The above LMI is *feasible*, if a vector \mathbf{x} exists which satisfies (8.1).

In some control system problems, it is more convenient to be formulated as the following LMI:

$$F(X_1, \dots, X_k) = M_0 + \sum_{i=1}^k G_i X_i H_i > 0 \quad (8.2)$$

where $X_i \in \Re^{p_i \times q_i}$ are matrix variables to be found and $G_i \in \Re^{n \times p_i}$, $H_i \in \Re^{q_i \times n}$ are known matrices. It is easy to see that the vector variable \mathbf{x} in (8.1) can be formed by stacking the columns of X_i in (8.2).

In general, an optimization problem can have an LMI as its cost function with several other LMI's of the same matrix variables as constraints.

Introduction of LMI can be as a matter of fact traced back to the work of Lyapunov. It has also been known for several decades that some control problems can be formulated as LMI problems (see [49, 184, 185], for example). The major impact behind the wide use of LMIs in control systems analysis and design in the last decade or so is due to the breakthrough of efficient numerical algorithms of interior-point methods in convex optimization [50, 130]. Such a development makes it practically possible to find solutions to LMI's representing many control systems problems which we will introduce next.

8.2 Control Problems Using LMI

8.2.1 Lyapunov Stability Criterion

For a continuous-time linear system \mathcal{S}

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\tag{8.3}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ the input (control) vector, and $y(t) \in \mathbb{R}^p$ the output (measurement) vector, the system is *asymptotically stable* if there exists a positive definite matrix $P > 0$ such that

$$A^T P + P A < 0\tag{8.4}$$

This is an LMI feasibility problem.

Remark Let $P = \{p_{ij}\}$, $p_{ij} = p_{ji}$. (8.4) can be re-written as

$$\sum_{i,j} p_{ij} (A^T E_{ij} + E_{ji} A) < 0\tag{8.5}$$

where E_{ij} is a zero matrix except its (i, j) element being 1. (8.5) is obviously in the form of (8.1).

8.2.2 Stabilization by State Feedback

Consider the system in (8.3) again. In the case that it is not asymptotically stable, a state feedback matrix F , in $u = v - Fx$, may be sought to form the following

closed-loop system:

$$\begin{aligned}\dot{x} &= (A - BF)x + Bv(t) \\ y(t) &= Cx(t) + Dv(t)\end{aligned}\tag{8.6}$$

where we use $v(t)$ to denote the external input and $u(t)$ in (8.3) is given as $u(t) = v(t) - Fx(t)$. To make the closed-loop system asymptotically stable, we require the state feedback matrix F to satisfy, following (8.4),

$$(A - BF)^T P + P(A - BF) < 0\tag{8.7}$$

or

$$A^T P + PA - F^T B^T P - PBF < 0\tag{8.8}$$

where $P > 0$ is to be found. Here we want to find two matrices, F and P , in (8.8). This is not a linear inequality, rather a quadratic one, due to the last two terms in the inequality. To make it linear, we may pre- and post-multiply P^{-1} on (8.8), which leads to

$$P^{-1}A^T + AP^{-1} - P^{-1}F^T B^T - BFP^{-1} < 0\tag{8.9}$$

By letting $P^{-1} =: Q$ and defining $R = FQ$, (8.9) becomes

$$QA^T + AQ - R^T B^T - BR < 0\tag{8.10}$$

which obviously is “linear”. Once the variables Q and R are found, the original variables P and F can be directly recovered.

Remark The technique used above is called “change of variables”, which is useful to derive a linear inequality from an originally nonlinear one. It is important to note that in the transformations the equivalence of inequalities should be retained and the original variables should be recoverable.

8.2.3 Computation of \mathcal{L}_2 Norm

For a general (stable) system with $u(t)$ as its input and $z(t)$ as the output, \mathcal{L}_2 norm can be defined as a number γ which is the minimum positive satisfying the following inequality:

$$\|y\|_2 < \gamma \|u\|_2 + \beta\tag{8.11}$$

where $\|\cdot\|_2$ denotes the standard 2-norm of a vector function, i.e.

$$\|x\|_2 = \left(\int_0^\infty x^T(t)x(t) dt \right)^{1/2}$$

as defined earlier in the book, and β is a positive constant. γ is also called the \mathcal{L}_2 -gain of the system which shows the maximum gain of the system output in terms of “size” with regard to that of the input, the RMS energy gain.

For a linear system in (8.3), the gain γ can be calculated from solving the following matrix inequality, for $P > 0$:

$$\begin{bmatrix} A^T P + P A + \frac{1}{\gamma} C^T C & P B + \frac{1}{\gamma} C^T D \\ B^T P + \frac{1}{\gamma} D^T C & -\gamma I + \frac{1}{\gamma} D^T D \end{bmatrix} < 0 \quad (8.12)$$

(8.12) can be derived as follows.

(8.12) is equivalent to (8.13) below, by pre- and post-multiplying $[x^T \ u^T]$ and $\begin{bmatrix} x \\ u \end{bmatrix}$, $\forall [x^T \ u^T]^T \neq 0$, respectively,

$$\begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} A^T P + P A + \frac{1}{\gamma} C^T C & P B + \frac{1}{\gamma} C^T D \\ B^T P + \frac{1}{\gamma} D^T C & -\gamma I + \frac{1}{\gamma} D^T D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} < 0 \quad (8.13)$$

which leads to

$$\begin{aligned} & x^T A^T P x + x^T P A x + \frac{1}{\gamma} x^T C^T C x + x^T \left(P B + \frac{1}{\gamma} C^T D \right) u \\ & + u^T \left(B^T P + \frac{1}{\gamma} D^T C \right) x + \frac{1}{\gamma} u^T D^T D u - \gamma u^T u < 0 \end{aligned} \quad (8.14)$$

and

$$x^T A^T P x + x^T P A x + 2x^T P B u + \frac{1}{\gamma} y^T y - \gamma u^T u < 0 \quad (8.15)$$

By defining $V = x^T P x$ and using $\dot{V} = x^T A^T P x + x^T P A x + 2x^T P B u$, (8.15) can be expressed as

$$\dot{V} + \frac{1}{\gamma} y^T y - \gamma u^T u < 0 \quad (8.16)$$

Integrating the above from 0 to ∞ yields

$$V_\infty - V_0 + \int_0^\infty \frac{1}{\gamma} y^T(t) y(t) dt - \int_0^\infty \frac{1}{\gamma} u^T(t) u(t) dt < 0$$

Further by shifting terms, taking the square root and applying the triangle inequality, the above inequality can be re-written in the form of (8.11).

8.2.4 Computation of \mathcal{H}_∞ Norm

The \mathcal{H}_∞ norm of a linear, stable system \mathcal{S} as in (8.3) is bounded by a constant number γ , i.e. $\|\mathcal{S}\|_\infty < \gamma$, if there exists $P > 0$ such that

$$\begin{bmatrix} A^T P + P A + C^T C & P B + C^T D \\ B^T P + D^T C & D^T D - \gamma^2 I \end{bmatrix} < 0$$

The above can be used iteratively to find the \mathcal{H}_∞ -norm of \mathcal{S} , the minimum γ .

This result is actually the well-known *bounded-real lemma* [184]. Such a system as in (8.3) is called *nonexpansive*; it has the property

$$\int_0^{t_f} y^T(t) y(t) dt \leq \gamma^2 \int_0^{t_f} u^T(t) u(t) dt$$

for any $t_f > 0$, when $x(0) = 0$.

8.2.5 Formulation of LQR in LMI

For the linear, time-invariant system in (8.3), the Linear-Quadratic Regulator (LQR) problem is, given an initial condition $x(0) = x_o$, to find a state feedback $u = -Fx$ that minimizes the cost function

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (8.17)$$

where $Q, R > 0$.

The solution to this problem is well known (for example see [8]) and $u = -Fx = -R^{-1}B^T P$, where $P > 0$ and solves the following Algebraic Riccati Equation (ARE):

$$A^T P + P A - P B R^{-1} B^T P + Q = 0$$

Furthermore,

$$J_{\min} = \min \int_0^\infty (x^T Q x + u^T R u) dt = x_o^T P x_o$$

The LQR problem can also be formulated as an LMI Problem. It can be shown by direct deductions that with a $P > 0$ satisfying the following matrix inequality:

$$A^T P + P A - P B R^{-1} B^T P + Q < 0 \quad (8.18)$$

and by setting the negative state feedback matrix $F = R^{-1}B^T P$, we have

$$J = \int_0^\infty (x^T Q x + u^T R u) dt < x_o^T P x_o$$

Hence we can reduce the cost function J if we can find smaller P . However, (8.18) is not a linear inequality in P . We need a few “tricks” here.

First, we define $L = P^{-1}$ and pre- and post-multiply L on both sides of the inequality (8.18), which changes it into

$$LA^T + AL - BR^{-1}B^T + LQL < 0 \quad (8.19)$$

Next, we use the following Schur Complement Formula to transform equivalently (8.19) into linear inequalities.

Schur Complement Formula For a symmetric matrix

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{12}^T & M_{22} \end{bmatrix} \quad (8.20)$$

$M < 0$ if and only if $M_{22} < 0$ and $M_{11} - M_{12}M_{22}^{-1}M_{12}^T < 0$.

Schur Complement Formula can be verified by pre- and post-multiplying

$$\begin{bmatrix} I & -M_{12}M_{22}^{-1} \\ 0 & I \end{bmatrix}$$

and

$$\begin{bmatrix} I & 0 \\ -M_{22}^{-1}M_{12}^T & I \end{bmatrix}$$

respectively, on both sides of the inequality $M < 0$.

Now by defining $M_{11} = LA^T + AL - BR^{-1}B^T$, $M_{12} = L$ and $M_{22} = -Q^{-1}$ and using the Schur Complement Formula, solving the quadratic matrix inequality (8.19) is equivalent to solving the following linear matrix inequality:

$$\begin{bmatrix} LA^T + AL - BR^{-1}B^T & L \\ L & -Q^{-1} \end{bmatrix} < 0 \quad (8.21)$$

Further, in order to find the minimal cost function, we may minimize γ , where $\gamma > x_o^T P x_o$ is required, i.e. $x_o^T P x_o - \gamma < 0$. To write this inequality in terms of L , the Schur Complement Formula can be used again. By defining $M_{11} = -\gamma$, $M_{12} = x_o^T$ and $M_{22} = -L = (-P^{-1})$, it is equivalent to

$$\begin{bmatrix} -\gamma & x_o^T \\ x_o & -L \end{bmatrix} < 0 \quad (8.22)$$

Therefore, the state feedback matrix F , in $u = -Fx$, which leads to the minimal cost function J can be found by solving the following LMI optimization problem for $L > 0$ and $\gamma > 0$:

$$\begin{aligned}
& \min \gamma \\
& \text{s.t.} \quad \begin{bmatrix} -\gamma & x_o^T \\ x_o & -L \end{bmatrix} < 0 \\
& \quad \quad \begin{bmatrix} LA^T + AL - BR^{-1}B^T & L \\ L & -Q^{-1} \end{bmatrix} < 0
\end{aligned}$$

and $F = R^{-1}B^T L^{-1}$.

8.3 A Few More Properties Concerning LMI

In formulations and solution procedures of LMIs, a number of existing results, such as the Schur Complement Formula, can be used to help solve the problems. In this section, we will introduce a few of them. Please note that proofs are not given here. Interested readers can consult the literature, for example [15, 49, 50] and references therein.

8.3.1 Congruence Transformation

For a given positive definite (real) matrix $M > 0$, the matrix $VMV^T > 0$ holds, where V is a real matrix of full row rank. This can be proved by using the definition of positive definiteness and is called *congruence transformation*. The positive definiteness is therefore invariant under congruence transformations. This property is useful, for example, to transform bilinear terms in a matrix inequality into a linear one, with variable changes as in the following.

We have the matrix inequality

$$M = \begin{bmatrix} A^T P + PA & PBF + C^T Q \\ F^T B^T P + QC & -2Q \end{bmatrix} < 0 \quad (8.23)$$

where the matrices A , B and C are given and $P > 0$, $Q > 0$ and (unstructured) F are matrix variables to be found. (8.23) is not a linear matrix inequality because of the “bilinear” terms involved. By defining $V = \begin{bmatrix} P^{-1} & 0 \\ 0 & Q^{-1} \end{bmatrix}$ and pre-multiplying V and post-multiplying $V^T (= V)$ on both sides of (8.23), yields

$$\begin{bmatrix} P^{-1}A^T + AP^{-1} & BFQ^{-1} + P^{-1}C^T \\ Q^{-1}F^T B^T + CP^{-1} & -2Q^{-1} \end{bmatrix} < 0 \quad (8.24)$$

Further, by introducing $X = P^{-1}$, $Y = Q^{-1}$ and $Z = FQ^{-1}$, we have the following linear matrix inequality instead of (8.24) or (8.23):

$$\begin{bmatrix} XA^T + AX & BZ + XC^T \\ Z^T B^T + CX & -2Y \end{bmatrix} < 0 \quad (8.25)$$

It is straightforward to recover P , Q and F from X , Y and Z .

8.3.2 Schur Complements for Nonstrict Inequalities

In the previous section, the Schur Complement Formula was introduced for the case of strict inequalities. For nonstrict inequalities, the Moore–Penrose pseudo inverse of constant matrix will have to be used [15].

For a symmetric matrix

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{12}^T & M_{22} \end{bmatrix}$$

$M \leq 0$ if and only if $M_{22} \leq 0$ and $M_{11} - M_{12}M_{22}^\dagger M_{12}^T \leq 0$, and $M_{12}(I - M_{22}M_{22}^\dagger) = 0$, where M_{22}^\dagger denotes the Moore–Penrose pseudo inverse of M_{22} .

8.3.3 Projection and Finsler's Lemmas

In analysis and synthesis of control systems using LMIs, a few results, listed below, can be used to test the solvability of the LMIs and further to find solutions. The first is a quite straightforward one.

8.3.3.1 Lemma 1

The inequality, with the symmetric variable X ,

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{12}^T & P_{22} + X & P_{23} \\ P_{13}^T & P_{23}^T & P_{33} \end{bmatrix} < 0 \quad (8.26)$$

has a solution *if and only*

$$\begin{bmatrix} P_{11} & P_{13} \\ P_{13}^T & P_{33} \end{bmatrix} < 0 \quad (8.27)$$

and, furthermore, any X satisfying the following inequality is a solution:

$$X < -P_{22} + \begin{bmatrix} P_{12}^T & P_{23} \end{bmatrix} \begin{bmatrix} P_{11} & P_{13} \\ P_{13}^T & P_{33} \end{bmatrix}^{-1} \begin{bmatrix} P_{12} \\ P_{23}^T \end{bmatrix}$$

It can be shown that the above lemma holds by first interchanging Rows 1 and 2 as well as Columns 1 and 2 of (8.26), and then applying the Schur Complement Formula.

8.3.3.2 Lemma 2

Consider the LMI below, with symmetric P_{11} , P_{22} and P_{33} and the unstructured variable X ,

$$\begin{bmatrix} P_{11} & P_{12} + X^T & P_{13} \\ P_{12}^T + X & P_{22} & P_{23} \\ P_{13}^T & P_{23}^T & P_{33} \end{bmatrix} < 0 \quad (8.28)$$

The LMI (8.28) has a solution X *if and only if*

$$\begin{bmatrix} P_{11} & P_{13} \\ P_{13}^T & P_{33} \end{bmatrix} < 0 \quad \text{and} \quad \begin{bmatrix} P_{22} & P_{23} \\ P_{23}^T & P_{33} \end{bmatrix} < 0$$

Furthermore, when the above two inequalities hold, a solution is given by

$$X = P_{23} P_{33}^{-1} P_{13}^T - P_{12}^T \quad (8.29)$$

The necessary condition is straightforward by the fact that all leading minors of a negative (positive) definite matrix are negative (positive) definite and by certain interchange of rows and columns. To show the sufficiency part, it is obvious that the LMI (8.28) is equivalent to the following LMI, by the Schur complement:

$$\begin{bmatrix} P_{11} & P_{12} + X^T \\ P_{12}^T + X & P_{22} \end{bmatrix} - \begin{bmatrix} P_{13}^T \\ P_{23}^T \end{bmatrix} P_{33}^{-1} \begin{bmatrix} P_{31} & P_{32} \end{bmatrix} < 0 \quad (8.30)$$

The matrix X defined in (8.29) simply cancels the off-diagonal terms in the left-hand side matrix above which leads to the inequality (8.30) due to the negativeness property of (8.28).

8.3.3.3 Lemma 3 (Projection Lemma)

For matrices A , B and a symmetric matrix P , the LMI below with unstructured variable X

$$A^T X B + B^T X^T A + P < 0 \quad (8.31)$$

has a solution *if and only if*

$$Ax = 0 \text{ or } Bx = 0 \quad \text{imply} \quad x^T P x < 0 \text{ or } x = 0 \quad (8.32)$$

Further, let A_\perp and B_\perp be matrices whose columns form a basis of $\ker(A)$ and $\ker(B)$, i.e. the orthogonal complements of A and B , respectively, (8.32) is equivalent to

$$A_\perp^T P A_\perp < 0 \quad \text{and} \quad B_\perp^T P B_\perp < 0 \quad (8.33)$$

A proof can be found in, for example, [36].

8.3.3.4 Lemma 4 (Finsler's Lemma)

It can also be shown that (8.33) is equivalent to the following two inequalities, for some σ :

$$P - \sigma A^T A < 0$$

$$P - \sigma B^T B < 0$$

This result is sometimes referred to as *Finsler's Lemma* [15, 36].

8.3.4 The *S-Procedure* for Quadratic Functions

Many control problems can be formulated as constrained optimization problems with the cost as well as constraint inequalities in quadratic forms. Some of such problems may be re-formulated as a single LMI, by so-called the *S-procedure*.

Let F_0, F_1, \dots, F_p be quadratic functions of the variable vector x ,

$$F_i(x) = x^T T_i x + 2u_i^T x + v_i, \quad i = 0, \dots, p$$

where T_i is symmetric. F_0, F_1, \dots, F_p are required to satisfy

$$F_0(x) \geq 0 \quad \text{for all } x \text{ such that } F_i(x) \geq 0, \quad i = 1, \dots, p \quad (8.34)$$

Obviously, a sufficient condition for (8.34) to be true is that there exist $\tau_1 \geq 0, \dots, \tau_p \geq 0$ such that

$$\text{for all } x, \quad F_0(x) - \sum_{i=1}^p \tau_i F_i(x) \geq 0 \quad (8.35)$$

The *vice versa* is not true, except when $p = 1$ and with the condition that F_1 is strictly positive for some x .

(8.35) can be written in the form of an LMI as

$$\begin{bmatrix} T_0 & u_0 \\ u_0^T & v_0 \end{bmatrix} - \sum_{i=1}^p \tau_i \begin{bmatrix} T_i & u_i \\ u_i^T & v_i \end{bmatrix} \geq 0$$

Similar results exist when the cost inequality in (8.34) is strict.

8.3.5 Dualization Lemma

Let P be a non-singular symmetric matrix in $\Re^{n \times n}$ and let \mathcal{U} and \mathcal{V} be two complementary subspaces whose (direct) sum forms \Re^n . Then

$$x^T P x < 0 \quad \text{for all } x \in \mathcal{U} \setminus \mathcal{O} \quad \text{and} \quad x^T P x \geq 0 \quad \text{for all } x \in \mathcal{V}$$

is equivalent to

$$x^T P^{-1} x > 0 \quad \text{for all } x \in \mathcal{U}_\perp \setminus \mathcal{O} \quad \text{and} \quad x^T P^{-1} x \leq 0 \quad \text{for all } x \in \mathcal{V}_\perp.$$

The above is called the “Dualization Lemma” and may in some cases lead to a simpler procedure of LMI solutions.

This page intentionally left blank

Part II
Introduction to Robust Control
Toolbox v3

This page intentionally left blank

Chapter 9

Building Uncertain Models

In this chapter we show how to build uncertain system models using the functions of Robust Control Toolbox[®]3. Building such models is an important step in the design of control systems whose plants possess some type of uncertainty. The corresponding functions of Robust Control Toolbox[®]3 allow to facilitate the process of building different uncertainty models and to analyze easily the properties of such models. First we describe how to build models of open-loop and closed-loop linear time-invariant systems (LTI models) and how to investigate their basic properties. Then we present various functions of Robust Control Toolbox[®]3 that allow to create models of systems with structured (real) uncertainties. The usage of these functions is illustrated for the simple case of a second order mass–damper–spring system. It is shown how to investigate several properties of uncertain models in the time domain and frequency domain. Finally, we show how to build models of systems with unstructured (complex) uncertainty. Creation of models with additive or multiplicative uncertainty is considered in detail.

The presentation in this and the next three chapters of Part II is based to some extent on the information given in the Robust Control Toolbox[®]3 User's Guide [12] and its previous editions.

9.1 LTI Models

In this section we describe how to create and manipulate models of linear time-invariant systems (LTI models) in MATLAB[®].

Creation of LTI models is done by the following commands:

- `ss`—State-space models (SS objects)
- `tf`—Transfer function matrices (TF objects)
- `zpk`—Zero-pole-gain models (ZPK objects)
- `frd`—Frequency response data models (FRD objects)

Further on we show how to create models of multivariable systems by using the commands `ss` and `tf`.

Creation of a state-space model of time-invariant continuous-time system is illustrated by the following example.

Example 9.1 Let us consider a two-input, two-output system described in the state-space by the equations

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

where

$$A = \begin{bmatrix} -1 & 0 & 5 \\ 2 & 1 & -4 \\ -6 & -3 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 0 \\ -4 & 1 \\ 0 & 6 \end{bmatrix}$$

$$C = \begin{bmatrix} -1 & 0 & -4 \\ 2 & 3 & 6 \end{bmatrix}, \quad D = \begin{bmatrix} 3 & -2 \\ -4 & 1 \end{bmatrix}$$

This system is completely controllable and completely observable so that its state-space model represents a minimal realization.

After entering the matrices A , B , C , and D in MATLAB®, the state-space model is obtained by the command line

```
Gss = ss(A,B,C,D)
```

Entering the line

```
Gss
```

produces the following result:

```
a =
```

	x1	x2	x3
x1	-1	0	5
x2	2	1	-4
x3	-6	-3	-2

```
b =
```

	u1	u2
x1	5	0
x2	-4	1
x3	0	6

```
c =
```

	x1	x2	x3
y1	-1	0	-4
y2	2	3	6

d =

```

      u1  u2
y1    3  -2
y2   -4   1

```

Continuous-time model.

To display the state matrix A of the model we enter the line

```
A = Gss.A
```

In a similar way it is possible to display other matrices of the state-space model.

The state-space model is converted to a transfer function matrix model by the command line

```
Gtf = tf(Gss)
```

As a result one obtains

Transfer function from input 1 to output...

```

      3 s^3 + s^2 + 118 s - 80
#1:  -----
      s^3 + 2 s^2 + 17 s - 14

      -4 s^3 - 10 s^2 - 172 s + 144
#2:  -----
      s^3 + 2 s^2 + 17 s - 14

```

Transfer function from input 2 to output...

```

      -2 s^3 - 28 s^2 - 52 s + 109
#1:  -----
      s^3 + 2 s^2 + 17 s - 14

      s^3 + 41 s^2 - 4 s + 46
#2:  -----
      s^3 + 2 s^2 + 17 s - 14

```

It follows from the result obtained that the transfer function matrix of the system is

$$G = \begin{bmatrix} \frac{3s^3+s^2+118s-80}{s^3+2s^2+17s-14} & \frac{-2s^3-28s^2-52s+109}{s^3+2s^2+17s-14} \\ \frac{-4s^3-10s^2-172s+144}{s^3+2s^2+17s-14} & \frac{s^3+41s^2-4s+46}{s^3+2s^2+17s-14} \end{bmatrix}$$

The system is not strictly proper since the degree of numerator polynomials is equal to the degree of denominator polynomials.

To determine the poles and transmission zeros we may use the state-space model implementing the commands `pole` and `zero`. As a result we obtain


```

p = pole(Gss)

p =

    -1.3681 + 4.1404i
    -1.3681 - 4.1404i
     0.7363

z = zero(Gss)

z =

    -0.9087 +11.1357i
    -0.9087 -11.1357i
     2.2174

```

Note that the transmission zeros are not necessarily the zeros of the polynomials in the numerators of transfer function matrix elements.

The commands `pole` and `zero` may be used also to determine the poles and zeros from the transfer function matrix model. However, in this case we obtain

```

p = pole(Gtf)

p =

    -1.3681 + 4.1404i
    -1.3681 - 4.1404i
     0.7363
    -1.3681 + 4.1404i
    -1.3681 - 4.1404i
     0.7363

z = zero(Gtf)

z =

    -0.9087 +11.1357i
    -0.9087 -11.1357i
    -1.3681 + 4.1404i
    -1.3681 - 4.1404i
     2.2174
     0.7363

```

i.e., the poles and zeros of the original system are repeated twice. In this case instead of finding the poles and zeros of the third order system the commands `pole` and `zero` determine the poles and zeros of the state-space realization of sixth order that is obtained as an intermediate result in the poles and zeros computation. (The

computation of these quantities is done in the state-space.) This fact is confirmed executing the command line

```
G = ss(Gtf)
```

which produces a state-space realization from the transfer function matrix computed previously. The result of this command is

a =

	x1	x2	x3	x4	x5	x6
x1	-2	-2.125	1.75	0	0	0
x2	8	0	0	0	0	0
x3	0	1	0	0	0	0
x4	0	0	0	-2	-4.25	3.5
x5	0	0	0	4	0	0
x6	0	0	0	0	1	0

b =

	u1	u2
x1	4	0
x2	0	0
x3	0	0
x4	0	8
x5	0	0
x6	0	0

c =

	x1	x2	x3	x4	x5	x6
y1	-1.25	2.094	-1.188	-3	-0.5625	2.531
y2	-0.5	-3.25	2.75	4.875	-0.6562	1.875

d =

	u1	u2
y1	3	-2
y2	-4	1

Continuous-time model.

It is seen that after the inverse conversion from transfer function matrix to state-space model we obtain a sixth order realization. This realization is obviously non-minimal.

The minimal state-space realization may be obtained in the given case by the command

```
G = ss(Gtf, 'min')
```

The result is

a =

	x1	x2	x3
x1	-1.784	-5.23	2.632
x2	3.298	-0.3631	-0.01393
x3	0.2271	1.286	0.147

b =

	u1	u2
x1	0.6277	6.158
x2	2.925	-0.03289
x3	-0.1801	0.2784

c =

	x1	x2	x3
y1	-4.016	-0.691	2.544
y2	6.238	-1.907	1.88

d =

	u1	u2
y1	3	-2
y2	-4	1

To create a state-space model of the discrete-time system

$$x_{k+1} = Ax_k + Bu_k,$$

$$y_k = Cx_k + Du_k,$$

one may use again the command `ss` setting in addition the sampling period T_s :

$$Gd = ss(A, B, C, D, Ts)$$

The discretization of a continuous-time state-space model G for a given sampling period T_s may be done by the command `c2d`.

$$Gd = c2d(G, Ts)$$

Consider now how to obtain a model of a multivariable system described by a transfer function matrix.

Example 9.2 Given is a two-input two-output fifth order system with transfer function matrix

$$G = \begin{bmatrix} \frac{6}{(0.9s+1)(0.1s+1)} & \frac{-0.05}{0.1s+1} \\ \frac{0.07}{0.3s+1} & \frac{5}{(1.8s-1)(0.06s+1)} \end{bmatrix}$$

To enter this matrix we use the command lines

```
s = tf('s');
g11 = 6/((0.9*s + 1)*(0.1*s + 1));
g12 = -0.05/(0.1*s + 1);
g21 = 0.07/(0.3*s + 1);
g22 = 5/((1.8*s - 1)*(0.06*s+1));
G = [g11 g12; g21 g22]
```

As a result we obtain

Transfer function from input 1 to output...

```

              6
#1:  -----
      0.09 s^2 + s + 1

      0.07
#2:  -----
      0.3 s + 1
```

Transfer function from input 2 to output...

```

      -0.05
#1:  -----
      0.1 s + 1

              5
#2:  -----
      0.108 s^2 + 1.74 s - 1
```

A state-space realization of this transfer function matrix may be obtained as shown in Example 9.1 always trying to obtain a minimal realization.

The singular value plot of the system frequency response is obtained by the command `sigma`. (The same command is used for SS and TF models.) To obtain the singular values in the range 10^{-2} – 10^3 rad/s we enter the command line

```
sigma(G, {10^(-2) 10^3})
```

The system singular values are shown in Fig. 9.1. It is seen that below the frequency 1 rad/s the system gain remains constant.

Given two systems with transfer function matrices G_1 and G_2 , the transfer matrix of their series connection (provided that the first system has the same number of outputs as the inputs of second system) is found by the command line

```
G = G2*G1
```

The parallel connection of two systems G_1 and G_2 with equal number of inputs and equal number of outputs is found by the command line

```
G = G1 + G2
```

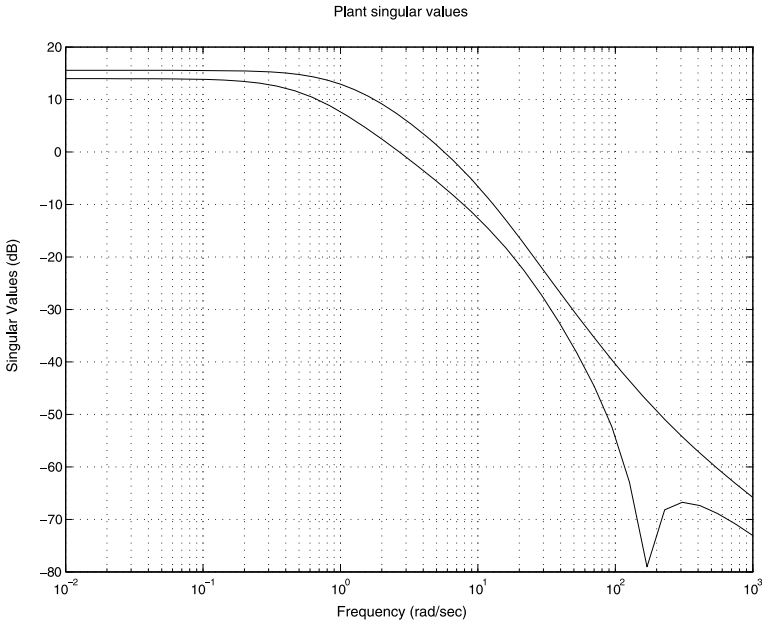


Fig. 9.1 System singular value plot

Coprime factorization of a transfer function matrix may be done by the command `ncfmr`.

The \mathcal{H}_∞ system norm may be obtained using the singular value plot of the frequency response. However, this may lead to errors in case of lightly damped systems so that it is better to use the function `norm(G, 'inf')`, which performs the computation of \mathcal{H}_∞ -norm of G in the state space.

Example 9.3 Consider the 2×2 transfer function matrix

$$G(s) = \begin{bmatrix} \frac{10(s+1)}{s^2+0.2s+100} & \frac{1}{s+1} \\ \frac{s+2}{s^2+0.1s+10} & \frac{5(s+1)}{(s+2)(s+3)} \end{bmatrix}$$

The singular values of $G(j\omega)$ are shown in Fig. 9.2. From the largest singular value we obtain the result that its maximum is about 34 dB which corresponds to \mathcal{H}_∞ norm equal to 50.12. More accurate computation of the norm may be done by the command line

```
norm(G, 'inf')
```

which gives

```
ans =
```

```
50.2471
```

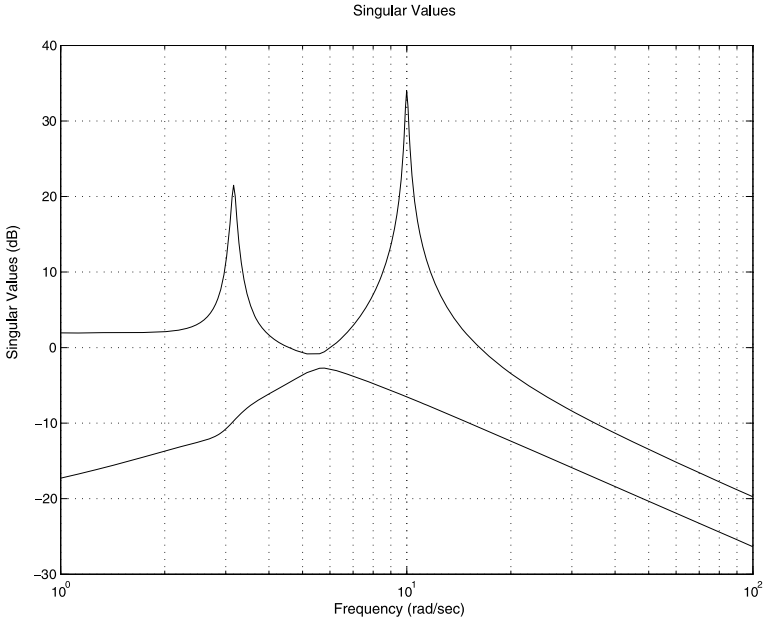


Fig. 9.2 Largest and smallest singular values of $G(j\omega)$

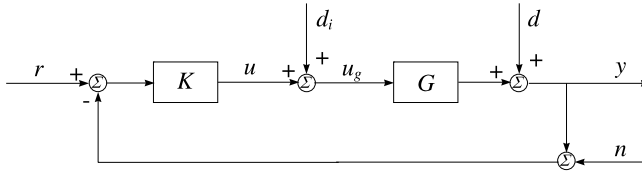


Fig. 9.3 Multivariable feedback system

Consider now the block-diagram of a multivariable feedback system shown in Fig. 9.3. The system consists of plant G and controller K and has reference r , sensor noise n , input disturbance d_i , and output disturbance d . All signals are assumed to be vector signals and the transfer function matrices have appropriate dimensions.

If the closed-loop system is internally stable, it satisfies the following equations:

$$y = T_o(r - n) + GS_id_i + S_o d \quad (9.1)$$

$$r - y = S_o(r - d) + T_o n - GS_id_i \quad (9.2)$$

$$u = KS_o(r - n) - KS_o d - T_i d_i \quad (9.3)$$

$$u_g = KS_o(r - n) - KS_o d + S_i d_i \quad (9.4)$$

where we make use of the following notation. The *input loop transfer matrix* L_i and the *output loop transfer matrix* L_o are defined, respectively, as

$$L_i = KG, \quad L_o = GK$$

The *input sensitivity matrix* is defined as the transfer function matrix from d_i to u_g :

$$S_i = (I + L_i)^{-1}$$

The *output sensitivity matrix* is defined as the transfer function matrix from d to y :

$$S_o = (I + L_o)^{-1}$$

The *input and output complementary sensitivity matrix* are defined as

$$\begin{aligned} T_i &= I - S_i = L_i(I + L_i)^{-1} \\ T_o &= I - S_o = L_o(I + L_o)^{-1} \end{aligned}$$

respectively.

The sensitivity transfer matrices are called also *sensitivity functions*.

The computation of these transfer function matrices and corresponding frequency responses is illustrated by the following example.

Example 9.4 Consider a two-channel system (i.e., a system with two inputs and two outputs) with a structure shown in Fig. 9.3. The plant is of fifth order with two inputs and two outputs and has a transfer function matrix

$$G = \begin{bmatrix} \frac{6}{(0.9s+1)(0.1s+1)} & \frac{-0.05}{0.1s+1} \\ \frac{0.07}{0.3s+1} & \frac{5}{(1.8s-1)(0.06s+1)} \end{bmatrix}$$

(How to enter this transfer function matrix was shown in Example 9.2.)

The system controller has the transfer function matrix

$$K = \begin{bmatrix} \frac{7(s+1)}{0.3s+1} & 0 \\ 0 & \frac{18(s+2)}{s+1} \end{bmatrix}$$

The closed-loop input and output sensitivity functions and input and output loop transfer matrices are found by the command lines

```
looptransfer = loopsens(G,K);
Si = looptransfer.Si;
Ti = looptransfer.Ti;
So = looptransfer.So;
To = looptransfer.To;
Li = looptransfer.Li;
Lo = looptransfer.Lo;
```

It should be taken into account that the command `loopsens` does not produce the transfer matrices themselves but their realizations in the state-space. If necessary, the corresponding transfer matrices may be obtained from these realizations by the command `tf`.

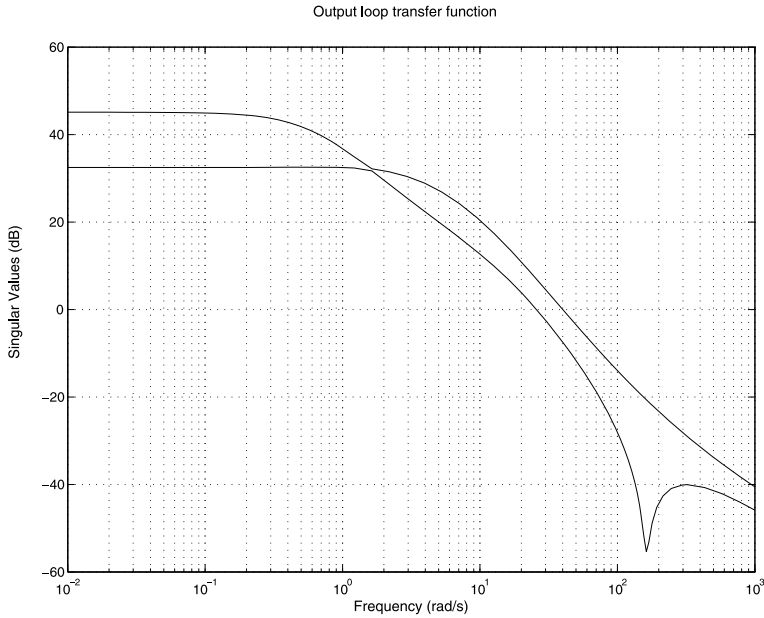


Fig. 9.4 Output loop transfer functions

Using the command line

```
poles = looptransfer.Poles
```

one finds the closed-loop poles and using the command line

```
stab = looptransfer.Stable
```

one checks the closed-loop system stability ($\text{stab} = 1$ means that the system is stable and $\text{stab} = 0$ means that the system is unstable).

The singular value plots of output loop transfer matrix L_o are shown in Fig. 9.4.

The singular value plots of input sensitivity and complementary sensitivity functions are shown in Fig. 9.5. In the low frequency range the singular values of the matrix $S_i(j\omega)$ are below -30 dB, which means that the input disturbance d_i is attenuated more than 30 times at plant input.

It should be noted that in the multivariable case the sensitivity functions with respect to the input and output are generally different ($S_o \neq S_i$, $L_o \neq L_i$).

The singular value plots of output sensitivity and complementary sensitivity functions are shown in Fig. 9.6. It is seen that for frequencies above 1000 rad/s the singular values of matrix $T_o(j\omega)$ are below -40 dB, i.e., the noise n in this frequency range is suppressed more than 100 times at plant output.

It is instructive to see also the sensitivity of the plant input to sensor noise. According to (9.4) this sensitivity is determined by the singular values of $K(j\omega)S_o(j\omega)$ which are shown in Fig. 9.7. It is seen that the maximum of the sensitivity is more than 30 dB for a frequency close to 40 rad/s and it is not less than 25 dB for frequencies greater than 100 rad/s. This means that *while the sensor noises are suppressed*

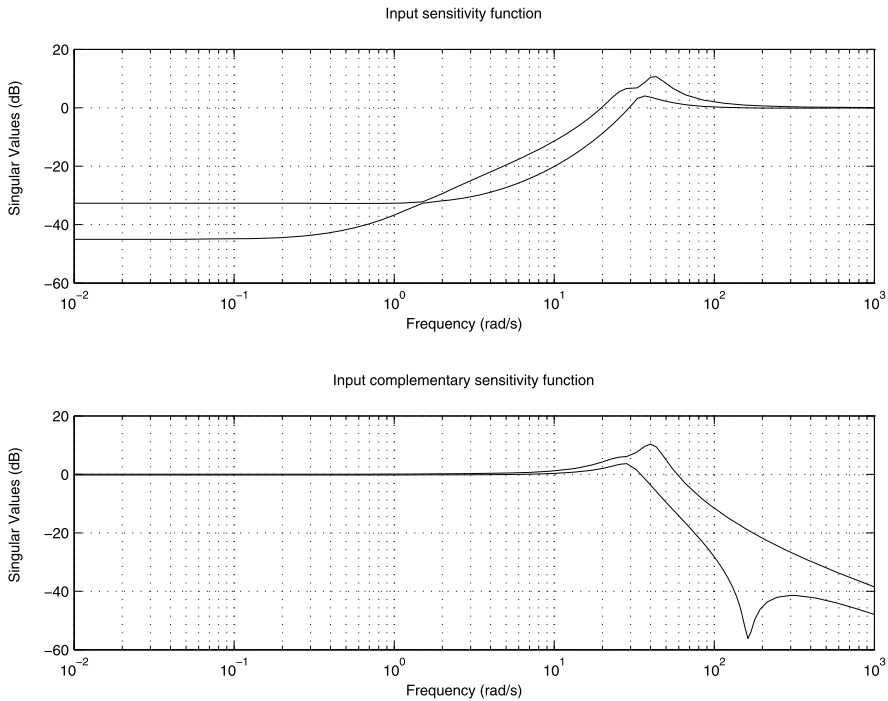


Fig. 9.5 Input sensitivity and complementary sensitivity

more than 100 times at plant output, they are amplified 18 times at plant input. The amplification of system noises at plant input is undesirable since it may lead to saturation of actuators and introduces nonlinear effect in closed-loop system behavior.

Consider now how to obtain the transient responses of the closed-loop system.

Example 9.5 Let us consider again the system described in Example 9.4. The transient response of the first output y_1 due to the step reference r_1 is obtained according to (9.2) by the command lines

```
tfin = 1;
time = 0:tfin/500:tfin;
nstep = size(time,2);
ref1(1:nstep) = 1.0;
ref2(1:nstep) = 0.0;
ref = [ref1' ref2'];
[y,t] = lsim(To(1:2,1:2),ref,time);
plot(t,y(:,1),'r-') grid
title('From ref 1 to outp 1')
xlabel('Time (secs)')
ylabel('y_1')
```

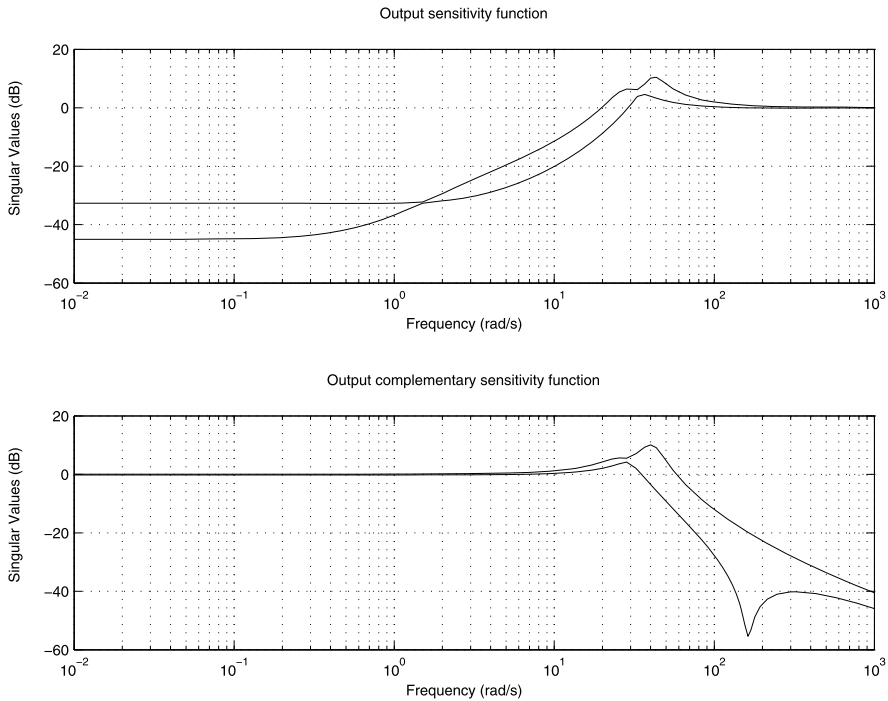


Fig. 9.6 Output sensitivity and complementary sensitivity

In $y(:,2)$ one obtains the transient response of the second output y_2 due to the first reference. Setting $\text{ref1}(1:\text{nstep}) = 0.0$; $\text{ref2}(1:\text{nstep}) = 1.0$; one obtains the transient responses due to the second reference.

The closed-loop transient responses due to step references are shown in Fig. 9.8. It is seen that there is some influence between the channels (i.e., they are not fully decoupled), due to the presence of off-diagonal elements in the plant transfer matrix and the diagonal structure of the controller transfer matrix. The overshoot of the first output does not exceed 50 %.

Since the relationship between the input disturbances and closed-system outputs is given by $y = GS_i d_i$ the transient response of the first output due to step input disturbance d_1 is obtained by the commands

```
time = 0:tfin/500:tfin;
nstep = size(time,2);
dist1(1:nstep) = 1.0;
dist2(1:nstep) = 0.0;
dist = [dist1' dist2'];
sys = G*Si;
[y,t] = lsim(sys(1:2,1:2),dist,time);
plot(t,y(:,1),'r-') grid
title('From dist 1 to output 1')
```

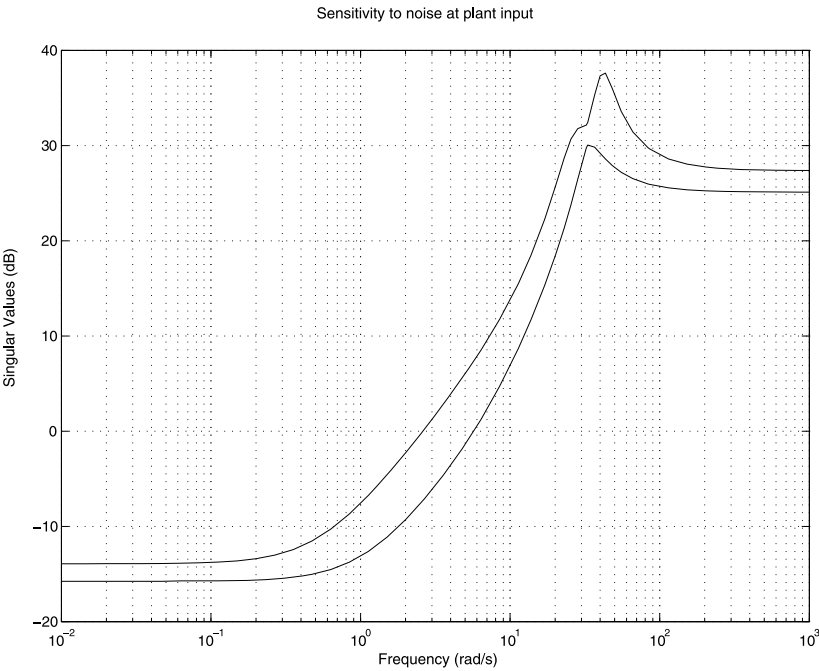


Fig. 9.7 Sensitivity to noise at plant input

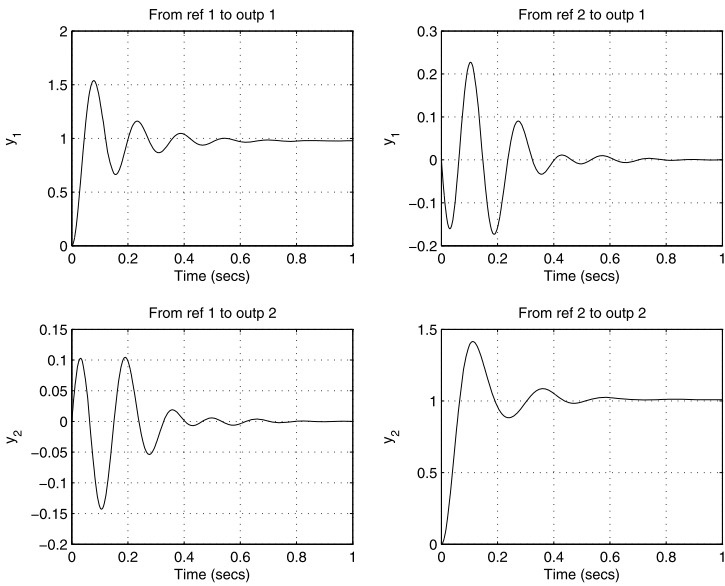


Fig. 9.8 Transient responses to reference

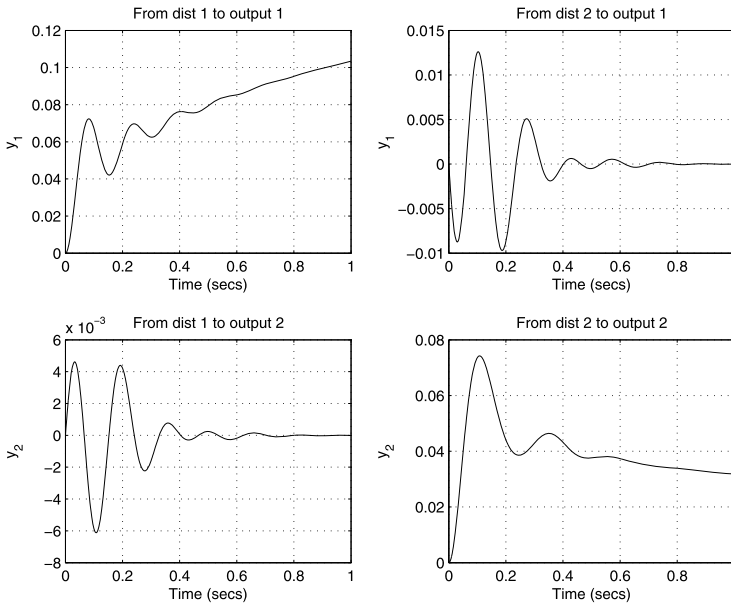


Fig. 9.9 Transient responses due to input disturbances

```
xlabel('Time (secs)')
ylabel('y_1')
```

The transient response of the second output due to the first input disturbance d_1 is obtained in $y(:, 2)$.

Similarly one may obtain the transient responses due to the output disturbances.

The closed-loop transient responses due to input and output disturbances are shown in Figs. 9.9 and 9.10, respectively.

9.2 Structured Uncertainty Models

In this section we shall consider how to build models with parametric (real) uncertainty by implementing various ways to describe such models available in Robust Control Toolbox[®]3. In our presentation we shall use a simple, second order, mechanical system, namely a *mass–damper–spring system*. The mass–damper–spring system is a common control experimental device frequently seen in an undergraduate teaching laboratory.

The *one-degree-of-freedom* (1DOF) mass–damper–spring system is depicted in Fig. 9.11.

The dynamics of such a system can be described by the following second order differential equation, by Newton's Second Law,

$$m\ddot{x} + c\dot{x} + kx = u \quad (9.5)$$

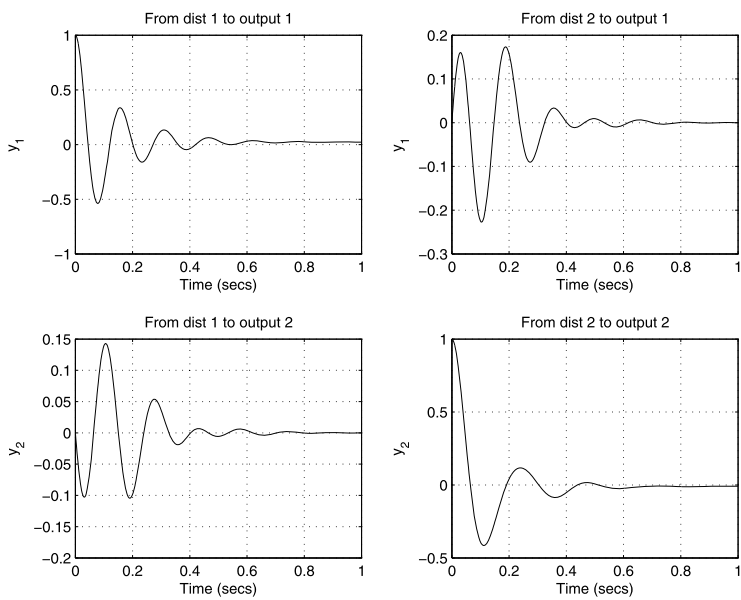


Fig. 9.10 Transient responses due to output disturbances

Fig. 9.11
Mass–damper–spring system

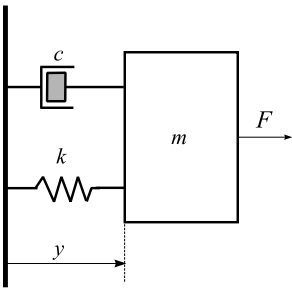
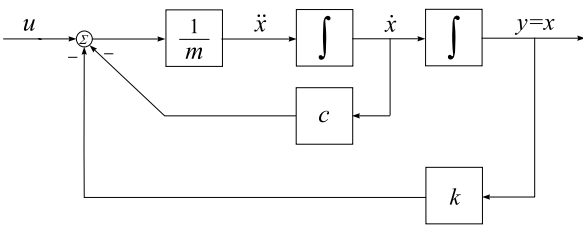


Fig. 9.12 Block diagram of the mass–damper–spring system



where x is the displacement of the mass block from the equilibrium position and $u = F$ is the force acting on the mass, with m the mass, c the damper constant and k the spring constant.

A block diagram of such a system is shown in Fig. 9.12, setting $y = x$.

Denoting $x_1 = x$, $x_2 = dx/dt$, the system (9.5) may be described in the state-space form

$$\dot{x} = Ax + Bu \quad (9.6)$$

$$y = Cx + Du \quad (9.7)$$

where

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0$$

The transfer function of this system is given by

$$\frac{y(s)}{u(s)} = \frac{1}{ms^2 + cs + k} \quad (9.8)$$

In a realistic system, the three physical parameters m , c , and k are not known exactly. However, it can be assumed that their values are within certain, known intervals. That is,

$$m = \bar{m}(1 + p_m\delta_m), \quad c = \bar{c}(1 + p_c\delta_c), \quad k = \bar{k}(1 + p_k\delta_k)$$

where $\bar{m} = 3$, $\bar{c} = 1$, $\bar{k} = 2$ are called *nominal values* of m , c , and k . The quantities p_m , p_c , and p_k , and δ_m , δ_c , and δ_k represent the possible (relative) perturbations on these three parameters. In the present case, we let $p_m = 0.4$, $p_c = 0.2$, $p_k = 0.3$ and $-1 \leq \delta_m, \delta_c, \delta_k \leq 1$. Note that this represents up to 40 % uncertainty in the mass, 20 % uncertainty in the damping coefficient and 30 % uncertainty in the spring stiffness.

The command lines for building uncertain models of mass-damper-spring system and accessing their properties may be found in the M-file `mds_models.m`.

9.2.1 Uncertain Real Parameters

The simplest way to describe uncertain real parameter in Robust Control Toolbox[®]3 is to use the function `ureal`. The syntax of this function is

```
p = ureal('Name',NominalValue,'Property1',Value1, ...
          'Property2',Value2,...)
```

The uncertain real parameter will have a name, set by 'Name' and a nominal value set by `NominalValue`.

The maximum deviation from `NominalValue` may be set in three different manners (by using the corresponding 'Property'):

- **PlusMinus**: the absolute deviation from `NominalValue`
- **Range**: the interval containing `NominalValue`
- **Percentage**: the percentage deviation from `NominalValue`

The `Mode` property specifies which one of these three descriptions remains unchanged if the `NominalValue` is changed. The possible values of `Mode` are 'Range', 'Percentage' and 'PlusMinus', the default value of `Mode` being 'PlusMinus' with value `[-1 1]`.

The property `AutoSimplify` of `ureal` specifies how to simplify expressions, containing uncertain parameters and is useful in cases when some parameters appear several times. Its default value is `basic` which means that elementary methods for algebraic simplification are implemented. Other possible values are `off` (without simplification) and `full` (model-reduction methods are applied to the uncertain object).

Using the function `ureal` the uncertain parameters of the mass-damper-spring system may be set by the following lines:

```
m = ureal('m',3,'Percentage',[-40, 40])
c = ureal('c',1,'Range',[0.8, 1.2])
k = ureal('k',2,'PlusMinus',[-0.6, 0.6])
```

As a result the following information is displayed:

```
Uncertain Real Parameter: Name m, NominalValue 3,
                           variability = [-40  40]%
Uncertain Real Parameter: Name c, NominalValue 1,
                           Range [0.8  1.2]
Uncertain Real Parameter: Name k, NominalValue 2,
                           variability = [-0.6  0.6]
```

9.2.2 Uncertain State-Space Systems

Once the uncertain parameters are set, there are several ways to obtain the uncertain system description.

First, it is possible to use the state-space description (9.6), (9.7). Entering the lines

```
A = [ 0      1
      -k/m  -c/m];
B = [0      1/m]';
C = [1      0];
D = 0;
uss1 = ss(A,B,C,D)
```

we obtain an object `uss1` from the class `uss`—*uncertain state space*. It should be noted that in the case of uncertain parameters the resulting system is always from the class `uss` no matter which function is used to build the system. In the given case we obtain

```
USS: 2 States, 1 Output, 1 Input, Continuous System
c: real, nominal = 1, range = [0.8  1.2], 1 occurrence
```

```

k: real, nominal = 2, variability = [-0.6  0.6],
                                     1 occurrence
m: real, nominal = 3, variability = [-40  40]%,
                                     3 occurrences

```

This result shows that the parameter m appears in the system `uss1` three times which means that the state-space realization obtained does not contain the minimum possible number of the parameter m . The usage of the command line

```
simplify(uss1, 'full')
```

does not help in the given case to produce a model in which the parameter m appears again three times.

Another possibility to obtain the uncertain state-space description is to use the transfer function (9.8). Entering the line

```
uss2 = tf(1,[m,c,k])
```

we obtain the uncertain object `uss2`. Here we use the extended capabilities of the function `tf` to work with uncertain parameters. In the given case we obtain

```

USS: 2 States, 1 Output, 1 Input, Continuous System
c: real, nominal = 1, range = [0.8  1.2], 1 occurrence
k: real, nominal = 2, variability = [-0.6  0.6],
                                     1 occurrence
m: real, nominal = 3, variability = [-40  40]%,
                                     1 occurrence

```

Note that the parameter m appears in the system `uss2` only once.

It is possible also to build the uncertain system describing directly the block-diagram presented in Fig. 9.12 by using the function `feedback`.

```

s = tf('s');
g1 = (1/s)/m;
int2 = 1/s;
uss3 = feedback(int2*feedback(g1,c),k)

```

The result is again an object of class `uss`:

```

USS: 2 States, 1 Output, 1 Input, Continuous System
c: real, nominal = 1, range = [0.8  1.2], 1 occurrence
k: real, nominal = 2, variability = [-0.6  0.6],
                                     1 occurrence
m: real, nominal = 3, variability = [-40  40]%,
                                     1 occurrence

```

Here the three uncertain parameters participate only once.

9.2.3 Properties of Uncertain Systems

The properties of uncertain object `uss1` may be seen by entering the line

```
get(uss1)
```


which gives

```

        a: [2x2 umat]
        b: [2x1 umat]
        c: [1 0]
        d: 0
    StateName: {2x1 cell}
        Ts: 0
    InputName: {' '}
    OutputName: {' '}
    InputGroup: [1x1 struct]
    OutputGroup: [1x1 struct]
        Name: ''
    NominalValue: [1x1 ss]
    Uncertainty: [1x1 atomlist]
        Notes: {}
    UserData: []

```

The command

```
uss1.Uncertainty
```

produces

```

    c: [1x1 ureal]
    k: [1x1 ureal]
    m: [1x1 ureal]

```

and the command

```
uss1.NominalValue
```

produces the nominal state-space description

a =

```

           x1      x2
x1         0      1
x2 -0.6667 -0.3333

```

b =

```

           u1
x1         0
x2  0.3333

```

c =

```

    x1  x2
y1   1   0

```

d =

```

    u1

```

```
y1    0
```

Continuous-time model.

The uncertain elements of uncertain object may be substituted by specific values using the function `usubs`. The command line

```
B = usubs(A,name1,value1,name2,value2,...)
```

instantiates the uncertain elements `name1`, `name2`, ... of `A` to the values `value1`, `value2`, In particular, value is set to 'NominalValue' or 'Random' in order to use the nominal value or a random instance of a particular element. For instance, the nominal state-space description of `uss1` may be obtained by

```
usubs(uss1,'m',3,'c',1,'k',2)
```

Another useful command implemented on uncertain objects is the command `usample` which randomly samples the uncertain system at a specified number of points. The command line

```
B = usample(A,n)
```

picks `n` random samples of the uncertainty in `A` and returns these samples in an array `B` of size `[size(A) n]`.

It is interesting to see the frequency responses of the mass-damper-spring system for different values of the parameters m , c , and k . For this aim we may use the extended capabilities of the function `bode` in MATLAB®7. For objects of the class `uss` this function plots a family of frequency responses for 20 random values of the uncertain elements as well as for the nominal values. In the given case we obtain

```
w = logspace(-1,1,200);
figure(1)
bode(uss1,w)
title('Bode plot of uncertain system')
grid
```

The result is shown in Fig. 9.13.

This sampling is done also when the functions `bodemag`, `impulse`, `nyquist` and `step` are called on a `uss` object. The command line

```
step(uss1), grid
```

produces the step responses of the mass-damper-spring system for 20 random values of the uncertain elements as well as for the nominal values. These step responses are shown in Fig. 9.14.

The command lines

```
frres = frd(uss2,w)
nyquist(frres), grid
```

produce the Nyquist diagram of the uncertain system `uss2`, shown in Fig. 9.15. The uncertain system `uss2` is sampled at 20 values, as well as its nominal value, and the Nyquist plots of these 21 systems are drawn.

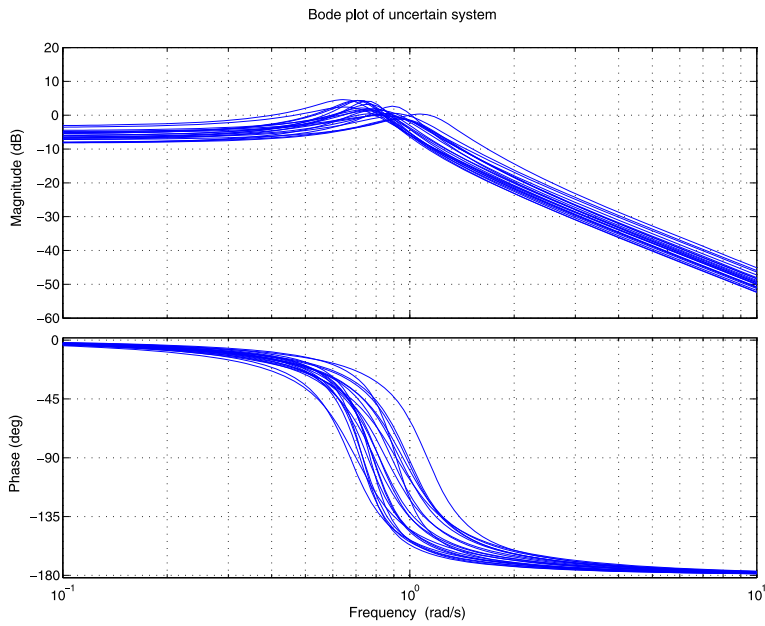


Fig. 9.13 Bode plot of uncertain system

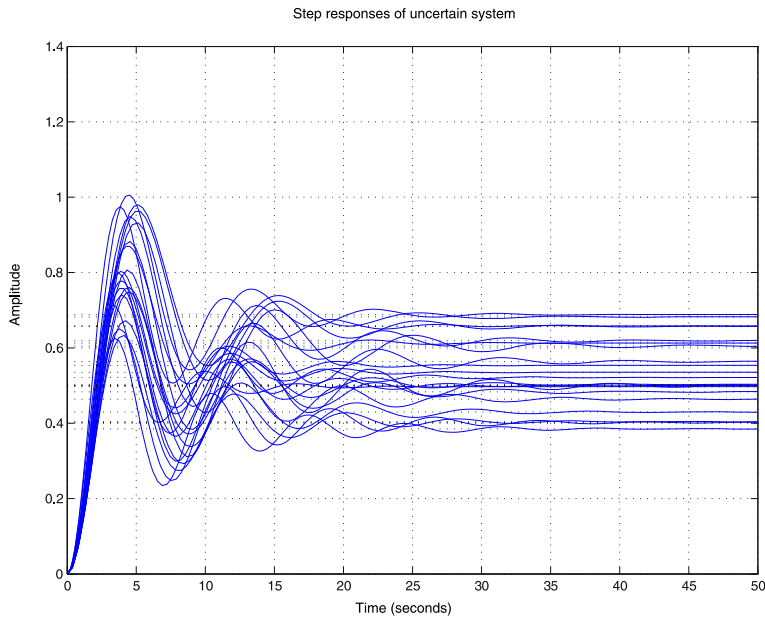


Fig. 9.14 Step responses of uncertain system

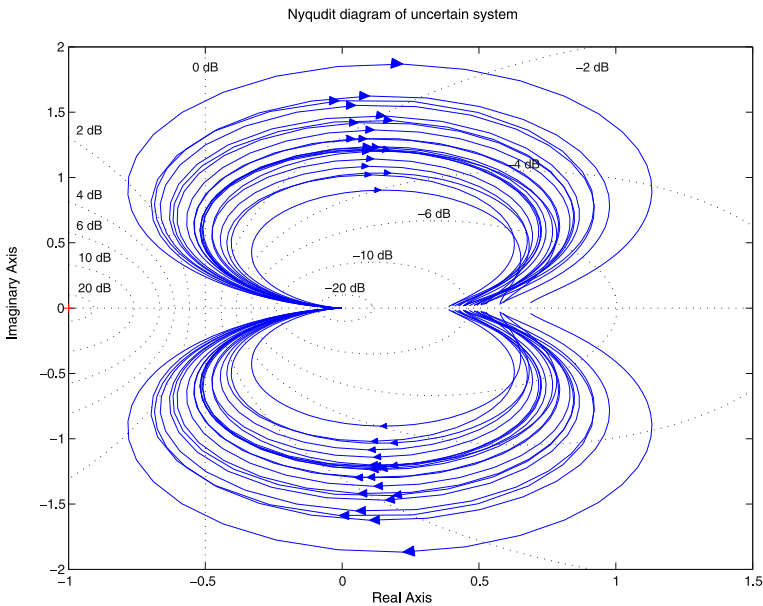


Fig. 9.15 Nyquist diagram of uncertain system

Sometimes it is necessary to analyze the behavior of an uncertain system for uniformly spaced values of its uncertain real parameters. In such a case it is appropriate to use the command `gridureal`. The command line

```
B = gridureal(A,n)
```

substitutes n uniformly spaced samples of the uncertain real parameters in uncertain object A . The size of array B is equal to $[size(A) \ n]$. The n samples are generated by uniformly gridding each `ureal` parameter in A across its range. If A includes uncertain objects other than `ureal`, then B is an uncertain object.

The command line

```
[B,SampleValues] = gridureal(A, names1, n1, names2, n2, ...)
```

takes $n1$ samples of the uncertain real parameters listed in `names1`, and $n2$ samples of the uncertain real parameters listed in `names2` and so on. `size(B)` will equal $[size(A) \ n1 \ n2 \ \dots]$.

As an example consider how to obtain the step response of the mass-damper-spring system for a grid of 50 values altogether of the uncertain parameters m , c and k . The line

```
step(gridureal(uss3,50)), grid
```

produces the step response of the uncertain system `uss3` for a grid of 50 values of the parameters m , c , and k , shown in Fig. 9.16.

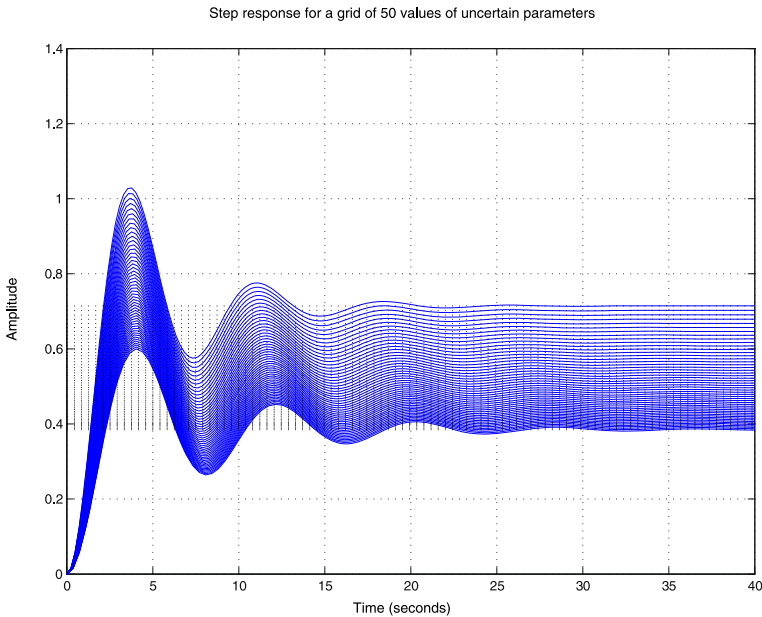


Fig. 9.16 Step response for a grid of 50 values of uncertain parameters

9.2.4 Other Functions to Build Uncertain Models

There are some additional functions that may be useful in building uncertain models.

The command `ucomplex` creates uncertain complex parameter. The syntax of this command resembles the syntax of command `ureal`.

```
B = ucomplex('Name',NominalValue,'Property1',Value1,...
             'Property2',Value2,...)
```

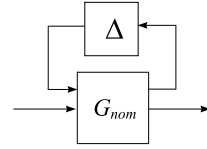
Here the uncertainty (maximum deviation) is specified in two different manners: by the property `Radius` (radius of disc centered at `NominalValue`), or by the property `Percentage` (disc size is percentage of magnitude of `NominalValue`).

Uncertain matrices (class `umat`) are built from uncertain parameters using MATLAB[®] matrix building syntax. These matrices can be added, subtracted, multiplied, inverted, transposed, etc., resulting in uncertain matrices. The rows and columns of an uncertain matrix are referenced in the same manner that MATLAB[®] references rows and columns of an array, using parentheses, and integer indices. Similarly to the case of real uncertain systems, specific values may be substituted for any of the uncertain parameters within a `umat` using the command `usubs`. The command `usample` generates a random sample of the uncertain matrix, substituting random samples (within their ranges) for each of the uncertain parameters.

Uncertain complex matrices are built with the command `ucomplexm`.

The command `frd` to compute frequency responses works also for objects of class `uss`. In this case one obtains object of class `ufrd`—uncertain frequency re-

Fig. 9.17 Decomposition of uncertain object



sponse. As an example, the frequency response of the system `uss1` may be obtained by the lines

```
w = logspace(-1,1,200);
freqs = frd(uss1,w)
```

The result is

```
UFRD: 1 Output, 1 Input, Continuous System, 200 Frequency
      points
      c: real, nominal = 1, range = [0.8 1.2], 1 occurrence
      k: real, nominal = 2, variability = [-0.6 0.6],
      1 occurrence
      m: real, nominal = 3, variability = [-40 40]%,
      3 occurrences
```

Now the line

```
bode(freqs)
```

reproduces the Bode plot, shown in Fig. 9.13.

Uncertain state-space system may be included in Uncertain State Space block and used in Simulink® to simulate and linearize uncertain systems. The reader is referred to [12, Chap. 5] for more information and examples on the representation and analysis of uncertain systems in Simulink®.

9.2.5 Decomposing Uncertain Objects

Each uncertain object (`umat`, `uss`, `ufrd`) may be represented as a linear fractional transformation of non-uncertain part and a matrix containing the uncertain parameters. Using the command `lftdata` an uncertain object may be decomposed into a nominal part and a normalized uncertainty matrix with \mathcal{H}_∞ norm, equal to 1. For a given uncertain object `usys` the command line

```
[Gnom,Delta,Blkstruct,Normunc] = lftdata(usys)
```

produces a nominal system G_{nom} and an uncertainty matrix Δ connected in a linear fractional transformation (Fig. 9.17). The variable `Blkstruct` returns $n \times 1$ structure, where `Blkstruct(i)` describes the i th normalized uncertain element. The variable `Normunc` returns an array of normalized uncertain elements.

Let us decompose the uncertain system `uss2`:

```
[Gnom,Delta,Blkstruct,Normunc] = lftdata(uss2)
```

The variable `Gnom` is obtained as a state-space description of class `ss`, since it does not contain uncertain parameters, and `Delta` is of class `umat` containing the uncertain elements `m`, `c` and `k`.

9.3 Building Uncertain Models Using `iconnect` and `sysic`

The class `iconnect` is used to build complex interconnections of uncertain matrices and systems. The `iconnect` objects have three fields that should be set by the user.

- Input: symbolic column vector, represents the input variables;
- Output: symbolic column vector, represents the output variables;
- Equation: cell-array of equality constraints (created using the command `equate`) describing the relations between input, output and intermediate variables.

Initially, an empty object of class `iconnect` is created, and then the fields `Input`, `Output`, and `Equation` are specified.

The input, output, and intermediate variables should be objects of class `icsignal`. They are symbolic vectors representing the signals in the interconnection. For instance, the command

```
icsignal(4)
```

creates a four element column vector for use by `iconnect`.

Let us describe the mass–damper–spring system interconnection, shown in Fig. 9.12, by using `iconnect` assuming that the uncertain parameters `m`, `c` and `k` are already defined. For intermediate variables we choose x and \dot{x} . The input, output, and intermediate variables are created by using `icsignal`.

```
u = icsignal(1)
x = icsignal(1)
xdot = icsignal(1)
iconnect1 = iconnect;
iconnect1.Input = u;
iconnect1.Output = x;
iconnect1.Equation{1} = equate(x, tf(1,[1,0])*xdot);
iconnect1.Equation{2} = equate(xdot, tf(1,[m,0])*
                                (u-k*x-c*xdot));
```

With `get(iconnect1)` one may see the properties of the system obtained:

```
Equation: {[1x1 icsignal] [1x1 icsignal]}
Input: [1x1 icsignal]
Output: [1x1 icsignal]
System: [1x1 uss]
```

As a result of the command

```
iconnect1.System
```

we see that system `iconnect1` is of the class `uss`:

```
USS: 2 States, 1 Output, 1 Input, Continuous System
  c: real, nominal = 1, range = [0.8 1.2], 1 occurrence
  k: real, nominal = 2, variability = [-0.6 0.6],
                                     1 occurrence
  m: real, nominal = 3, variability = [-40 40]%,
                                     1 occurrence
```

As an alternative of the command `iconnect` to build complex interconnections of uncertain systems, one may use the command `sysic`. In order to implement this command, it is necessary to enter into the workspace of MATLAB® the following variables: `systemnames`, `inputvar`, and `outputvar`. These variables are character strings (`char`'s) with the following meaning.

- `systemnames` is a `char` containing the names of systems that should be connected. They have to be separated by spaces with no additional punctuation. The systems specified in the `char` should already exist in the workspace.
- `inputvar` is a `char`, specifying the external inputs to the interconnection (the names are formal, i.e., it is not necessary that these variables exist). The `char` should have the form of a column vector (i.e., enclosed in square brackets `[]`). The dimension of each input in the list may be set in braces `{ }`, for instance `U1{3}`.
- `outputvar` is a `char`, containing the system outputs or actually the equations, used to produce the output variables (they have no names); the outputs are obtained as linear combinations of external inputs and subsystem outputs described in `systemnames`. For multivariable subsystems, arguments within parentheses specify which subsystem outputs are to be used and in what order. For instance, the expression `G(2:3,5)` specifies output 2, 3, and 5 from the subsystem `G`.

For each of the subsystems, specified in `systemnames`, it is necessary to define a separate variable describing its inputs in a way valid for `outputvar`. The names of these variables should begin with `input_to_` and to finish with the subsystem name.

Then the command

```
SysOutName = sysic
```

reads the subsystem names specified in `systemnames`, the interconnection descriptions, and computes the description of the resulting system with name `SysOutName`. Also a check of the signal number is done.

For the mass–damper–spring system, description with `sysic` may be obtained as follows (it is assumed that the uncertain parameters `m`, `c` and `k` already exist in the workspace).

```
m1 = inv(m);
int1 = 1/s;
int2 = tf(1,[1,0]);
systemnames = 'int1 int2 c k m1';
```



```

inputvar = '[u]';
outputvar = '[int2]';
input_to_int1 = '[m1]';
input_to_int2 = '[int1]';
input_to_c = '[int1]';
input_to_k = '[int2]';
input_to_m1 = '[u-c-k]';
uss4 = sysic

```

Note that to obtain $\frac{1}{m}$ we use the function `inv`, since m contains uncertainty and the direct division $\frac{1}{m}$ leads to an error. Also, two different but equivalent ways to define the integrators `int1` and `int2` are implemented.

The system `uss4` obtained is also of class `uss`.

The interconnections built with `sysic` are componentwise minimal in the sense that the state dimension of the interconnection equals the sum of the state dimensions of the components. This makes `sysic` appropriate for building complex uncertainty models.

9.4 Unstructured Uncertainty Models

The unstructured (complex) uncertainty models are built using the function `ultidyn`. The uncertain linear, time-invariant dynamics object `ultidyn` represents an unknown linear system whose only known attribute is a magnitude bound on its frequency response. The syntax of this function is

```

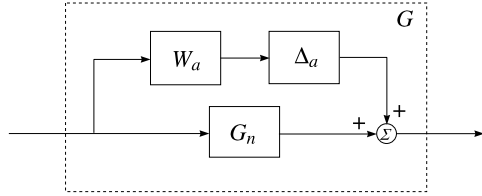
H = ultidyn('Name',ioSize,'Property1',Value1, ...
            'Property2',Value2,...)

```

The uncertain dynamic object will have a name, set by `'Name'` and size `ioSize`, [number-of-outputs number-of-inputs]. The property `Type` specifies whether the known attributes about the frequency response are related to gain or phase and has value `'GainBounded'` or `'PositiveReal'`, respectively. The default value is `'GainBounded'`. The property `Bound` is a single number, which, along with `Type`, completely specifies what is known about the uncertain system frequency response. Specifically, if Δ is an `ultidyn` object, and if γ denotes the value of the `Bound` property, then the object represents the set of all stable, linear, time-invariant systems whose frequency response satisfies the conditions:

- If `Type` is `'GainBounded'`, $\sigma_{\max}[\Delta(\omega)] \leq \gamma$ for all frequencies, where $\sigma_{\max}[\Delta(\omega)]$ is the maximum singular value of $\Delta(\omega)$. When `Type` is `'GainBounded'`, the default value for `Bound` (i.e., γ) is 1.
- If `Type` is `'PositiveReal'`, $\Delta(\omega) + \Delta^*(\omega) \geq \gamma$ for all frequencies, where $\Delta^*(\omega)$ denotes the Hermitian conjugate matrix of $\Delta(\omega)$. When `Type` is `PositiveReal`, the default value for `Bound` (i.e., γ) is 0.

Fig. 9.18 Plant with additive uncertainty



The property `SampleStateDim` is a positive integer, defining the state dimension of random samples of the uncertain object when sampled with `usample`. The default value is 1.

The property `AutoSimplify` of `ultidyn` is similar to the property `AutoSimplify` of the function `ureal`.

For instance, the command line

```
uns = ultidyn('uns',[3 2],'Type','GainBounded','Bound',1.8)
```

creates a `ultidyn` object with internal name `uns`, dimension 3-by-2 (three outputs, two inputs), norm bounded by 1.8.

Objects of class `ultidyn` may be used together with `uss` objects to create uncertain systems by using `interconnect` or `sysic` functions along with basic system interconnection functions defined in Control System Toolbox (`append`, `blkdiag`, `series`, `parallel`, `feedback`, `lft`, and `stack`).

9.4.1 Models with Additive Uncertainty

The following example illustrates the using of Robust Control Toolbox^{®3} commands to obtain a model of SISO system with additive uncertainty.

Example 9.6 Consider a first order plant with uncertainty in the gain and time constant:

$$G(s) = \frac{K}{Ts + 1}, \quad 0.8 \leq K \leq 1.2, \quad \frac{0.7}{15} \leq T \leq \frac{1.3}{15}$$

The nominal model is taken as

$$G_n(s) = \frac{1}{\frac{1}{15}s + 1}$$

The model with additive uncertainty is chosen as

$$G(s) = G_n(s) + \Delta_a(s)W_a(s)$$

where

$$|\Delta_a(j\omega)| \leq 1$$

The block-diagram of the plant with additive uncertainty is shown in Fig. 9.18.

To determine the weighting transfer function $W_a(s)$ we compute the error frequency response $G(j\omega) - G_n(j\omega)$ for different values of parameters K and T . This is done with commands

```
omega = logspace(-1,3,100);
%
K0 = 1.0; T0 = 1/15;
Gnom = tf([K0],[T0 1]);
Gnom_frd = frd(Gnom,omega);
figure(1)
hold off
for K = 0.8*K0:0.08*K0:1.2*K0
    for T = 0.7*T0:0.06*T0:1.3*T0
        G = tf([K],[T 1]);
        G_frd = frd(G,omega);
        diff = G_frd - Gnom_frd;
        bodemag(diff,'c--',omega)
    end
end
grid
temp1 = 'Approximation of uncertain transfer function';
temp2 = ' by additive uncertainty'; title([temp1 temp2])
legend(' |W_a(j\omega)| ', ' |G(j\omega)-G_{nom}(j\omega)| ',3)
```

The computation of the frequency responses is done by the command `frd`.

The error frequency responses are shown in Fig. 9.19.

We have

$$|G(j\omega) - G_n(j\omega)| = |W_a(j\omega)\Delta_a(j\omega)| \leq |W_a(j\omega)|$$

so that $|W_a(j\omega)|$ represents an upper bound on the error frequency response. By the commands

```
ord = 2;
wfit
```

one finds a sufficiently accurate second order approximation of this bound. By the commands

```
[freq,resp_db] = ginput(20); % pick 20 points
for i = 1:20 % Converts the logarithmic
    resp(i) = 10^(resp_db(i)/20); % response to magnitude
end % response
sys = frd(resp,freq); % creates frd object
W = fitmagfrd(sys,ord); % fits the frequency
% response
Wtf = tf(W); % converting into transfer
% function form
```

stored in the file `wfit.m`, one finds a stable and minimum-phase approximation of the transfer function for a given logarithmic frequency response. For this aim we

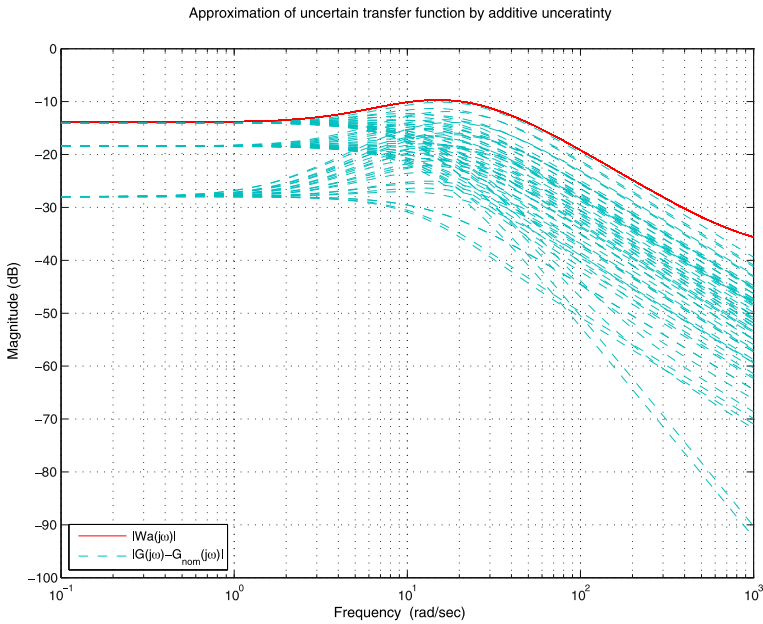


Fig. 9.19 Approximation with additive uncertainty

use 20 data points of the upper bound, positioning the cursor using a mouse. Data points are entered by pressing the left mouse button.

As a result we obtain

```
Transfer function:
0.0121 s^2 + 11.4 s + 56.95
-----
s^2 + 36.25 s + 283.7
```

As a disadvantage of this model one may note the fact that to present the uncertainty in a first order model it is necessary to use a weighted transfer function of a second order system.

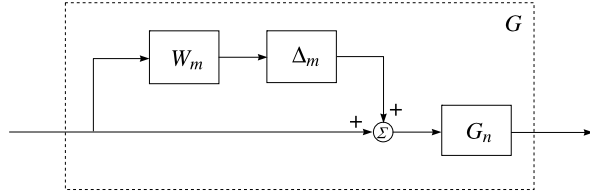
The frequency response of W_a is also shown in Fig. 9.19.

The model with additive uncertainty is obtained by the command lines

```
Delta_a = ultidyn('Delta_a',[1 1]);
G = Gnom + Wa*Delta_a
```

In the given case the complex scalar uncertainty Δ_a is set by the function `ultidyn`.

Fig. 9.20 Plant with multiplicative uncertainty



9.4.2 Models with Multiplicative Uncertainty

The following example illustrates how to obtain a model of SISO plant with multiplicative uncertainty.

Example 9.7 Consider the uncertain plant presented in Example 9.6.

The model with multiplicative uncertainty is chosen as

$$G(s) = G_n(s)[1 + W_m(s)\Delta_m(s)]$$

where

$$|\Delta_m(j\omega)| \leq 1$$

The block-diagram of the plant with multiplicative uncertainty is shown in Fig. 9.20.

In the given case we construct the magnitude frequency responses of the relative error

$$\frac{|G(j\omega) - G_n(j\omega)|}{|G_n(j\omega)|}$$

which is done by the following command lines:

```
omega = logspace(-1,3,100);
%
K0 = 1.0; T0 = 1/15;
Gnom = tf([K0],[T0 1]);
Gnom_frd = frd(Gnom,omega);
figure(1)
hold off
for K = 0.8*K0:0.08*K0:1.2*K0
    for T = 0.7*T0:0.06*T0:1.3*T0
        G = tf([K],[T 1]);
        G_frd = frd(G,omega);
        reldiff = (G_frd - Gnom_frd)/Gnom_frd;
        bodemag(reldiff,'c--',omega)
    end
end
grid
temp1 = 'Approximation of uncertain transfer function';
temp2 = ' by multiplicative uncertainty';
```

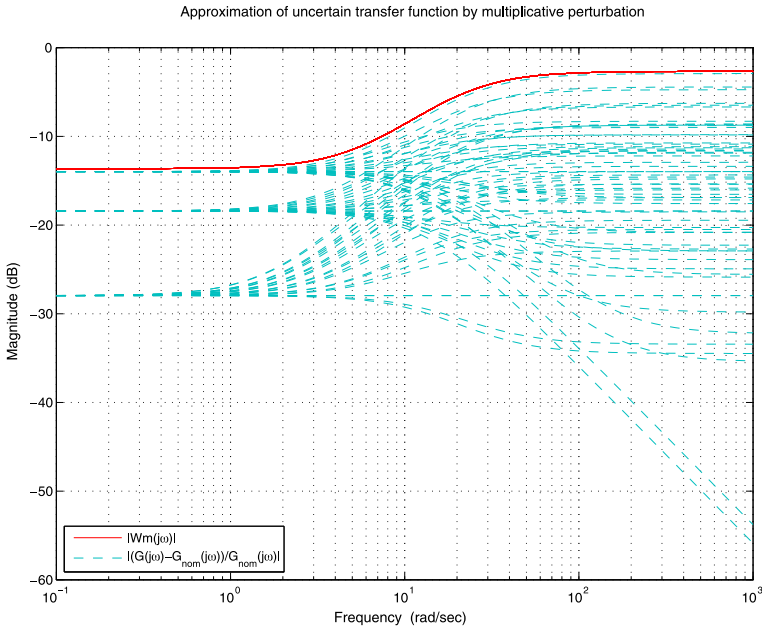


Fig. 9.21 Approximation with multiplicative uncertainty

```
title([temp1 temp2]);
legend(' |Wm(j\omega)| ', ...
       ' |(G(j\omega)-G_{nom}(j\omega))/G_{nom}(j\omega)| ', 3)
```

The frequency responses obtained are shown in Fig. 9.21.

Since

$$\frac{|G(j\omega) - G_n(j\omega)|}{|G_n(j\omega)|} \leq |W_m(j\omega)|$$

determining $W_m(j\omega)$ is equivalent to finding of the upper bound of the magnitude response of relative error. This is done by the commands

```
ord = 1;
wfit
Wm = Wtf
```

using again the file `wfit.m` with approximation order equal to 1. After executing these commands we find a stable, minimum-phase approximation of the upper bound:

```
Transfer function:
0.7351 s + 4.288
-----
      s + 20.65
```

The frequency response of W_m is shown in Fig. 9.21.

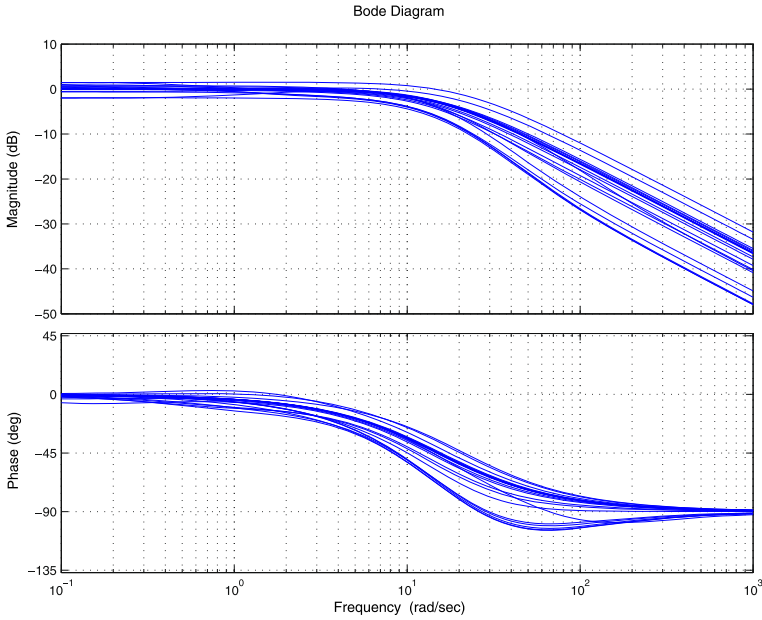


Fig. 9.22 Frequency responses of the uncertain model

Obtaining the model with multiplicative uncertainty is done by the commands

```
Delta_m = ultidyn('Delta_m',[1 1]);
G = Gnom*(1 + Wm*Delta_m)
```

which results in second order uncertain state-space model

```
USS: 2 States, 1 Output, 1 Input, Continuous System
Delta_m: 1x1 LTI, max. gain = 1, 1 occurrence
```

the frequency responses of the uncertain model are obtained by the command line

```
bode(Gnom,'r--',G,'b-',omega)
```

and are shown in Fig. 9.22.

9.4.3 Unmodeled Dynamics

Consider the set of plants

$$G(s) = G_n(s)f(s)$$

where $G_n(s)$ is a fixed (and known) transfer function. We desire to neglect the term $f(s)$ (which may be a fixed transfer function or belongs to an uncertainty set) and to represent $G(s)$ by multiplicative uncertainty with a nominal model G_n in the form

$$G(s) = G_n(s)(1 + W_m(s)\Delta_m(s))$$

where $\Delta_m(j\omega) \leq 1$.

Since

$$\frac{G(s) - G_n(s)}{G_n(s)} = f(s) - 1$$

then the magnitude frequency response of the relative uncertainty due to the neglect of the dynamics of $f(s)$, is

$$\frac{|G - G_n|}{|G_n|} = |f(j\omega) - 1|$$

From this expression one obtains that

$$|W_m(j\omega)| = \max \left| \frac{G - G_n}{G_n} \right| = \max |f(j\omega) - 1|$$

This procedure is illustrated by the following example, in which the neglect time delay is represented by a multiplicative uncertainty.

Example 9.8 Given is the plant $G = G_n(s)e^{-\tau s}$, where $0 \leq \tau \leq 0.1$ and $G_n(s)$ does not depend on τ . Our desire is to represent the plant by a multiplicative uncertainty and nominal model $G_n(s)$. For this aim first we compute the magnitude response of the relative error

$$|f(j\omega) - 1| = |e^{-j\omega\tau} - 1| = \sqrt{(\cos(\omega\tau) - 1)^2 + \sin(\omega\tau)^2}$$

for values of τ between 0 and 0.1. This is done by the commands

```
omega = logspace(-1,3,200);
figure(1)
hold off
for tau = 0:0.01:0.1;
    for i = 1:200
        om = omega(i);
        pert(i) = sqrt((cos(om*tau)-1)^2 + sin(om*tau)^2);
    end
    magg = frd(pert,omega);
    bodemag(magg,'c--')
    hold on
end
grid
temp1 = 'Approximation of uncertain time delay';
temp2 = ' by multiplicative uncertainty';
title([temp1 temp2])
legend(' |Wm(j\omega)| ', ...
       ' |(G(j\omega)-G_{nom})(j\omega))/G_{nom}(j\omega)| ', 2)
```

The frequency responses obtained are shown in Fig. 9.23. After that we find a stable and minimum-phase approximation of the multiplicative uncertainty as in Example 9.7 using the commands

```
ord = 2;
wfit
Wm = Wtf
```

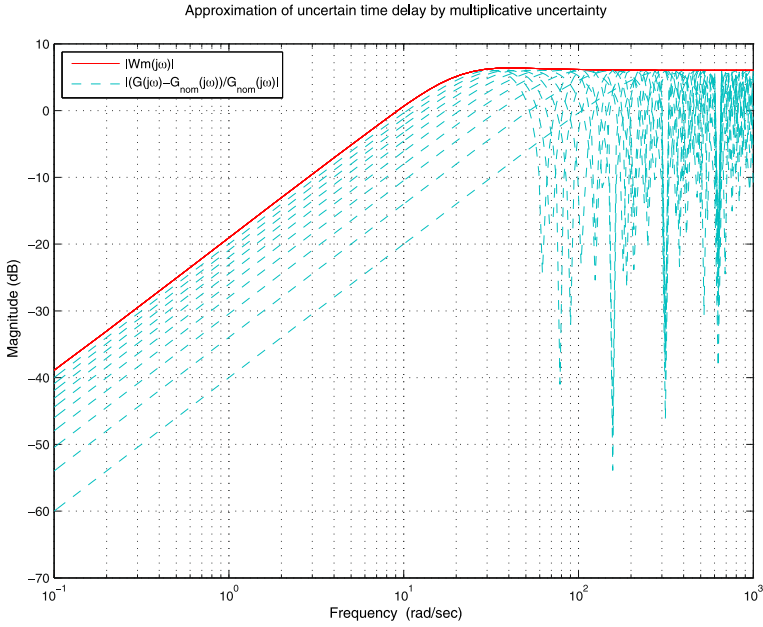



Fig. 9.23 Approximation of uncertain time delay

As a result, for the weighting transfer function $W_m(s)$ we obtain

Transfer function:

$$\frac{2s^2 + 54s + 1.1}{s^2 + 37s + 485}$$

9.4.4 Multivariable Plants with Unstructured Uncertainty

Obtaining models of multivariable plants with unstructured uncertainty may be done using the methods considered previously in this section, finding for each element of the transfer function matrix SISO model with additive or multiplicative uncertainty. The scalar models obtained are then combined in a transfer matrix in order to determine the uncertainty model of the multivariable plant.

Consider for example the model of two-input, two-output uncertain plant with transfer function matrix

$$G = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$$

where g_{11} , g_{12} , g_{21} , g_{22} are scalar transfer function containing uncertainty. These transfer functions are represented by the corresponding models approximating separately the maximum error of each element.

If each element is represented by multiplicative uncertainty model, the model of the whole system is obtained as

$$G = \begin{bmatrix} g_{n11}(1 + W_{11}\Delta_{11}) & g_{n12}(1 + W_{12}\Delta_{12}) \\ g_{n21}(1 + W_{21}\Delta_{11}) & g_{n22}(1 + W_{22}\Delta_{22}) \end{bmatrix}$$

where g_{n11} , g_{n12} , g_{n21} , g_{n22} are nominal transfer functions, W_{11} , W_{12} , W_{21} , W_{22} are the weighting functions obtained after approximation and Δ_{11} , Δ_{12} , Δ_{21} , Δ_{22} are complex scalar uncertainties of class `ultidyn`.

Note that the uncertainty in model obtained is characterized by a certain structure independently on the fact that in the individual elements the uncertainty is unstructured.

Finally, it is necessary to point out that models with mixed uncertainty (structured and unstructured) may be created from the corresponding objects by using `interconnect` or `sysic` functions.

9.5 Exercises

Exercise 9.1 For a two-channel system with plant and controller transfer matrices

$$G = \begin{bmatrix} \frac{1}{s-1} & 0 \\ 0 & \frac{1}{s+1} \end{bmatrix}, \quad K = \begin{bmatrix} \frac{s-1}{s+1} & 1 \\ 0 & 1 \end{bmatrix}$$

obtain the sensitivity matrices S_i , S_o , complementary sensitivity matrices T_i , T_o and loop transfer matrices L_i , L_o and compute the singular value plots of the corresponding frequency responses.

Exercise 9.2 For the system with plant and controller transfer matrices

$$G = \begin{bmatrix} \frac{2}{0.015s^2+0.8s+1} & \frac{0.02}{0.3s+1} & \frac{-0.08}{0.4s+1} \\ \frac{0.04}{0.1s+1} & \frac{1}{0.04s^2+1.2s+1} & \frac{-0.03}{s+1} \end{bmatrix}$$

$$K = \begin{bmatrix} \frac{20(0.3s+1)}{0.1s+1} & 0 \\ 0 & \frac{30(0.5s+1)}{0.2s+1} \\ \frac{s+1}{0.2s+1} & \frac{-0.4}{0.5s+1} \end{bmatrix}$$

obtain the closed-loop transient responses due to step reference r and step input and output disturbances d_i , d .

Exercise 9.3 For the system in the previous problem analyze the influence of controller elements K_{11} and K_{22} gains on the closed-loop frequency responses and transient responses. What is the influence of these gains on the magnitude of the steady-state errors in both channels?

Exercise 9.4 Build an uncertain model for the system

$$G = \begin{bmatrix} \frac{K}{10s+1} \\ \frac{1}{Ts+1} \end{bmatrix}$$

where the nominal values of the parameters K , T are 5 and 0.5, respectively, and the uncertainty in K is $\pm 30\%$, and in T — $\pm 10\%$.

Exercise 9.5 For the system, described by the differential equation

$$a_0\ddot{x} + 2a_0\dot{x} + a_1x = b_0\dot{u} + b_1u$$

where the nominal values of the parameters a_0 , a_1 , b_0 , b_1 are 0.9, 2, 0.2, 1, respectively, and all parameters have uncertainty of $\pm 30\%$, build an uncertain state-space model (class `uss`). Note the repeated parameter a_0 .

Exercise 9.6 For the SISO system with transfer function

$$W(s) = \frac{K}{T^2s^2 + 2\xi Ts + 1}$$

where K changes in the range $[10, 12]$, T changes in the range $[0.1, 0.15]$ and ξ changes in the range $[0.3, 0.4]$, find a model with additive uncertainty.

Exercise 9.7 For the system from the previous exercise find a model with multiplicative uncertainty.

Exercise 9.8 For the two-channel plant with nominal transfer function matrix

$$G(s) = \begin{bmatrix} \frac{10}{s^2+0.4s+16} & \frac{1}{s+1} \\ \frac{1}{s+2} & \frac{5}{(s+2)(s+3)} \end{bmatrix}$$

and uncertain delays in the first and second input in the range $[0, 0.2]$, find a model with unstructured uncertainty.

Chapter 10

Robust Stability and Performance

This chapter is devoted to the important subject of robust stability and robust performance analysis using MATLAB[®]. First we show how to implement functions from Robust Control Toolbox[®]3 to analyze the robust stability of systems with unstructured, structured and mixed uncertainty. Then we consider the implementation of the function `robustperf` to analyze the robust performance of closed-loop systems and show how to interpret the results obtained by this function. Similarly to robust stability margin, we use the notion of performance margin that shows the uncertainty level, up to which the system possess specified performance. Finally, we consider how to determine the so called “worst-case” gain, which is determined as the largest value of the frequency response maximum for the allowed uncertainty.

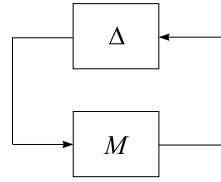
10.1 Robust Stability Analysis

For the aim of robust stability analysis, it is convenient to represent the uncertain control system by the $M-\Delta$ loop, shown in Fig. 10.1. In this loop the transfer function of the nominal part (denoted by M) is separated from uncertain part (denoted by Δ). In the simpler case of unstructured Δ it is possible to use the small-gain theorem (see Sect. 3.1 of Part I). In the general case of structured $\Delta(s)$ the robust stability analysis is done by using the structured singular value μ , (see Sect. 3.3 of Part I).

Except some simple cases, the structured singular value μ , which is a frequency function, cannot be computed exactly. However, there exist efficient algorithms to determine upper and lower bounds of μ . That is why the conclusions about system robust stability should be drawn in terms of these bounds. Specifically, let the maximums along frequency of the upper and lower bound of the structured singular values be denoted respectively by β_u and β_l . Then

- The uncertain system under consideration is guaranteed stable for all structured uncertain matrices Δ with $\|\Delta(s)\|_\infty < 1/\beta_u$.

Fig. 10.1 M - Δ loop for robust stability analysis



- There exists a specific structured transfer matrix Δ with $\|\Delta(s)\|_\infty = 1/\beta_l$ that destabilizes the system.

In addition, for the case of normalized uncertainty

$$\|\Delta\|_\infty \leq 1$$

it follows that

- If $\beta_u < 1$, the system is robustly stable with respect to the modeled uncertainty.
- If $\beta_l > 1$, the robust stability is not achieved.
- If $\beta_l < 1$ and $\beta_u > 1$, it is not possible to draw with certainty a conclusion about stability; it is possible that the system is not robustly stable.

The main tool in Robust Control Toolbox^{®3} for robust stability analysis, based on computing of structured singular value, is the command `robuststab`. This command may be used for stability analysis of systems with unstructured, structured or mixed uncertainty. The command `robuststab` does not require the system to be presented in some special form (for instance, as an M - Δ loop). Its syntax is

```
[stabmarg,destabunc,report] = robuststab(sys,opt)
[stabmarg,destabunc,report,info] = robuststab(sys,opt)
```

Input arguments:

- `sys`: model of the uncertain system under investigation, may be of class `uss` or of class `ufrd`. If `sys` is of class `uss`, the computations are done for appropriately chosen by the command `robuststab` frequency values. If it is of class `ufrd`, the associated with `sys` frequency vector is used;
- `opt`: optional input arguments. It represents an object, created by the command `opt = robopt('name1',value1,'name2',value2,...)`. Some of the most frequently used properties are
 - `Display`: displays progress of computations; default is `'off'`;
 - `Sensitivity`: computes the influence of individual uncertainties on the stability margin; default is `'on'`.

Output arguments:

- `stabmarg`: structure with the following fields:
 - `UpperBound`: upper bound on stability margin;
 - `LowerBound`: lower bound on stability margin;
 - `DestabilizingFrequency`: frequency at which system instability occurs. Corresponds to the upper bound of stability margin;

- `destabunc`: structure containing a combination of uncertain parameter values closest to their nominal values that cause system instability—it corresponds to the upper bound of stability margin;
- `report`: variable containing text description of robustness analysis results;
- `info`: structure with several fields:
 - `Frequency`: frequency vector used in the computations;
 - `MussvBnds`: `frd` object containing the computed values of the upper bound (`MussvBnds(1,1)`) and lower bound (`MussvBnds(1,2)`) of the structured singular value;
 - `Sensitivity`: structure with number of fields equal to the number of uncertain elements. In each field is stored a number, indicating the influence of the corresponding uncertain element on the stability margin. For instance, the number 40 means that if the uncertainty range is increased with 25 %, the stability margin will decrease 10 % (25 % of 40).

Consider in some detail how to interpret the results obtained by command `robuststab`. If the uncertain system `sys` is of class `uss`, the command `robuststab` checks the stability of the nominal system. In case the system is unstable, to `UpperBound` and to `LowerBound` is assigned the value $-\infty$. If, however, `sys` is of class `ufrd`, such a check is not done assuming that the nominal system is stable.

The robust stability analysis done by the command `robuststab` is based on the computation of bounds on the structured singular value, decomposing the uncertain system before that into $M-\Delta$ form with $\|\Delta(s)\|_\infty \leq 1$, i.e., the results from analysis pertain to normalized uncertainty.

The results of stability analysis are obtained in terms of upper and lower bounds of the stability margin. Remember that in the given case the stability margin is defined as the reciprocal value of the maximum of structured singular values with respect to the frequency. Hence, the upper bound of stability margin is equal to the reciprocal value of the maximum along frequency of the lower bound of the structured singular value (i.e., $\text{UpperBound} = 1/\beta_l$) and the lower bound of the stability margin is obtained in the same way from the upper bound of the structured singular value ($\text{LowerBound} = 1/\beta_u$).

Taking into account the stability conditions, derived by the bounds of structured singular value, it is possible to draw the following conclusions.

- If $\text{LowerBound} > 1$, the uncertain system is robustly stable with respect to modeled uncertainty.
- If $\text{UpperBound} < 1$, the system does not achieve robust stability.
- If $\text{LowerBound} < 1$ and $\text{UpperBound} > 1$, it is not possible to draw a definite conclusion about stability; the system may not be robustly stable.

Let for instance $\text{LowerBound} = 1.2$ and $\text{UpperBound} = 1.3$. Apart from the fact that the system is robustly stable for the given uncertainty level, in this case it is guaranteed that

- The system remains stable for uncertainty levels smaller than 120 % of the given one.

- There exist at least one collection of uncertainty values with level 130 % of the given one which causes instability. (One of these destabilizing sets is included in the structure `destabunc`.)

Apart by `robuststab`, the robust stability analysis may be done also by the command `mussv` which calculates upper and lower bound on the structured singular values $\mu_{\Delta}(M)$ as a function of frequency. In order to implement this command, the uncertain system should be decomposed in an M – Δ loop (Fig. 10.1) with normalized uncertainty $\|\Delta(s)\|_{\infty} \leq 1$. The basic syntax of `mussv` is

```
bounds = mussv(M,BlkStructure)
```

Description of arguments:

- `M`: model of the known part of the uncertain system that may be of class `frd` (this is usual in implementing the function `mussv`) or a complex matrix. If it is of class `frd`, the structured singular value bounds are calculated for the associated with `M` frequency vector;
- `BlkStructure`: input argument, specifying the block-structure of the uncertainty matrix $\Delta(s)$. It represents a matrix with two columns and as many rows as the number of uncertain system blocks. Each row contains information which specifies the corresponding uncertainty block.
 - Scalar real parameter (uncertain element of class `ureal`) is described by $\begin{bmatrix} -1 & 0 \end{bmatrix}$, and n times repeated scalar real parameter—by $\begin{bmatrix} -n & 0 \end{bmatrix}$;
 - Scalar complex block (1×1 element of class `ultidyn` or element of class `ucomplex`) is set by $\begin{bmatrix} 1 & 0 \end{bmatrix}$, and n times repeated element—by $\begin{bmatrix} n & 0 \end{bmatrix}$;
 - $n \times m$ full complex block ($n \times m$ element of class `ultidyn`) is specified by $\begin{bmatrix} n & m \end{bmatrix}$.
- `bounds`: output argument. If `M` is of class `frd`, then `bounds` is a `frd` object, containing the computed values of the upper bound (`bounds(1,1)`) and lower bound (`bounds(1,2)`) of the structured singular value of `M` with respect to the uncertainty structure specified by `BlkStructure`.

To represent the uncertain system in the form required by `mussv` one usually implements the function `lftdata`,

```
[M,Delta,BlkStruct] = lftdata(sys)
```

which determines the M – Δ decomposition (with normalized uncertain elements) of the uncertain system with model `sys`. The output argument `BlkStruct` describes the block-diagonal structure of `Delta` and it is in the form that allows its direct use as an input argument of the function `mussv` (avoiding in this way to specify the block-structure of the uncertainty, as already described). Also, it is necessary to take into account that in the robust stability analysis it is necessary to take only the part of `M` that is connected to `Delta`.

Note that the functions `lftdata` and `mussv` are used by the command `robuststab` that facilitates the robust stability analysis.

The commands `robuststab` and `mussv` may be implemented in the analysis of continuous-time as well as discrete-time systems. The type of the system is not

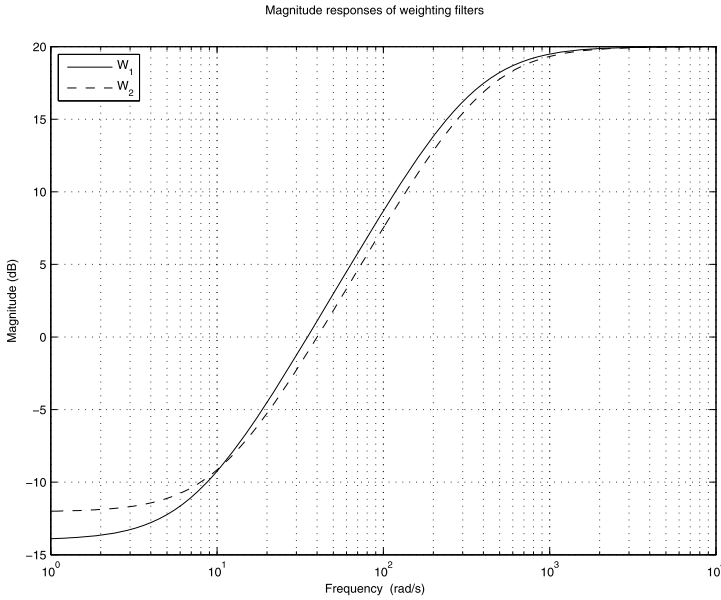


Fig. 10.2 Magnitude responses of weighting filters

necessary to be indicated by the user since this information is stored in `uss` and `ufrd` models.

Example 10.1 Consider a fifth order two channel system with two inputs, two outputs and nominal transfer function

$$G_{\text{nom}} = \begin{bmatrix} \frac{6}{(0.9s+1)(0.1s+1)} & \frac{-0.05}{0.1s+1} \\ \frac{0.07}{0.3s+1} & \frac{5}{(1.8s-1)(0.06s+1)} \end{bmatrix}$$

(This transfer function was used already in Examples 9.2 and 9.4.) The system has an uncertainty at the first input that consists of 20 % error in the low-frequency range, increases to 100 % at 35 rad/s, and reaches 1000 % in the high-frequency range. The uncertainty at the second input is 25 % in the low-frequency range, 100 % at 40 rad/s and increases to 1000 % in the high-frequency range. These uncertainties may be represented as input multiplicative uncertainties by using uncertain `ulydin` objects and appropriate weighting filters, created by the function `makeweight`.

```
W1 = makeweight(0.20,35,10);
W2 = makeweight(0.25,40,10);
```

The magnitude responses of weighting filters are shown in Fig. 10.2.

Next, the model with input multiplicative uncertainty is obtained by the command lines

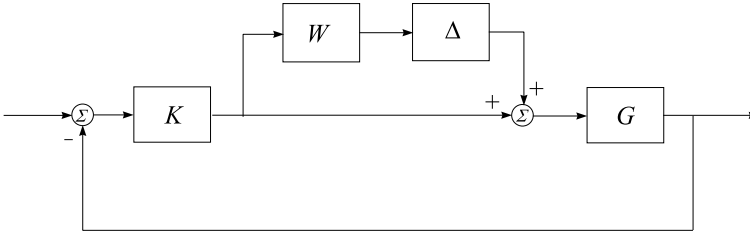


Fig. 10.3 Closed-loop system with input multiplicative uncertainty

```
Delta1 = ultidyn('Delta1',[1 1]);
Delta2 = ultidyn('Delta2',[1 1]);
W = blkdiag(W1,W2);
Delta = blkdiag(Delta1,Delta2);
G = Gnom*(eye(2) + Delta*W)
```

As a result we obtain an eighth order uncertain object,

```
USS: 8 States, 2 Outputs, 2 Inputs, Continuous System
Delta1: 1x1 LTI, max. gain = 1, 1 occurrence
Delta2: 1x1 LTI, max. gain = 1, 1 occurrence
```

The robust stability of the closed-loop system will be analyzed for the controller

$$K = \begin{bmatrix} \frac{2(s+1)}{s} & \frac{-s}{3s+1} \\ \frac{-5(s+1)}{0.8s+1} & \frac{4(0.7s+1)}{s} \end{bmatrix}$$

The block-diagram of the closed-loop system is shown in Fig. 10.3.

Since the plant has unstructured uncertainties Δ_1 and Δ_2 it is instructive to see what result will produce the small gain theorem if we assume as unstructured the plant uncertainty matrix

$$\Delta = \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix}$$

For this aim we present the system as an M - Δ loop, shown in Fig. 10.1 with

$$M = -W(I + KG)^{-1}KG = -WT_i$$

where

$$W = \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix}$$

and T_i is the input complementary sensitivity. Now according to the results presented in Sect. 3.1 of Part I, the condition for robust stability, based on small-gain theorem, is

$$\|WT_i\|_\infty < 1$$

The input sensitivity is determined by the command lines

```
looptransfer = loopsens(G,K);
Ti = looptransfer.Ti;
```

The \mathcal{H}_∞ -norm of the transfer function matrix WT_i (the nominal transfer function matrix is used in the computations) is determined as

```
[PeakNorm,freq] = norm(W*Ti.Nominal,'inf')
```

As a result one obtains

```
PeakNorm =
```

```
1.2353
```

```
freq =
```

```
4.8763
```

This result shows that the closed-loop system does not achieve robust stability assuming that the uncertain matrix Δ is unstructured. However, the matrix Δ actually has a block-diagonal structure which affects very much the robustness of the closed-loop system. Further on, taking into account the structure of Δ , we show that the closed-loop system in fact achieves robust stability even for larger input multiplicative uncertainties than the given ones.

The robust stability analysis is done by the function `robuststab` with input argument the uncertain frequency response corresponding to the output complementary sensitivity.

```
omega = logspace(-1,2,200);
To_g = ufrd(To,omega);
opt = robust('Display','on');
[stabmarg,destabunc,report,info] = robuststab(To_g,opt);
```

As a result we obtain

```
stabmarg =
```

```
UpperBound: 2.020374929266748
```

```
LowerBound: 2.020373919079788
```

```
DestabilizingFrequency: 3.696912707195026
```

It follows that

- for the given structure and level of uncertainty the closed-loop system is robustly stable (the lower bound of stability margin satisfies `LowerBound > 1`);
- the system remains stable for uncertainty levels smaller than 202.03 % from the given one;
- there exists uncertainty with level 202.03 % from the given one which destabilizes the system.

These conclusions about the robust stability of the system are contained in the variable report. It contains also an information about the influence of the uncertain elements on the stability margin. The influence of the uncertainty Δ_2 is stronger.

report =

```
Assuming nominal UFRD system is stable ...
Uncertain System is robustly stable to modeled uncertainty.
-- It can tolerate up to 202% of the modeled uncertainty.
-- A destabilizing combination of 202% of the modeled
   uncertainty exists, causing an instability at 3.7 rad/s.
-- Sensitivity with respect to uncertain element ...
   'Delta1' is 63%. Increasing 'Delta1' by 25% leads
                        to a 16% decrease in the margin.
   'Delta2' is 90%. Increasing 'Delta2' by 25% leads
                        to a 23% decrease in the margin.
```

In this way we see that taking into account the block-diagonal structure of the uncertainty Δ will lead us to the conclusion that the closed-loop system achieves robust stability as opposite to the case when we assumed Δ as unstructured. This confirms the well known fact that neglecting the uncertainty structure leads to pessimistic conclusions about the system stability.

The command lines

```
destabunc.Delta1
destabunc.Delta2
```

produce the state-space realizations of the perturbations Δ_1 and Δ_2 that destabilize the closed-loop system. The corresponding transfer functions are

```
tf(destabunc.Delta1)
```

```
Transfer function:
-10.56 s^2 + 14.35 s + 6.374e-015
-----
s^3 + 6.587 s^2 + 20.77 s + 18.57
```

and

```
tf(destabunc.Delta2)
```

```
Transfer function:
-10.56 s^2 + 9.88 s + 2.194e-015
-----
s^3 + 6.164 s^2 + 18.56 s + 12.78
```

If these perturbations are substituted in the transfer function matrix T_o it is possible to find the poles of the closed-loop system:

```
pole(usubs(To,destabunc))
```

As a result we obtain

ans =

```

1.0e+002 *

-4.117820048554725
-3.552895319158590
-0.090818247953519 + 0.087226740827892i
-0.090818247953519 - 0.087226740827892i
-0.1000000000000026
-0.088879355942136
-0.021603036018368 + 0.048364783432694i
-0.021603036018368 - 0.048364783432694i
 0.0000000000000000 + 0.036969127071950i
 0.0000000000000000 - 0.036969127071950i
-0.046004083820114 + 0.016746382352412i
-0.046004083820114 - 0.016746382352412i
-0.033451309908770
-0.003583579386419 + 0.000780496584728i
-0.003583579386419 - 0.000780496584728i
-0.010671435327352
-0.009480508626777
-0.006893138594670

```

We see that the destabilizing poles which correspond to the destabilizing perturbations are $\pm j3.697$. In Fig. 10.4 we show the frequency responses of the upper and lower bounds of the structured singular value μ obtained by the command lines

```

semilogx(info.MussvBnds(1,1), 'r-', info.MussvBnds(1,2),
         'b--')
grid
title('Robust stability')
xlabel('Frequency (rad/s)')
ylabel('mu')
legend('\mu-upper bound', '\mu-lower bound', 2)

```

Consider now the robust stability of the given system by using the command `mussv`. For this aim we decompose the output sensitivity function T_o into M - Δ loop.

```
[M, Delta, BlockStructure] = lftdata(To);
```

To do the stability analysis it is necessary to take only the part of M that is connected to Δ ,

```

size_Delta = size(Delta);
M11 = M(1:size_Delta(2), 1:size_Delta(1));

```

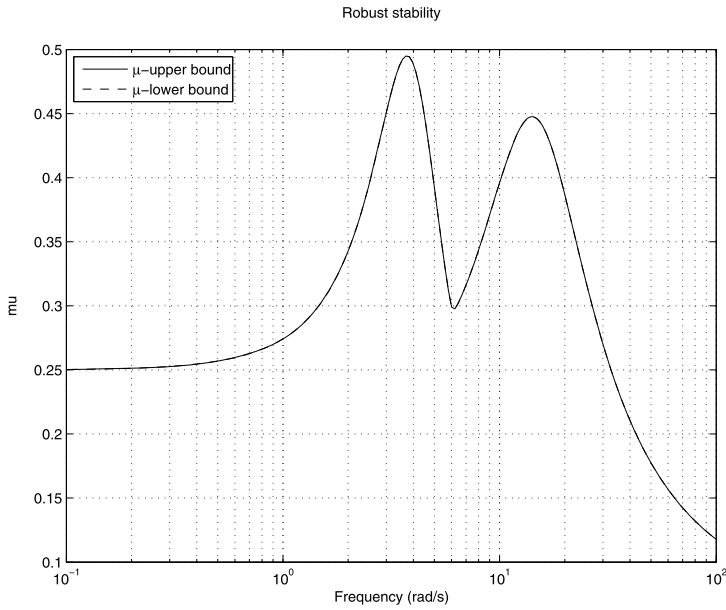


Fig. 10.4 Robust stability

The frequency response of $M11$ is determined for the frequency vector used by `robuststab`:

```
omega = info.Frequency;
M11_g = frd(M11,omega);
```

Now we are in position to calculate the upper and lower bounds on μ using the function `mussv`.

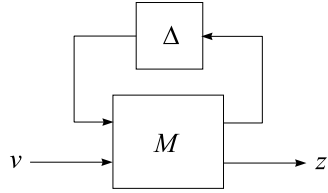
```
rbounds = mussv(M11_g,BlockStructure);
```

The bounds obtained are the same as the bounds determined in case of using the function `robuststab` that are shown in Fig. 10.4.

10.2 Robust Performance Analysis

For the aim of robust performance analysis the closed-loop system is represented by the block-diagram, shown in Fig. 10.5. The known system part M includes the model of the nominal control system, weighting functions to represent the uncertainty and weighting functions used to specify performance requirements. The unknown part represents a set of transfer function matrices Δ with given block-diagonal structure, corresponding to the uncertainty structure of the system under consideration. The input vector v contains all exogenous signals to the system: references, disturbances, and sensor noises. The output vector z includes signals having

Fig. 10.5 Block-diagram for robust performance analysis



the meaning of “errors” which characterize the system performance: for instance, the weighted (by weighting transfer functions) tracking errors, control actions and so on.

Let $T_{zv}(s)$ denotes the transfer function matrix from v to z . It is appropriate as a system performance index to use the quantity

$$\|T_{zv}(s)\|_{\infty} \quad (10.1)$$

Smaller values of this index means smaller “errors” z due to the “worst” input signals v and hence better system performance.

The problem of robust performance analysis may be reduced to the problem of robust stability analysis of the closed-loop which consists of the block M and the extended uncertainty block

$$\Delta_P = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_F \end{bmatrix} \quad (10.2)$$

In (10.2) Δ_F is a fictitious complex (unstructured) uncertain block with size $n_v \times n_z$ where n_v and n_z are the size of vectors v and z , respectively. The robust performance analysis of the system is done by using the structured singular value of M with respect to the extended uncertainty Δ_P (see Sect. 6.1 of Part I).

Since it is only possible to calculate upper and lower bounds of the structured singular value, the conclusions about robust performance should be drawn on the basis of these bounds. Let the maxima with respect to the frequency of the upper and lower bounds of the structured singular value $\mu_{\Delta_P}(M)$ be denoted by β_u and β_l , respectively. Then

- For all uncertainty matrices Δ with the given structure satisfying $\|\Delta\|_{\infty} < 1/\beta_u$, the closed-loop system is stable and $\|T_{zv}(s)\|_{\infty} \leq \beta_u$.
- There exists a specific matrix Δ with the given structure satisfying $\|\Delta\|_{\infty} = 1/\beta_l$ for which either $\|T_{zv}(s)\|_{\infty} \geq \beta_l$ or the system is unstable.

Customary, the weighting transfer functions, used to specify the performance requirements, are chosen so that when the condition

$$\|T_{zv}(s)\|_{\infty} < 1 \quad (10.3)$$

is fulfilled, the control system has the desired performance.

It is said that a given uncertain system achieves robust performance if for the uncertainty allowed the system is stable and condition (10.3) is satisfied.

monotonically and for a specific uncertainty level the system loses stability. Given are also the upper bound and lower bound of the performance degradation curve that are obtained by using the structured singular value. The points of intersection of these curves with the hyperbole $y = 1/x$ have abscissas equal to the performance margin pm and its bounds pm_l and pm_u , and ordinates β , β_u and β_l , respectively.

The robust performance analysis is done by the command `robustperf`. The results of this command are obtained in terms of the upper and lower bounds of the performance margin. The syntax of this command, the input and output arguments are similar to the command `robuststab` for robust stability analysis.

```
[perfmarg,perfmargunc,report] = robustperf(sys,opt)
[perfmarg,perfmargunc,report,info] = robustperf(sys,opt)
```

Input arguments.

- `sys`: model of the uncertain system (extended by the weighting transfer functions used to specify the performance requirements, i.e., $T_{zv}(s)$). If `sys` is of class `uss`, the computations are done for frequency values appropriately chosen by the function `robustperf`. If it is of class `ufrd`, the associated with `sys` frequency vector is used. The system may be either continuous-time or discrete-time;
- `opt`: optional input argument created by command `robopt` in the same way as for `robuststab`.

Output arguments.

- `perfmarg`: structure with the following fields.
 - `UpperBound`: upper bound of performance margin;
 - `LowerBound`: lower bound of performance margin;
 - `CriticalFrequency`: frequency value corresponding to the upper bound of performance margin;
- `perfmargunc`: a combination of uncertain element values corresponding to upper bound of performance margin;
- `report`: char array with text description of the results from robust performance analysis;
- `info`: structure with several fields.
 - `Frequency`: frequency vector used in the computations;
 - `MussvBnds`: `frd` object containing the computed values of the upper bound (`MussvBnds(1,1)`) and lower bound (`MussvBnds(1,2)`) of the structured singular value corresponding to the case of robust performance;
 - `Sensitivity`: structure containing information for the influence of uncertain elements on the performance margin.

The robust performance analysis by the command `robustperf` is based on the calculation of the structured singular value bounds determined for normalized uncertainty $\|\Delta(s)\|_\infty \leq 1$.

Let for instance the performance margin bounds are `LowerBound` = 1.25 and `UpperBound` = 1.30. In this case it is guaranteed that for uncertainty level less than 125 % of the given one (i.e., 1.25 normalized units), the closed-loop system is

stable and the value of performance index (\mathcal{H}_∞ -norm of `sys`) is less or equal to 0.8. Also, there exists at least one combination of uncertainty values with level 130 % of the given one, for which either the performance index is greater or equal to 0.769 (i.e., $1/1.30$) or the closed-loop system is unstable. (One of these combinations is included in the structure `perfmargin`.)

For a performance requirement (10.3) one may derive the following conclusions: if `LowerBound` > 1, the system achieves robust performance for the modeled uncertainty; if `UpperBound` < 1 the robust performance is not achieved; if `LowerBound` < 1 and `UpperBound` > 1 it is not possible to draw conclusions with certainty.

In addition to `robustperf`, the robust performance analysis may be done computing with the command `mussv` the upper and lower bound of the structured singular value $\mu_{\Delta p}(M)$ as a function of frequency. This approach, however, is not recommended.

Example 10.2 Consider the control system described in Example 10.1 and let G denotes the plant transfer function matrix and K —the controller transfer function matrix. Let the requirements to system performance are to ensure reference (r) tracking with admissible error (e) and limited control (u) magnitude in the presence of disturbance d . According to the first requirement, the output sensitivity function S_o connecting e with r and d should be sufficiently small in the low-frequency range (r and d are low-frequency signals). According to the second requirement the transfer function matrix $K S_o$ connecting u with r and d should be sufficiently small. To take into account both requirements it is appropriate to use as a system performance index the quantity

$$\left\| \begin{bmatrix} W_p S_o \\ W_u K S_o \end{bmatrix} \right\|_\infty \quad (10.4)$$

where W_p and W_u are weighting transfer function matrices. Let

$$W_p = \begin{bmatrix} w_p & 0 \\ 0 & w_p \end{bmatrix}, \quad w_p = \frac{0.04(s+10)}{(s+0.005)}$$

and

$$W_u = \begin{bmatrix} w_u & 0 \\ 0 & w_u \end{bmatrix}, \quad w_u = \frac{4.0 \cdot 10^{-2}(0.01s+1)}{(0.005s+1)}$$

are chosen so that when the condition

$$\left\| \begin{bmatrix} W_p S_o \\ W_u K S_o \end{bmatrix} \right\|_\infty < 1 \quad (10.5)$$

is fulfilled, the control system has desired performance, i.e., desired accuracy of reference tracking and limited magnitude of control action are achieved.

The problem is to analyze the system robust performance taking into account the plant uncertainty.

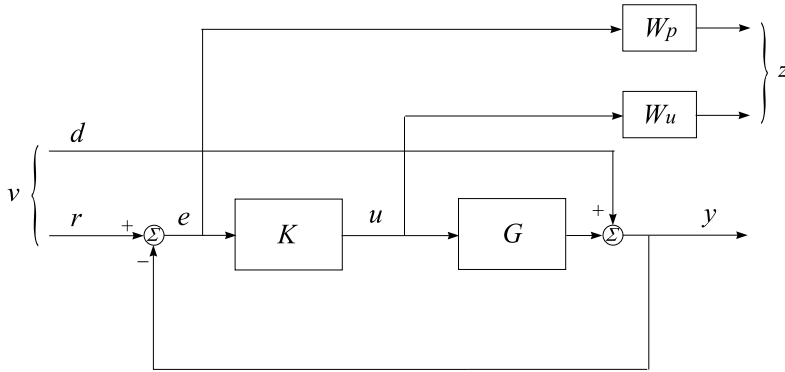


Fig. 10.7 Block-diagram of the closed-loop system with performance requirements

The block-diagram of the closed-loop system including the weighting functions W_p and W_u is shown in Fig. 10.7. In the given case

$$T_{zv} = \begin{bmatrix} W_p S_o \\ W_u K S_o \end{bmatrix}$$

the output vector z containing the weighted “errors” and the input vector v being the difference $v = r - d$.

The system model is built by the command `sysic` saving the transfer function matrix $T_{zv}(s)$ into the variable `clp` of class `uss`:

```
wp = 0.04*(s + 10)/(s + 0.005);
Wp = blkdiag(wp,wp);
wu = 4.0*10^(-2)*(0.01*s+1)/(0.005*s+1);
Wu = blkdiag(wu,wu);
systemnames = ' G K Wp Wu ';
inputvar = '[ ref{2}; dist{2} ]';
outputvar = '[ Wp; Wu ]';
input_to_G = '[ K ]';
input_to_K = '[ ref-G-dist ]';
input_to_Wp = '[ ref-G-dist ]';
input_to_Wu = '[ K ]';
clp = sysic;
```

For the system with nominal plant model

```
norm(clp.Nom,'inf')
ans =
    0.8081
```

condition (10.5) is satisfied and hence the closed-loop system achieves nominal performance.

The system robust performance is analyzed by the command `robustperf` setting appropriate frequency vector:

```
omega = logspace(-1,2,200);
clp_g = ufrd(clp,omega);
opt = robopt('Display','on');
[perfmarg,perfmargunc,report,info] = robustperf(clp_g,opt)
```

As a result one obtains

```
perfmarg =
```

```
UpperBound: 1.007341638920202
LowerBound: 1.007047914437433
CriticalFrequency: 15.885651294280526
```

Since $\text{LowerBound} > 1$ the performance requirement (10.5) is satisfied for the given uncertainty, i.e., the system achieves robust performance.

It follows from the result obtained that for uncertainty levels less than 100.70 % of the given one the value of the performance index (10.4) is less than or equal to $1/1.0070 = 0.9930$.

Also, the analysis gives a combination of uncertainties with level 100.73 % of the given one for which the value of (10.4) is greater than or equal to $1/1.0073 = 0.9928$.

Part of the conclusions drawn about the system robust performance are contained in the variable `report`. It contains also information about the influence of the uncertain elements on the performance margin. The influence of the uncertainty Δ_2 is stronger:

```
report =

Uncertain System achieves a robust performance margin of
1.007.
-- A model uncertainty exists of size 101% resulting in a
performance margin of 0.993 at 15.9 rad/s.
causing an instability at 15.9 rad/s.
-- Sensitivity with respect to uncertain element ...
'Delta1' is 30%. Increasing 'Delta1' by 25% leads to a
8% decrease in the margin.
'Delta2' is 44%. Increasing 'Delta2' by 25% leads to a
11% decrease in the margin.
```

In Fig. 10.8 we show the frequency responses of the upper and lower bounds of μ obtained by the command lines

```
semilogx(info.MussvBnds(1,1),'r-',info.MussvBnds(1,2),
'b--')
grid
title('Robust performance')
xlabel('Frequency (rad/s)')
```

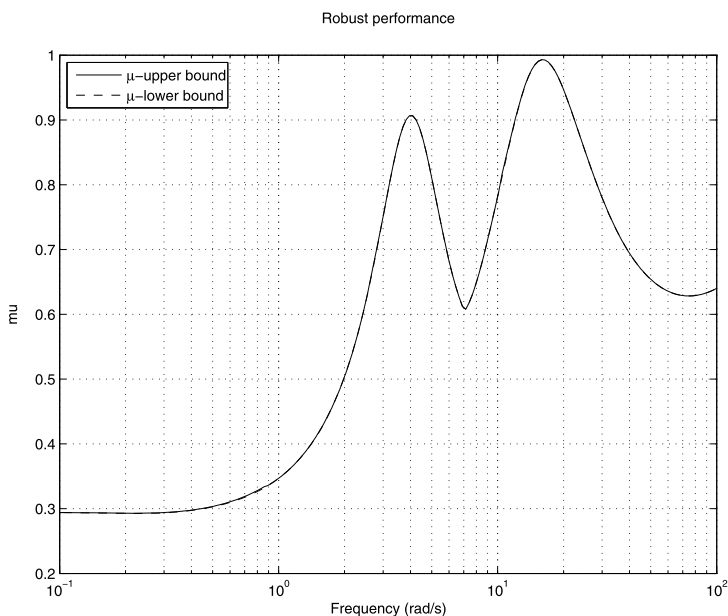


Fig. 10.8 Upper and lower bounds of μ

```
ylabel('mu')
legend('\mu-upper bound', '\mu-lower bound', 2)
```

The maxima of upper and lower bounds of structured singular values with respect to frequency are found by the lines

```
[pk1,pklidx] = max(info.MussvBnds(1,2).ResponseData(:));
[pku,pkuidx] = max(info.MussvBnds(1,1).ResponseData(:));
pkmu.UpperBound = pku;
pkmu.LowerBound = pk1;
```

and are equal to

```
pkmu =
```

```
UpperBound: 0.9930
```

```
LowerBound: 0.9927
```

The output variable `perfmargunc` contains a combination of uncertain element values corresponding to the upper bound of performance margin. Substitution of these values in the closed-loop transfer function matrix allows to find the worst-case performance for the given level of uncertainty. In Fig. 10.9 we show the singular value plots corresponding to closed-loop transfer function matrix $T_{zv}(s)$ for random value and worst-case uncertainties, obtained by the command lines

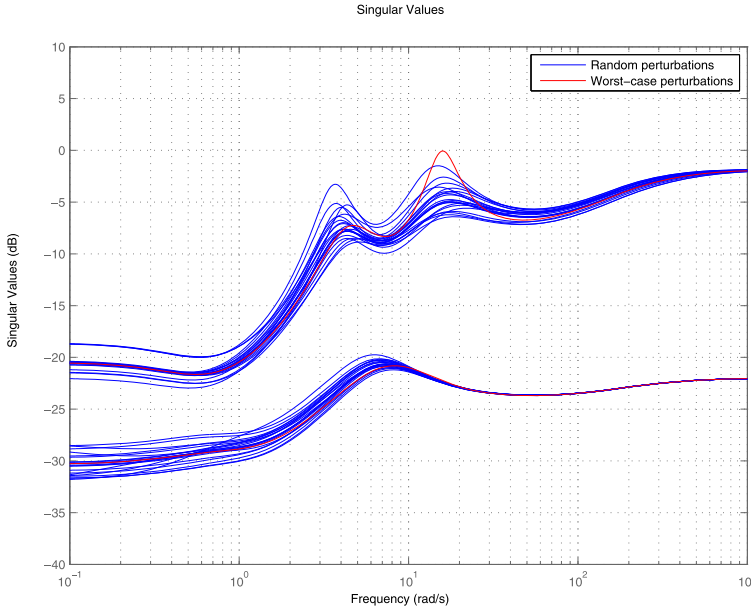


Fig. 10.9 Worst-case performance

```
omega = logspace(-1,3,500);
sigma(clp,'b-',usubs(clp,perfmargin),'r-',omega)
```

The worst-case performance does not exceed the value 0.9930, as expected.

To illustrate the influence of uncertainty on the closed-loop system, in Fig. 10.10 we show the singular value plots of output complementary sensitivity for random value and worst-case uncertain elements.

In Fig. 10.11 we show the magnitude frequency response of the inverse weighting function W_p^{-1} and the singular value plot of the output sensitivity S_o for random values of uncertainties. In order to satisfy condition (10.5), it is necessary that the magnitude responses of S_o to lie below the magnitude response W_p^{-1} in the whole frequency range.

In Fig. 10.12 the inverse control action weighting function W_u^{-1} and the singular value plot of KS_o , which shows the sensitivity of control action to references and disturbances. As in the case of output sensitivity, condition (10.5) means that the magnitude responses of KS_o should lie below the response of W_u^{-1} .

Finally, we obtain the transient responses with respect to the unit step first reference for 30 random values of uncertainties by using the command lines

```
tfin = 5;
nsample = 30;
[To30,samples] = usample(To,nsample);
time = 0:tfin/500:tfin;
nstep = size(time,2);
```

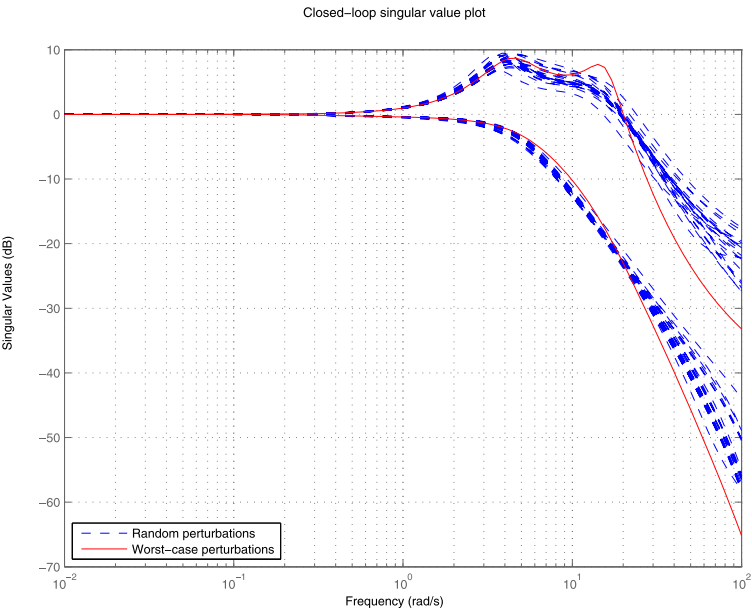


Fig. 10.10 Singular values of output complementary sensitivity

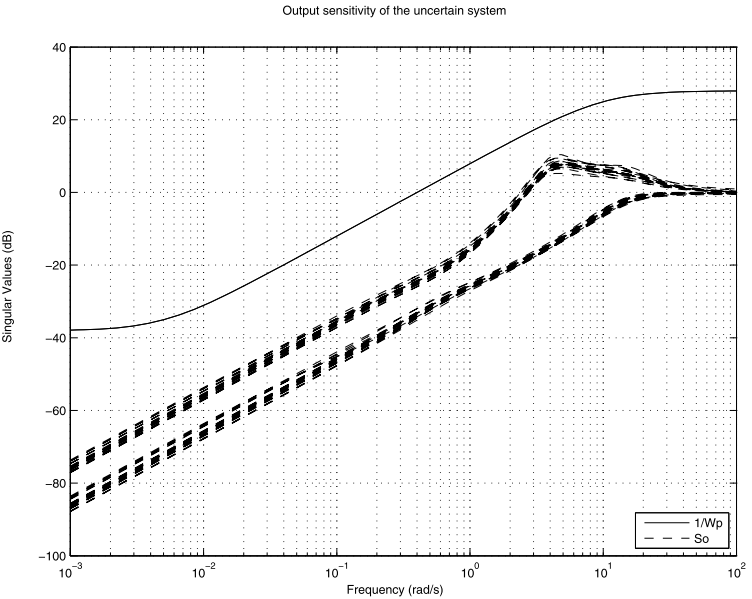


Fig. 10.11 Singular value plot of output sensitivity

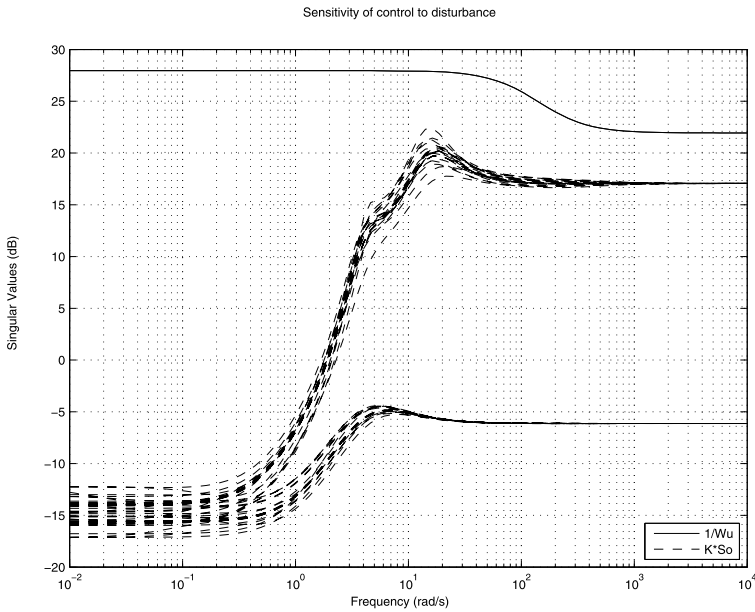


Fig. 10.12 Sensitivity of control to disturbances

```

ref1(1:nstep) = 1.0;
ref2(1:nstep) = 0.0;
ref = [ref1' ref2'];
nsample = 30;
[To30,samples] = usample(To,nsample);
figure(4)
subplot(2,2,1)
hold off
for i = 1:nsample
    [y,t] = lsim(To30(1:2,1:2,i),ref,time);
    plot(t,y(:,1),'r-')
    hold on
end
grid
title('From inp 1 to outp 1') xlabel('Time (secs)')
ylabel('y_1')
figure(4)
subplot(2,2,3)
hold off
for i = 1:nsample
    [y,t] = lsim(To30(1:2,1:2,i),ref,time);
    plot(t,y(:,2),'b-')
    hold on

```

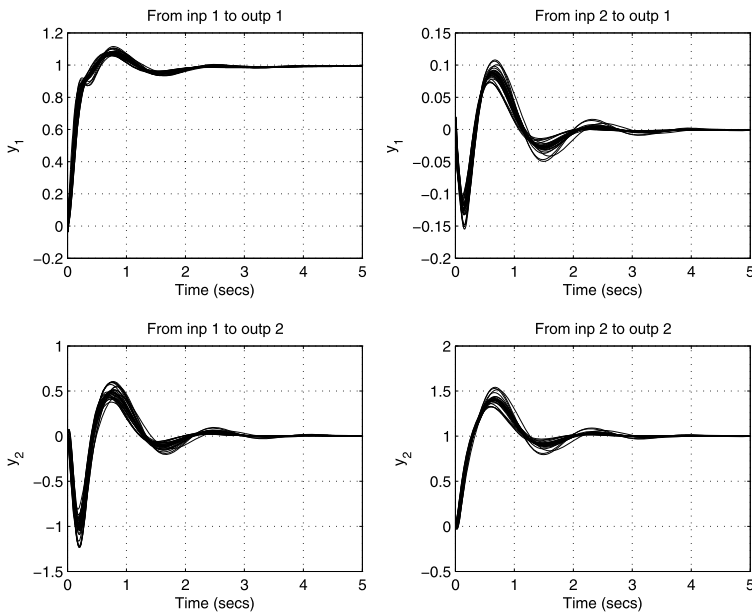


Fig. 10.13 Transient responses of the uncertain system

```

end
grid
title('From inp 1 to outp 2')
xlabel('Time (secs)')
ylabel('y_2')

```

In a similar way the transient responses are found with respect to the unit step reference at the second input. The transient responses to references are shown in Fig. 10.13.

10.3 Worst-Case Gain

The system performance may be assessed approximately by the maximum of the frequency response of the largest singular value (\mathcal{H}_∞ norm) of the sensitivity or complementary sensitivity transfer function matrix. For uncertain systems, it is of interest to determine the largest value of this maximum for the allowed uncertainty. This value, representing the largest possible gain in the frequency domain, is defined as the “worst-case” gain.

In Robust Control Toolbox[®]3 the worst-case gain of uncertain systems may be determined by the command `wcgain`. It calculates upper and lower bounds on the worst gain and determines the uncertainty corresponding to this gain. The syntax, input, and output arguments of `wcgain` are


```
[maxgain,wcunc] = wcgain(sys)
[maxgain,wcunc,info] = wcgain(sys)
```

- `sys`: model of the uncertain system under investigation that may be of class `uss` or of class `ufrd`. If `sys` is of class `uss` the computations are done for appropriately chosen by the function `wcgain` frequency values and if it is of class `ufrd` the associated with `sys` frequency vector is used;
- `maxgain`: structure with following fields.
 - `LowerBound`: lower bound on the worst-case gain;
 - `UpperBound`: upper bound on the worst-case gain;
 - `CriticalFrequency`: frequency value corresponding to `LowerBound`;
- `wcunc`: structure containing a combination of uncertain element values which maximize the system gain—it corresponds to the lower bound of the worst-case gain;
- `info`: structure with the following fields.
 - `Frequency`: frequency vector for which the uncertain system analysis is done;
 - `Sensitivity`: structure containing information for the influence of the uncertain elements on the worst-case gain. For instance, if in the given field is stored the number 20, this means that if the uncertainty range of the corresponding element is increased by 25 % then the value of the worst-case gain is increased by 5 % (25 % of 20).

The command `wcgain` may be used for worst-case gain analysis of continuous-time as well as discrete-time systems. The system type is not necessary to be indicated by the user since this information is contained in `uss` and `ufrd` models.

Example 10.3 Consider the uncertain control system described in Example 10.1. Our task is to determine the worst-case gain for the allowable plant uncertainty.

To determine the worst-case gain we shall make use of the function `wcgain` with output argument the output complementary sensitivity T_o . This is done by the command line

```
[maxgain,wcunc,info] = wcgain(To);
```

As a result one obtains

```
maxgain =
```

```
LowerBound: 4.6934
UpperBound: 4.6943
CriticalFrequency: 3.9226
```

```
wcunc =
```

```
Delta1: [1x1 ss]
Delta2: [1x1 ss]
```

The peak of the maximum singular value of the output complementary sensitivity is equal to 4.6934 (13.43 dB) and is obtained for the uncertain element transfer functions

```
tf(wcunc.Delta1)
```

```
Transfer function:
```

```
-5.547 s^2 + 4.932 s + 1.095e-015
```

```
-----
```

```
s^3 + 6.436 s^2 + 20.32 s + 13.68
```

```
tf(wcunc.Delta2)
```

```
Transfer function:
```

```
-5.547 s^2 + 9.875 s - 8.77e-015
```

```
-----
```

```
s^3 + 7.327 s^2 + 25.26 s + 27.39
```

The second uncertainty has stronger influence on the worst-case gain;

```
info.Sensitivity
```

```
ans =
```

```
Delta1: 30.1655
```

```
Delta2: 80.2187
```

Consider the properties of the closed-loop system with nominal and worst-case gain.

The nominal and worst-case gain of the output complementary sensitivity are obtained by the commands

```
Twc = usubs(To,wcunc);
```

```
omega = logspace(-1,3,100);
```

```
sigma(Twc,'r-',To.Nominal,'b--',omega)
```

```
legend('Worst-case gain','Nominal system',3)
```

and are shown in Fig. 10.14.

The step responses of the nominal and worst-case gain systems are obtained by the lines

```
step(Twc,'r-',To.Nominal,'b--'), grid
```

```
legend('Worst-case gain','Nominal system',4)
```

The step responses are shown in Fig. 10.15. It is seen from the figure that the plant uncertainty may lead to significant deterioration of system behavior. For the worst-case gain the transient responses are very oscillatory with much larger settling time.

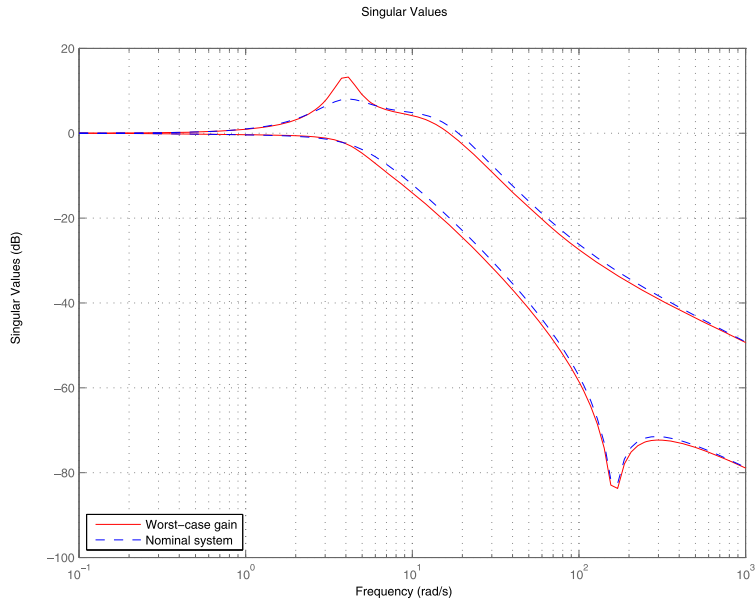


Fig. 10.14 Closed-loop singular values for nominal and worst-case gain

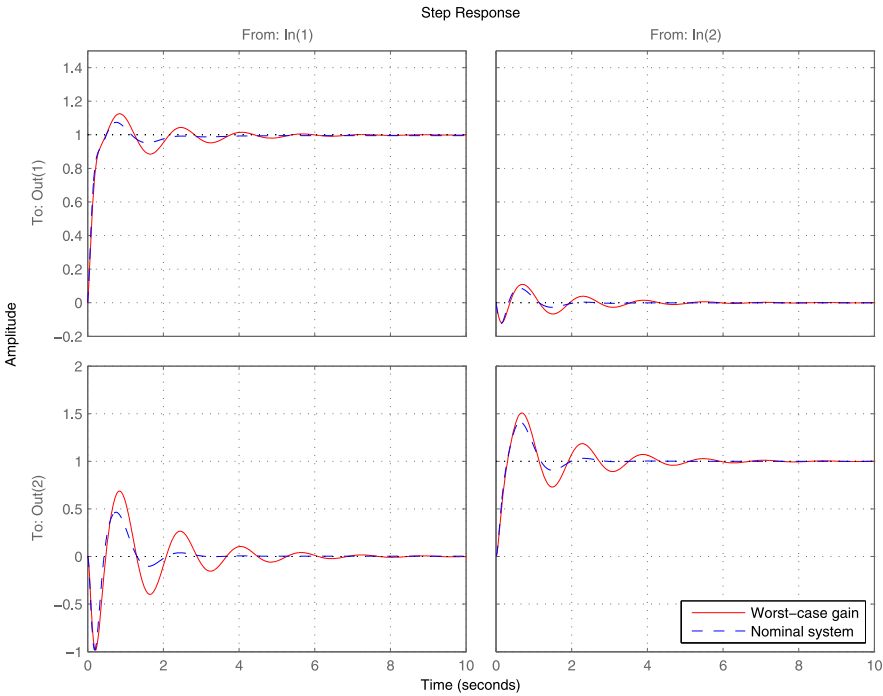


Fig. 10.15 Step responses of the nominal and worst-case gain systems

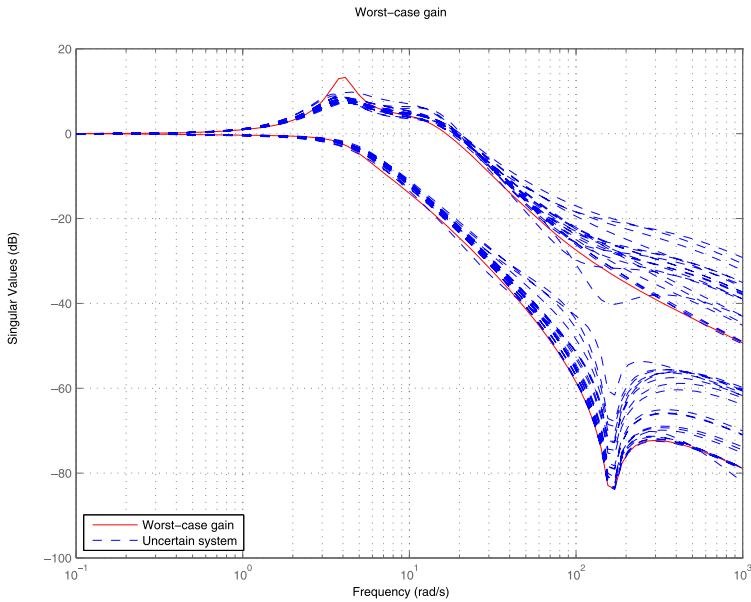


Fig. 10.16 Random and worst-case singular value plots

Consider now the properties of the closed-loop system with random uncertainty values and system with worst-case gain.

The singular value plots of the output complementary sensitivity for systems with random uncertainty values and with worst-case gain are obtained by the lines

```
sigma(Twc,'r-',To,'b--',omega), grid
title('Worst-case gain')
legend('Worst-case gain','Uncertain system',3)
```

These singular plots are shown in Fig. 10.16. The worst-case gain produces the largest pick of the maximum singular value as expected.

The step responses of systems with random uncertainty values and with worst-case gain are obtained by the command lines

```
step(Twc,'r-',To,'b--'), grid
legend('Worst-case gain','Uncertain system',4)
```

and are shown in Fig. 10.17. The step responses determined for the worst-case gain have the worst performance.

Similar results are obtained if instead of the complementary sensitivity function one uses as an input argument to `wcgain` the sensitivity function of the closed-loop system.

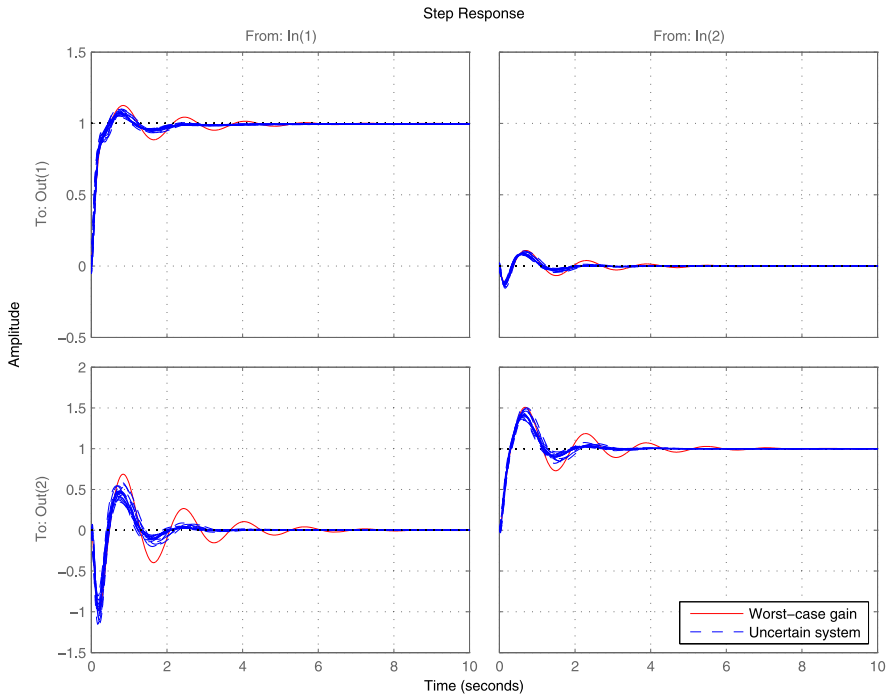


Fig. 10.17 Random and worst-case gain step responses

10.4 Exercises

Exercise 10.1 Given is a two-input/two-output plant with transfer function matrix

$$G = \begin{bmatrix} \frac{k_1}{T_1 s + 1} & \frac{k_2}{T_2 s + 1} \\ \frac{k_3}{T_3 s + 1} & \frac{k_4}{T_4 s - 1} \end{bmatrix}$$

The nominal parameter values are $k_1 = 7.2$, $T_1 = 0.9$, $k_2 = -3$, $T_2 = 1.2$, $k_3 = 2$, $T_3 = 3$, $k_4 = 5$ and $T_4 = 0.7$, the relative parameter uncertainty being 45 %.

The controller transfer matrix is

$$K = \begin{bmatrix} \frac{10(s+1)}{0.3s+1} & 0 \\ 0 & \frac{15(s+2)}{s+1} \end{bmatrix}$$

Analyze the robust stability of the closed-loop system using the function `robuststab`.

Exercise 10.2 Given is a two-input/two-output plant with nominal transfer function matrix

$$G = \frac{1}{s^2 + 2s + 4} \begin{bmatrix} -s + 2 & 2s + 1 \\ -3 & -s + 2 \end{bmatrix}$$

and controller

$$K = \frac{2(s + 2)(s^2 + 2s + 4)}{s(s + 1)(s^2 + 2s + 7)} \begin{bmatrix} -s + 2 & -2s - 1 \\ 3 & -s + 2 \end{bmatrix}$$

The plant has input uncertainties that do not exceed 2 % in the low frequency range, increasing gradually to 100 % at frequency 10 rad/s and reaching 200 % in high frequency range.

Construct a plant model with input multiplicative uncertainty and analyze the robust stability of the closed-loop system.

Exercise 10.3 For the systems presented in Exercises 10.1 and 10.2, analyze the closed-loop robust performance using the function `robustperf`.

Exercise 10.4 Given is a plant with uncertain parameters and input uncertainty whose transfer function is

$$G = \frac{k}{Ts + 1}$$

The gain k and time constant T have nominal values 1 and 2, respectively, and relative uncertainty 25 %. The input plant uncertainty is 5 % in the low frequency range, increasing gradually to 100 % at frequency 2.5 rad/s and reaching 500 % in the high frequency range.

The controller transfer function is

$$K = \frac{k_c(T_c s + 1)}{T_c s}$$

where $k_c = 0.15$, $T_c = 0.4$.

- Construct a plant model representing the input uncertainty as a multiplicative uncertainty.
- Determine the worst case gain of the closed-loop system by using the function `wcgain`;
- Obtain the magnitude responses and transient responses of the closed-loop system for the nominal gain and worst-case gain.
- Obtain the magnitude responses and transient responses of the closed-loop system for certain number of random uncertainty values and compare them with the corresponding responses for the worst-case gain.

Exercise 10.5 For the system described in Exercise [10.2](#)

- (a) Determine the worst closed-loop performance using the command `wcgain` with the output sensitivity as an input argument.
- (b) Obtain the singular value plot of the output sensitivity and the transient responses of the closed-loop system in the nominal and worst case.

Chapter 11

\mathcal{H}_∞ Design

In this chapter we present two important techniques for \mathcal{H}_∞ design of linear control systems, namely the *Loop Shaping Design* and the *mixed sensitivity design*. It is shown how to use the corresponding functions `loopsyn` and `mixsyn` from Robust Control Toolbox^{®3} that allows the controller design to be done easily. We describe also some more sophisticated design methods implemented by the function `hinfsyn` that may produce better nominal performance and robustness of the closed-loop system. It is demonstrated in examples that the \mathcal{H}_∞ design does not always ensure robust stability and robust performance of the closed-loop system which is the main disadvantage of this important design method.

11.1 \mathcal{H}_∞ Loop-Shaping Design

Consider the block-diagram shown in Fig. 11.1. The connection of plant G and controller K is driven by the references r , output disturbances d , and sensor noise n . Here, y are the outputs that have to be manipulated and u are the control signals. In terms of the sensitivity function $S(s) = (I + L(s))^{-1}$ and complementary sensitivity function $T(s) = L(s)(I + L(s))^{-1} = I - S(s)$, where $L(s) = G(s)K(s)$, we have the following important relationships:

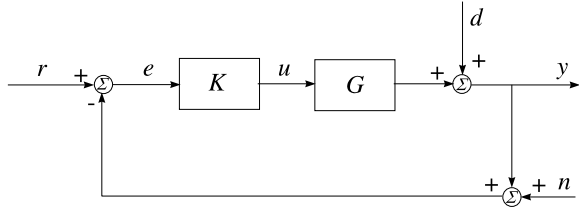
$$y(s) = T(s)r(s) + S(s)d(s) - T(s)n(s) \quad (11.1)$$

$$u(s) = K(s)S(s)[r(s) - n(s) - d(s)] \quad (11.2)$$

and these relationships determine the following goals with respect to the closed-loop system, in addition to the requirement that $K(s)$ should stabilize $G(s)$.

1. For *disturbance attenuation* $\bar{\sigma}(S(j\omega))$ should be made small.
2. For *noise suppression* $\bar{\sigma}(T(j\omega))$ should be made small.
3. For *good reference tracking* we should have $\bar{\sigma}(T(j\omega)) \approx \underline{\sigma}(T(j\omega)) \approx 1$.
4. For *control energy saving* $\bar{\sigma}(R(j\omega))$, where $R(s) = K(s)S(s)$, should be made small.

Fig. 11.1 Closed-loop system



If the unstructured uncertainty in the plant model G is represented by additive perturbation, i.e., $G_p(s) = G(s) + \Delta_a(s)$ then the size of the smallest destabilizing additive perturbation is

$$\bar{\sigma}(\Delta_a(j\omega)) = \frac{1}{\bar{\sigma}(R(j\omega))}$$

5. For *robust stability* in the presence of additive perturbation $\bar{\sigma}(R(j\omega))$ should be made small.

If the uncertainty is modeled by output multiplicative perturbation such that $G_p(s) = (I + \Delta_m(s))G(s)$ then the size of the smallest destabilizing multiplicative perturbation is

$$\bar{\sigma}(\Delta_m(j\omega)) = \frac{1}{\bar{\sigma}(T(j\omega))}$$

That is why

6. For *robust stability* in presence of output multiplicative perturbation $\bar{\sigma}(T(j\omega))$ should be made small.

The requirements 1 to 6 cannot be satisfied simultaneously. For this reason the design of feedback systems is a trade-off between conflicting goals in the frequency domain. Fortunately, it is possible to find a compromise due to the different requirements in the different frequency ranges. For instance, the disturbance attenuation is usually achieved in the low frequency range while the noise suppression is to be met in the high frequency range.

Setting of desired attenuation in the requirement 1 above, for example, may be specified as

$$\bar{\sigma}(S(j\omega)) \leq |W_1^{-1}(j\omega)| \quad (11.3)$$

where $|W_1^{-1}(j\omega)|$ is the desired factor of disturbance attenuation. Making $W_1(j\omega)$ dependent on the frequency ω allows to set different attenuation for different frequency ranges.

The stability margins of the closed-loop system are set by singular values inequalities as

$$\bar{\sigma}(R(j\omega)) \leq |W_2^{-1}(j\omega)| \quad (11.4)$$

or

$$\bar{\sigma}(T(j\omega)) \leq |W_3^{-1}(j\omega)| \quad (11.5)$$

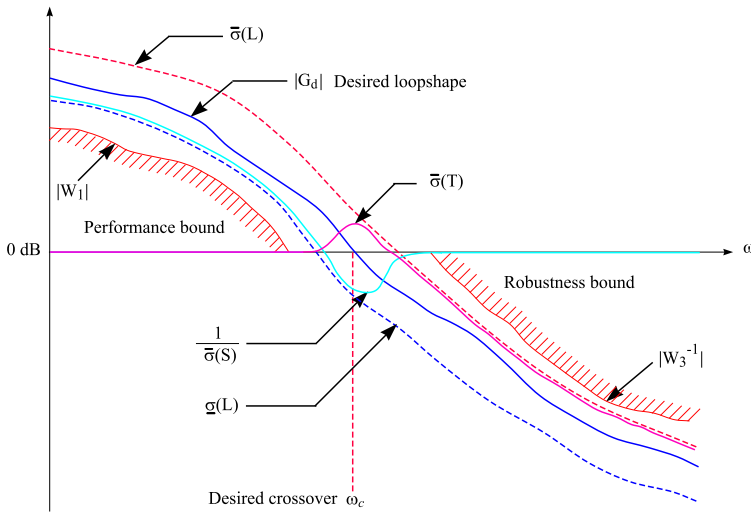


Fig. 11.2 Specifying the desired singular values of L , S , and T

where $|W_2(j\omega)|$ and $|W_3(j\omega)|$ are the largest additive and multiplicative plant perturbations, respectively.

Usually, the effect of all plant uncertainties is represented by a unique, fictitious, multiplicative uncertainty Δ_m so that the design requirements are written as

$$\frac{1}{\sigma_i(S(j\omega))} \leq |W_1(j\omega)|; \quad \sigma_i(T(j\omega)) \leq |W_3^{-1}(j\omega)|$$

These conditions are equivalent to

$$\frac{1}{\underline{\sigma}(S(j\omega))} \leq |W_1(j\omega)|; \quad \bar{\sigma}(T(j\omega)) \leq |W_3^{-1}(j\omega)|$$

as is shown in Fig. 11.2. Note that above the axis 0 dB

$$\underline{\sigma}(L(j\omega)) \approx \frac{1}{\bar{\sigma}(S(j\omega))}$$

while below this axis it is fulfilled that

$$\bar{\sigma}(L(j\omega)) \approx \bar{\sigma}(T(j\omega))$$

It follows from Fig. 11.2 that the controller transfer function matrix $K(s)$ should be determined so that the frequency responses $\bar{\sigma}(L(j\omega))$ and $\underline{\sigma}(L(j\omega))$ avoid the hatched zones. For good performance $\underline{\sigma}(L(j\omega))$ should lie above the performance bound and for robust stability $\bar{\sigma}(L(j\omega))$ should lie below the robustness boundary. The idea of Loop-Shaping Design is to shape the singular values of $L(s)$ by suitable choice of $K(s)$ guaranteeing at the same time the stability of the closed-loop system.

In Robust Control Toolbox[®]3, the Loop-Shaping Design is done by the function `loopsyn`. This function allows to design a stabilizing controller for which the

open-loop frequency responses are optimized so that to match closely the desired frequency responses G_d (see Fig. 11.2). The basic syntax of the function `loopsyn` for Loop-Shaping Design is

$$K = \text{loopsyn}(G, G_d)$$

where G is the plant transfer matrix, G_d corresponds to the desired frequency responses of the open-loop system $L = GK$ and K is the optimal controller. The controller K has the property that it shapes the open-loop $L = GK$ in a such way that it matches as close as possible the frequency responses of G_d under the constraint that the controller should stabilize the closed-loop system.

The desired frequency responses G_d should satisfy the following conditions.

- **Robust stability.** The desired loop G_d should have low gain (as small as possible) in the high frequency range where the model is so inaccurate that its phase responses are completely inaccurate with errors approaching $\pm 180^\circ$.
- **Performance.** The desired loop G_d should have high gain (as great as possible) in the low-frequency range, where the model is adequate, in order to achieve high accuracy in steady state and good disturbance attenuation.
- **Crossover frequency and roll-off.** The desired frequency responses G_d should have crossover frequency ω_c between the frequency ranges described above and should have roll-off -20 or -40 dB/decade below the crossover frequency, which allows the designer to keep the frequency delay less than -180° inside the bandwidth ($0 < \omega < \omega_c$).

Other considerations that may affect the choice of G_d are the poles and zeros of the plant G which are located in the right-half of the complex plane. These poles and zeros impose fundamental limitations on the crossover frequency ω_c . For instance, the crossover frequency ω_c should be greater than the module of each unstable pole and less than the module of each unstable zero, i.e.,

$$\max_{\text{Re}(p_i) > 0} |p_i| < \omega_c < \min_{\text{Re}(z_i) > 0} |z_i|$$

If these requirements are not fulfilled in the selection of G_d , the function `loopsyn` will compute controller K , but the open loop $L = GK$ will have a poor fit to G_d , so that the performance requirements are not satisfied.

Example 11.1 Given a system of fourth order two-input, two-output plant with output multiplicative uncertainty, the transfer function matrix is

$$G = (I_2 + \Delta W)G_{\text{nom}}$$

where

$$G_{\text{nom}} = \begin{bmatrix} \frac{12}{0.2s+1} & -\frac{0.05}{0.1s+1} \\ \frac{0.1}{0.3s+1} & \frac{5}{0.7s-1} \end{bmatrix}$$

$$\Delta = \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix}, \quad |\Delta_1| < 1, |\Delta_2| < 1$$

and

$$W = \begin{bmatrix} w_1 & 0 \\ 0 & w_2 \end{bmatrix}$$

is the matrix of weighting functions determined by the model errors in the two channels. The uncertainty in the first output is 10 % in the low frequency range increases to 100 % at $\omega = 20$ rad/s and reaches 1000 % in the high frequency range. The uncertainty in the second output is 20 % in the low frequency range, 100 % at $\omega = 25$ rad/s and 1000 % in the high frequency range.

The uncertain plant model is obtained by the command lines

```
s = tf('s');
g11 = 12/(0.2*s + 1);
g12 = -0.05/(0.1*s + 1);
g21 = 0.1/(0.3*s + 1);
g22 = 5/(0.7*s - 1);
Gnom = [g11 g12; g21 g22];
w1 = makeweight(0.1,20,10);
w2 = makeweight(0.2,25,10);
W = blkdiag(w1,w2);
Delta_1 = ultidyn('Delta_1',[1 1]);
Delta_2 = ultidyn('Delta_2',[1 1]);
Delta = blkdiag(Delta_1,Delta_2);
G = (eye(2) + Delta*W)*Gnom;
```

The plant singular values are shown in Fig. 11.3. It is seen that after the frequency of 20 rad/s there is a significant uncertainty in the plant which may reach up to 20 dB at frequencies larger than 100 rad/s. Hence, the controller should be designed so that the open-loop system has gain that is smaller than -20 dB for $\omega > 100$ rad/s.

The requirements to the singular values of the open-loop transfer function matrix are.

- **Robustness requirements:** Roll-off -20 dB/decade and a gain of -20 dB at frequency 100 rad/s,
- **Performance requirements:** Maximize $1/\sigma(S)$ in the high frequency range.

Both requirements may be satisfied taking the desired open-loop transfer function matrix as

$$G_d(s) = 10/s$$

(Note that the coefficient in the numerator of $G_d(s)$ is exactly equal to the crossover frequency ω_c .)

The \mathcal{H}_∞ Loop-Shaping Design is done by the commands

```
Gd = 10/s;
[K,cls,gam] = loopsyn(Gnom,Gd);
```

The desired frequency responses and the obtained responses of the open-loop system are shown in Fig. 11.4. They are produced by the command lines

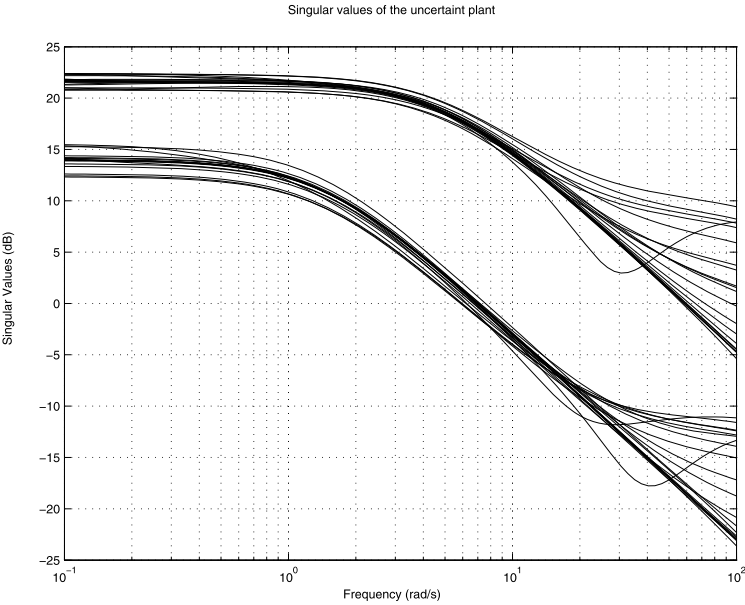


Fig. 11.3 Plant singular values

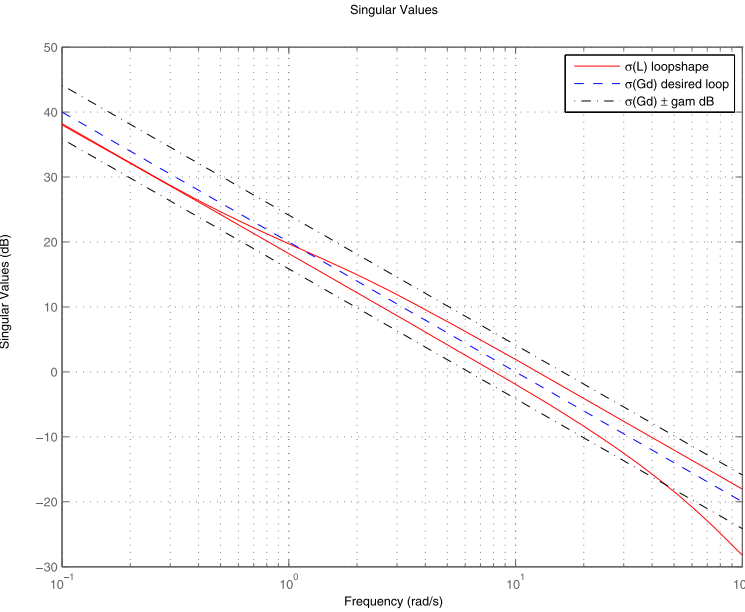


Fig. 11.4 Design results obtained by `loopsyn`

```

looptransfer = loopsens(Gnom,K);
L = looptransfer.Lo;
sigma(L, 'r-', Gd, 'b--', Gd/gam, 'k-', Gd*gam, 'k-', omega)

```

It is seen that both singular values of $L = GK$ are close to the desired frequency response G_d . The number $\pm gam$, dB (i.e., $20 \log_{10}(gam)$) shows the accuracy with which the open-loop frequency responses match the desired frequency responses,

$$\bar{\sigma}(GK), \text{ dB} \geq |G_d| - gam, \text{ dB}, \quad \omega < \omega_c$$

and

$$\bar{\sigma}(GK), \text{ dB} \leq |G_d| + gam, \text{ dB}, \quad \omega > \omega_c$$

In the given case we have

```
gam =
```

```
1.6097
```

The quantities $1/\sigma_i(S)$, characterizing the performance, and the quantities $\sigma_i(T)$, characterizing system robustness, are obtained by the command lines

```

omega = logspace(-1,3,100);
I = eye(size(L));
sigma(I+L, 'r-', T, 'b--', omega);
grid
legend('1/\sigma(S) performance', '\sigma(T) robustness')

```

(Note that the quantities $1/\sigma_i(S)$ are the singular values of the matrix $S^{-1} = I + L$.) The results are shown in Fig. 11.5.

The controller singular values are shown in Fig. 11.6. The controller is of 10th order and has an integrating effect in the low frequency range.

The sensitivity function and complementary sensitivity function of the closed-loop uncertain system are shown in Figs. 11.7 and 11.8, respectively.

The transient responses of the closed-loop uncertain system to unit step inputs are shown in Fig. 11.9. Clearly, the mutual interaction between two channels is very small (i.e., the channels are *decoupled*), which is due to the closeness of both singular values of $L = GK$ to the desired frequency response G_d , which is a diagonal matrix.

11.2 Mixed Sensitivity Design

Another opportunity for Loop-Shaping Design is the \mathcal{H}_∞ *mixed sensitivity design*, which is done by the command

```
K = mixsyn(G,W1,[],W3)
```

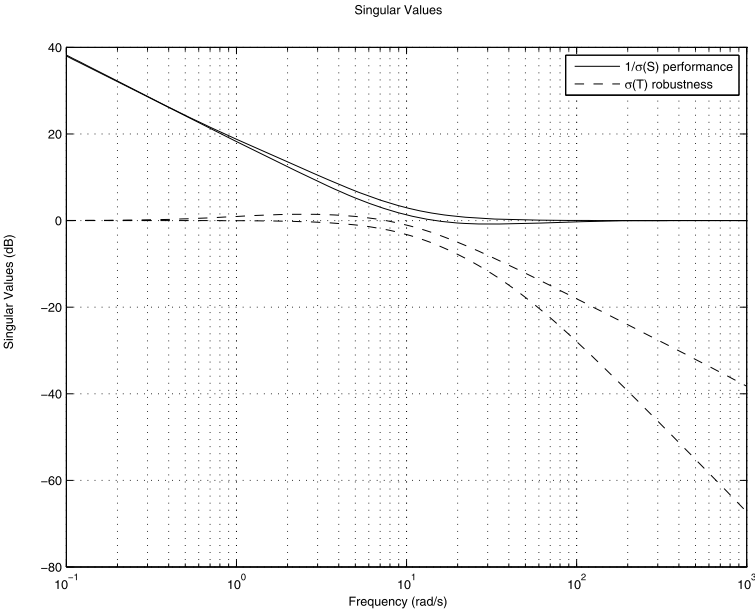


Fig. 11.5 Robustness and performance of the closed-loop system

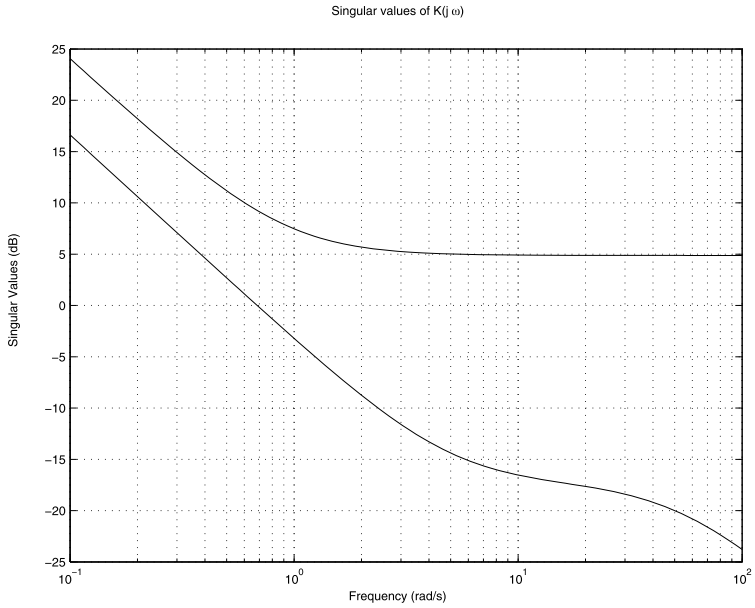


Fig. 11.6 Controller singular value plot

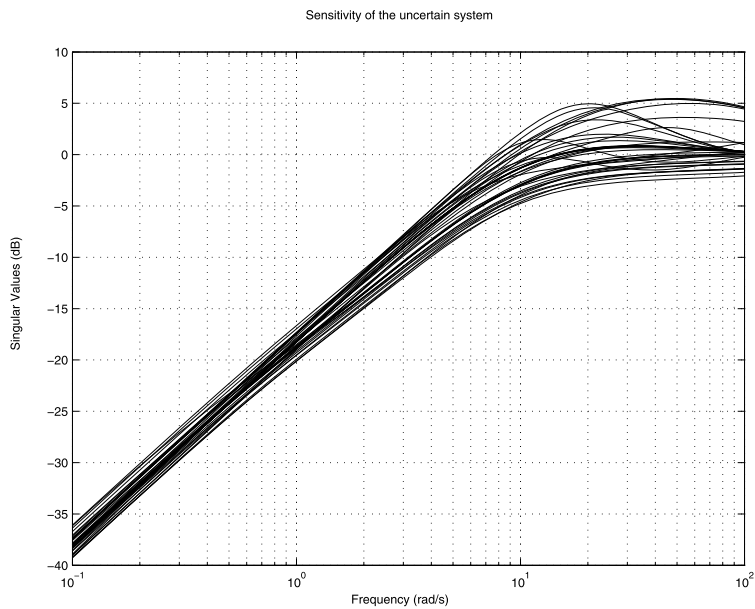


Fig. 11.7 Sensitivity of the uncertain system

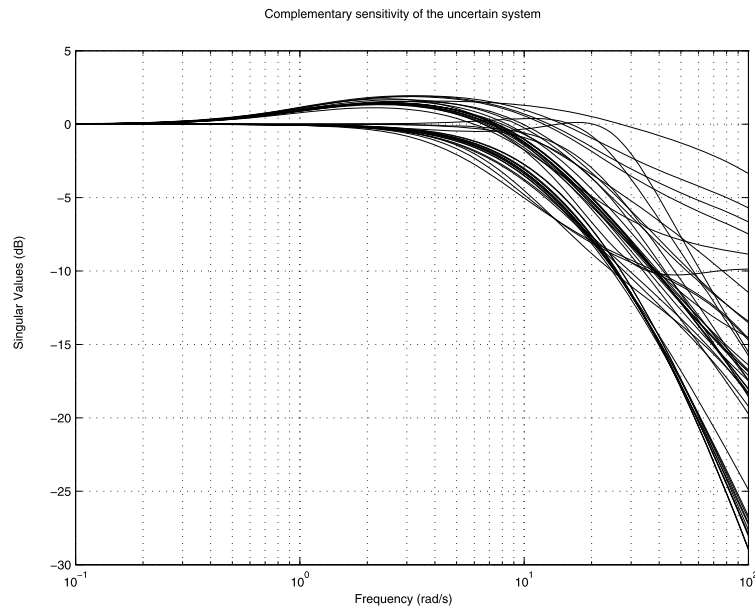


Fig. 11.8 Complementary sensitivity of the uncertain system

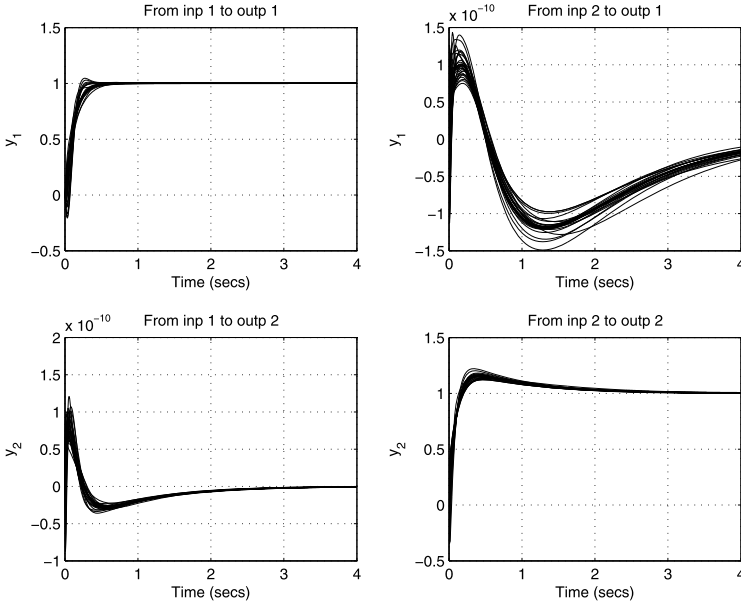


Fig. 11.9 Transient responses of the closed-loop system

In this design, the requirements for robust stability and performance, set by (11.3) and (11.5), are combined in a single requirement with respect to the \mathcal{H}_∞ closed-loop norm

$$\|T_{zr}\|_\infty < 1$$

where (see Fig. 11.10)

$$T_{zr} \stackrel{\text{def}}{=} \begin{bmatrix} W_1 S \\ W_3 T \end{bmatrix}$$

The quantity $\|T_{zr}\|_\infty$ is called *mixed sensitivity cost function*, since it “penalizes” simultaneously sensitivity $S(s)$ and complementary sensitivity $T(s)$. The mixed sensitivity design is achieved when the function W_1 has the desired shape for frequencies $\omega < \omega_c$ and the function W_3^{-1} has the desired shape for frequencies $\omega > \omega_c$. In choosing the weighting functions W_1 and W_3 for a mixsyn design it is necessary to ensure that the crossover frequency for the magnitude response of W_1 at 0 dB is below the crossover frequency of the magnitude response of W_3^{-1} at 0 dB, as is shown in Fig. 11.2. This allows to obtain a “gap” for the desired loop shape G_d to pass between the performance bound W_1 and robustness bound W_3^{-1} . In the opposite case the performance and robustness requirements will not be satisfied.

The function `mixsyn` may be implemented in the more general form

$$[K, cl, gam] = \text{mixsyn}(G, W1, W2, W3)$$

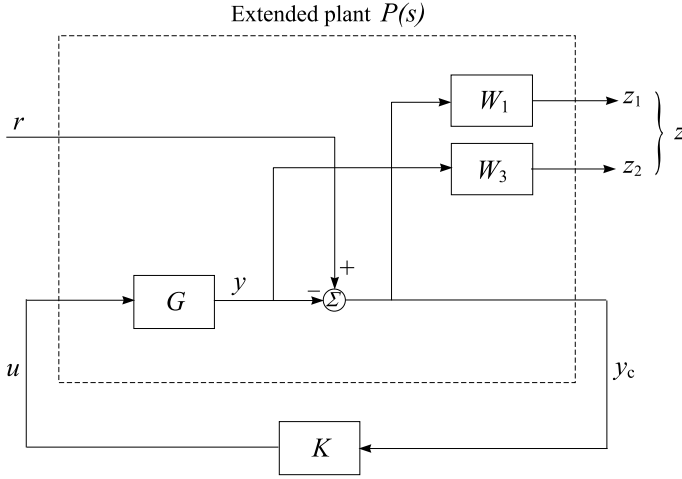


Fig. 11.10 Mixed sensitivity \mathcal{H}_∞ design

which makes it possible to find a controller K minimizing the \mathcal{H}_∞ norm of the weighted mixed sensitivity

$$\left\| \begin{bmatrix} W_1 S \\ W_2 R \\ W_3 T \end{bmatrix} \right\|_\infty$$

The inclusion of the term $W_2 R$ allows to “penalize” the control action $u(t)$ so that to minimize the control energy. The weighting functions W_1 , W_2 , W_3 may be scalar variables, matrices with appropriate dimensions or empty variables.

The matrices S , R , and T obtained on the basis of the computed K satisfy the inequalities

$$\begin{aligned} \bar{\sigma}(S(j\omega)) &\leq \gamma \underline{\sigma}(W_1^{-1}(j\omega)) \\ \bar{\sigma}(R(j\omega)) &\leq \gamma \underline{\sigma}(W_2^{-1}(j\omega)) \\ \bar{\sigma}(T(j\omega)) &\leq \gamma \underline{\sigma}(W_3^{-1}(j\omega)) \end{aligned}$$

where $\gamma = gam$. If $\gamma < 1$ is fulfilled then $\|T_{zr}\|_\infty < 1$.

The weighting functions W_1 , W_2 , and W_3 are used to “shape” the closed-loop transfer functions S , R , and T , respectively, which is different from shaping the open-loop frequency response in \mathcal{H}_∞ Loop-Shaping Design, discussed in Sect. 11.1.

Example 11.2 Consider the system with multiplicative uncertainty given in Example 11.1. The performance and robust stability bounds are specified by the weighting functions

$$W_1 = \frac{s + 10}{2s + 0.3}, \quad W_3 = \frac{s + 10}{0.05s + 20}$$

The design of \mathcal{H}_∞ mixed sensitivity optimal controller is done by the command lines

```
W1 = (s + 10)/(2*s + 0.3);
W3 = (s + 10)/(0.05*s + 20);
[K_h,cl_h,gam] = mixsyn(Gnom,W1,[],W3);
```

The design results are analyzed with the aid of the command lines

```
looptransfer = loopsens(Gnom,K_h);
L = looptransfer.Lo;
T = looptransfer.To;
I = eye(size(L));
figure(1)
omega = logspace(-1,3,100);
sigma(I+L,'b-',W1/gam,'r--',T,'b-.',gam/W3,'r.',omega)
grid
legend('1/\sigma(S) performance', ...
       '\sigma(W1) performance bound', ...
       '\sigma(T) robustness', ...
       '\sigma(1/W3) robustness bound',3)
figure(2)
omega = logspace(-1,3,100);
sigma(L,'b-',W1/gam,'r--',gam/W3,'r.',omega)
grid
legend('\sigma(L)', '\sigma(W1) performance bound', ...
       '\sigma(1/W3) robustness bound',3)
```

In Fig. 11.11 we show the singular values of S and T and the performance and robustness bounds W_1 and $1/W_3$, respectively. It is seen that the minimum singular value of S^{-1} lies below the magnitude response of W_1 and the maximum singular value of T is below the magnitude response of $1/W_3$. This means that the performance and robustness requirements specified by the weighting functions W_1 and W_3 , are satisfied. Actually, we have

```
gam =

    0.7891
```

In Fig. 11.12 we show separately the singular values of the open-loop system L with respect to performance bound W_1 and robustness bound $1/W_3$. As a result of satisfying performance and robustness requirements, the smallest singular value of L lies above the bound W_1 in the low frequency range and the largest singular value of L is below the bound $1/W_3$ in the high frequency range.

The controller singular values are shown in Fig. 11.13. The controller is of eighth order and has an integrating effect in the low frequency range.

The transient responses of the uncertain closed-loop system are shown in Fig. 11.14. Obviously, with this design also the mutual influence between the channels is very small.

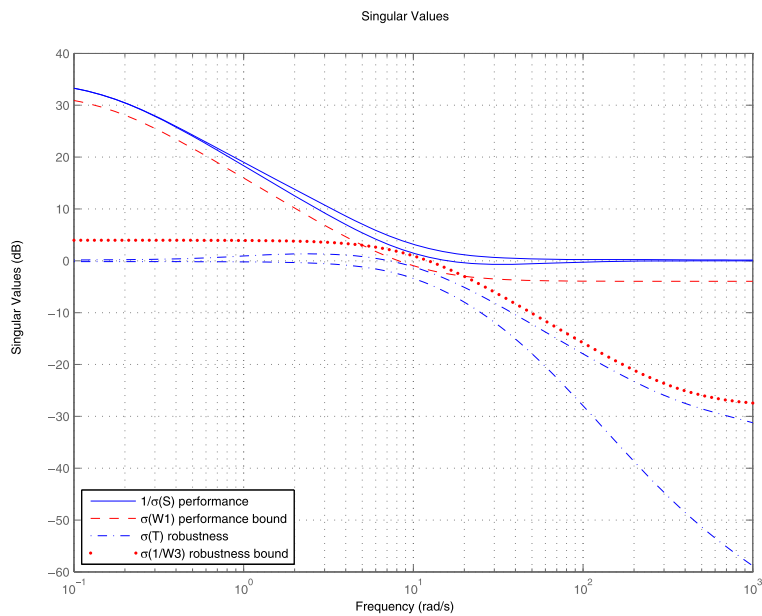


Fig. 11.11 Design results obtained by `mixsyn`

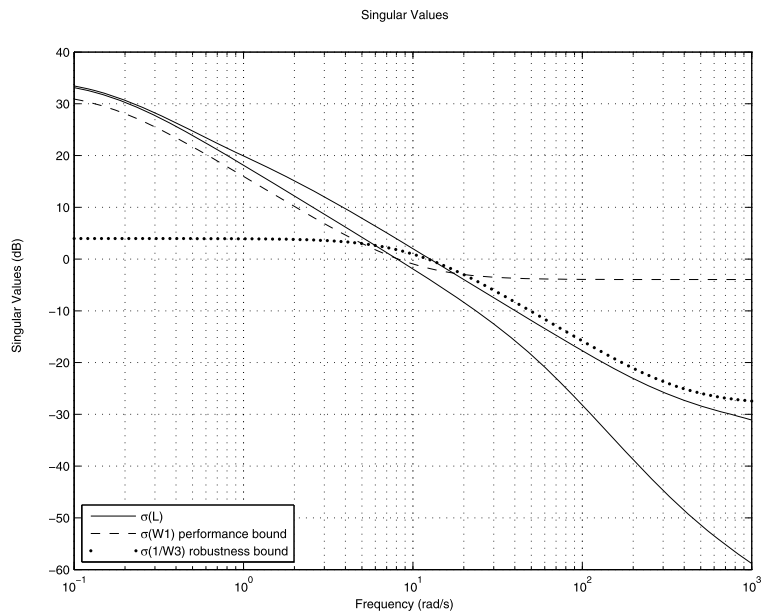


Fig. 11.12 Singular values of the open-loop system

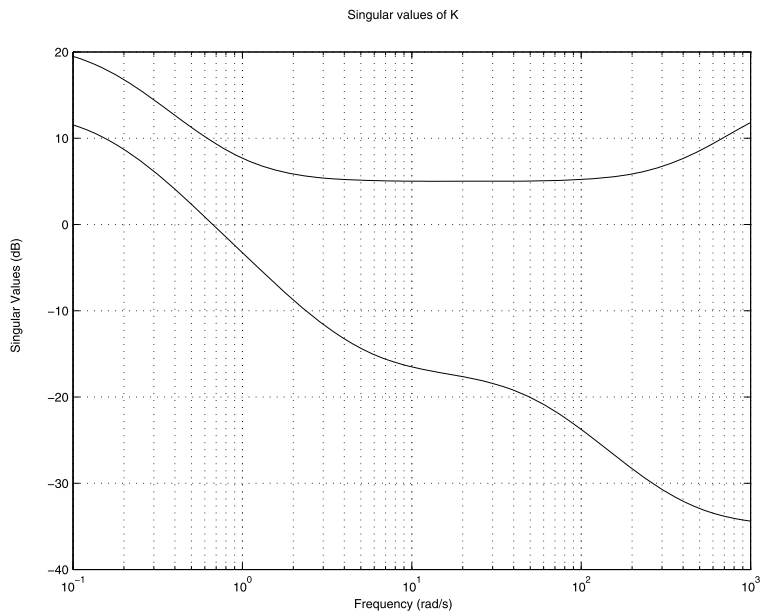


Fig. 11.13 Controller singular value plot

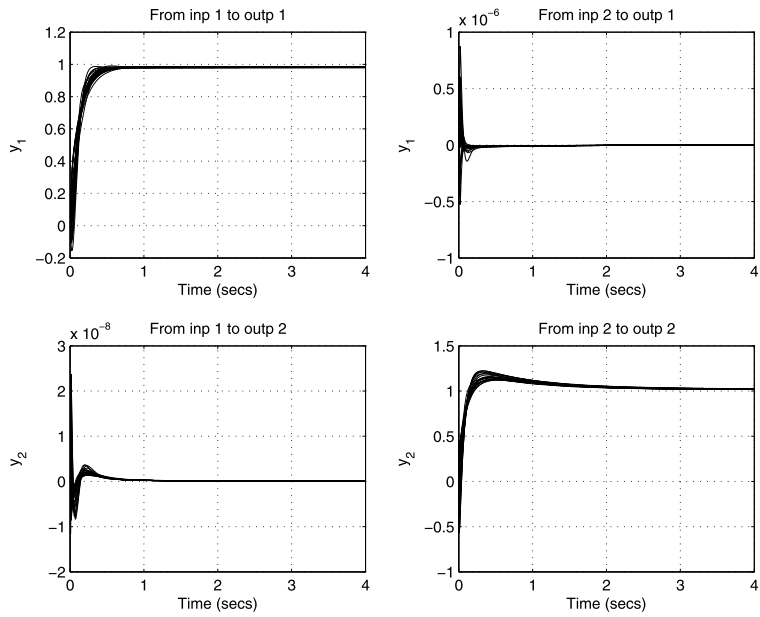


Fig. 11.14 Transient responses of the closed-loop system

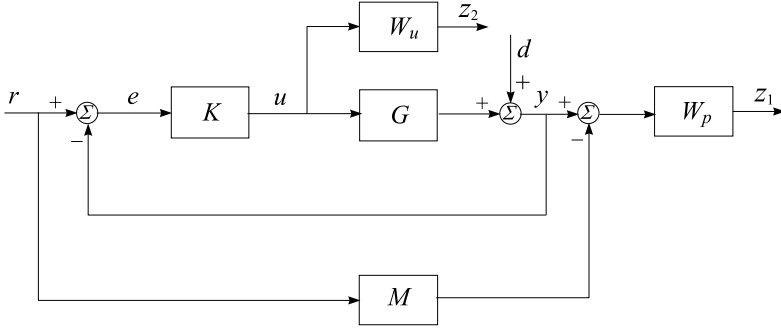
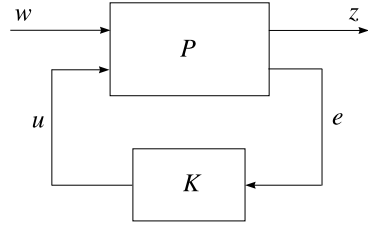


Fig. 11.15 Closed-loop system with model

Fig. 11.16 Standard \mathcal{H}_∞ optimization problem

11.3 Other Versions of \mathcal{H}_∞ Design

11.3.1 \mathcal{H}_∞ Control with Models

\mathcal{H}_∞ control with models may be considered as a generalization of the mixed sensitivity design. The usage of this method allows the robustness and performance requirements to be fulfilled, including also performance requirements to specified model M . The block-diagram of the closed-loop system is shown in Fig. 11.15.

The closed-loop system may be represented in the form of the standard \mathcal{H}_∞ optimization problem shown in Fig. 11.16. The open-loop system is described by

$$\begin{bmatrix} z_1 \\ z_2 \\ e \end{bmatrix} = \begin{bmatrix} -W_p M & W_p & W_p G_{\text{nom}} \\ 0 & 0 & W_u \\ I & -I & -G_{\text{nom}} \end{bmatrix} \begin{bmatrix} r \\ d \\ u \end{bmatrix}, \quad (11.6)$$

which defines the extended transfer matrix P . It may be shown that the closed-loop system is described by

$$z = T_{zw} w$$

where

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \quad w = \begin{bmatrix} r \\ d \end{bmatrix}$$

Table 11.1 \mathcal{H}_∞ functions to be minimized

Function	Description
$W_p(T_o - M)$	Weighted difference between the real and ideal closed-loop system
$W_p S_o$	Weighted output sensitivity
$W_u S_i K$	Weighted control action due to reference
$W_u K S_o$	Weighted control action due to disturbance

and

$$T_{zw} = \begin{bmatrix} W_p(T_o - M) & W_p S_o \\ W_u S_i K & -W_u K S_o \end{bmatrix} \quad (11.7)$$

In the given case we have

$$S_i = (I - K G_{\text{nom}})^{-1}, \quad S_o = (I - G_{\text{nom}} K)^{-1}, \quad T_o = S_o G_{\text{nom}} K$$

and

$$S_i K = K S_o$$

The goal is to determine a stabilizing controller K that minimizes the \mathcal{H}_∞ norm of T_{zw} . The four functions that have to be minimized are described in Table 11.1.

In the table, W_p and W_u are frequency dependent weighting functions that will be called weighting performance function and weighting control function, respectively. The role of the function W_p is to ensure closeness between the closed-loop system and the model in the desired low frequency range and the role of the function W_u is to limit the magnitude of control actions.

The model M is usually set in the diagonal form

$$M = \begin{bmatrix} M_1 & 0 & \dots & 0 \\ 0 & M_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & M_r \end{bmatrix}$$

so that to achieve decoupling of the system outputs with the blocks M_i , $i = 1, \dots, r$ usually being set as second order lags with desired time constants and damping.

Example 11.3 Consider the design of mass–damper–spring system described in Sect. 9.2. The system is of second order and has three uncertain parameters m , c , and k , which vary in the ranges $1.8 \leq m \leq 4.2$, $0.8 \leq c \leq 1.2$, $1.4 \leq k \leq 2.6$.

The design goal is to find \mathcal{H}_∞ control law $u(s) = K(s)y(s)$ for the configuration shown in Fig. 11.15, which fulfills the nominal performance condition

$$\|T_{zw}\|_\infty < 1 \quad (11.8)$$

where the matrix T_{zw} is given by (11.7). The desired closed-loop system model is chosen as a second order lag

$$M = \frac{1}{T^2 s^2 + 2\xi T s + 1}$$

where $T = 1$ s and $\xi = 0.7$. In the given case the weighting performance function is a scalar function $W_p(s) = w_p(s)$ and is chosen as

$$w_p(s) = 0.5 \frac{2s + 1}{2s + tol}$$

where $tol = 0.001$. This weighting function has the purpose to ensure the gain of the loop from r and d to the error $y - y_M$ to be of order tol in the low frequency range which will ensure closeness between the system and model and sufficient disturbance attenuation at the system output. To limit the magnitude of the high frequency components of control action, the weighting control function is chosen as

$$w_u(s) = 0.5 \frac{0.05s + 1}{0.0001s + 0.01}$$

This weighted function ensures attenuation of components with frequency over 10 rad/s.

The specification of model and weighting functions is done by the command lines

```
nuM = 1;
dnM = [1.0^2  2*0.7*1.0  1];
gainM = 1.0;
M = gainM*tf(nuM,dnM);
tol = 10^(-2);
nuWp = [2  1];
dnWp = [2  tol];
gainWp = 5*10^(-1);
Wp = gainWp*tf(nuWp,dnWp);
nuWu = [0.05  1];
dnWu = [0.0001  1];
gainWu = 5.0*10^(-2);
Wu = gainWu*tf(nuWu,dnWu);
```

The matrix P of the extended open-loop system may be found from (11.6). However, the expression obtained in this way contains two times the plant transfer function matrix G . As a consequence, the plant uncertainty will participate twice in the closed-loop system transfer matrix which introduces unnecessary pessimism in the robustness analysis. That is why it is better to determine the transfer matrix of the extended system from the open-loop structure obtained by the commands `interconnect` or `sysic`.

The open-loop system structure is presented in Fig. 11.17. This structure is obtained by using the command `sysic` implementing the following lines:

```
systemnames = ' G M Wp Wu ';
inputvar = '[ ref; dist; control ]';
outputvar = '[ Wp; Wu; ref-G-dist ]';
input_to_G = '[ control ]';
input_to_M = '[ ref ]';
```

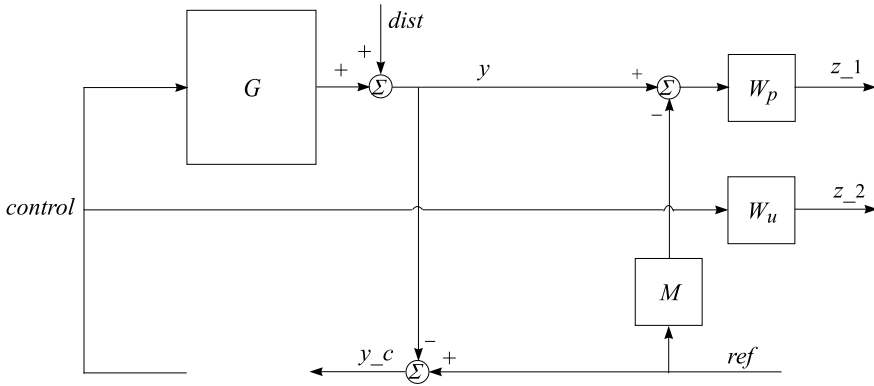
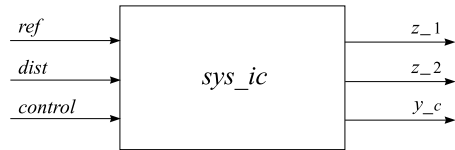



Fig. 11.17 Open-loop system structure

Fig. 11.18 Generalized block-diagram of the open-loop system



```
input_to_Wp = '[ G+dist-M ]';
input_to_Wu = '[ control ]';
sys_ic = sysic;
```

The open-loop system is obtained in the variable `sys_ic` and has three inputs and three outputs (Fig. 11.18).

The design is done by the function `hinfscn` that computes suboptimal \mathcal{H}_∞ controller based on the given open-loop structure. Since the uncertainty is neglected in the \mathcal{H}_∞ design, the nominal value of the open-loop interconnection is used.

Below we give the commands used in the design of \mathcal{H}_∞ controller K followed by the design results.

```
nmeas = 1;
ncont = 1;
gmin = 0.1;
gmax = 10;
tol = 0.001;
[K,clp] = hinfscn(sys_ic.Nominal,nmeas,ncont,gmin,gmax,tol);
get(K)
```

Resetting value of Gamma min based on D_11, D_12, D_21 terms

```
Test bounds:      0.5000 < gamma <=      10.0000
```

```
gamma  hamx_eig  xinf_eig  hamy_eig  yinf_eig  nrho_xy  p/f
10.000  4.8e-001  2.7e-008  5.0e-003  0.0e+000  0.0001   p
```

5.250	4.8e-001	2.7e-008	5.0e-003	0.0e+000	0.0003	p
2.875	4.8e-001	2.8e-008	5.0e-003	0.0e+000	0.0010	p
1.688	4.8e-001	3.0e-008	5.1e-003	0.0e+000	0.0033	p
1.094	4.8e-001	3.4e-008	5.1e-003	0.0e+000	0.0101	p
0.797	4.8e-001	4.3e-008	5.3e-003	0.0e+000	0.0324	p
0.648	4.8e-001	6.3e-008	5.4e-003	0.0e+000	0.1293	p
0.574	4.8e-001	-1.1e+001#	5.5e-003	0.0e+000	16.6897#	f
0.611	4.8e-001	7.6e-008	5.5e-003	0.0e+000	0.2901	p
0.593	4.8e-001	8.6e-008	5.5e-003	0.0e+000	0.6308	p
0.583	4.8e-001	9.3e-008	5.5e-003	0.0e+000	1.3623#	f
0.588	4.8e-001	9.0e-008	5.5e-003	0.0e+000	0.9129	p
0.587	4.8e-001	9.1e-008	5.5e-003	0.0e+000	0.9903	p
0.586	4.8e-001	9.1e-008	5.5e-003	0.0e+000	1.0480#	f
0.586	4.8e-001	9.1e-008	5.5e-003	0.0e+000	1.0013#	f

Gamma value achieved: 0.5866

The number of measurements and the number of controls are equal to 1. The range for γ -iteration is chosen between the numbers 0.1 and 10 with tolerance $tol = 0.001$. At each iteration, the program shows the current value of γ and the results from five tests for existence of suboptimal controller. At the end of each iteration is shown the symbol p or f which indicates whether the current value of γ is accepted or dismissed. The symbol # is used to show which of the five conditions for existence of \mathcal{H}_∞ suboptimal controller is violated for the γ used. When the iteration procedure ends, the minimal achievable value of γ is given. The transfer function matrix of the closed-loop system from `dist` to `e` is saved in the variable `clp`.

The minimal achievable value of γ is 0.586 with the suboptimal controller obtained being of order six, the same as the order of the open-loop system. Since the value of the \mathcal{H}_∞ -norm of the closed-loop system is less than 1, the condition (11.8) for nominal performance is satisfied.

The controller Bode plot is shown in Fig. 11.19.

The robust stability analysis of the closed-loop system is done by the command lines

```
clp_ic = lft(sys_ic,K);
omega = logspace(-2,2,100);
clp_g = ufrd(clp_ic,omega);
opt = robopt('Display','on');
[stabmarg,destabu,report,info] = robuststab(clp_g,opt);
report
semilogx(info.MussvBnds(1,1),'r-',info.MussvBnds(1,2),'b--')
```

The closed-loop system does not achieve robust stability since the smallest value of the structured singular value is about 1.05.

The robust performance analysis is done by the commands

```
opt = robopt('Display','on');
[perfmarg,perfmargunc,report,info] = robustperf(clp_g,opt);
report
semilogx(info.MussvBnds(1,1),'r-',info.MussvBnds(1,2),'b--')
```

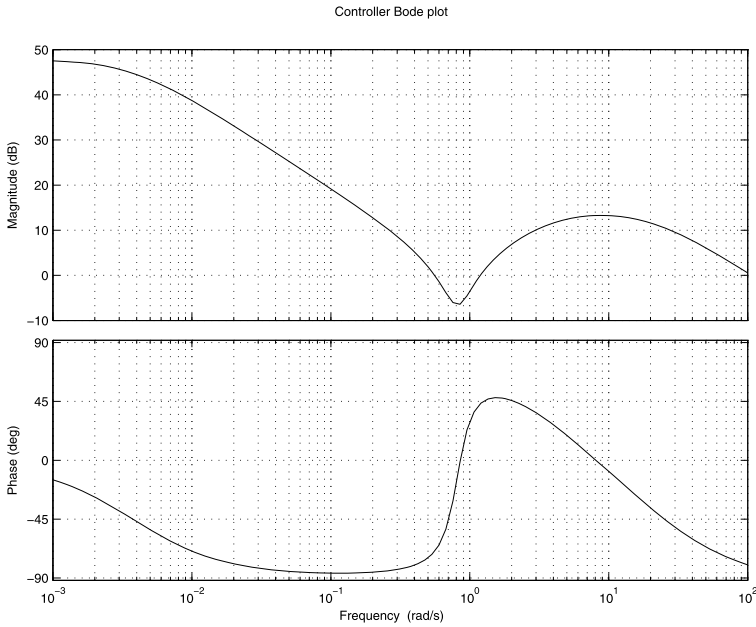


Fig. 11.19 Frequency responses of the \mathcal{H}_∞ controller

the closed-loop system does not achieve robust performance since the performance margin is only 0.6015. This value shows that for uncertainty greater than 60.15 % from the given one the robust performance is violated.

In this way, in the given case the \mathcal{H}_∞ controller designed does not ensure neither robust stability nor robust performance of the closed-loop system. Since the smallest value of γ is less than 1, the closed-loop system achieves only nominal performance. This fact significantly limits the application of \mathcal{H}_∞ control laws in the design of robust systems.

The frequency responses and transient responses of the uncertain system is done for the system without weighting filters. This system is obtained in the variable `cls` after implementing the commands

```
systemnames = ' G ';
inputvar = '[ ref; dist; control ]';
outputvar = '[ G+dist; control; ref-G-dist ]';
input_to_G = '[ control ]';
sim_ic = sysic;
cls = lft(sim_ic,K);
```

The transfer function matrices T_o , S_o , and KS_o are obtained from the structure `cls` by the commands

```
To = cls(1,1);
So = cls(1,2);
```

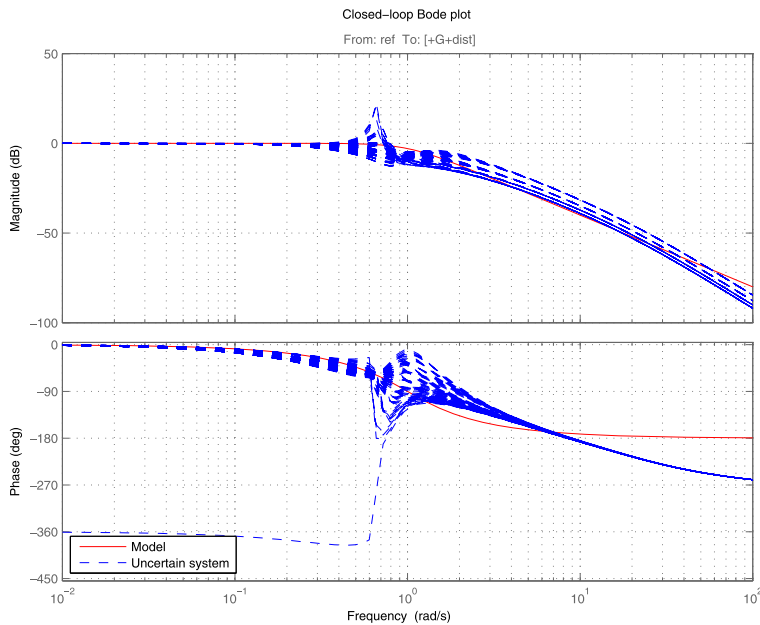


Fig. 11.20 Frequency responses of the closed-loop uncertain system

```
KSo = -cls(2,2);
```

The frequency responses of the uncertain system is done setting by the command `gridureal` four values to each of the three uncertain parameters, which allows to obtain altogether $4^3 = 64$ responses. The magnitude and phase frequency responses of the closed-loop system under variation of the uncertain parameters are obtained by the command lines

```
omega = logspace(-2,2,100);
T64 = gridureal(To,'c',4,'k',4,'m',4);
bode(M,'r-',T64,'b--',omega)
```

and are shown together with model frequency responses in Fig. 11.20. It is seen that the closed-loop responses differ significantly from the model responses around the plant resonant frequency.

The comparison between the sensitivity function of the uncertain system and the inverse performance weighting function is done by the command lines

```
omega = logspace(-3,2,100);
S64 = gridureal(So,'c',4,'k',4,'m',4);
bodemag(1/Wp,'r-',S64,'b--',omega)
```

The result is shown in Fig. 11.21. It is seen that for certain values of the uncertain parameters the magnitude of the sensitivity function around the resonant frequency

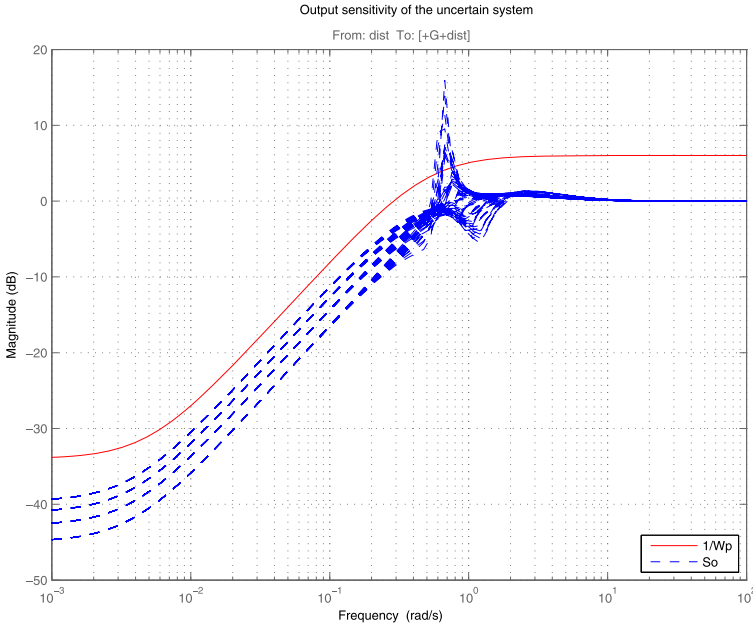


Fig. 11.21 Sensitivity function of the uncertain system

exceeds significantly the limit determined by $1/W_p$, which leads to the violation of the robust performance condition.

The transient responses of the closed-loop system are obtained for the nominal plant since for some values of the uncertain parameters this system becomes unstable.

The transient response with respect to the reference is shown in Fig. 11.22. The process is aperiodic and is slow in comparison to the model.

The corresponding control action is shown in Fig. 11.23.

11.3.2 Two-Degree-of-Freedom \mathcal{H}_∞ Control

The technique of \mathcal{H}_∞ control with model, considered above, may be extended to the design of 2-degree-of-freedom controllers. The block-diagram of the closed-loop system with 2-degree-of-freedom controller is shown in Fig. 11.24. The system has a reference (r), output disturbance (d) and two output errors (z_1) and (z_2). The system M is the ideal model, to which the closed-loop system should match.

In the given case the controller K consists of a feedback controller K_y for disturbance attenuation and a pre-filter K_u to achieve the desired closed-loop performance and is represented as

$$K = [K_r \quad K_y]$$

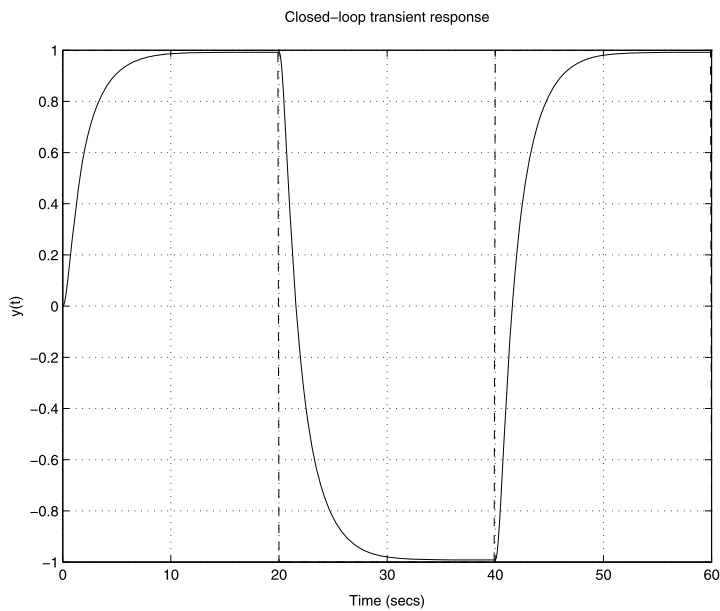


Fig. 11.22 Transient response of the nominal system with respect to reference

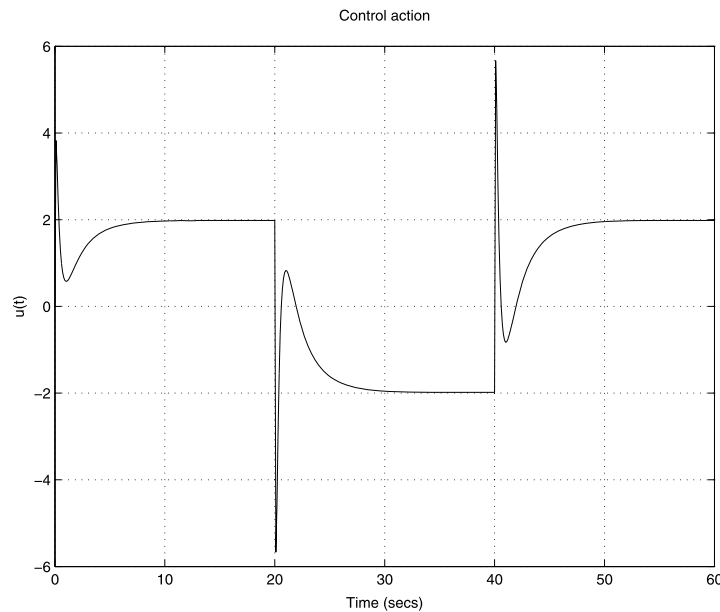


Fig. 11.23 Control action of the nominal system

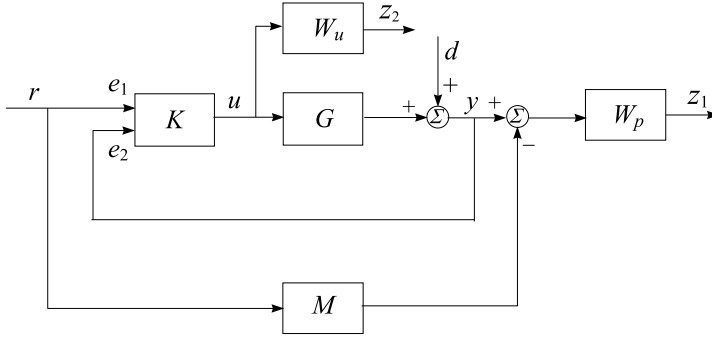


Fig. 11.24 Closed-loop system with 2-degree-of-freedom controller

Table 11.2 \mathcal{H}_∞ function to be minimized

Function	Description
$W_p(S_o G K_r - M)$	Weighted difference between the real and ideal closed-loop system
$W_p S_o$	Weighted output sensitivity
$W_u S_i K_r$	Weighted control action due to reference
$W_u K_y S_o$	Weighted control action due to disturbance

The transfer function matrices K_r and K_y may be obtained as follows.

The closed-loop system may be represented in the form of the standard problem shown in Fig. 11.16. The system is described by

$$\begin{bmatrix} z_1 \\ z_2 \\ e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} -W_p M & W_p & W_p G \\ 0 & 0 & W_u \\ I & 0 & 0 \\ 0 & I & G \end{bmatrix} \begin{bmatrix} r \\ d \\ u \end{bmatrix}$$

The closed-loop transfer function matrix is given by

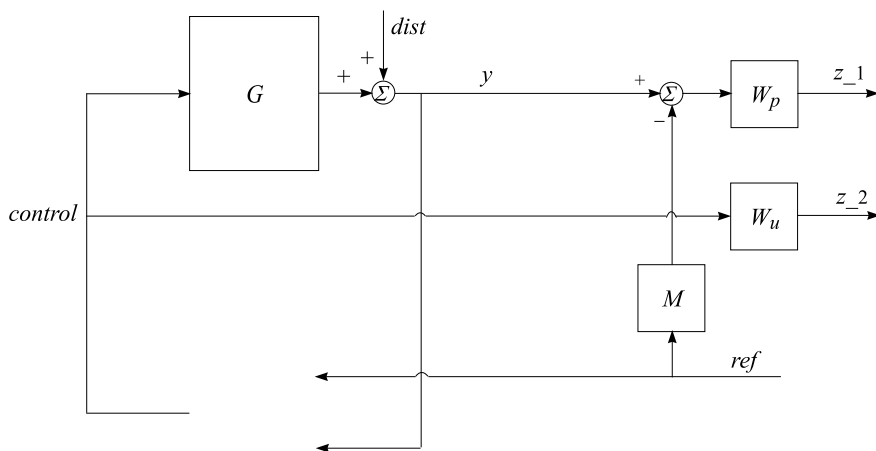
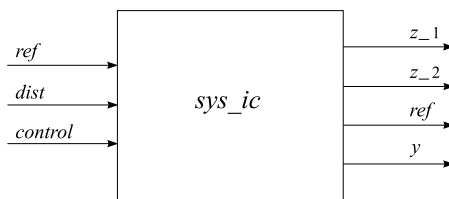
$$T_{zw} = \begin{bmatrix} W_p(S_o G K_r - M) & W_p S_o \\ W_u S_i K_r & W_u K_y S_o \end{bmatrix}$$

where the input and output sensitivities are equal to

$$S_i = (I - K_y G)^{-1}, \quad S_o = (I - G K_y)^{-1}$$

respectively. Again, the goal is to obtain a stabilizing controller K that minimizes the \mathcal{H}_∞ norm of T_{zw} . The four blocks of the matrix T_{zw} , whose magnitude should be minimized, have the same meaning as shown in Table 11.1. The four function that should be minimized are described in Table 11.2.

As previously, W_p and W_u are frequency dependent filters chosen as described above.

**Fig. 11.25** Open-loop system structure**Fig. 11.26** Generalized block-diagram of the open-loop system

The design of 2-degree-of-freedom controller is done taking as input to the controller the vector $y_c = [r^T \ y^T]^T$ instead of the vector $y_c = r - y$ as in the case of one degree-of-freedom.

Example 11.4 Consider the mass-damper-spring system for which a one-degree-of-freedom controller with model was designed in Example 11.3. The desired model M and the weighting functions W_p and W_u are the same as the previously used. The block-diagram of the open-loop system is given in Fig. 11.25. The system structure is obtained by the command lines

```
systemnames = ' G M Wp Wu ';
inputvar = '[ ref; dist; control ]';
outputvar = '[ Wp; Wu; ref; G+dist ]';
input_to_G = '[ control ]';
input_to_M = '[ ref ]';
input_to_Wp = '[ G+dist-M ]';
input_to_Wu = '[ control ]';
sys_ic = sysic;
```

The open-loop system has three inputs and four outputs (Fig. 11.26). In this case the \mathcal{H}_∞ design is done by the commands

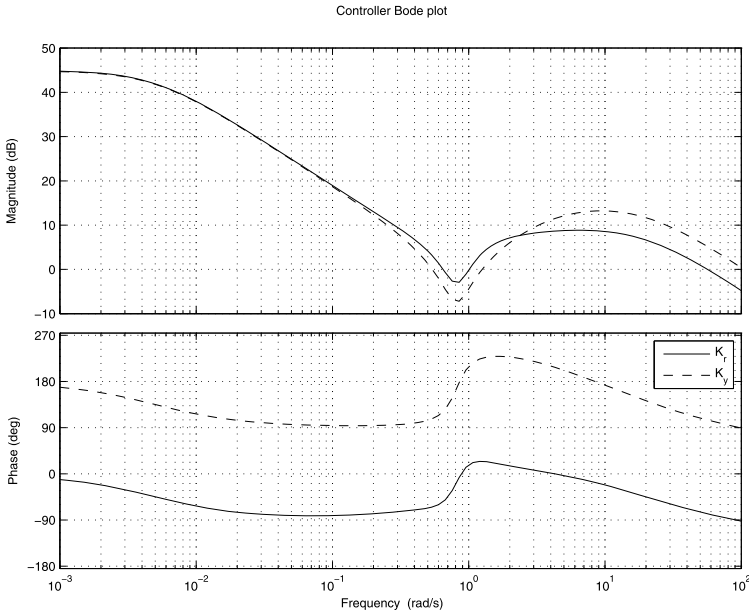


Fig. 11.27 Frequency responses of the \mathcal{H}_∞ 2-degree-of-freedom controller

```
nmeas = 2;
ncont = 1;
gmin = 0.1;
gmax = 10;
tol = 0.001;
[K,clp] = hinfsyn(sys_ic.Nominal,nmeas,ncont,gmin,gmax,tol);
```

The controller obtained is again of sixth order. The frequency responses of the two parts of the controller are shown in Fig. 11.27.

The robust properties of the closed-loop system are close to the properties of the system with one-degree-of-freedom controller. Again, in the given case the \mathcal{H}_∞ controller does not ensure robust stability and robust performance of the closed-loop system.

The transient response of the closed-loop system is shown in Fig. 11.28. the transient response is significantly faster comparing to the previous example (compare with Fig. 11.22).

The closed-loop control action has nearly 2 times smaller magnitude in comparison with the control action of the system with one-degree-of-freedom controller (compare Figs. 11.29 and 11.23).

In this way the 2-degree-of-freedom controller allows to achieve better nominal performance of the closed-loop system for a smaller magnitude of the control action. Both controllers, however, do not ensure robust stability and robust performance of the closed-loop system.

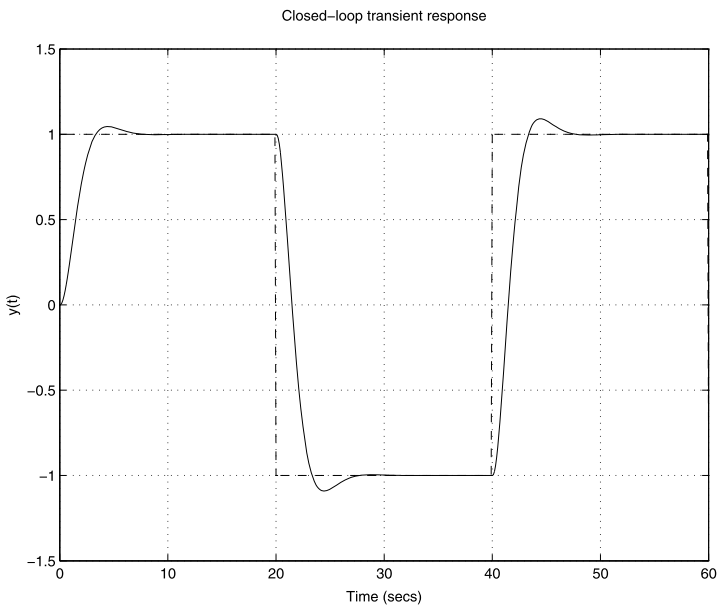


Fig. 11.28 Transient responses of the nominal system with respect to reference

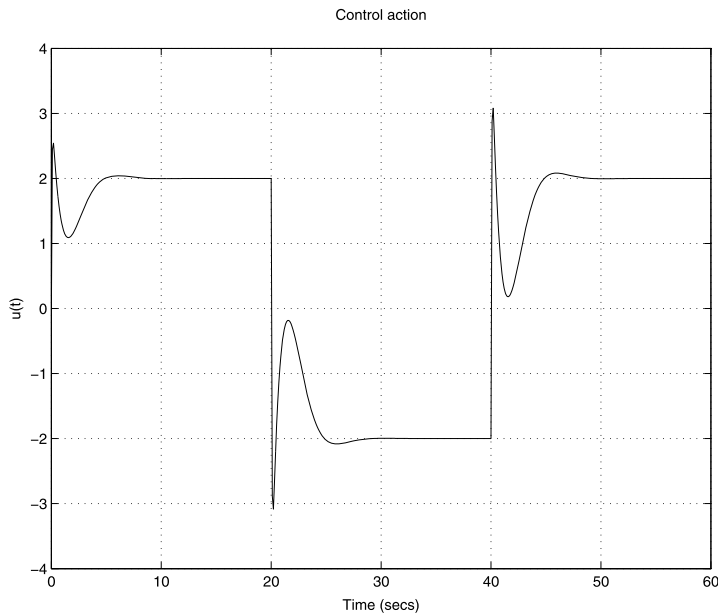


Fig. 11.29 Control action of the nominal system

11.4 Exercises

Exercise 11.1 Given a two-input, two-output system with transfer function matrix

$$G = \begin{bmatrix} \frac{k_1}{T_1 s + 1} & -\frac{0.05}{0.1s + 1} \\ \frac{0.1}{0.3s + 1} & \frac{k_2}{T_2 s + 1} \end{bmatrix}$$

where the coefficients k_1 and k_2 have nominal values 12 and 5, respectively, and relative uncertainty 15 %, and the time constants T_1 and T_2 have nominal values 0.2 and 0.7, respectively, and relative uncertainty 20 %.

- Build uncertainty model of the given system;
- design a loop shaping controller that fulfills the following requirements:
 - robustness requirements: roll off -20 dB/decade and attenuation of -20 dB for the frequency 100 rad/s,
 - performance requirements: maximize $1/\sigma(S)$ in the low frequency range;
- obtain the singular value plot of the open-loop system and compare with the desired frequency response;
- obtain the singular value plots of the sensitivity matrix and complementary sensitivity function of the uncertain closed-loop system;
- obtain the transient response of the uncertain closed-loop system.

Exercise 11.2 For the system from Exercise 11.1 do the following.

- Design a mixed sensitivity \mathcal{H}_∞ controller with performance and robustness bounds specified by

$$W_1 = \frac{s + 10}{2s + 0.3}, \quad W_3 = \frac{s + 10}{0.05s + 20}$$

- obtain the singular value plots of the sensitivity matrix and complementary sensitivity matrix and compare them with the performance and robustness bounds;
- obtain the singular value plots of the sensitivity matrix and complementary sensitivity matrix of the uncertain closed-loop system;
- obtain the transient responses of the uncertain closed-loop system.

Exercise 11.3 Given a plant with transfer function

$$G = \frac{s - 10}{(s + 1)(s + 10)}$$

design a \mathcal{H}_∞ 2-degree-of-freedom controller minimizing the performance index

$$\left\| \begin{bmatrix} W_p S G \\ W_u K S G \end{bmatrix} \right\|_\infty$$

where

$$W_p = \frac{1}{s + 0.001}, \quad W_u = \frac{s + 2}{s + 10}$$

For this aim, build the extended system structure by using the command `sysic` and implement the function `hinfsyn` for \mathcal{H}_∞ design.

Obtain the frequency responses of the sensitivity function and the transient response of the closed-loop system.

Exercise 11.4 Given below is a system with plant transfer function matrix

$$G = \begin{bmatrix} \frac{k_1}{T_1 s + 1} & -\frac{0.05}{0.1s + 1} \\ \frac{0.1}{0.3s + 1} & \frac{k_2}{T_2 s - 1} \end{bmatrix}$$

where the coefficients k_1 and k_2 have nominal values 12 and 5, respectively, and relative uncertainty 15 % and the time constants T_1 and T_2 have nominal values 0.2 and 0.7, respectively, and relative uncertainty 20 %. (The plant is the same as in Exercise 11.1.)

- Build uncertain model of the given system.
- Design an \mathcal{H}_∞ 2-degree-of-freedom controller minimizing the performance index

$$\left\| \begin{bmatrix} W_p S(G_{\text{nom}}) \\ W_u K S(G_{\text{nom}}) \end{bmatrix} \right\|_\infty$$

where

$$W_p = \begin{bmatrix} w_p & 0 \\ 0 & w_p \end{bmatrix}, \quad W_u = \begin{bmatrix} w_u & 0 \\ 0 & w_u \end{bmatrix}$$

and

$$w_p = \frac{0.95(s^2 + 2000s + 4000)}{s^2 + 1900s + 10}, \quad w_u = \frac{10^{-6}(0.1s + 1)}{0.001s + 1}$$

- Analyze the robust stability of the closed-loop system.
- Analyze the robust performance of the closed-loop system.
- Obtain the singular value plots of the sensitivity function and complementary sensitivity function of the uncertain closed-loop system.
- Obtain the transient response of the uncertain closed-loop system.

This page intentionally left blank

Chapter 12

μ -Synthesis

This chapter is devoted to the implementation aspects of one of the most important techniques in Robust Control Design, namely the μ -synthesis. For properly chosen weighting functions this design method usually produces a controller that ensures both robust stability and robust performance of the closed-loop system. After a short statement of the μ -synthesis problem, we describe the usage of the function `dkssyn` from Robust Control Toolbox[®]3 that determines iteratively a controller which minimizes the closed-loop structured singular value. Some versions of the μ -synthesis are considered next, including the design of 2-degree-of-freedom controller that usually produces the best results with respect to the closed-loop robust performance. At the end of the chapter we discuss some practical aspects of the μ -analysis and μ -synthesis.

12.1 The μ -Synthesis Problem

Consider a control problem in the Linear Fractional Transformation shown in Fig. 12.1.

The system denoted by P is the open-loop connection and represents all known elements including the nominal system model and the performance weighting functions, as well as the uncertainty weighting functions. The block Δ is the uncertain element from the set $\mathbf{\Delta}$, which parameterizes all supposed model uncertainty. The controller is denoted by K . Inputs to P are three sets of signals: inputs u_Δ due to the uncertainty, references and disturbances w and controls u . Three sets of outputs are generated: outputs y_Δ due to the uncertainty, errors or controlled outputs z and measurements y .

The set of systems to be controlled is described by the LFT

$$\left\{ F_U(P, \Delta) : \Delta \in \mathbf{\Delta}, \max_{\omega} \bar{\sigma}[\Delta(j\omega)] \leq 1 \right\}$$

Fig. 12.1 Block-diagram of the closed-loop system

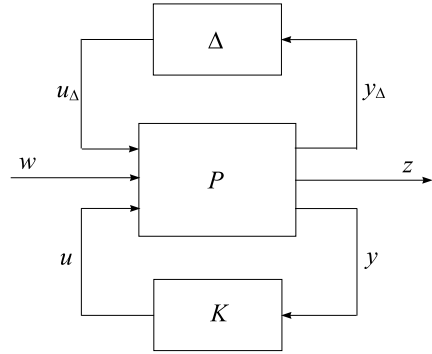
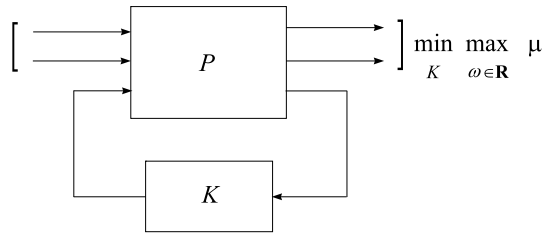


Fig. 12.2 μ -Synthesis



The design goal is to determine a controller K , stabilizing the nominal system; also for all $\Delta \in \mathbf{\Delta}$, $\max_{\omega} \bar{\sigma}[\Delta(j\omega)] \leq 1$, the closed-loop system is stable and satisfies

$$\|F_U[F_L(P, K), \Delta]\|_{\infty} < 1$$

For given arbitrary K this performance criterion may be tested by using the robust performance test on the Linear Fractional Transformation $F_L(P, K)$. The robust performance test should be performed with respect to the extended uncertain structure

$$\Delta_P \stackrel{\text{def}}{=} \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_F \end{bmatrix} : \Delta \in \mathbf{\Delta}, \Delta_F \in \mathcal{C}^{n_w \times n_z} \right\}$$

where Δ_F is a fictitious complex (unstructured) uncertainty. The system with controller K achieves robust performance if and only if

$$\mu_{\Delta_P}(F_L(P, K)(j\omega)) < 1$$

The aim of the μ -synthesis is to minimize the peak value of the structured singular value $\mu_{\Delta_P}(\cdot)$ of the closed-loop transfer function matrix $F_L(P, K)$ over the set of all stabilizing controllers K . This is written as

$$\min_{K \text{ stabilizing}} \max_{\omega} \mu_{\Delta_P}(F_L(P, K)(j\omega)) \quad (12.1)$$

and is shown in Fig. 12.2.

12.2 μ -Synthesis by D - K Iterations

The μ -synthesis method by D - K iterations (Sect. 6.2) is realized in Robust Control Toolbox[®]3 by the function `dksyn`. It is implemented by the command line

```
[K, clp, bnd] = dksyn(P, nmeas, ncont, opt)
```

Input arguments

- `P` is the uncertainty plant model (should be of class `uss`) obtained, for instance, by the command `sysic` (different from the command for \mathcal{H}_∞ design `hinfsyn` that may work with the nominal system only);
- `nmeas`: the number of plant measured outputs (corresponds to the last `nmeas` measurement outputs);
- `ncont`: the number of plant control inputs (corresponds to the last `ncont` control inputs);
- `opt`: optional parameter. It is an object created by the command `opt = dkitopt` and has the following elements:
 - `opt.FrequencyVector`: vector with frequency values used in the μ -analysis; if not set, it is chosen automatically;
 - `opt.InitialController`: controller used to initiate first iteration, default is an empty SS object;
 - `opt.AutoIter`: automated μ -synthesis mode, default is 'on';
 - `opt.DisplayWhileAutoIter`: displays iteration progress in `opt.AutoIter` mode, default is 'off';
 - `opt.StartingIterationNumber`: starting iteration number, default is 1;
 - `opt.NumberOfAutoIterations`: number of D - K iterations to perform, default is 10;
 - `opt.AutoScalingOrder`: maximum state order for fitting D -scaling data, default is 5;
 - `opt.AutoIterSmartTerminate`: automatic termination of iteration procedure based on progress of design iteration, default is 'on';
 - `opt.AutoIterSmartTerminateTol`: tolerance used by `opt.AutoIterSmartTerminate`, default is 0.005.

Output arguments

- `K`: controller designed (SS object);
- `clp`: closed-loop system model (USS object)—may be obtained also as `clp = lft(P, K)`;
- `bnd`: upper bound on the robust performance of the closed-loop system `clp`;
- `dkinfo`: N -by-1 cell array where N is the total number of iterations performed. The i th cell contains a structure with several fields, most significant of which are:
 - `K`: Controller at i th iteration, SS object.
 - `Bnds`: Robust performance bound for the closed-loop system.
 - `MussvBnds`: Upper and lower μ bounds, an FRD object computed by the auxiliary function `mussv`.

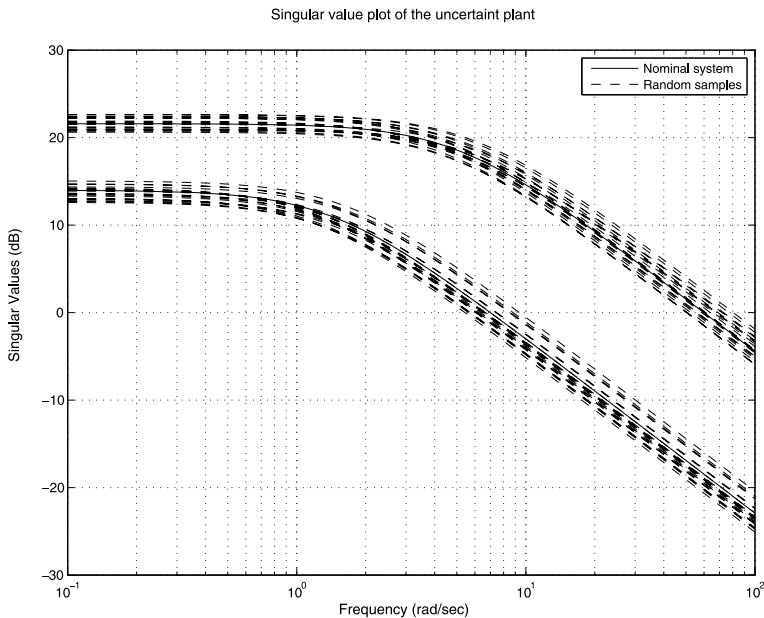


Fig. 12.3 Frequency responses of the uncertain plant

- **MussvInfo**: Structure returned from the function `mussv` at each iteration.

The information contained in these fields may be very important in some cases. If, for instance, one wants to take the controller from the fourth iteration, this may be done by the line

```
K_4 = dkinfo{4}.K;
```

The command `dksyn` works with continuous-time as well as discrete-time systems. It is not necessary to be indicated by the user since this information is contained in the `ss` and `uss` objects.

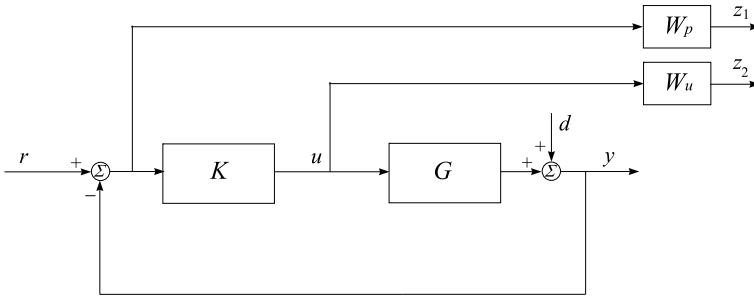
Example 12.1 Consider a two-input, two-output system with transfer function matrix

$$G = \begin{bmatrix} \frac{k_1}{T_1 s + 1} & -\frac{0.05}{0.1s + 1} \\ \frac{0.1}{0.3s + 1} & \frac{k_2}{T_2 s - 1} \end{bmatrix}$$

where the coefficients k_1 and k_2 have nominal values 12 and 5, respectively, and relative uncertainty 15 %, and the time constants T_1 and T_2 have nominal values 0.2 and 0.7, respectively, and relative uncertainty 20 %.

The frequency responses of the plant singular values are shown in Fig. 12.3.

The design goal is to achieve robust stability and robust performance of the closed-loop system shown in Fig. 12.4, in the presence of plant uncertainty and output disturbances.

**Fig. 12.4** Block-diagram of the closed-loop system

The closed-loop system is described by

$$z = T_{zw} w$$

where

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \quad w = \begin{bmatrix} r \\ d \end{bmatrix}$$

and

$$T_{zw} = \begin{bmatrix} W_p S_o & -W_p S_o \\ W_u K S_o & -W_u K S_o \end{bmatrix}$$

where

$$S_o = (I - GK)^{-1}$$

is the output sensitivity transfer function matrix.

To achieve closed-loop robust performance, i.e.

$$\mu_{\Delta P}(T_{zw}(j\omega)) < 1$$

means that the condition

$$\|T_{zw}\|_{\infty} < 1$$

will be fulfilled for each possible plant uncertainty. This, in turn, guarantees fulfillment of the conditions

$$\|W_p S_o\|_{\infty} < 1, \quad \|W_u K S_o\|_{\infty} < 1$$

In the given case the performance weighting and control weighting functions are taken in the form

$$W_p = \begin{bmatrix} w_p & 0 \\ 0 & w_p \end{bmatrix}, \quad W_u = \begin{bmatrix} w_u & 0 \\ 0 & w_u \end{bmatrix}$$

where

$$w_p(s) = 0.5 \frac{s + 10}{s + 0.3}$$

and

$$w_u(s) = 0.1 \frac{0.001s + 1}{0.0001s + 1}$$

The open-loop transfer function matrix P is obtained by the command lines

```
systemnames = ' G Wp Wu ' ;
inputvar = '[ ref{2}; dist{2}; control{2} ]';
outputvar = '[ Wp; Wu; ref-G-dist ]';
input_to_G = '[ control ]';
input_to_Wp = '[ ref-G-dist ]';
input_to_Wu = '[ control ]';
sys_ic = sysic;
```

and is stored in the variable `sys_ic`. The open-loop system is of order 8.

The μ -synthesis is done by the commands

```
nmeas = 2;
ncont = 2;
fv = logspace(-3,3,100);
opt = dkitopt('FrequencyVector',fv, ...
             'DisplayWhileAutoIter','on', ...
             'NumberOfAutoIterations',3)
[K,cl_mu,bnd_mu,dkinfo] = dksyn(sys_ic,nmeas,ncont,opt);
```

setting three steps of the DK-iterations.

After performing three iterations one obtains the report

Iteration Summary

Iteration #	1	2	3
Controller Order	8	16	12
Total D-Scale Order	0	8	4
Gamma Achieved	1.990	1.046	1.218
Peak μ -Value	1.433	1.001	1.173

This report shows that at the second iteration the closed-loop system practically achieves robust performance for a value of μ being close to 1. The controller order at this iteration becomes equal to 16. (Different from \mathcal{H}_∞ design, the controller order here may be obtained significantly higher than the order of the open-loop system due to the use of scaling matrices.) Since at the third iteration the value of μ is greater than the value at the second iteration, the program chooses automatically the controller from the second step.

The singular value plot of the controller is shown in Fig. 12.5.

The robust stability and robust performance analysis are done by the command lines

```
clp_ic = lft(sys_ic,K);
omega = logspace(-2,2,100);
```

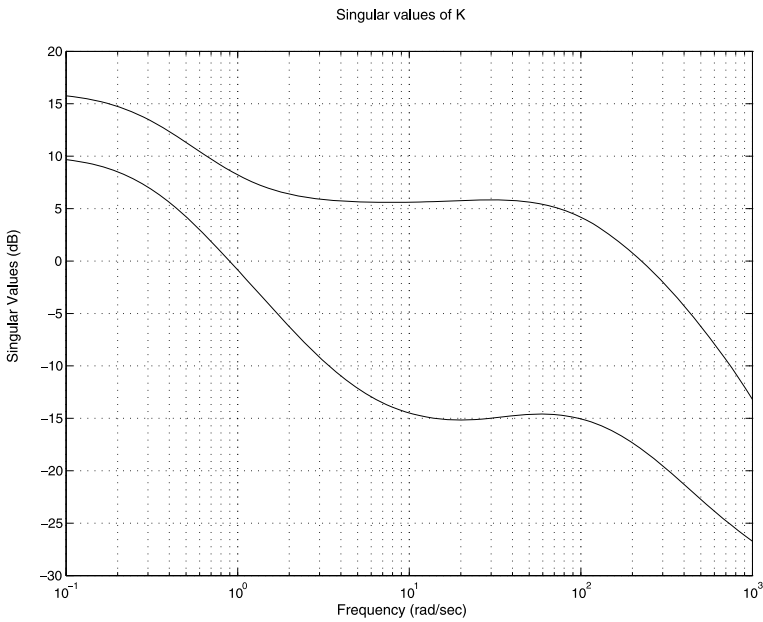


Fig. 12.5 Controller singular values

```

clp_g = ufrd(clp_ic,omega);
%
% Robust stability
opt = robopt('Display','on');
[stabmarg,destabu,report,info] = robuststab(clp_g,opt);
report
%
% Robust performance
opt = robopt('Display','on');
[perfmarg,perfmargunc,report,info] = robustperf(clp_g,opt);
report

```

As is seen from Fig. 12.6, the closed-loop system remains stable even for uncertainties that are 4 times greater than the plant.

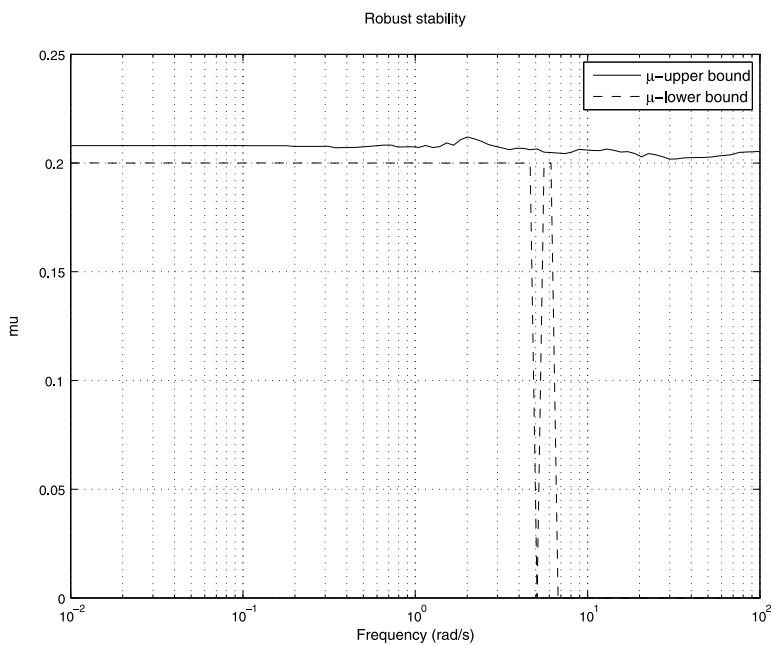
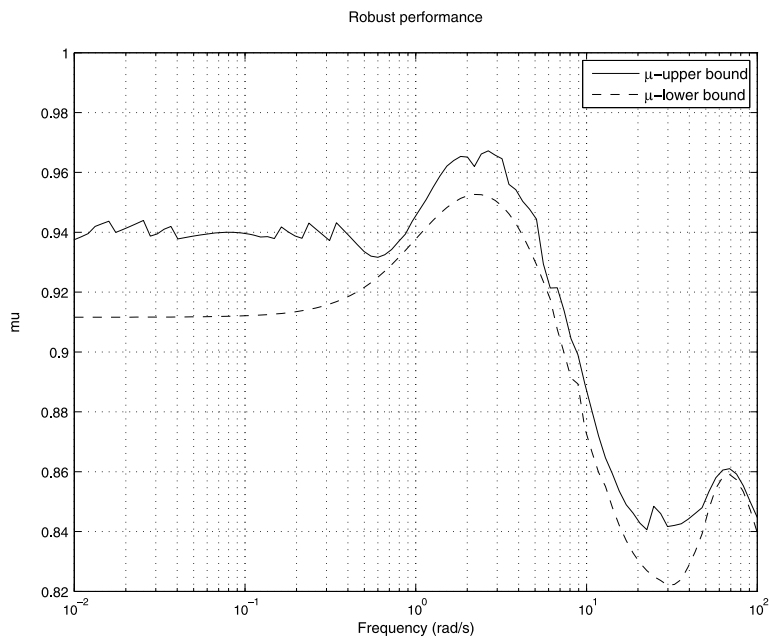
The frequency response of the structured singular value in case of robust performance is shown in Fig. 12.7.

The frequency responses of the closed-loop system for random perturbations of the uncertain parameters are shown in Fig. 12.8.

Since the closed-loop system achieves robust performance, the condition

$$\|W_p S_o\|_{\infty} < 1$$

is satisfied for each uncertainty and the singular values of the output sensitivity S_o lie below the singular values of W_p^{-1} (Fig. 12.9). Similarly, the singular values of

**Fig. 12.6** Closed-loop robust stability**Fig. 12.7** Closed-loop robust performance

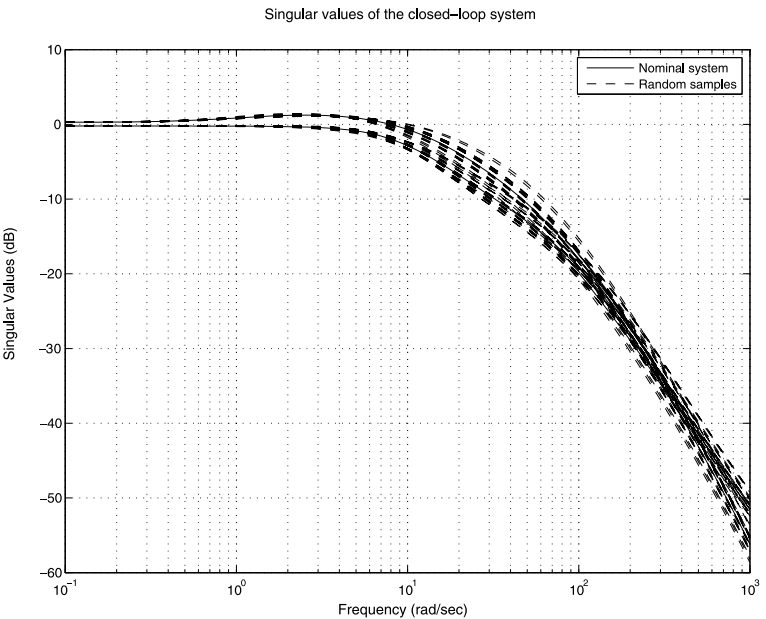


Fig. 12.8 Frequency responses of the closed-loop system

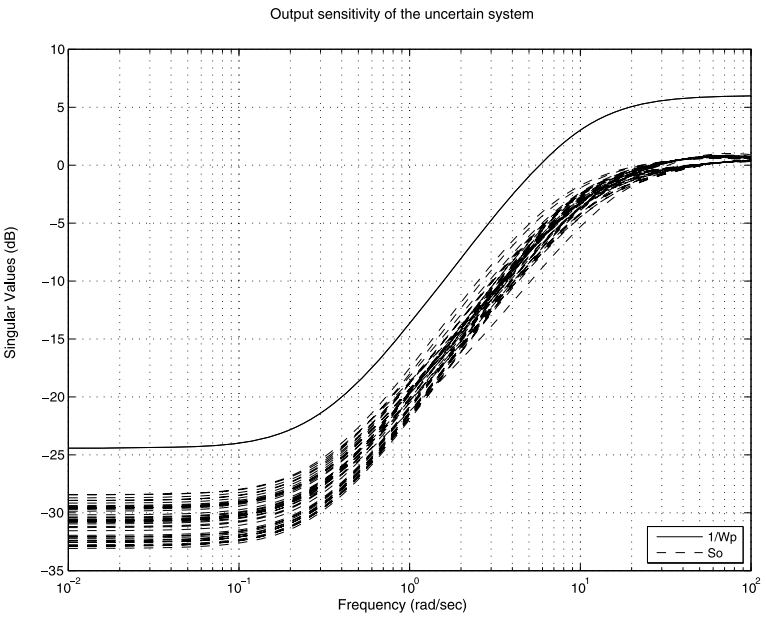


Fig. 12.9 Sensitivity functions of the closed-loop system

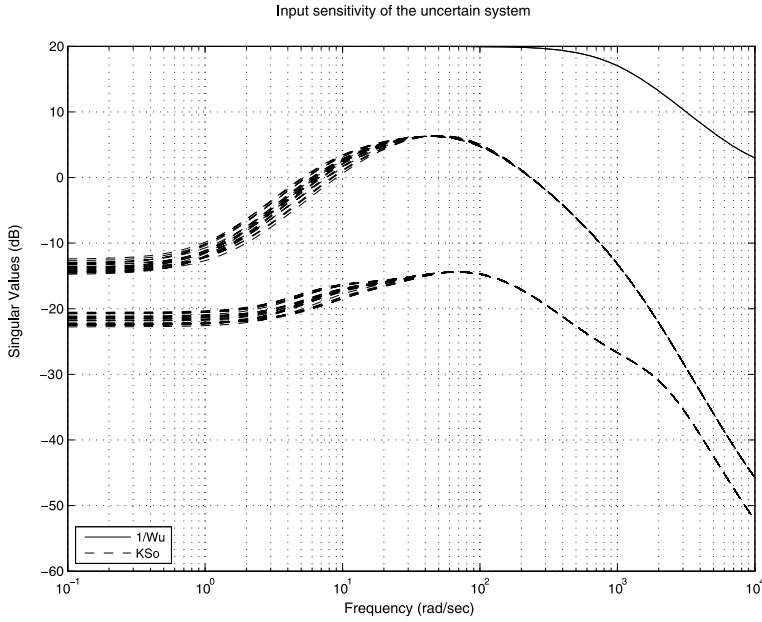


Fig. 12.10 Input sensitivity of the closed-loop system

the transfer matrix $S_i K = K S_o$ lie below the singular values of W_u^{-1} (Fig. 12.10), i.e., the constraints on the control action are respected.

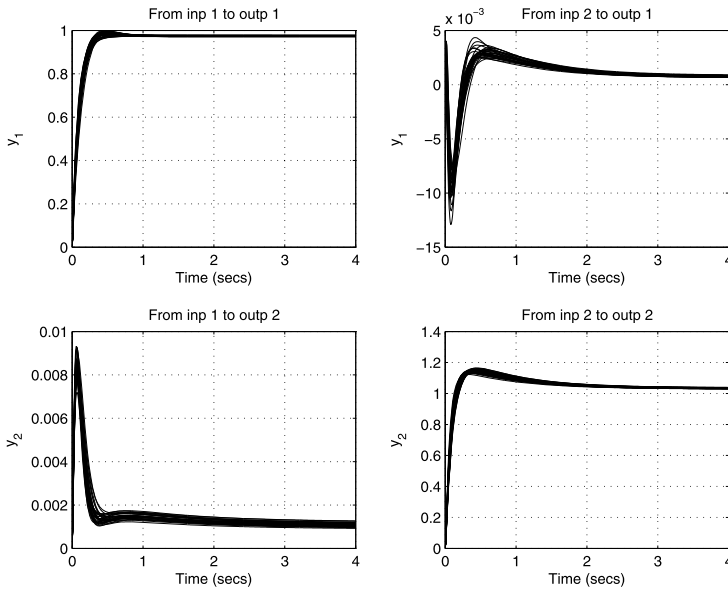
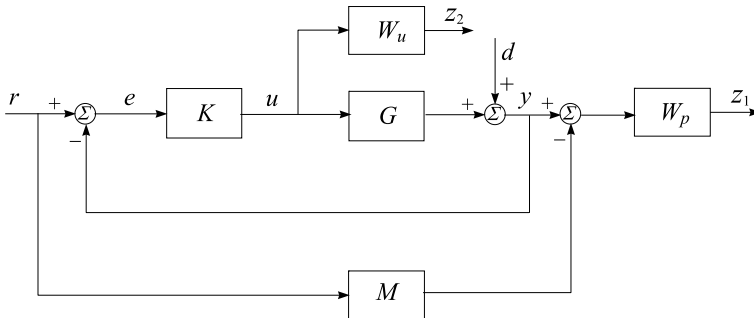
The transient responses of the closed-loop system due to the reference for 30 random values of the uncertain parameters are shown in Fig. 12.11.

12.3 Versions of μ -Synthesis

12.3.1 μ -Synthesis with Model

The μ -synthesis with a model corresponds to the block-diagram of the closed-loop system shown in Fig. 12.12. This diagram is the same as the diagram shown in Fig. 11.15, but in this case instead of \mathcal{H}_∞ norm of the closed-loop system one minimizes the structured singular value μ . The usage of μ -synthesis allows to suppress the effect of the disturbances on the system output and to achieve closeness of the closed-loop system behavior to the model for all possible plant uncertainties. The open-loop and closed-loop transfer function matrices are the same as those shown in Sect. 11.3, using instead of the plant nominal model G_{nom} the uncertain plant G .

Example 12.2 Consider the μ -synthesis of the mass–damper–spring system whose \mathcal{H}_∞ design was presented in Example 11.3. For this aim we use the same model and weighting functions as given in Example 11.3:

**Fig. 12.11** Transient responses of the closed-loop system**Fig. 12.12** Closed-loop system with model

$$M = \frac{1}{T^2 s^2 + 2\xi T s + 1}, \quad T = 1, \xi = 0.7$$

$$W_p(s) = 0.5 \frac{2s + 1}{2s + 0.001}$$

$$W_u(s) = 0.5 \frac{0.05s + 1}{0.0001s + 0.01}$$

In the given case, after three iterations one obtains a controller of 20th order, for which the maximum value of μ decreases to 1.116, i.e., the performance margin is 0.899. Despite the fact that the robust performance is not achieved the closed-loop system is robustly stable with the maximum of the structured singular value

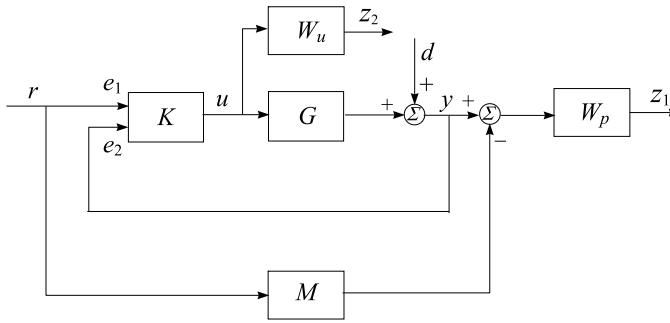


Fig. 12.13 Closed-loop system with 2-degree-of-freedom controller

being equal to 0.578. In this way the μ -synthesis produces better results compared to \mathcal{H}_∞ design in which the controller does not ensure even robust stability of the closed-loop system. When the μ -synthesis does not lead to robust performance, as in the case under consideration, it is necessary either to change the performance requirements to the closed-loop system (i.e., the weighting functions) or to try 2-degree-of-freedom controller design.

12.3.2 μ -Synthesis of 2-Degree-of-Freedom Controller

The block-diagram of the closed-loop system in the given case is shown in Fig. 12.13 and is the same as the block-diagram for \mathcal{H}_∞ design, shown in Fig. 11.24. The usage of μ -synthesis allows to achieve better robustness of the closed-loop system improving usually the nominal performance as well. The implementation of 2-degree-of-freedom controller usually lead to better design results in comparison with the usage of 1-degree-of-freedom controller.

Example 12.3 Consider again the mass–damper–spring system for which an 1-degree-of-freedom controller was design in the previous example. For the same system, in Example 11.4 an \mathcal{H}_∞ design of 2-degree-of-freedom controller was obtained but the robust stability and robust performance of the closed-loop system were not achieved. In the given case, by aid of the μ -synthesis, an 18th order controller is obtained for which the maximum value of μ is decreased to 0.984 thus achieving closed-loop robust stability and robust performance.

The controller frequency responses are shown in Fig. 12.14.

The frequency response of μ in the case of robust stability analysis is shown in Fig. 12.15 and in the case of robust performance analysis in Fig. 12.16. Obviously, the system achieves both robust stability and robust performance.

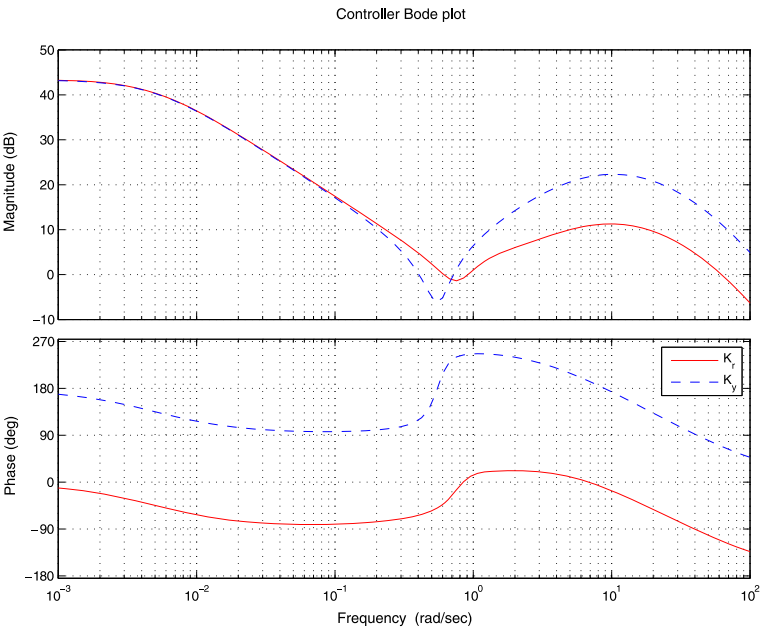


Fig. 12.14 Frequency responses of the controller

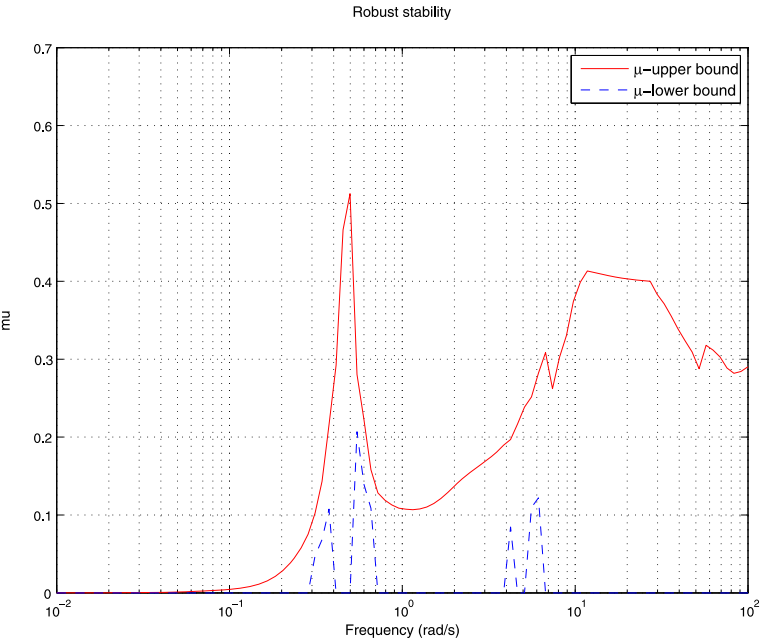


Fig. 12.15 Robust stability for 2-degree-of-freedom controller

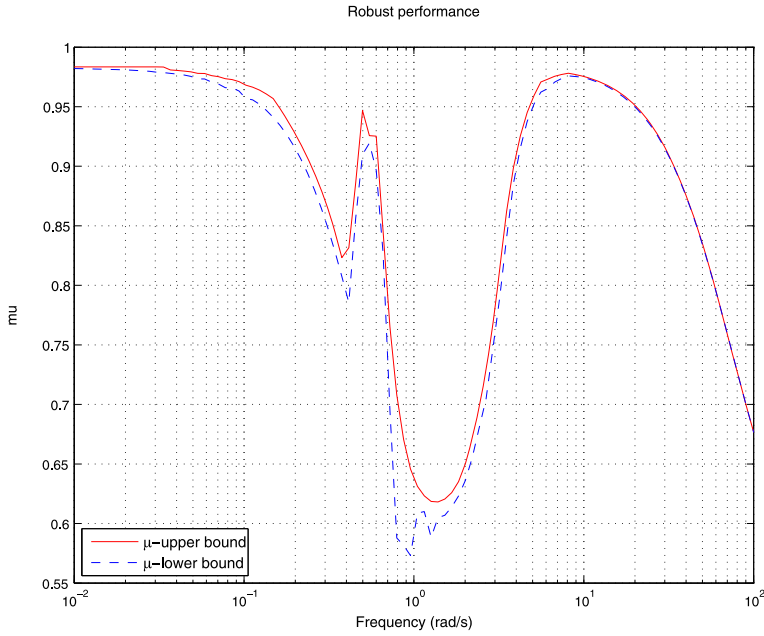


Fig. 12.16 Robust performance for 2-degree-of-freedom controller

The worst-case gain analysis of the closed-loop system is done by the command line

```
[maxgain,maxgainunc] = wcgain(To);
```

where T_o is the closed-loop transfer function.

The magnitude response of the worst-case gain is shown in Fig. 12.17.

Transient responses to the reference of the uncertain closed-loop system are shown in Fig. 12.18 and the corresponding control actions in Fig. 12.19. The transient responses are characterized by satisfactory performance for all possible plant uncertainties.

The results of \mathcal{H}_∞ design and μ -synthesis of the mass–damper–spring system obtained in Examples 11.3, 11.4, 12.2, and 12.3 show that the best closed-loop system robustness is achieved for the controller in the form of 2-degrees-of-freedom and is designed by the aid of μ -synthesis.

12.4 Practical Aspects of μ -Analysis and μ -Synthesis

In this section we give some recommendations about how to use in practice the structured singular value μ .

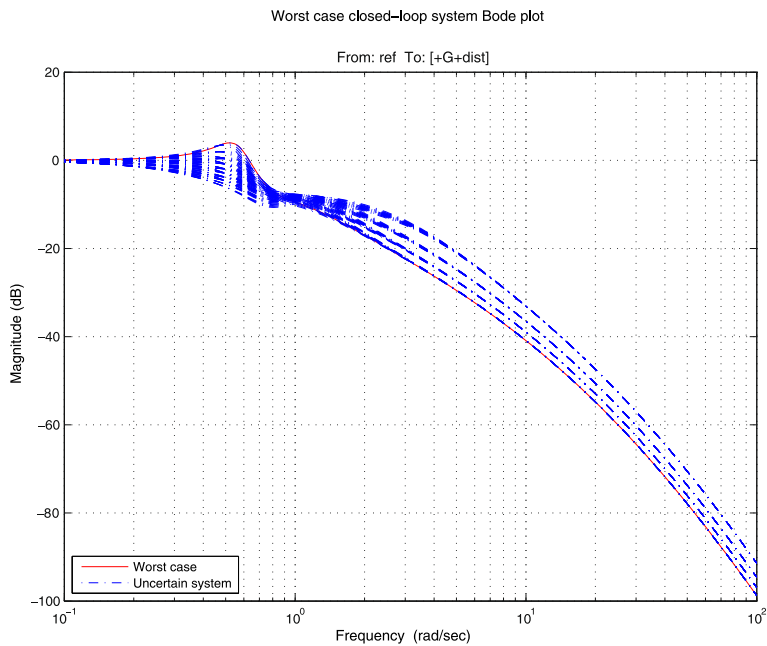


Fig. 12.17 Worst-case gain magnitude frequency response

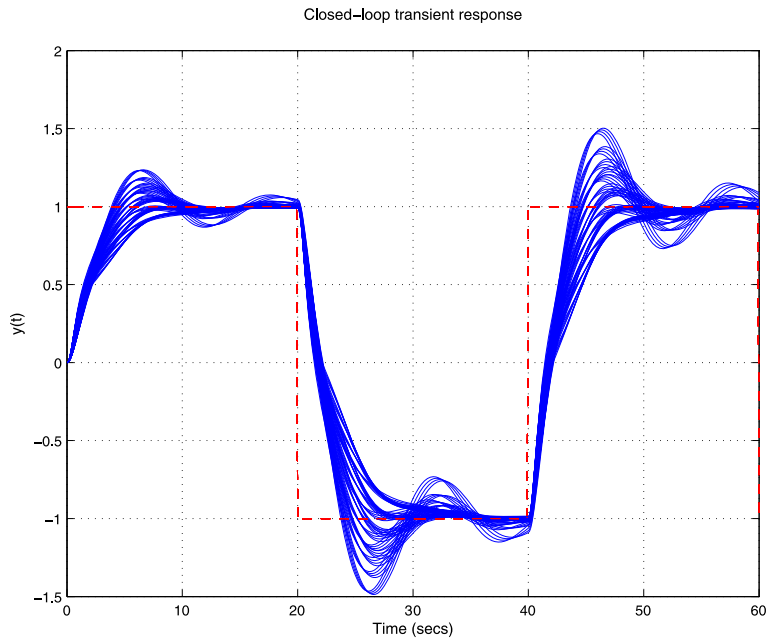


Fig. 12.18 Transient responses of the closed-loop system

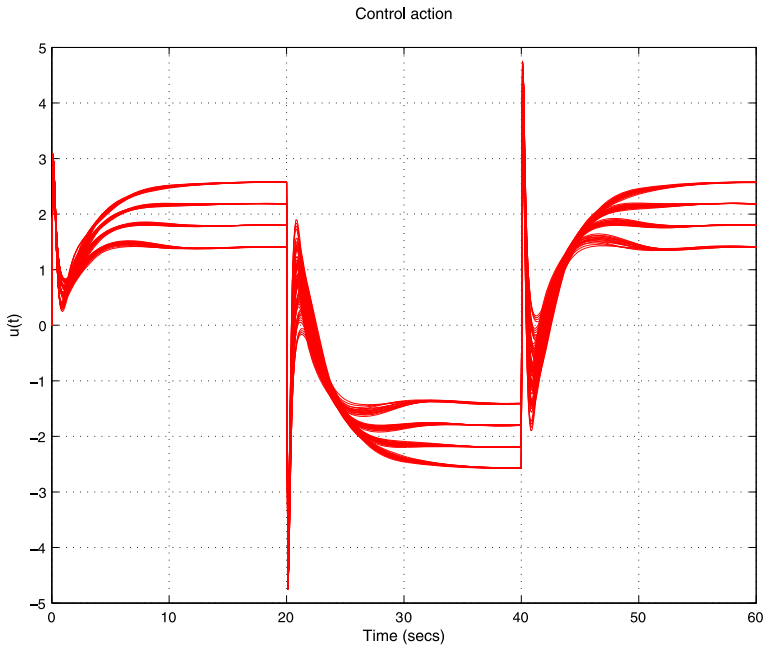


Fig. 12.19 Closed-loop control actions

1. Due to the efforts necessary to derive the uncertain plant model and the almost unavoidable complication of the controller designed it is appropriate to begin with simplified uncertainty description in order to see whether the performance requirements can be met. Only in the case when these requirements are satisfied, it is appropriate to consider more complicated uncertainty descriptions including, for instance, parametric uncertainties to “sharpen” the design with more accurate uncertainty model.
2. The usage of μ means worst-case analysis, so one should be cautious when introducing many sources of uncertainties, disturbances and noises. In such a case it becomes less and less possible for the worst case to appear and the analysis and design performed may become unnecessarily conservative.
3. There is always uncertainty with respect to inputs and outputs so that it is reasonable in the general case to include diagonal input and output uncertainties. The relative (multiplicative) uncertainty is very appropriate for this aim.
4. In the practical design it is customary to obtain values of μ that exceed 1. This may result from very high requirements to the closed-loop performance which are impossible to satisfy for the given plant. In such a case it is necessary to loosen the requirements setting other performance and/or control action weighting functions. Finding a controller that ensures value of μ less than one means that the requirements set are possible to achieve. In other cases it may be necessary to design controller with different structure, for instance 2-degree-of-freedom controller, in order to fulfill the requirements posed.

5. In case of a discrete-time system it is appropriate to perform the controller design initially in the continuous-time case. This is due to the fact that the best possible performance may be obtained in the continuous-time case which can then be considered as a limit for discrete-time designs. Also, in the continuous-time case it is easier to find appropriate performance weighting functions that again may be implemented in the discrete-time design.

12.5 Exercises

Exercise 12.1 For the system from Example 12.1 do the following.

- (a) Design a μ -controller minimizing the performance index

$$\left\| \begin{bmatrix} W_p S G \\ W_u K S G \end{bmatrix} \right\|_{\infty}$$

where

$$W_p = \begin{bmatrix} w_p & 0 \\ 0 & w_p \end{bmatrix}, \quad W_u = \begin{bmatrix} w_u & 0 \\ 0 & w_u \end{bmatrix}$$

and

$$w_p = \frac{0.95(s^2 + 2000s + 4000)}{s^2 + 1900s + 10}, \quad w_u = \frac{10^{-6}(0.1s + 1)}{0.001s + 1}$$

- (b) Analyze the robust performance of the closed-loop system.
 (c) Obtain the singular value plots of the sensitivity function and complementary sensitivity function of the uncertain closed-loop system.
 (d) Obtain the transient response of the uncertain closed-loop system.

Exercise 12.2 For the same plant and weighting functions as in Exercise 12.1 do the following.

- (a) Design a 2-degree-of-freedom μ -controller.
 (b) Analyze the robust performance of the closed-loop system.
 (c) Obtain the singular value plots of the sensitivity function and complementary sensitivity function of the uncertain closed-loop system.
 (d) Obtain the transient response of the uncertain closed-loop system.

Compare the results with the results from Exercise 12.1.

This page intentionally left blank

Chapter 13

Analysis and Design of Parameter-Dependent Systems

The structured uncertainty models presented in Chap. 9 are not the unique way to represent systems with uncertain parameters. These models are usually appropriate to describe systems whose uncertainties are relatively small in size and may not be convenient to implement in the case of large parameter variations. In such cases it might be better to use models in the form of the so called *parameter-dependent systems* which are appropriate for use in the more general case of large uncertainties. The linear parameter-dependent systems, called also *Linear Parameter-Varying Systems* (LPV systems), arise in many applications, for instance in robotics and aerospace systems. In this chapter we show how to build models of parameter-dependent systems and how to use these models in robust stability analysis and controller design. Especially important is the so called *gain scheduling design* which allows to obtain satisfactory closed-loop performance in case of widely varying plant parameters.

The presentation in this chapter follows closely the corresponding chapters of [50].

13.1 Representation of Parameter-Dependent Systems

13.1.1 SYSTEM Matrix

In many cases the linear system models are obtained in the general form of the so called *descriptor systems* which are described by state-space equations of the form

$$E \frac{dx}{dt} = Ax(t) + Bu(t) \quad (13.1)$$

$$y(t) = Cx(t) + Du(t) \quad (13.2)$$

In these equations the matrix E is a square matrix that may be singular in the general case. If this matrix is invertible and well conditioned, (13.1), (13.2) are easily converted to the usual state-space description multiplying on the left both sides

of (13.1) by the matrix E^{-1} . However, if the matrix E is singular or poorly conditioned this conversion is not possible and one has to use the descriptor system model (13.1), (13.2). The descriptor system models arise naturally in the description of many physical systems and are especially appropriate to represent systems with large parameter variations.

In the discrete-time case the descriptor systems are described by the state-space equations

$$\begin{aligned} E x_{k+1} &= A x_k + B u_k \\ y_k &= C x_k + D u_k \end{aligned}$$

Consider as an example the mass–damper–spring system introduced in Sect. 9.2 and depicted in Fig. 9.11. The dynamics of this system is described by the 2nd-order differential equation

$$m\ddot{x}_s + c\dot{x}_s + kx_s = u \quad (13.3)$$

where x_s is the displacement of the mass block from the equilibrium position and u is the force acting on the mass, with m the mass, c the damper constant and k the spring constant. Denoting $x_1 = x_s$, $x_2 = dx_s/dt$, the system (13.3) may be described by the two first order differential equations

$$\dot{x}_1 = x_2 \quad (13.4)$$

$$m\dot{x}_2 = -kx_1 - cx_2 + u \quad (13.5)$$

Taking the output as $y = x_s$, (13.4), (13.5) are represented as a descriptor system in the form (13.1), (13.2) with matrices

$$E = \begin{bmatrix} 1 & 0 \\ 0 & m \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ -k & -c \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [1 \quad 0], \quad D = 0$$

In MATLAB[®], a continuous-time or discrete-time descriptor system model is stored for convenience as a single matrix called SYSTEM matrix. This matrix has the form

$$\left[\begin{array}{cc|c} A + j(E - I) & B & n \\ & & 0 \\ & C & \vdots \\ & D & 0 \\ \hline 0 & 0 & -\text{Inf} \end{array} \right]$$

where $j = \sqrt{-1}$, n is equal to the number of states of (13.1) and the entry Inf is used to distinguish SYSTEM matrices from the regular matrices.

The SYSTEM matrices are created by the function `ltisys`. For given matrices A, B, C, D, E of the descriptor system formulation (13.1), (13.2), the command line

```
sys = ltisys(A,B,C,D,E)
```

produces the SYSTEM matrix `sys`. When omitted, D and E are set to the default values $D = 0$ and $E = I$. The autonomous system $E\dot{x} = Ax$ is thus specified by the line

```
sys = ltisys(A,E)
```

Given a SYSTEM matrix `sys` created with `ltisys`, the function `ltiss` allows to extract the corresponding state-space realization. This is done by the command line

```
(A,B,C,D,E) = ltiss(sys)
```

It is important to note that when the output argument E is not referenced, `ltiss` returns the state-space realization $(E^{-1}A, E^{-1}B, C, D)$ provided the matrix E is nonsingular.

The number of states, inputs, and outputs of a linear system are extracted from the SYSTEM matrix `sys` with the function `sinfo`,

```
sinfo(sys)
```

The time and frequency response plots of descriptor systems represented by SYSTEM matrices may be obtained by the function `splot`.

Series and parallel interconnections of SYSTEM matrices are performed by the functions `sadd` and `smult` and feedback interconnection can be done by the function `sloop`.

13.1.2 Affine Parameter-Dependent Models

Many physical systems may be described by linearized equations in descriptor form

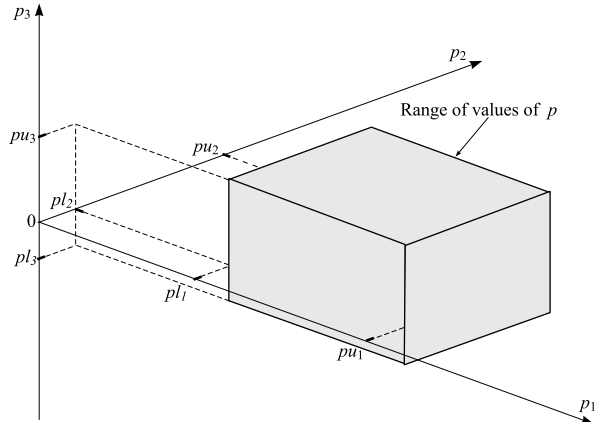
$$E(p)\dot{x} = A(p)x + B(p)u \quad (13.6)$$

$$y = C(p)x + D(p)u \quad (13.7)$$

where $A(\cdot), \dots, E(\cdot)$ are known functions of some parameter vector $p = (p_1, \dots, p_s)$. The parameters p_1, \dots, p_s may vary slowly with the time within some prescribed boundaries. In the case of *affine parameter-dependent systems* the matrices of the state-space description are affine functions of the parameters, which means that these matrices may be represented as

$$\begin{aligned} A(p) &= A_0 + p_1 A_1 + \dots + p_s A_s, & B(p) &= B_0 + p_1 B_1 + \dots + p_s B_s \\ C(p) &= C_0 + p_1 C_1 + \dots + p_s C_s, & D(p) &= D_0 + p_1 D_1 + \dots + p_s D_s \\ E(p) &= E_0 + p_1 E_1 + \dots + p_s E_s \end{aligned}$$

where $A_0, \dots, A_s, B_0, \dots, B_s$ and so on are constant matrices that do not depend on the parameters p_1, \dots, p_s . Note that for a particular system some of these matrices may be zero matrices.

Fig. 13.1 Parameter box

At first glance, the affine parameter-dependent systems constitute a very small set of systems with varying parameters. However, many real world systems may be approximated with sufficient accuracy as affine parameter-dependent systems. The advantage of such a system model is the possibility to derive easily several results concerning closed-loop stability and controller design.

Using the notation

$$S(p) = \begin{bmatrix} A(p) + jE(p) & B(p) \\ C(p) & D(p) \end{bmatrix}, \quad S_i = \begin{bmatrix} A_i + jE_i & B_i \\ C_i & D_i \end{bmatrix}, \quad i = 0, \dots, s$$

the affine parameter-dependent system (13.6), (13.7) is written compactly in SYSTEM matrix terms as

$$S(p) = S_0 + p_1 S_1 + \dots + p_s S_s \quad (13.8)$$

The constant matrices S_0, \dots, S_s may be considered as system coefficients and may have not physical meaning by themselves.

The parameter uncertainty range can be described as a box in the parameter space. This corresponds to cases where each uncertain or slowly varying parameter p_i ranges between two known bounds pl_i and pu_i ,

$$p_i \in [pl_i, pu_i], \quad i = 1, \dots, s \quad (13.9)$$

If $p = (p_1, \dots, p_s)$ is the vector of all uncertain parameters, (13.9) constitutes a hyperrectangle in the parameter space called the *parameter box*. The parameter box is set by the function `pvec`. For instance, in case of three parameters ($s = 3$), the parameter box is specified by the command lines

```
range = [pl_1 pu_1, pl_2 pu_2, pl_3 pu_3]
p = pvec('box', range)
```

The parameter box for the case of three parameters is shown in Fig. 13.1.

The function `pvec` allows also to specify bounds on the rates of variation \dot{p}_i of the parameter vector components $p_i(t)$. For instance, if the rates of the three parameters $p_1(t)$, $p_2(t)$, $p_3(t)$ vary in the ranges $[vl_1 \ vu_1]$, $[vl_2 \ vu_2]$, $[vl_3 \ vu_3]$, respectively, these rates may be specified by adding a third argument `rate` to the calling list of `pvec` as shown below

```
rate = [vl_1 vu_1, vl_2 vu_2, vl_3 vu_3]
p = pvec('box', range, rate)
```

When the argument `rate` is omitted, all parameters are assumed to be time-invariant.

Provided the coefficient matrices S_0, \dots, S_s are known and the domains where the parameters and their rates vary are specified, the affine parameter-dependent model (13.8) may be obtained by the function `psys`. For instance, the system

$$S(p) = S_0 + p_1 S_1 + p_2 S_2 + p_3 S_3$$

depending on three parameters, is defined by

```
S0 = ltisys(A0,B0,C0,D0,E0)
S1 = ltisys(A1,B1,C1,D1,E1)
S2 = ltisys(A2,B2,C2,D2,E2)
S3 = ltisys(A3,B3,C3,D3,E3)
pds = psys(pv, [S0 S1 S2 S3])
```

where `pv` is the parameter description returned by the function `pvec`.

It is necessary to stress that, by default, the function `ltisys` sets the $E(p)$ matrix to the identity matrix I . That is why if the arguments `E0`, `E1`, `E2`, `E3` in the above lines are omitted then the matrix $E(p)$ will be set to $E(p) = I + (p_1 + p_2 + p_3)I$. To specify an affine parameter-dependent system with a parameter-independent E matrix (e.g., $E(p) = I$), it is necessary to explicitly set $E_1 = E_2 = E_3 = 0$ by entering

```
S0 = ltisys(A0,B0,C0,D0)
S1 = ltisys(A1,B1,C1,D1,0)
S2 = ltisys(A2,B2,C2,D2,0)
S3 = ltisys(A3,B3,C3,D3,0)
```

Example 13.1 Consider the mass–damper–spring system whose state-space equations (13.4), (13.5) were obtained earlier in this section in descriptor form. The three uncertain parameters m, k, c of the system vary in the intervals

$$m \in [1.8, 4.2], \quad k \in [1.4, 2.6], \quad c \in [0.8, 1.2]$$

The parameter-dependent matrices of the mass–damper–spring system may be represented as affine functions of the uncertain parameters m, k, c as

$$A(m, k, c) = \begin{bmatrix} 0 & 1 \\ -k & -c \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + m \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + k \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} + c \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$$

$$E(m, k, c) = \begin{bmatrix} 1 & 0 \\ 0 & m \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + m \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} + k \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + c \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The affine model of the mass–damper–spring system is specified by the following command lines:

```
A_0 = [0 1;0 0]; B_0 = [0 1]'; C_0 = [1 0]; D_0 = 0;
E_0 = [1 0;0 0];
S_0 = ltisys(A_0,B_0,C_0,D_0,E_0);
%
A_m = [0 0;0 0]; B_m = [0 0]'; C_m = [0 0]; D_m = 0;
E_m = [0 0;0 1];
S_m = ltisys(A_m,B_m,C_m,D_m,E_m);
%
A_k = [0 0;-1 0]; B_k = [0 0]'; C_k = [0 0]; D_k = 0;
E_k = [0 0;0 0];
S_k = ltisys(A_k,B_k,C_k,D_k,0);
%
A_c = [0 0;0 -1]; B_c = [0 0]'; C_c = [0 0]; D_c = 0;
E_c = [0 0;0 0];
S_c = ltisys(A_c,B_c,C_c,D_c,0);
%
pv = pvec('box',[1.8 4.2;1.4 2.6;0.8 1.2]);
pds_mds = psys(pv,[S_0 S_m S_k S_c]);
```

The first SYSTEM matrix S_0 contains the state-space data for $m = k = c = 0$ while S_m , S_k , S_c define the coefficient matrices of m, k, c . The range of parameter values in the parameter box is specified by the `pvec` command. The results obtained can be checked with the commands `psinfo` and `pvinfos`:

```
psinfo(pds_mds)
```

```
Affine parameter-dependent model with 3 parameters
(4 systems)
Each system has 2 state(s), 1 input(s), and
1 output(s)
```

```
pvinfos(pv_mds)
```

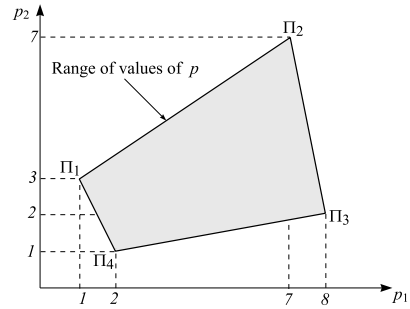
```
Vector of 3 parameters ranging in a box
```

The command line

```
nom_sys = psinfo(pds_mds,'eval',[3.0 2.0 1.0])
```

allows to evaluate the parameter-dependent model for the nominal values $m = 3.0, k = 2.0, c = 1.0$.

Fig. 13.2 Polytopic parameter range



Apart from the possibility to describe the parameter uncertainty range as a box, uncertain parameter vectors can be specified as ranging in a polytope of the parameter space. This possibility is illustrated in Fig. 13.2 for the simple case of $s = 2$. The polytope is characterized by the four vertices

$$\Pi_1 = (1, 3), \quad \Pi_2 = (7, 7), \quad \Pi_3 = (8, 2), \quad \Pi_4 = (2, 1)$$

An uncertain parameter vector p with this range of values is defined by

```
pi1=[1,3], pi2=[7,7], pi3=[8,2], pi4=[2,1]
p = pvec('pol',[pi1,pi2,pi3,pi4])
```

The string 'pol' indicates that the parameter range is defined as a polytope.

13.1.3 Polytopic Models

Polytopic system is a linear time-varying system

$$\begin{aligned} E(t)\dot{x} &= A(t)x + B(t)u \\ y &= C(t)x + D(t)u \end{aligned}$$

whose SYSTEM matrix

$$S(t) = \begin{bmatrix} A(t) + jE(t) & B(t) \\ C(t) & D(t) \end{bmatrix}$$

varies within a fixed polytope of matrices

$$P = \left\{ \sum_{i=1}^q \alpha_i S_i : \alpha_i \geq 0, \sum_{i=1}^q \alpha_i = 1 \right\}$$

In the above expression S_1, S_2, \dots, S_q are given *vertex systems*

$$S_i = \begin{bmatrix} A_i + jE_i & B_i \\ C_i & D_i \end{bmatrix}, \quad i = 1, \dots, q$$

corresponding to the vertices of $S(t)$.

In this way the polytopic system $S(t)$ is a convex combination (linear combination with nonnegative coefficients) of the SYSTEM matrices S_1, \dots, S_q , i.e. $S(t) = \alpha_1 S_1 + \dots + \alpha_q S_q$. The nonnegative numbers $\alpha_1, \dots, \alpha_q$ are called the *polytopic coordinates* of $S(t)$.

Polytopic models are fully specified by the list of their vertex systems, i.e., by the SYSTEM matrices S_1, \dots, S_q and may be created by the function `psys`. For instance, a polytopic model taking values in the convex combination of the three linear time-invariant systems `S1`, `S2`, `S3` is created by the line

```
polysys = psys([S1 S2 S3])
```

Affine parameter-dependent systems

$$S(p) = \begin{bmatrix} A(p) + jE(p) & B(p) \\ C(p) & D(p) \end{bmatrix}$$

may be converted easily to polytopic systems. If each parameter p_i varies in some interval $[pl_i, pu_i]$, $i = 1, \dots, s$, then the parameter vector $p = (p_1, \dots, p_s)$ takes values in a parameter box with $q = 2^s$ corners $\Pi_1, \Pi_2, \dots, \Pi_q$. According to the results of functional analysis, if the function $S(p)$ is affine in p , then it maps this parameter box to some polytope of SYSTEM matrices. Specifically, this polytope is the convex linear combination of the images $S_1 = S(\Pi_1), S_2 = S(\Pi_2), \dots, S_q = S(\Pi_q)$ of the parameter box corners $\Pi_1, \Pi_2, \dots, \Pi_q$. These images are the vertex systems of the polytopic model.

An affine parameter-dependent system is converted to an equivalent polytopic model using the function `aff2pol`. The syntax is of this function is

```
polysys = aff2pol(affsys)
```

where `affsys` is the affine model and `polysys` is the resulting polytopic model. The polytopic model consists of the instances of `affsys` at the vertices of the parameter box.

For illustration, if the function `aff2pol` is applied to the mass–damper–spring system, considered in Example 13.1, one obtains

```
polys_mds = aff2pol(pds_mds);  
psinfo(polys_mds)
```

Polytopic model with 8 vertex systems

```
Each system has 2 state(s), 1 input(s), and  
1 output(s)
```

In the given case the affine model depends on three parameters, which results in $q = 2^3 = 8$ vertex systems of the polytopic model.

13.2 Analysis of Parameter-Dependent Systems

In this section we consider the implementation of MATLAB[®] functions for stability analysis of parameter-dependent systems and evaluation of the transient response of such systems.

Stability analysis of parameter-dependent systems is done by the second method of Lyapunov. According to this method a linear time-varying system

$$E(t)\dot{x} = A(t)x \quad (13.10)$$

is asymptotically stable if there exists a positive definite quadratic Lyapunov function

$$V(x) = x^T P x$$

such that its derivative with respect to time is negative definite, i.e.

$$\frac{dV(x(t))}{dt} < 0$$

along all state trajectories. If it is possible to find such a Lyapunov function we say that the system (13.10) is *quadratically stable*. The existence of quadratic Lyapunov function guarantees stability for arbitrarily fast time variations of the system parameters which is a very stringent condition in practice.

The quadratic stability of affine or polytopic parameter-dependent system may be tested by the MATLAB[®] function `quadstab` which involves solution of several Linear Matrix Inequalities (LMI) in an attempt to find a suitable Lyapunov function $V(x)$. However, `quadstab` may produce pessimistic results due to the allowance of arbitrarily fast parameter variations. Less conservative results in case of constant or slowly varying parameters may be obtained by the function `pdlstab` which is considered next.

For a parameter-dependent system the function `pdlstab` seeks a parameter-dependent Lyapunov function to establish the stability of uncertain state-space models over some parameter range or polytope of systems. Specifically, for an affine parameter-dependent system

$$E(p)\dot{x} = A(p)x + B(p)u$$

$$y = C(p)x + D(p)u$$

with $p = (p_1, p_2, \dots, p_s)$ `pdlstab` seeks a Lyapunov function of the form

$$V(x, p) = x^T Q^{-1} x, \quad Q(p) = Q_0 + p_1 Q_1 + \dots + p_s Q_s$$

with symmetric matrices Q_0, Q_1, \dots, Q_s such that $dV(x, p)/dt < 0$ along all admissible trajectories.

For a time-invariant polytopic system

$$E\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

with

$$\begin{bmatrix} A + jE & B \\ C & D \end{bmatrix} = \sum_{i=1}^q \alpha_i \begin{bmatrix} A_i + jE_i & B_i \\ C_i & D_i \end{bmatrix}, \quad \alpha_i \geq 0, \quad \sum_{i=1}^q \alpha_i = 1 \quad (13.11)$$

`pdlstab` seeks a single Lyapunov function of the form

$$V(x, \alpha) = x^T Q(\alpha)^{-1} x, \quad Q(\alpha) = \alpha_1 Q_1 + \dots + \alpha_q Q_q$$

such that $dV(x, \alpha)/dt < 0$ for all polytopic decompositions (13.11). Note that for this type of models one of the matrices A, E should be constant and the other uncertain.

Similarly to `quadstab`, the function `pdlstab` determines whether the sufficient LMI stability conditions are feasible.

The syntax of the function `pdlstab` is

```
[tau, Q0, Q1, ..., Qs] = pdlstab(affsys, options)
```

for the case of affine parameter-dependent systems and

```
[tau, Q1, Q2, ..., Qq] = pdlstab(polysys, options)
```

for the case of polytopic parameter-dependent systems.

Input arguments

- `affsys` is the affine system description specified with the function `psys`, which contains information about the range of values and rate of variation of each parameter p_i ;
- `polysys` is the polytopic system description specified with the function `psys` or obtained by the function `aff2pol` from an affine model;
- `options`: optional parameter.
 - Setting `options(1) = 0` tests robust stability (default).
 - When `options(2) = 0`, `pdlstab` uses simplified sufficient conditions for faster execution. Set `options(2) = 1` to use the least conservative conditions.

Output arguments

- `tau`: a scalar parameter which shows whether the LMI stability conditions are feasible. If `tau` is negative then the system is stable;
- `Q0, Q1, ..., Qs`: contain the coefficient matrices Q_0, Q_1, \dots, Q_s of the matrix $Q(p)$ associated with the Lyapunov function $V(x, p)$ in case of affine model;
- `Q1, Q2, ..., Qq`: contain the coefficient matrices Q_1, Q_2, \dots, Q_q of the matrix $Q(\alpha)$ associated with the Lyapunov function $V(x, \alpha)$ in case of polytopic model.

Example 13.2 Consider the mass–damper–spring system whose affine parameter-dependent model `pds_mds` was created in Example 13.1. For this model the command line

```
[tmin,Q0,Q1,Q2,Q3] = pdlstab(pds_mds)
```

produces

```
Solver for LMI feasibility problems  $L(x) < R(x)$ 
This solver minimizes  $t$  subject to  $L(x) < R(x) + t \cdot I$ 
The best value of  $t$  should be negative
for feasibility
```

```
Iteration      :      Best value of  $t$  so far
```

```
1              -0.024610
```

```
Result:  best value of  $t$ :      -0.024610
         f-radius saturation:  0.000% of  $R = 1.00e+007$ 
```

```
This system is stable for the specified parameter
                                trajectories
```

```
tmin =
```

```
-0.0246
```

```
Q0 =
```

```
1.6750    -0.3994
-0.3994    1.1642
```

```
Q1 =
```

```
0.8456    -0.0451
-0.0451    -0.0271
```

```
Q2 =
```

```
0.0543    -0.0989
-0.0989    0.9498
```

```
Q3 =
```

```
0.6545    -0.7452
-0.7452    -0.0235
```

The results shows that the mass–damper–spring system remains stable for all parameter values that belong to the specified parameter box. Note that although the matrices Q_1 and Q_3 are not positive definite, the matrix $Q(p) = Q_0 + mQ_1 + kQ_2 + cQ_3$ remains positive definite for the whole parameter range.

The time response of an affine parameter-dependent system

$$\begin{aligned} E(p)\dot{x} &= A(p)x + B(p)u \\ y &= C(p)x + D(p)u \end{aligned}$$

along a given parameter trajectory $p(t)$ and for an input signal $u(t)$ is simulated by the function `pdsimul`. The parameter vector $p(t)$ is required to range in a box (type 'box' of the function `pvec`). The function `pdsimul` also accepts the polytopic representation of affine parameter-dependent systems as returned by `aff2pol(pds)`.

The syntax of the function `pdsimul` is

```
pdsimul(pds, 'traj', tf, 'ut', xi, options)
```

or

```
[t, x, y] = pdsimul(pds, pv, 'traj', tf, 'ut', xi, options)
```

When invoked without output arguments, `pdsimul` plots the output response $y(t)$ of the affine parameter-dependent system `pds`. Otherwise, it returns the vector of integration time points t as well as the state and output responses x , y . The parameter trajectory and input signals are specified by two time functions `p = traj(t)` and `u = ut(t)`. If 'ut' is omitted, the response to a step input is computed by default. The final time and initial state vector can be reset through `tf` and `xi` (their respective default values are 5 seconds and zero vector). Finally, the input argument `options` gives access to the parameters controlling the integration of the system differential equations.

Example 13.3 Consider the simulation of the step response of mass–damper–spring system for a period of 30 seconds. The parameter trajectory is chosen as

$$\begin{aligned} m(t) &= 3 + 1.2e^{-0.3t} \cos(10t) \\ k(t) &= 2 + 0.6e^{-0.3t} \sin(10t) \\ c(t) &= 0.8 + 0.01t \end{aligned}$$

and is specified in the file `traj_mds.m`.

```
function p = traj_mds(t)
%
p = [3.0 + 1.2*exp(-0.3*t)*cos(10*t); ...
     2.0 + 0.6*exp(-0.3*t)*sin(10*t); ...
     0.8 + 0.01*t];
```

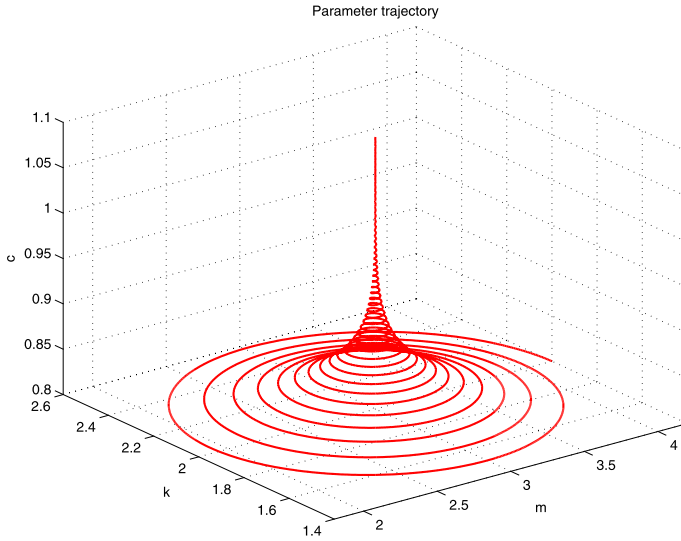


Fig. 13.3 Parameter trajectory for system simulation

The parameter trajectory is shown in Fig. 13.3.

The time response of the mass–damper–spring system for the given parameter trajectory along the time response of the nominal system are obtained by the lines

```
[t,x,y] = pdsimul(pds_mds,'traj_mds',30);
mds_nom = psinfo(pds_mds,'eval',[3 2 1]);
[a,b,c,d] = ltiss(mds_nom); mds_ss = ss(a,b,c,d);
tt = 0:0.01:30;
y_nom = step(mds_ss,tt);
figure(1)
plot(t,y(:,1),'b-',tt,y_nom,'r--'), grid
xlabel('Time (secs)')
ylabel('y')
title('System transient response')
legend('Parameter-dependent system','Nominal system')
```

Both time responses are shown in Fig. 13.4. Comparing with the step responses of the mass–damper–spring system for 20 random values of the uncertain elements, shown in Fig. 9.14, we see that the response of the parameter-dependent system is slightly different from the response of the nominal system. This is explained by the fact that in the latter case the parameters m , k , c correspond to a given trajectory in the parameter space and do not vary independently.

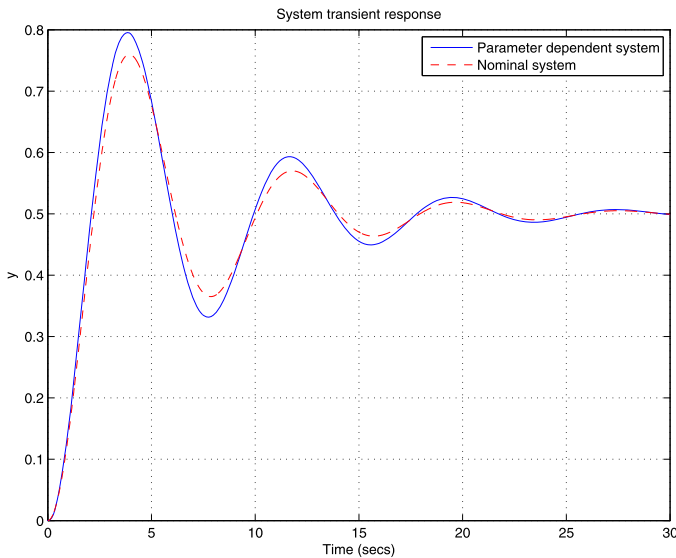


Fig. 13.4 Time response of the parameter-dependent system

13.3 Gain Scheduling Design for Parameter-Dependent Systems

When the system parameters undergo large variations it is often impossible to achieve high performance and even closed-loop stability over the entire operating range with a single time-invariant controller. In this case it is appropriate to apply a technique called *gain scheduling*, which is successfully used in the control of uncertain or time-varying systems. This technique involves implementation of a family of controllers designed for different regions of the parameter space so as to guarantee stability and performance in that region. During the system operation the controllers are changed according to a physical parameter measured in real time which detects in what region the system is working in the corresponding moment of the time. The change of controllers is done either by interpolation of certain parameters or by switching. Such controllers are said to be *scheduled* by the parameter measurements. In this section we consider implementation of gain-scheduled \mathcal{H}_∞ controllers which ensures higher performance in the face of large variations in operating conditions and provides smooth changes between controllers.

The synthesis technique presented in this section is applicable to affine parameter-dependent plants with state-space description

$$P(., p) \begin{cases} \dot{x} = A(p)x + B_1(p)w + B_2u, \\ z = C_1(p)x + D_{11}(p)w + D_{12}u \\ y = C_2x + D_{21}w + D_{22}u \end{cases} \quad (13.12)$$

where

$$p(t) = (p_1(t), \dots, p_s(t)), \quad p_i \in [p_i^l, p_i^u], \quad i = 1, \dots, s$$

is a time-varying vector of physical parameters and $A(\cdot)$, $B_1(\cdot)$, $C_1(\cdot)$, $D_{11}(\cdot)$ are affine functions of $p(t)$. Note that $p(t)$ may include part of the state vector x itself provided that the corresponding states are accessible to measurement. Although this description is not valid in the general case when all matrices are functions of $p(t)$, it may be used for many systems arising in practice.

If the parameter vector $p(t)$ takes values in a box with corners Π_i , $i = 1, \dots, q$ ($q = 2^s$) then the plant SYSTEM matrix

$$S(p) = \left[\begin{array}{c|cc} A(p) & B_1(p) & B_2 \\ \hline C_1(p) & D_{11}(p) & D_{12} \\ C_2 & D_{21} & D_{22} \end{array} \right]$$

ranges in a matrix polytope with vertices $S(\Pi_i)$. Decomposing the parameter vector

$$p(t) = \alpha_1 \Pi_1 + \dots + \alpha_q \Pi_q, \quad \alpha_i \geq 0, \quad \sum_{i=1}^q \alpha_i = 1 \quad (13.13)$$

over the corners Π_i of the parameter box, the system matrix $S(p)$ is represented as

$$S(p) = \alpha_1 S(\Pi_1) + \dots + \alpha_q S(\Pi_q)$$

This representation suggests to seek a parameter-dependent controller

$$K(., p) \begin{cases} \dot{\zeta} = A_K(p)\zeta + B_K(p)y, \\ u = C_K(p)\zeta + D_K(p)y \end{cases}$$

whose SYSTEM matrix has the polytopic representation

$$\begin{bmatrix} A_K(p) & B_K(p) \\ C_K(p) & D_K(p) \end{bmatrix} = \sum_{i=1}^q \alpha_i \begin{bmatrix} A_K(\Pi_i) & B_K(\Pi_i) \\ C_K(\Pi_i) & D_K(\Pi_i) \end{bmatrix} \quad (13.14)$$

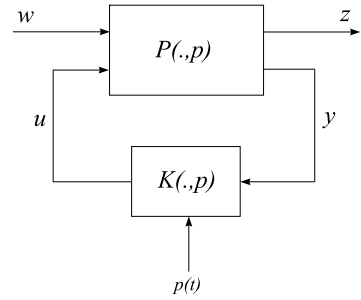
In this way, for a given convex decomposition (13.13) of the current parameter vector $p(t)$, the controller state-space matrices at the operating point $p(t)$ are obtained by convex interpolation of the linear time-invariant *vertex controllers*

$$K_i = \begin{bmatrix} A_K(\Pi_i) & B_K(\Pi_i) \\ C_K(\Pi_i) & D_K(\Pi_i) \end{bmatrix}$$

The values of controller matrices $A_K(p)$, $B_K(p)$, $C_K(p)$, $D_K(p)$ are derived from the values $A_K(\Pi)$, $B_K(\Pi)$, $C_K(\Pi)$, $D_K(\Pi)$ at the corners of the parameter box according to

$$A_K(p) = \alpha_1 A_K(\Pi_1) + \dots + \alpha_q A_K(\Pi_q)$$

Fig. 13.5 Gain-scheduled \mathcal{H}_∞ problem



$$B_K(p) = \alpha_1 B_K(\Pi_1) + \cdots + \alpha_q B_K(\Pi_q)$$

$$C_K(p) = \alpha_1 C_K(\Pi_1) + \cdots + \alpha_q C_K(\Pi_q)$$

$$D_K(p) = \alpha_1 D_K(\Pi_1) + \cdots + \alpha_q D_K(\Pi_q)$$

This interpolation yields a smooth scheduling of the controller matrices by the parameter measurements $p(t)$.

The closed-loop interconnection involving the parameter-dependent plant $P(., p)$ and parameter-dependent controller $K(., p)$ is shown in Fig. 13.5 where w is the vector of external input signals and z is the vector characterizing the closed-loop system behavior. The design goal is to determine a gain-scheduled controller with the vertex representation (13.14) that stabilizes the closed-loop system for all admissible parameter trajectories $p(t)$ and such the worst-case closed-loop gain from w to z does not exceed some prescribed level $\gamma > 0$. Controllers that satisfy this aim are called \mathcal{H}_∞ gain scheduled controllers.

The design of \mathcal{H}_∞ gain-scheduled controllers is done by the function `hinfgs` which finds the desired controller by solving 2s LMIs (see for details [9, 50, Chap. 7]). The syntax of this function is

```
[gopt,pdK] = hinfgs(pdP,r)
```

The input of the function consists of the affine or polytopic model `pdP` of the plant 13.12 and the vector `r=[p2 m2]` specifying the dimensions of the output y and control u , respectively. On output, `gopt` is the best achievable with respect to parameter variations closed-loop \mathcal{H}_∞ norm and `pdK` is a polytopic system consisting of the vertex controllers

$$K_i = \begin{bmatrix} A_K(\Pi_i) & B_K(\Pi_i) \\ C_K(\Pi_i) & D_K(\Pi_i) \end{bmatrix}$$

For a given value p of the parameter vector $p(t)$, the corresponding gain-scheduled controller

$$K(p) = \begin{bmatrix} A_K(p) & B_K(p) \\ C_K(p) & D_K(p) \end{bmatrix}$$

is determined by entering the command lines

```
alpha = polydec(pv,p)
kp = psinfo(pdK,'eval',alpha)
```

Here, the function `polydec` computes the convex decomposition $p(t) = \sum_{i=1}^q \alpha_i \Pi_i$ producing as an output the vector `alpha = (alpha_1, ..., alpha_q)`. The controller $K(p) = \sum_{i=1}^q \alpha_i K_i$ corresponding to the vector `p` is obtained as a `SYSTEM` matrix by the function `psinfo`.

A polytopic model of the closed-loop system may be found by the function `slft` using the command line

```
pcl = slft(pdP,pdK)
```

For a given parameter value `p` of $p(t)$ the closed-loop system is evaluated by the line

```
pcl_t = psinfo(pcl,'eval',polydec(pv,p))
```

The closed-loop time response is obtained by

```
pdsimul(pcl,'trafun',tf,'inputfun')
```

where `pcl` is the polytopic representation of the closed-loop system and `tf` is the desired duration of the simulation. The functions `trafun` `inputfun` define the desired parameter trajectory and input signals, respectively, as function of the time. If the parameter '`inputfun`' is omitted, the `pdsimul` plots the step response by default.

Example 13.4 This example presents the design of an \mathcal{H}_∞ gain-scheduled controller for a hydro turbine power unit in the case of wide range power variation.

Assuming a rigid conduit and incompressible flow, the turbine and penstock characteristics are determined by three basic equations relating to the velocity of the water in the penstock, the turbine mechanical power, and the acceleration of water column [88]. We have

$$\begin{aligned} U &= k_u G \sqrt{H} \\ P_m &= k_p H U \\ (\rho L A_p) \frac{dU}{dt} &= -A_p (\rho a_g) (H - H_0) \end{aligned} \quad (13.15)$$

where U is the water velocity; G gate opening, H hydraulic head at gate, k_u a proportionality coefficient, P_m turbine mechanical power, k_p turbine efficiency, H_0 initial steady-state value of H , A_p pipe area, L length of the penstock, ρ mass density, a_g acceleration due to gravity, $\rho a_g (H - H_0)$ static pressure deviation at turbine gate.

To describe the dynamics of the hydro turbine in wide range of operational conditions, it is convenient to represent (13.15) in normalized form, based on steady-state operating values of the corresponding variables. Setting

$$\bar{U} = \frac{U}{U_0}, \quad \bar{G} = \frac{G}{G_0}, \quad \bar{H} = \frac{H}{H_0}, \quad \bar{P}_m = \frac{P_m}{P_{m0}}$$

where the subscript 0 refers to initial steady-state values, (13.15) are written as

$$\begin{aligned}\bar{H} &= (\bar{U}/\bar{G})^2 \\ \bar{P}_m &= \bar{U} \bar{H} \\ T_w \frac{d\bar{U}}{dt} &= -(\bar{H} - 1)\end{aligned}\tag{13.16}$$

where

$$T_w = \frac{LU_0}{a_g H_0}\tag{13.17}$$

is the *water starting time at any load*. It represents the time required for a head H_0 to accelerate the water in the penstock from standstill to the velocity U_0 . The value of T_w at rated load is called *water starting time at rated load* and is denoted by T_W . This time constant has a fixed value for a given turbine-penstock unit and can be obtained from

$$T_W = \frac{LQ_r}{a_g A_p H_r}$$

where $Q_r = A_p U_r$ is the water-flow rate at rated load and U_r, H_r are the water velocity and hydraulic head at rated load, respectively.

In case of small signal deviations about the operation point, the linearized representation of (13.16) is

$$\begin{aligned}\Delta \bar{U} &= \Delta \bar{G} + 0.5 \Delta \bar{H} \\ \Delta \bar{P}_m &= \Delta \bar{H} + \Delta \bar{U} \\ T_w \frac{d\Delta \bar{U}}{dt} &= -\Delta \bar{H}\end{aligned}\tag{13.18}$$

After some algebraic manipulations of (13.18) the transfer function between the gate opening position and the turbine mechanical power is obtained as

$$\Delta \bar{P}_m = \frac{-T_w s + 1}{0.5 T_w s + 1} \Delta \bar{G}\tag{13.19}$$

It can be seen that the linearized model transfer function depends on the time constant T_w , which varies with load, i.e., it has to be considered as a varying system parameter.

Denoting the power reference value as $P_0 = k_p H_0 U_0$, it is possible to obtain from (13.17) a relationship between the time constant T_w and the normalized power reference $\bar{P}_0 = P_0/P_r$, where $P_r = k_p H_r U_r$ is the turbine power at rated load. As a result we have

$$T_w(\bar{P}_0) = k \frac{\bar{P}_0}{\bar{H}_0^2}, \quad k = \frac{L P_r}{a_g k_p H_r^2}\tag{13.20}$$

where $\bar{H}_0 = H_0/H_r$. Equation (13.20) shows that the time constant T_w is a function of the power reference value \bar{P}_0 which changes during the hydro turbine operation. The variation of this time constant leads to variations of turbine model which should be considered as a parameter-dependent model. Taking into account (13.20) the following plant model is obtained:

$$\Delta \bar{P}_m = \frac{-T_w(\bar{P}_0)s + 1}{0.5T_w(\bar{P}_0) + 1} \Delta \bar{G} \quad (13.21)$$

Further on we shall make use of the following data related to the turbine, penstock and generator of the hydro generating unit: penstock length 900 m; rated hydraulic head 150 m; initial hydraulic head 1 m; piping area 8.5 m²; water velocity at rated load 7.65 m/s; water flow rate at rated load 65 m³/s; gate opening at rated load 0.96; gate opening at no load 0.02.

The servo system opening the turbine gate is assumed to have the description

$$\Delta \bar{G}(s) = \frac{1}{T_s s + 1} \Delta \bar{u}(s)$$

where \bar{u} is the normalized servo system reference signal and $T_s = 3$ s. Incorporating this dynamics in the plant equations we obtain

$$\bar{P}_m(s) = W(s, \bar{P}_0) \bar{u}(s)$$

where the transfer function $W(s, \bar{P}_0)$ relating the control signal and the hydro turbine power is obtained as

$$W(s, \bar{P}_0) = \frac{1}{(T_s s + 1)} \frac{(-T_w(\bar{P}_0)s + 1)}{(0.5T_w(\bar{P}_0)s + 1)} \quad (13.22)$$

Introducing the state and output vectors

$$x(t) = [\Delta \bar{G}(t) \quad \Delta \bar{P}_m(t)]^T, \quad y(t) = \bar{P}_m(t)$$

one obtains the state space description

$$\dot{x}(t) = A(\bar{P}_0)x(t) + Bu(t) \quad (13.23)$$

$$y(t) = Cx(t) \quad (13.24)$$

where

$$A(\bar{P}_0) = \begin{bmatrix} -\frac{1}{T_s} & 0 \\ \frac{2}{T_s} + \frac{2}{T_w} & -\frac{2}{T_w} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1}{T_s} \\ -\frac{2}{T_s} \end{bmatrix}, \quad C = [0 \quad 1]$$

The system (13.23),(13.24) may be represented as the parameter-dependent system

$$\dot{x} = A(p)x + B(p)u \quad (13.25)$$

$$y = C(p)x + D(p)u \quad (13.26)$$

which is affine in the uncertain parameter

$$p = \frac{2}{T_w} \quad (13.27)$$

In fact, using the setting (13.27), the system matrices may be written as

$$\begin{aligned} A(p) &= A_0 + pA_1, & B(p) &= B_0 + pB_1 \\ C(p) &= C_0 + pC_1, & D(p) &= D_0 + pD_1 \end{aligned}$$

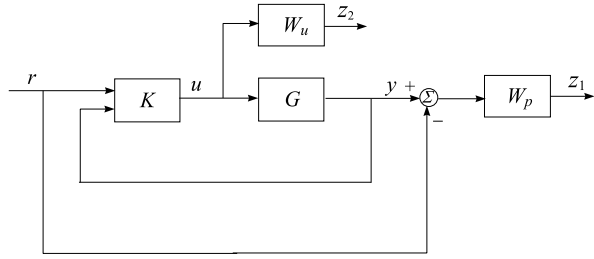
where

$$\begin{aligned} A_0 &= \begin{bmatrix} \frac{1}{T_s} & 0 \\ \frac{1}{T_s} & 0 \end{bmatrix}, & A_1 &= \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix} \\ B_0 &= \begin{bmatrix} \frac{1}{T_s} \\ \frac{2}{T_s} \end{bmatrix}, & B_1 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ C_0 &= [0 \quad 1], & C_1 &= [0 \quad 0] \\ D_0 &= 0, & D_1 &= 0 \end{aligned}$$

Assuming that the normalized power reference varies in the interval $[0.1, 1.0]$, the time constant T_w varies in the range $[0.46771, 4.6771]$. Hence the uncertain parameter p varies between the values $2/4.6771$ and $2/0.46771$.

The affine parameter-dependent system model is obtained in the variable `pds` by the lines

```
A_0 = [-1/Ts 0; 2/Ts 0]; B_0 = [1/Ts -2/Ts]';
C_0 = [0 1]; D_0 = 0;
S_0 = ltisys(A_0,B_0,C_0,D_0,1);
%
A_p = [0 0; 1 -1]; B_p = [0 0]';
C_p = [0 0]; D_p = 0;
S_p = ltisys(A_p,B_p,C_p,D_p,0);
%
% determine the range of time constant Tw
P0 = 1.0;
Tw1 = L*P0*Pr/(ag*kp*Hr^2);
P0 = 0.1;
Tw2 = L*P0*Pr/(ag*kp*Hr^2);
%
pv = pvec('box',[2.0/Tw1 2.0/Tw2]);
pds = psys(pv,[S_0 S_p]);
```

Fig. 13.6 Closed-loop system

The design of gain-scheduled controller of the hydro turbine system is done according to the block diagram shown in Fig. 13.6 and resembles the mixed-sensitivity \mathcal{H}_∞ design. A 2-degree-of-freedom controller is used to achieve better performance. The open-loop plant interconnection is obtained by the command line

```
[pdP,nmc] = sconnect('r','e=r-G(1);K','K:[r;G]','G:K',pds);
```

The arguments of the function `sconnect` have the following meaning.

- The first input argument `'r'` is the scalar reference signal r .
- The second argument `'e=r-G(1);K'` specifies two output signals separated by a semicolon. These signals are the difference $r - y$ between reference and plant output, and the control action u .
- The third argument `'K:[r;G]'` names the controller and specifies its inputs. In the given case a 2-degrees-of-freedom controller is used with inputs r and y .
- The remaining arguments come in pairs and specify, for each system in the loop, its input list and its `SYSTEM` matrix. Here `'G:K'` means that the input of G is the output of K , and `tt pds` is the `SYSTEM` matrix of $G(s)$.

The design aim is to fulfill the mixed-sensitivity requirement

$$\left\| \begin{bmatrix} W_p S \\ W_u K S \end{bmatrix} \right\|_\infty < 1$$

The performance and control weighting functions are chosen as

$$W_p(s) = 1.6 \frac{4s + 1}{60s + 0.01}, \quad W_u(s) = 0.4 \frac{1.2s + 1}{0.0024s + 1}$$

respectively, in order to achieve good transient response respecting the constraint on the control action. These weighting functions are appended to the plant interconnection by the line

```
Paug = smult(pdP,sdiag(Wp,Wu,eye(2)));
```

where the function `sdiag` forms a block-diagonal matrix with W_p , W_u and the unit 2×2 matrix on the diagonal and `smult` multiplies the parameter-dependent plant `pdP` with this matrix. Now the polytopic gain-scheduled controller with two fourth

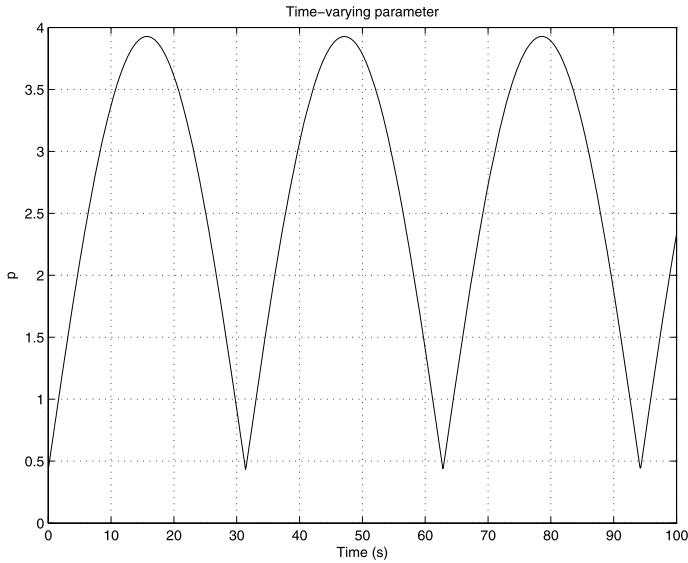


Fig. 13.7 Variation of the system parameter

order vertex systems K_1 and K_2 is obtained in the variable `pdK` by the line

```
[gopt,pdK] = hinfgs(Paug,nmc)
```

and the corresponding closed-loop system is

```
pCL = slft(pdP,pdK)
```

The simulation of the closed-loop system is done by the function `pdsimul`. In order to set difficult conditions for controller work, the parameter trajectory is chosen to vary as shown in Fig. 13.7. The trajectory is set in the file `traj_ht`. The transient response of the closed-loop system for this parameter trajectory is computed for a time period of 100 s by the lines

```
tf = 100;
[t,x,y] = pdsimul(pCL,'traj_ht',tf);
```

The step response of the closed-loop system is shown in Fig. 13.8. Although the parameter T_w varies rapidly which leads to quick variations of the controller, the transient response is satisfactory. The corresponding control action is shown in Fig. 13.9. The Simulink® model `Hinf_GS_Control` of the closed-loop system involving the nonlinear plant model and the gain-scheduled controller is shown in Fig. 13.10. The model of the gain-scheduled controller

$$K(p) = \alpha_1 K_1 + \alpha_2 K_2$$

including the two vertex time-invariant controllers K_1 and K_2 , is shown in Fig. 13.11. For given value of the scheduling parameter T_w , computed from the

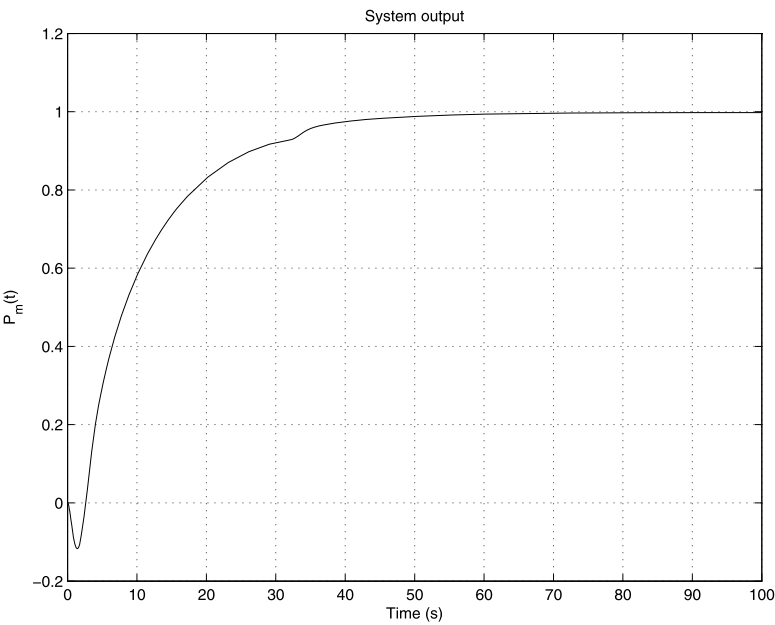


Fig. 13.8 Step response of the closed-loop system

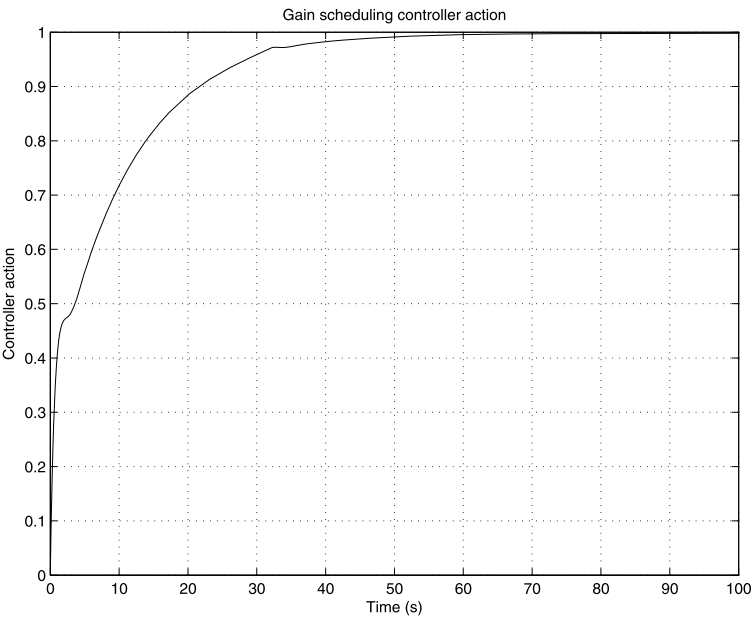


Fig. 13.9 Closed-loop control action

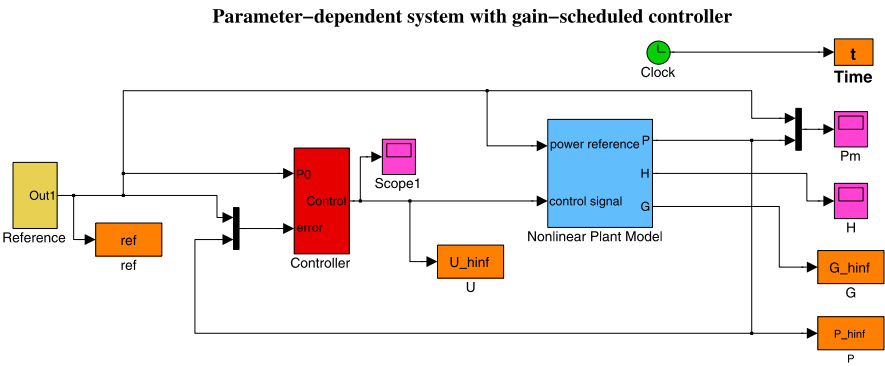


Fig. 13.10 Simulink model of the closed-loop system

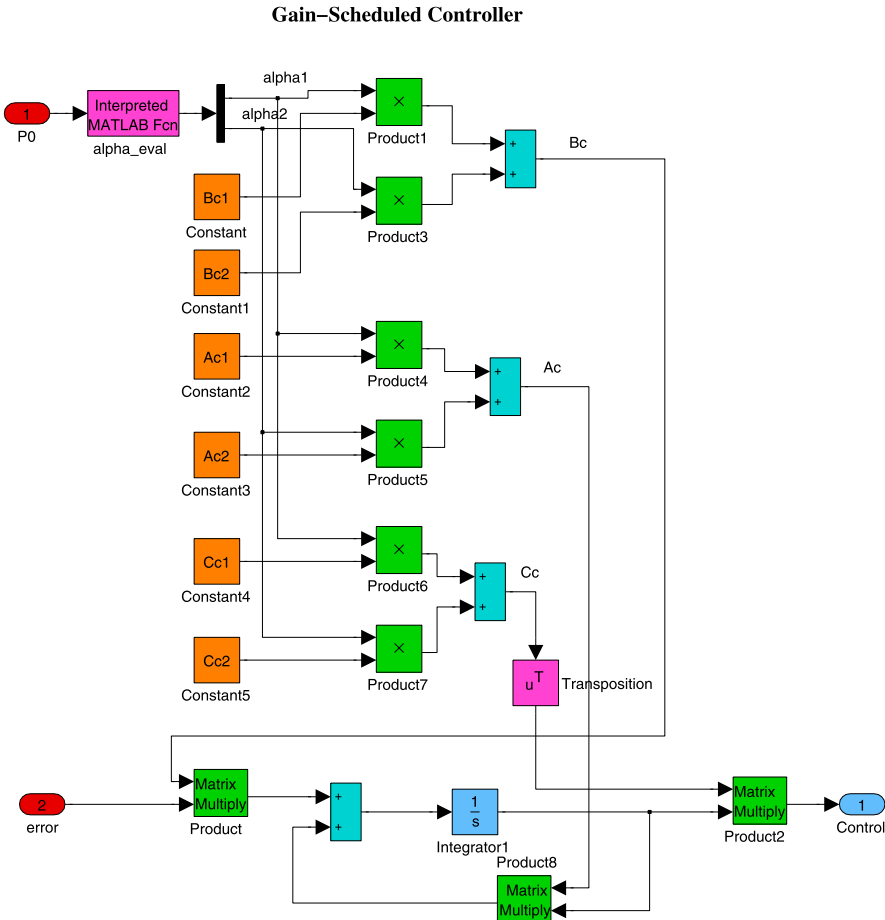


Fig. 13.11 Simulink model of the gain-scheduled controller

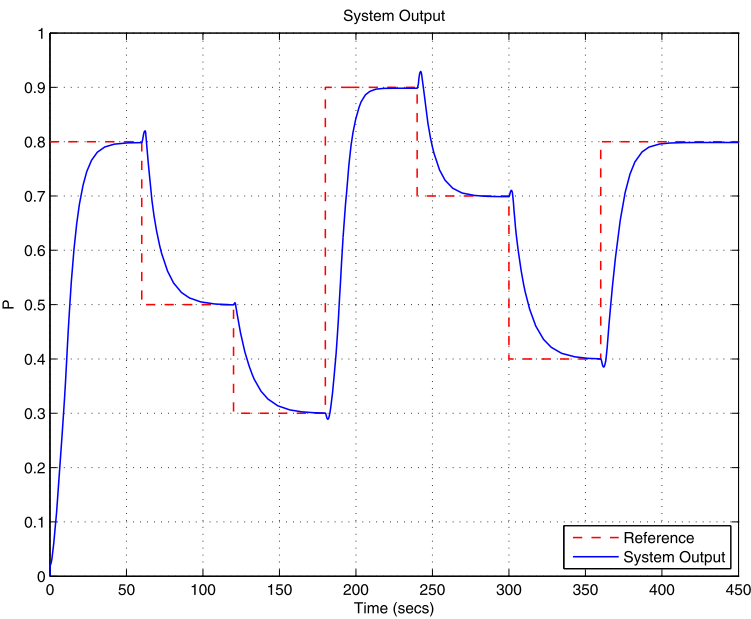


Fig. 13.12 Transient response of the closed-loop system

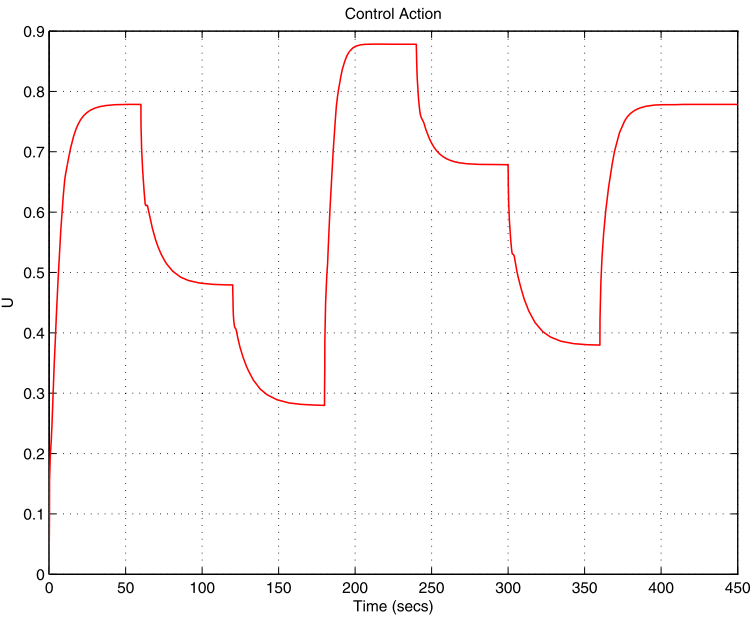


Fig. 13.13 Closed-loop control action

desired power reference P_0 , the polytopic coordinates α_1 and α_2 are evaluated in the function `alpha_eval` by the line

```
alpha = polydec(pv, 2.0/Tw);
```

The closed-loop system simulation is done for different values of the power reference in the range 0.3–0.9 and for a period of time equal to 450 s. The simulation results are plotted by the function `plot_simul_results`. The variation of the reference signal and the corresponding closed-loop transient response are shown in Fig. 13.12. It is seen that the behavior of the nonlinear closed-loop system is close to the behavior of the linear system whose step response was shown in Fig. 13.8. The closed-loop control action, shown in Fig. 13.13, varies in the interval $[0, 0.9]$.

13.4 Exercises

Exercise 13.1 Analyze the quadratic stability of the mass–damper–spring system using the function `quadstab`.

Exercise 13.2 For the plant described in Example 13.4 try to find a μ -controller that ensures robust stability and robust performance of the closed-loop system for the same range of variation of the time constant T_w .

Part III

Design Examples

This page intentionally left blank

Chapter 14

Robust Control of a Hard Disk Drive

This chapter considers the design of a robust servo system of a *hard disk drive* (HDD). Three robust design methods are applied, namely, the μ -synthesis, \mathcal{H}_∞ optimal design and the \mathcal{H}_∞ loop-shaping design procedure (LSDP). After a description of the HDD servo-system dynamics in Sect. 14.1, it is shown (in Sect. 14.2) how to derive the plant model that involves several uncertain parameters. Then we consider the synthesis of continuous-time controllers using the available methods for robust control design. These controllers are compared in aspects of robustness of closed-loop system stability and of performance in the frequency domain and in the time-domain. The design of discrete-time controller is also included, for two sampling frequencies. Finally, we present the simulation results of the nonlinear continuous-time and discrete-time closed-loop systems, followed by conclusions at the end of the chapter.

The prevalent trend in hard-disk design is towards smaller disks with increasingly larger capacities. This implies that the track width has to be smaller leading to lower error tolerance in the positioning of the read/write heads. The controller for track following has to achieve tighter regulation in the presence of parameter variations, nonlinearities and noises. Hence, it is appropriate to use advanced design methods like μ -synthesis and \mathcal{H}_∞ optimization in order to achieve robust stability and robust performance of the closed-loop servo system.

14.1 Hard Disk Drive Servo System

The schematic diagram of a typical hard disk drive is shown in Fig. 14.1. The disk assembly consists of several flat disks called *platters* coated on both sides with very thin layers of magnetic material (thin-film media). The magnetic material is used to store the data in the form of magnetic patterns. The platters rotate at high speed, driven by a spindle motor. Recently, the spindle speed has become 5400 r.p.m., 7200 r.p.m. or even 10,000 and 15,000 r.p.m. The data are retrieved from, or recorded onto, the platters by electromagnetic *read/write* (R/W) heads that

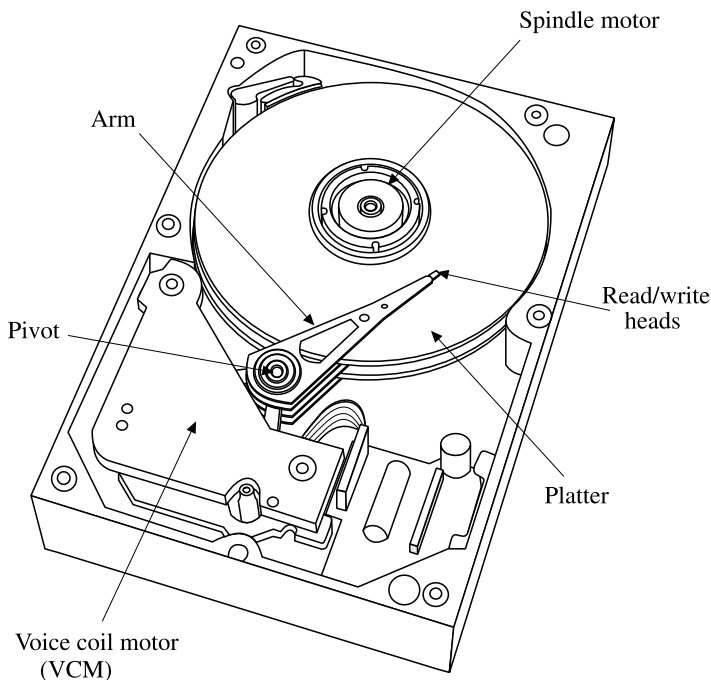


Fig. 14.1 Schematic diagram of a hard disk drive

are mounted at the bottom of *sliders*. Today's hard disks read data using *giant-magneto-resistive* (GMR) heads and write data with thin-film inductive heads. The sliders with read/write heads are mounted onto head arms. The arms are lightweight rigid constructions allowing them to be moved rapidly on the platter surface. There is one arm per read/write head and all arms are mounted in a stack assembly that moves over multiple disk surfaces simultaneously. The heads are suspended less than 1 microinch above the disk surface. The appropriate flying height of the heads is achieved thanks to the air flow generated by the spinning disk.

The data recorded on the platters are in concentric circles called *tracks*. Modern hard disks have tens of thousands of tracks resulting in track density 35,000 *tracks per inch* (TPI) or more. Thus, the distance between adjacent tracks is less than a microinch. Each track is divided into smaller pieces called *sectors* that contain 512 bytes of information. There may be several hundreds or even thousands of sectors in a track. The drive density may reach several hundreds GB per platter ($1 \text{ GB} = 10^9 \text{ bytes}$).

The head arms are moved on the surface of the platter by a rotary voice coil actuator frequently called the *Voice Coil Motor* (VCM). The VCM consists of a voice coil, mounted at the end of the head arm assembly, and permanent electromagnets. By controlling the current in the coil, the heads can move in one direction or the other in order to follow precisely the data track.

The goal of the hard disk drive servo control system is to achieve a precise positioning of the read/write heads on the desired track (track-following mode) while data are being written or read, and a quick transition from one track to another target track (seeking mode). As the drive initiates a seek command, it switches to a seek control algorithm that is a type of time-optimal (bang-bang) control algorithm. When the head is positioned over the desired track, the drive switches into track-following mode. In the present work we consider the servo controller of the track-following mode.

All modern hard disk drives read the relative position of the head to track directly from the disk media by using a method called *embedded servo* or *sector servo*. For this method the servo information is interleaved with user data across the entire surface of all platters. Position information is placed on the disk surfaces in servo frames during the manufacturing of the disk and cannot be rewritten. This is why the disk heads are locked out by the drive controller from writing to the areas where servo information is written. Because of the interleaving of servo information with data, the embedded servo system is a sampled-data system. Increasing the number of servo frames within a track improves the performance of the servo system due to the higher sample rate but limits the maximum data storage. The servo information is read by the same head that reads the data. The position information usually consists of two parts: coarse position giving track number and fine position information relative to each track. The position error, representing the difference between the reference track position and the head position, is measured by making use of *position bursts* that are part of the servo information. The position bursts are patterns of alternating magnetic polarity written on the disk surface with a particular frequency. The periodic signal, obtained from the read head passing over a burst pattern, has an amplitude that is proportional to how much the read head is directly over the burst pattern. The signals corresponding to two or four bursts of position information are demodulated by a servo demodulator in order to compose the *position error signal* (PES). This signal is used by the servo controller to change the voice coil current appropriately and hence the read/write heads position. It should be pointed out that the absolute head position is not generally known from the servo information that is read off of the disk. Only a signal proportional to the position error is available, which means that for track following the reference signal is ideally equal to zero.

The failure of the head to follow the track on a platter surface faithfully as the disk spins is referred to as *runout*. The runout could have serious consequences, especially during writing where data in adjacent tracks might be overwritten. There are two types of runout, namely the *repeatable runout* (RRO) and *nonrepeatable runout* (NRRO). Repeatable runout is caused by both imbalance in the spindle and imperfections in the servowriting process that result in noncircular position information. This information is encoded at the spindle frequency, yielding a repeatable noncircular track for the disk servo to follow. The nonrepeatable runout is caused by a windage induced actuator arm, slider motion and mechanical vibrations that can arise from various sources such as ball bearing defects, spindle motor vibrations and slider vibrations. The RRO may be reduced through feedforward compensation at the corresponding frequency by using *harmonic correctors*.

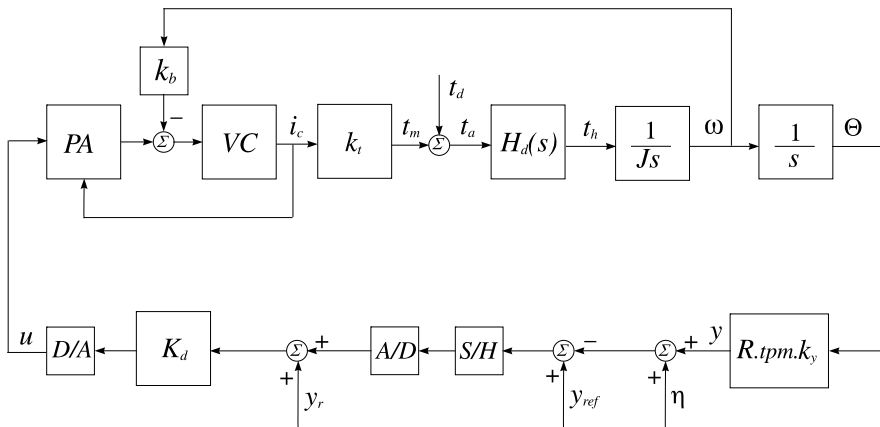


Fig. 14.2 Block-diagram of the hard disk drive servo system

The block-diagram of the HDD servo system is shown in Fig. 14.2. The R/W heads are moved by a VCM that is driven by the output current i_c of a *power amplifier* (PA). The actual position signal y is compared with the signal y_{ref} , which represents the desired head position. For the track following, the reference input y_{ref} is theoretically equal to zero and $y(t)$ appears as an error signal. In practice, y_{ref} must be set equal to the signal representing both RRO and NRRO. The digital signal y_r is a reference for the desired track and is used during the seeking mode. The error signal is sampled by an *analogue-to-digital* (A/D) converter and serves as part of an input to the digital servo controller K_d that is typically implemented on a *digital signal processor* (DSP) chip. The output of the controller is converted to analogue form by a *digital-to-analog* (D/A) converter and amplified by the PA. Since the motor torque is proportional to the voice coil current, the amplifier is configured as a current source. The exogenous signal t_d is the torque disturbance due to external shock and vibrations, power amplifier noise, digital-to-analogue converter noise, pivot bearing friction and flex cable bias. The increase of the spindle speed increases the air flow inside the disk (windage) that in turn increases the disturbance torque on the actuator. The disturbance is a low-frequency signal with spectral content usually below 500 Hz. The position noise signal η includes quantization errors due to servo demodulator noise, finite resolution of the analogue-to-digital converter, media noise and preamplifier noise. The position noise is a high-frequency signal with spectral content usually above 1 kHz. Since the measured PES is contaminated with noise, the true PES, $y_{\text{ref}} - y$, is not available.

One of the limitations inherent in the design of servo controllers for high track density HDD is the influence of actuator mechanical resonant modes on the head-positioning servo. If the actuator input contains a periodic component with frequency equal to a resonance frequency, this component may be amplified greatly that results in large off-track deviation of the read/write heads. Usually, the actuator is mechanically designed in such a way that the resonant modes occur at frequencies that will be attenuated by the servo system. However, as servo bandwidth increases

Table 14.1 Rigid body model parameters and tolerances

Parameter	Description	Value	Units	Tolerance
J	arm moment of inertia	6.3857×10^{-6}	kg m ²	±10.0 %
R	arm length	5.08×10^{-2}	m	±0.1 %
k_{PA}	amplifier gain	10.0	V/V	±0.0 %
k_t	VCM torque constant	9.183×10^{-2}	N m/A	±10.0 %
k_b	back e.m.f. constant	9.183×10^{-2}	N m/A	±10.0 %
tpm	tracks per meter	10^6	tracks	–
k_y	position measurement gain	1.2	V/track	±5.0 %
R_{coil}	coil resistance	8.00	Ω	±20.0 %
R_s	sense resistance in the power amplifier feedback	0.2	Ω	±1.0 %
L_{coil}	coil inductance	0.001	H	+0, –15 %
e_{max}	saturated power amplifier voltage	12.0	V	–0, +5 %
RPM	disk rotation rate	7200	rev/min	±1.0 %
t_w	track width	1	μm	±1.0 %

to meet higher performance requirements, this attenuation may not be achieved due to mechanical design constraints. It is also important to note that the presence of resonant modes may limit the servo bandwidth via stability margin constraints. With a reduced bandwidth, the servo system may not be able to achieve the desired performance.

The rotary actuators of hard disks may have tens of resonances that may lead to a high order model. However, in practice only 3 to four main resonances are taken into account. Usually, these are the first and second torsion modes (in the range 1500–2500 Hz) as well as the first sway mode (in the range 8000–12 000 Hz).

A common approach to reduce the effect of resonance modes is to put notch filters in the servoloop that attenuates or filters out vibrations at selected major resonant frequencies. However, each notch filter introduces phase margin loss at low frequencies thus reducing the system robustness. Also, the presence of uncertainty in the resonant modes may decrease significantly the efficiency of those filters.

Our goal in this chapter is to design a robust track-following servo control system for a 3.5-inch HDD with track density 25,400 TPI. The desired settling time is about 1 ms in the presence of four resonances, several uncertain parameters, position sensing noise and disturbances. The parameters of the rigid-body model and their tolerances are given in Table 14.1. For dimensional compatibility, the track density is given in tracks per meter, instead of in TPI.

We now first consider the derivation of a HDD servo system model.

The dynamics of the rotary arm is described by the equation

$$J \frac{d^2 \Theta}{dt^2} = t_m + t_d$$

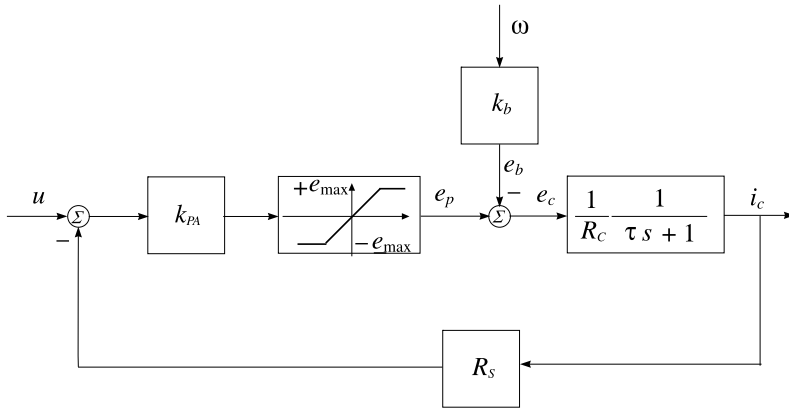


Fig. 14.3 Block diagram of the power amplifier with voice coil

where J is the arm moment of inertia, Θ is the angle of arm rotation, t_m is the VCM torque, and t_d is the disturbance torque.

The VCM torque is given by

$$t_m = k_t i_c$$

where k_t is the motor torque constant and i_c is the current through the VCM coil.

The voice coil has a resistance R_{coil} and an inductance L_{coil} . An additional current sense resistance R_s is connected in serial with the voice coil to implement a feedback from the power amplifier output. Hence the voice coil admittance is described by the transfer function

$$G_{\text{vca}}(s) = \frac{i_c(s)}{e_c(s)} = \frac{1/R_c}{\tau s + 1}$$

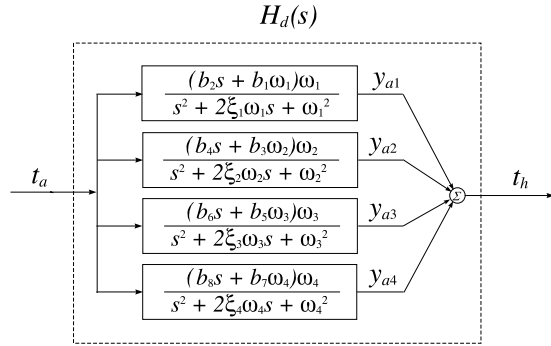
where e_c is the input voltage to the voice coil, $\tau = L_{\text{coil}}/R_c$ and $R_c = R_{\text{coil}} + R_s$.

The block diagram of the power amplifier with voice coil is shown in Fig. 14.3. The input of the voice coil is the difference $e_c = e_p - e_b$, where e_p is the output voltage of the amplifier and $e_b = k_b \omega$ is the back electromotive force (e.m.f.) that is generated during the moving of the coil in the magnetic field. Since the saturation voltage of the amplifier is e_{max} , in the absence of back e.m.f., the amplifier will saturate for an input voltage greater than $e_{\text{max}}/k_{\text{PA}}$. In the study limited to linear systems, the amplifier saturation is neglected.

The length of the arc, corresponding to the arm rotation angle Θ , is equal to $R\Theta$. For small values of Θ the number of tracks contained in the arc is $R \cdot \Theta \cdot \text{tpm}$. This gives an output signal $y = R \cdot \text{tpm} \cdot k_y \cdot \Theta$.

If we neglect the dynamics of the voice coil, the transfer function of the servomotor considered as a rigid body is obtained in the form of a double integrator. Such a model oversimplifies the system dynamics and cannot produce reliable results if used in the design.

Fig. 14.4 Transfer functions of resonant modes



The next step is to take into account the high-frequency resonant modes of the head disk assembly represented by the transfer function $H_d(s)$. In the given case $H_d(s)$ consists of four resonant modes and is obtained as

$$H_d(s) = \sum_{j=1}^4 \frac{b_{2j}\omega_j s + b_{2j-1}\omega_j^2}{s^2 + 2\xi_j\omega_j s + \omega_j^2}$$

(see Fig. 14.4).

Here ω_j , ξ_j and b_{2j} , b_{2j-1} are, respectively, the resonance frequency, the damping coefficient and the coupling coefficients of the j th mode, for $j = 1, \dots, 4$. The resonance parameters are usually determined experimentally and for the servo system under consideration their values are shown in Table 14.2.

It is important to note that all model parameters are known with some tolerances and may vary with the changing of working conditions as well as with time. Also, the closed-loop system can be very sensitive to the external disturbance t_d and the position-sensing noise η . Both factors would lead to an actual system dynamics far away from the dynamics of the nominal closed-loop system. Thus, it is necessary to use control-system design methods that ensure the desired closed-loop stability and performance in the presence of uncertain parameters, noises, and disturbances.

14.2 Derivation of Uncertainty Model

In order to implement robust control design methods we must have a plant model that incorporates uncertain parameters. As is seen from Tables 14.1 and 14.2, the total number of uncertain parameters is greater than 25, which complicates the analysis and design of a HDD servo system. In this study, we shall concentrate on those uncertainty parameters that influence the closed-loop system behavior most. By using the function `ureal` we define the real uncertain parameters

$$J, K_t, K_y, \omega_1, \omega_2, \omega_3, \omega_4, \xi_1, \xi_2, \xi_3, \xi_4$$

The block-diagram of the plant is shown in Fig. 14.5.

Table 14.2 Resonance parameters and tolerances

Parameter	Description	Value	Units	Tolerance
ω_1	pivot bearing resonance	$2\pi 50$	rad/s	$\pm 5.0\%$
ω_2	first torsional resonance	$2\pi 2200$	rad/s	$\pm 12.0\%$
ω_3	second torsional resonance	$2\pi 6400$	rad/s	$\pm 8.0\%$
ω_4	first sway resonance	$2\pi 8800$	rad/s	$\pm 15.0\%$
b_1	first resonance coupling	0.006	—	$\pm 7.0\%$
b_2	first resonance coupling	0	1/s	$\pm 7.0\%$
b_3	second resonance coupling	0.013	—	$\pm 10.0\%$
b_4	second resonance coupling	-0.0018	1/s	$\pm 7.0\%$
b_5	third resonance coupling	0.723	—	$\pm 5.0\%$
b_6	third resonance coupling	-0.0015	1/s	$\pm 10.0\%$
b_7	fourth resonance coupling	0.235	—	$\pm 5.0\%$
b_8	fourth resonance coupling	-0.0263	1/s	$\pm 10.0\%$
ξ_1	first resonance damping	0.05	—	$\pm 5.0\%$
ξ_2	second resonance damping	0.024	—	$\pm 8.0\%$
ξ_3	third resonance damping	0.129	—	$\pm 10.0\%$
ξ_4	fourth resonance damping	0.173	—	$\pm 10.0\%$

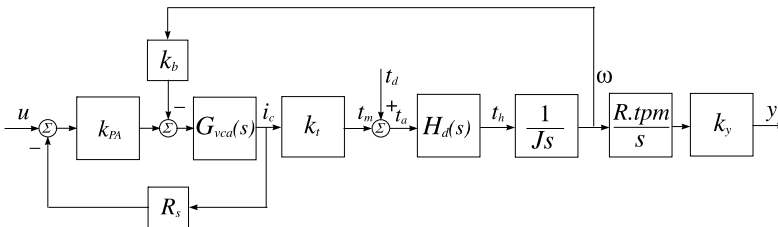


Fig. 14.5 Block-diagram of the plant

Consider first the derivation of the uncertainty model for the resonant modes. All four modes have similar transfer functions. The uncertainty model may be obtained in a different way. For instance, on the basis of the representation, shown in Fig. 14.5, one may implement the function \mathbf{tf} in order to obtain uncertain transfer function for each of the resonant modes. In this case, however, the uncertain parameter ω is repeated five times, which leads to a large number of the model uncertain parameters. This complicates very much the analysis and especially the synthesis of the system. That is why in the given case we use a state space representation in which ω is repeated only twice.

The model of a resonant mode in the state space is

$$\dot{x}_1 = \omega x_2$$

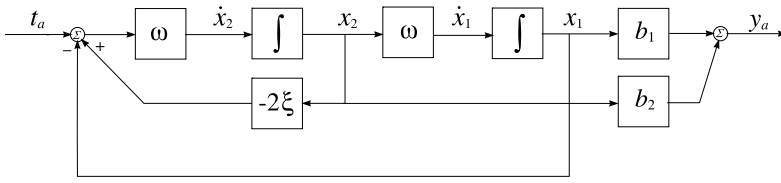


Fig. 14.6 Block-diagram of a resonant mode

Fig. 14.7 Parallel connection of the four resonant modes

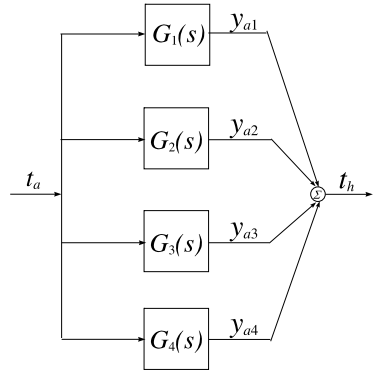
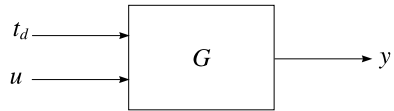


Fig. 14.8 Block-diagram of the hard disk considered as a control plant



$$\dot{x}_2 = \omega(-x_1 - 2\xi x_2 + t_a)$$

$$y_a = b_1 x_1 + b_2 x_2$$

The block-diagram corresponding to the state equations of a resonant mode is shown in Fig. 14.6. The uncertain model of the resonant mode in the given case is obtained by using the function `iconnect`. In this model the uncertain parameter ω is repeated twice.

The parallel connection of the four resonant modes, which has transfer function $H_d(s)$, is shown in Fig. 14.7.

After obtaining the transfer function of the resonant modes, the derivation of the uncertain plant model is done by the function `iconnect`, using the block diagram, shown in Fig. 14.5. The model obtained is of 11th order and has 11 independent uncertain parameters, four of them (ω_1 , ω_2 , ω_3 and ω_4) being repeated twice, such that the total number of the uncertain parameters becomes equal to 15. The uncertain model of the hard disk servo system is obtained by the M-file `mod_hdd.m` and is saved in the variable G .

The hard disk model has two inputs and one output (Fig. 14.8).

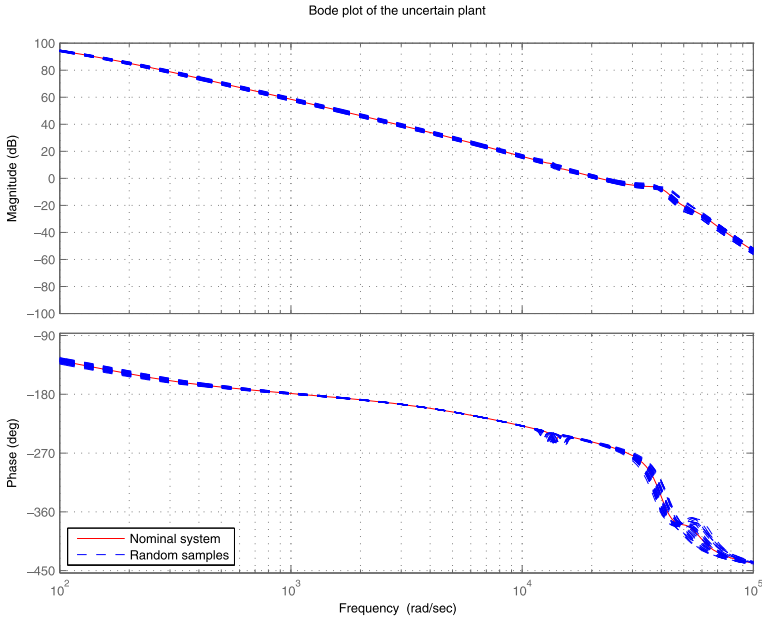


Fig. 14.9 Bode plot of the hard disk with uncertain parameters

Let us introduce the representation

$$G = [G_d \quad G_u]$$

such that

$$y = G_d t_d + G_u u$$

where G_d and G_u are the plant scalar transfer functions with respect to the disturbance and control action, respectively.

In Fig. 14.9 we show the Bode plot of the hard disk, obtained from the transfer function G_u for random values of the uncertain parameters.

14.3 Closed-Loop System Design Specifications

The design of the HDD servo controller will be first conducted in the continuous-time case. In general, it may obtain best possible performance in the continuous-time case that can then be considered as a limit for discrete-time designs. Also, in the continuous-time case it is easier to find appropriate performance weighting functions that again may be implemented in the discrete-time design.

The block-diagram of the closed-loop system, which includes the feedback structure and the controller as well as the elements representing the model uncertainty and the performance objectives, is shown in Fig. 14.10.

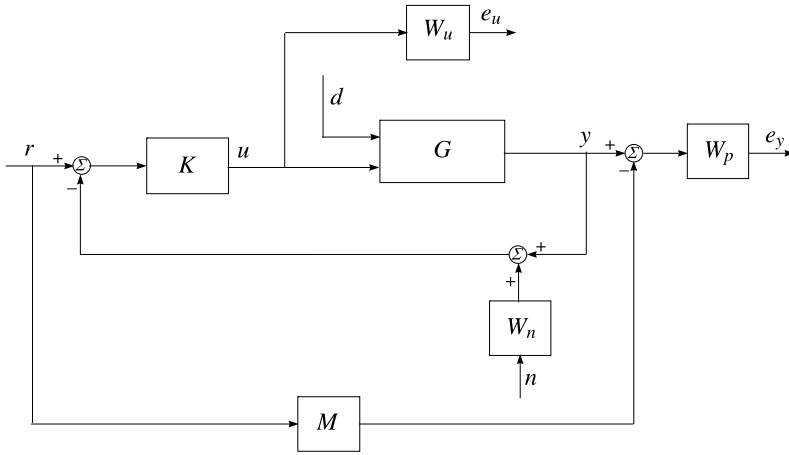


Fig. 14.10 Block-diagram of the closed-loop system with performance specifications

The system has a reference input (r), an input disturbance (d), a noise (n) and two output costs (e_y and e_u). The system M is an ideal model of performance, to which the designed closed-loop system tries to match. The transfer function of the uncertain plant is denoted by G . The performance objective requires the transfer matrix from r , d , and n to e_y and e_u to be small in the sense of $\|\cdot\|_\infty$, for all possible values of the uncertain parameters. The position noise signal is obtained by passing the unit-bounded signal n through the weighting transfer matrix W_n . The transfer matrices W_p and W_u are used to reflect the relative significance of the different frequency ranges for which the performance is required. Hence, the performance objective can be recast, with possible slight conservativeness, as that the $\|\cdot\|_\infty$ of the transfer function matrix from r , d , and n to e_y and e_u is less than 1.

It is straightforward to show that

$$\begin{bmatrix} e_y \\ e_u \end{bmatrix} = \begin{bmatrix} W_p(S_o G_u K - M) & W_p S_o G_d & -W_p S_o G_u K W_n \\ W_u S_i K & -W_u K S_o G_d & -W_u K S_o W_n \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix} \quad (14.1)$$

where $S_i = (I + KG)^{-1}$, $S_o = (I + GK)^{-1}$ are the input and output sensitivities, respectively. In (14.1) we make use of the fact that $S_i K = K S_o$. Note that $S_o G_d$ is the transfer function between d and y .

This objective is similar to the usual mixed S/KS sensitivity optimization and it would meet both robust stability and performance criteria by incorporating performance specifications in the matching model M . The six functions to be minimized are described in Table 14.3.

The controller synthesis problem of the Hard Disk Drive Servo System is to find a linear, output feedback controller $K(s)$ that has to ensure the following properties of the closed-loop system.

Table 14.3 \mathcal{H}_∞ functions to be minimized

Function	Description
$W_p(S_o G_u K - M)$	Weighted difference between the ideal and actual closed-loop systems
$W_p S_o G_d$	Weighted disturbance sensitivity
$W_p S_o G_u K W_n$	Weighted noise sensitivity
$W_u S_i K$	Weighted control effort due to reference
$W_u K S_o G_d$	Weighted control effort due to disturbance
$W_u K S_o W_n$	Weighted control effort due to noise

14.3.1 Nominal Performance

The closed-loop system achieves nominal performance if the performance objective is satisfied for the nominal plant model.

The nominal performance objective is to satisfy the inequality

$$\left\| \begin{bmatrix} W_p(S_o G_{u \text{ nom}} K - M) & W_p S_o G_{d \text{ nom}} & -W_p S_o G_{u \text{ nom}} K W_n \\ W_u S_i K & -W_u K S_o G_{d \text{ nom}} & -W_u K S_o W_n \end{bmatrix} \right\|_\infty < 1 \quad (14.2)$$

where the transfer functions $G_{d \text{ nom}}$, $G_{u \text{ nom}}$ are obtained for the nominal plant parameters.

14.3.2 Robust Stability

The closed-loop system achieves robust stability if the closed-loop system is internally stable for each possible plant dynamics G .

14.3.3 Robust Performance

The closed-loop system must remain internally stable for each G and in addition the performance criterion

$$\left\| \begin{bmatrix} W_p(S_o G_u K - M) & W_p S_o G_d & -W_p S_o G_u K W_n \\ W_u S_i K & -W_u K S_o G_d & -W_u K S_o W_n \end{bmatrix} \right\|_\infty < 1 \quad (14.3)$$

has to be satisfied for each G . This means that the structured singular value corresponding to the transfer function matrix from $\begin{bmatrix} r \\ d \\ n \end{bmatrix}$ to $\begin{bmatrix} e_y \\ e_u \end{bmatrix}$ (Fig. 14.10) should be less than 1, with regard to the parametric uncertainty and the complex fictitious uncertainty, connecting the inputs and outputs in the robust performance analysis.

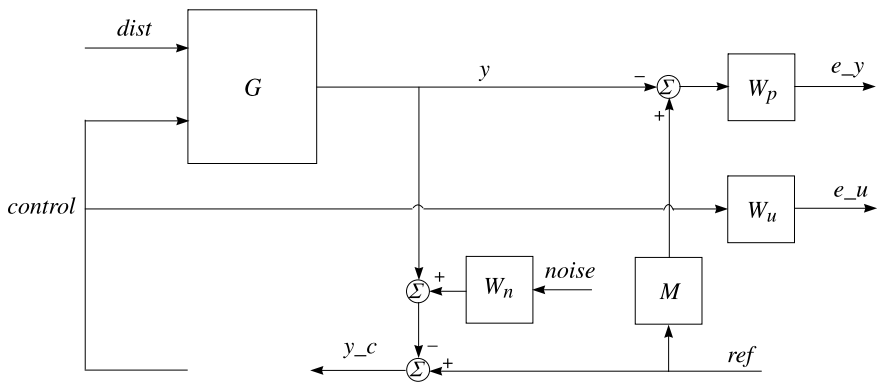
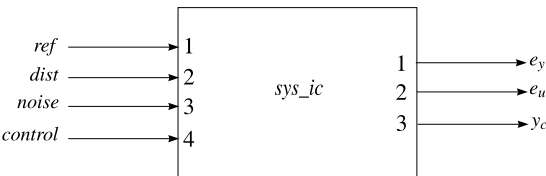


Fig. 14.11 Block-diagram of the open-loop system with performance specifications

Fig. 14.12 Schematic diagram of the open-loop connection



In addition to these requirements it is desirable that the controller designed would have acceptable complexity, i.e. it is of reasonably low order.

According to the above considerations, the aim of the design is to determine a controller *K*, such that for all stable perturbations, the perturbed closed-loop system remains stable and the performance objective is satisfied for all such perturbations.

14.4 System Interconnections

The internal structure of the four-input, three-output system, which is saved in the variable *sys_ic*, is shown in Fig. 14.11. The reference, the disturbance and the noise are saved in the variables *ref*, *dist*, and *noise*, respectively, and the control signal—in the variable *control*.

The variables *ref*, *dist*, *noise*, *y*, *y_c*, *e_y*, and *e_u* are scalar variables.

The open-loop connection is obtained by the M-file *olp_hdd*. The schematic diagram, showing the specific input/output ordering for the variable *sys_ic*, is given in Fig. 14.12.

The block-diagram used in the simulation of the closed-loop system is shown in Fig. 14.13. The corresponding closed-loop interconnection, which is saved in the variable *sim_ic*, is obtained by the M-file *sim_hdd*.

The schematic diagram showing the specific input/output ordering for the variable *sim_ic* is shown in Fig. 14.14.

Fig. 14.13 Block-diagram of the closed-loop system

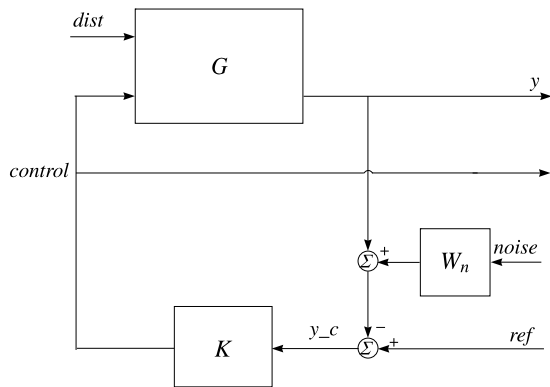
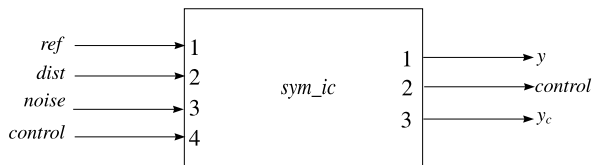


Fig. 14.14 Schematic diagram of the closed-loop connection



14.5 Controller Design in Continuous-Time

There are a few further “hard” constraints for the controller design, as listed below.

Peak closed-loop gain	<4 dB
Open-loop gain	>20 dB at 100 Hz
Steady-state error	<0.1 μm
Settling time	<1.5 ms
Closed-loop bandwidth	>1000 Hz
Gain margin	>5 dB
Phase margin	>40 deg

The designed control system must achieve good disturbance rejection and noise attenuation. In addition, it is necessary to have control action smaller than 1.2 V in order to avoid power-amplifier saturation.

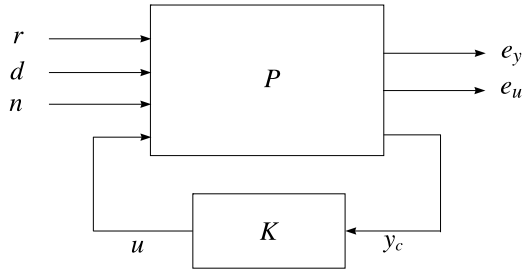
To design the controller we shall use μ -synthesis, \mathcal{H}_∞ optimization and the \mathcal{H}_∞ loop shaping design procedure (LSDP) in this exercise.

In the case of μ -synthesis and \mathcal{H}_∞ optimization design, we have to specify the model transfer function M and the weighting transfer functions W_n , W_p , and W_u .

The model transfer function is chosen so that the time response to the reference signal would have an overshoot less than 20 % and a settling time less than 1 ms. A possible model satisfying the requirements is

$$M = \frac{1}{3.75 \times 10^{-9}s^2 + 1.2 \times 10^{-4}s + 1}$$

Fig. 14.15 Block-diagram of the closed-loop system with μ -controller



The noise shaping function W_n is determined on the basis of the spectral density of the position noise signal. In the given case it is taken as the high-pass filter

$$W_n = 6 \times 10^{-4} \frac{0.1s + 1}{0.001s + 1}$$

whose output has a significant spectral content above 500 Hz. For this shaping filter, the position noise signal is only 0.6 mV in the low-frequency range but it is 60 mV in the high-frequency range that corresponds to a position error of 5 % of the track width.

The weighting functions W_p and W_u have to be chosen so as to ensure an acceptable trade-off between the nominal performance and the robust performance of the closed-loop system. They are selected in the course of the μ -synthesis, since this particular design method allows us to achieve maximum performance of the perturbed, closed-loop system.

14.5.1 μ -Design

In the μ -synthesis we shall take into account only the uncertainty in the rigid-body model (i.e. only the uncertainty in the parameters k_t , J , and k_y) will be considered. The inclusion of the uncertainties of resonant modes would make the D - K iterations difficult to converge which is the reason to ignore these uncertainties. This confirms that the resonant modes may create difficulties in the controller design. However, these resonant modes (with nominal values) are included in the plant dynamics and will be considered, with parametric variations, in the assessment of designed controllers in Sect. 14.6.

The block-diagram of the closed-loop system used in the μ -synthesis is shown in Fig. 14.15. Denote by $P(s)$ the transfer function matrix of the four-input, three-output, open-loop system. This matrix is obtained from the open-loop connection `sys_ic` by the command line

```
mu_ic = usubs(sys_ic, 'w1', 'nom', 'w2', 'nom', 'w3', 'nom', ...
               'w4', 'nom', 'z1', 'nom', 'z2', 'nom', ...
               'z3', 'nom', 'z4', 'nom')
```

and is saved in the variable `mu_ic`. During the execution of this command the parameters $\omega_1, \omega_2, \omega_3, \omega_4, \xi_1, \xi_2, \xi_3$, and ξ_4 are substituted by their nominal values. The transfer function matrix $P(s)$ may be represented as the upper Linear Fractional Transformation

$$P = F_u(P_{\text{nom}}, \Delta_r)$$

where the matrix P_{nom} is the nominal transfer matrix and Δ_r contains the three uncertainties in the rigid body model. The robust performance of the closed-loop system is considered with regard to the block uncertainty

$$\Delta_P := \left\{ \begin{bmatrix} \Delta_r & 0 \\ 0 & \Delta_F \end{bmatrix} : \Delta_r \in \mathcal{R}^{3 \times 3}, \Delta_F \in \mathcal{C}^{3 \times 2} \right\}$$

where the second block Δ_F is a fictitious uncertainty block that connects the inputs and outputs of the closed-loop system. In order to satisfy the robust performance requirements it is necessary to find a stabilizing controller $K(s)$, such that for each frequency $\omega \in [0, \infty]$ the structured singular value satisfies the condition

$$\mu_{\Delta_P}[F_L(P_{\text{nom}}, K)(j\omega)] < 1$$

The fulfillment of this condition guarantees robust performance of the closed-loop system, i.e.

$$\left\| \begin{bmatrix} W_p(S_o G_u K - M) & W_p S_o G_d & -W_p S_o G_u K W_n \\ W_u S_i K & -W_u K S_o G_d & -W_u K S_o W_n \end{bmatrix} \right\|_{\infty} < 1 \quad (14.4)$$

The μ -synthesis is conducted by using the M-file `ms_hdd`. It should be noted that the robust performance achieved during the D - K iteration is only with respect to the uncertainties in the rigid-body model, since only these uncertainties are taken into account in the design. Hence it is necessary to make an additional robust stability and robust performance analysis that takes into account the other uncertainties.

The closed-loop system performance specifications are reflected by the weighting performance function $W_p(s)$. Three performance weighting functions are considered in the design. They are

$$W_{p1}(s) = 10^{-4} \frac{s^2 + 8 \times 10^4 s + 10^8}{s^2 + 7 \times 10^4 s + 2.5 \times 10^4}$$

$$W_{p2}(s) = 10^{-4} \frac{s^2 + 4 \times 10^5 s + 2.5 \times 10^9}{s^2 + 3.9 \times 10^5 s + 6.25 \times 10^5}$$

and

$$W_{p3}(s) = 10^{-4} \frac{s^2 + 1.15 \times 10^6 s + 3.6 \times 10^{10}}{s^2 + 1.05 \times 10^6 s + 9 \times 10^6}$$

In Fig. 14.16 we show the magnitude responses of the inverses of these three weighting functions, i.e. W_{p1}^{-1} , W_{p2}^{-1} , and W_{p3}^{-1} . It is seen that in all three selections,

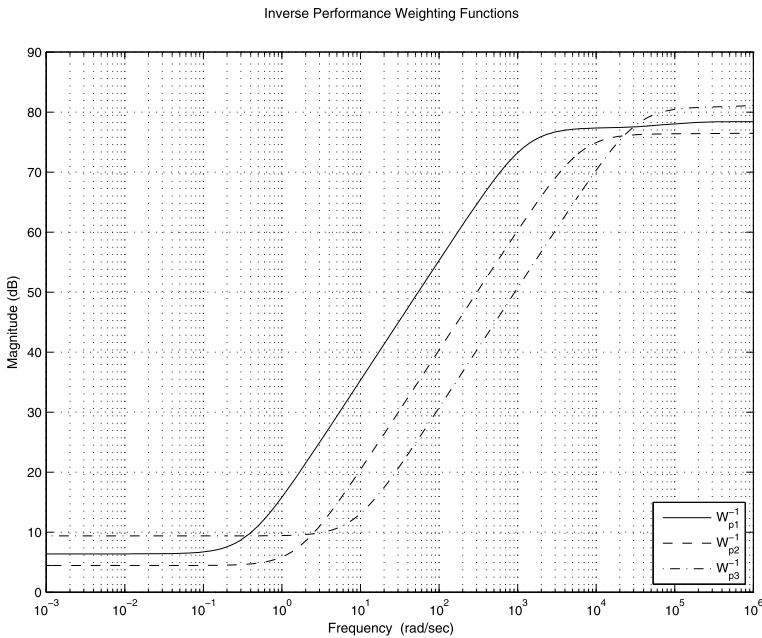


Fig. 14.16 Magnitude responses of the inverse of W_p

the aim is to achieve a small difference between the system and model outputs, and a small effect of the disturbance on the system outputs. This will ensure good tracking of the reference input and small error due to low-frequency disturbances. Changing the performance weighting function from W_{p1} to W_{p3} moves the inverse weighting frequency response to the right (to higher frequencies) which forces the system to match the model in over a larger frequency range.

The control weighting function is usually chosen as high-pass filters in order to ensure that the control action will not exceed 1.2 V. Again, three such weighting functions are considered in the design and are listed below:

$$W_{u1}(s) = 10^{-6} \frac{0.385s^2 + s + 1}{10^{-4}s^2 + 2 \times 10^{-3}s + 1}$$

$$W_{u2}(s) = 10^{-6} \frac{0.55s^2 + s + 1}{10^{-4}s^2 + 2.1 \times 10^{-3}s + 1}$$

and

$$W_{u3}(s) = 3 \times 10^{-6} \frac{4.05s^2 + s + 1}{10^{-4}s^2 + 2 \times 10^{-3}s + 1}$$

These three control weighting functions are paired with the three performance weighting functions, in the given order, in the μ -synthesis.

Table 14.4 Robust stability and robust performance for three controllers

Controller	Order	Robust stability stabmarg	Robust performance perfmarg
1	26	2.82	2.33
2	34	2.19	1.94
3	26	1.94	1.19

Table 14.5 Gain and phase margins for three controllers

Controller	Gain margin dB	Phase margin deg
1	15.1	50.9
2	11.1	44.1
3	10.9	36.6

The final choice of the appropriate performance and control weighting functions is obtained by comparing the results from the corresponding μ -designs. (Three D - K iterations are used in the first version and four in the second and third version.) The robust stability and robust performance analysis are conducted by the file `mu_hdd`, using the functions `robuststab` and `robustperf`. The results are shown in Table 14.4.

It is seen from the table that the closed-loop system achieves robust stability and robust performance for all three controllers; however, the best results, in terms of stability and performance margins, come from the first controller.

The gain and phase margins for all three design cases are found from the transfer function $G_{u\text{nom}}K$ of the nominal open-loop system by the function `margin`. The results are listed in Table 14.5 and confirm the rule that higher gain and phase margins mean better robustness.

The closed-loop transient responses are shown in Fig. 14.17, for a simulated runout of one track width ($1\text{ }\mu\text{m}$) and a torque disturbance $t_d = 0.0005\text{ N m}$.

It is seen from Fig. 14.17 that the first controller yields a transient response with the smallest undershoot (less than 30 %), but this response is the slowest one. The third controller yields the fastest response but the undershoot in this case is the largest one (greater than 50 %).

Figure 14.18 shows that as a result of the appropriate tuning of the control weighting functions all three controllers produce a control action whose amplitude is slightly less than 1.2 V.

The comparison of the transient responses to disturbance, shown in Fig. 14.19, reveals that the worst disturbance rejection is found in the first controller case and the best in the third controller case. This is a result of the tightest closed-loop bandwidth of $9 \times 10^3\text{ rad/s}$ (measured at -3 dB) in the first controller case compared with the bandwidth of $15.5 \times 10^3\text{ rad/s}$ in the third controller case (see Fig. 14.20). Note that the largest peak of the magnitude is found in the third controller case, which results in the largest undershoot of the transient response.

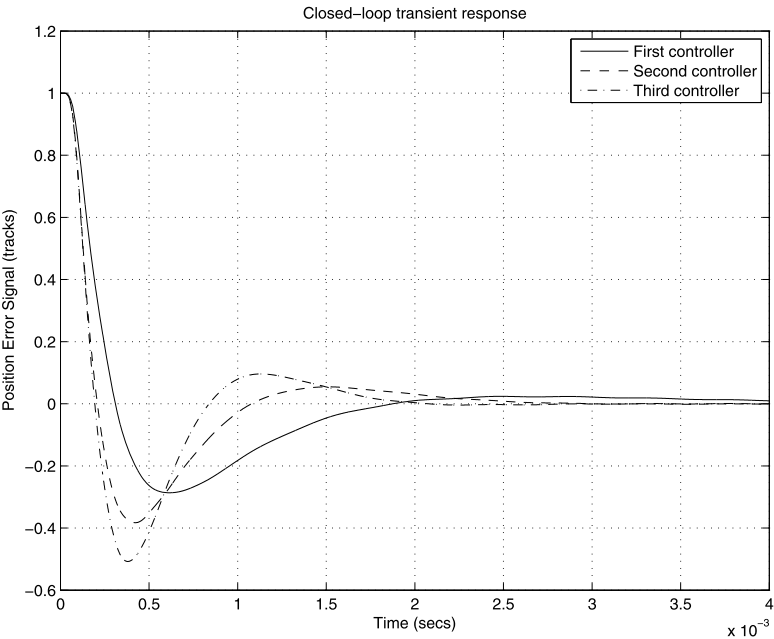


Fig. 14.17 Transient responses for three μ -controllers

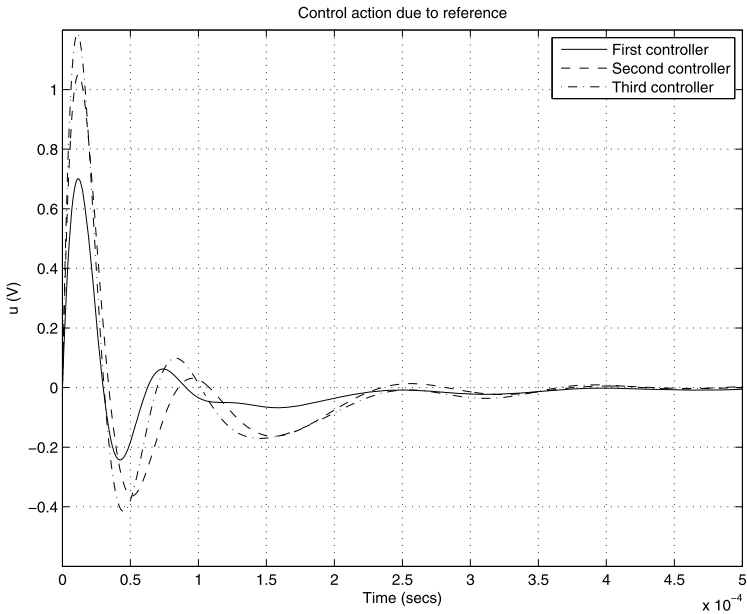


Fig. 14.18 Control actions of three μ -controllers

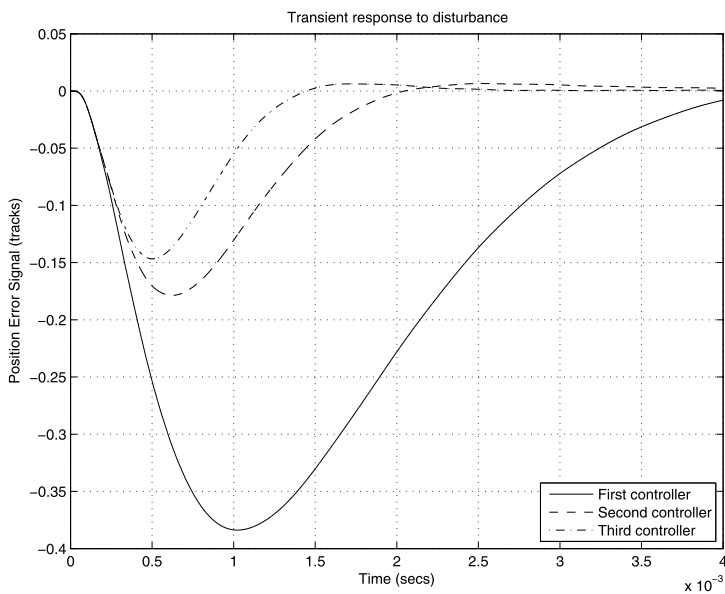


Fig. 14.19 Transient responses to disturbance of three μ -controllers

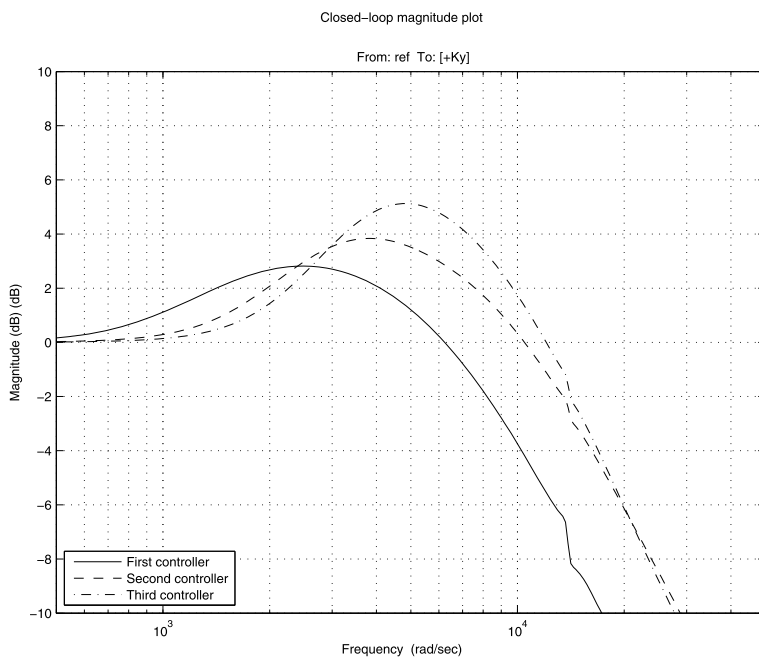
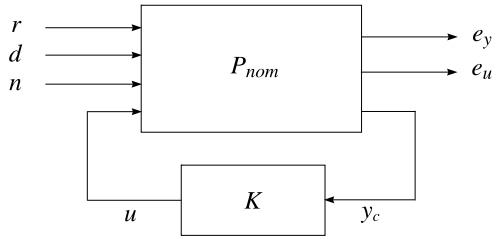


Fig. 14.20 Closed-loop magnitude plots of three μ -controllers

Fig. 14.21 Closed-loop system with \mathcal{H}_∞ controller



The results obtained from different weighting functions show that moving the frequency response of the inverse performance weighting function to the right would lead to larger closed-loop system bandwidth, and consequently, faster time-responses of the closed-loop system, though may introduce larger over(under)shoot. However, at the same time, this may reduce the robustness of the closed-loop system. Hence, one has to compromise between the different objectives in the design. In the present design case, it seems that the second controller leads to a good trade-off between the requirements in terms of transient response, disturbance rejection, and robustness. Hence, we will use the weighting functions W_{p2} and W_{u2} in both the μ -synthesis and \mathcal{H}_∞ design.

14.5.2 \mathcal{H}_∞ Design

The aim of the design in this case is to find an \mathcal{H}_∞ (sub)optimal, output controller for the interconnection shown in Fig. 14.21 in which we ignore the plant uncertainty.

The variable `hin_ic` that corresponds to the transfer function P_{nom} of the nominal open-loop system is obtained by the command line

```
hin_ic = sys_ic.Nom
```

The \mathcal{H}_∞ optimal control minimizes the ∞ -norm of $F_L(P_{\text{nom}}, K)$ with respect to the transfer function K of the controller. In the given case $F_L(P_{\text{nom}}, K)$ (as the transfer function matrix in (14.2)) is the nominal closed-loop transfer function matrix from the reference, disturbance and noise signals (the variables `ref`, `dist` and `noise`) to the weighted outputs `e_y` and `e_u`. The design is conducted using the M-file `hinf_hdd.m`, which computes a (sub)optimal \mathcal{H}_∞ control law for a given open-loop system. The value of γ is chosen 10 % higher than the minimum possible value. The controller such obtained is of 18th order.

14.5.3 \mathcal{H}_∞ Loop-Shaping Design

The design of a robust control for the Disk Drive System can be successfully accomplished using the \mathcal{H}_∞ loop-shaping design procedure (LSDP) as well, as described

in this subsection. Note that in the case of \mathcal{H}_∞ LSDP, we do not use the performance weighting function implemented in the cases of μ and \mathcal{H}_∞ designs. Instead, we use a prefilter W_1 and a postfilter W_2 in order to shape appropriately the frequency response of the augmented, open-loop transfer function $W_2 G W_1$ (see Chap. 5).

In the present case we choose a prefilter with transfer function

$$W_1 = 4 \frac{0.05s + 1}{s}$$

The gain of 4 is chosen to ensure a steady-state error, due to the disturbance, of less than 10 % of the track width. Larger gain leads to smaller steady-state errors but possibly worse transient response. The postfilter is taken simply as $W_2 = 1$.

The magnitude plots of the original and shaped systems are shown in Fig. 14.22. The design of the \mathcal{H}_∞ LSDP controller uses the M-file `lsh_hdd` that implements the function `ncfsyn`. The controller obtained is of order 12.

14.6 Comparison of Designed Controllers

The comparison of the closed-loop system with μ , \mathcal{H}_∞ , and \mathcal{H}_∞ LSDP controllers begins with the robust stability and performance analysis.

The robust stability is tested on the closed-loop transfer function matrix, the computation of the structured singular value $\mu_\Delta(\cdot)$ being done by the command `mussv`. The uncertainty matrix Δ , which is necessary in the computation of $\mu_\Delta(\cdot)$, is pulled out from the closed-loop transfer matrix by the command `lftdata` and has dimension 15×15 (the uncertainties in the resonant modes participate twice). To achieve robust stability it is necessary that the μ -values are less than 1 over the frequency range. The robust stability test is done by the file `rbs_hdd`.

In Fig. 14.23 we compare the upper bounds of the structured singular values, for the robust stability analysis, of the closed-loop systems with the three controllers (μ , \mathcal{H}_∞ , and \mathcal{H}_∞ LSDP).

It is seen from the figure that all three closed-loop systems achieve the robust stability. The best robustness is obtained by the μ -controller.

The comparison of the nominal performance for the three controllers, in Fig. 14.24, shows that the performance in the \mathcal{H}_∞ LSDP controller case over the low-frequency range is much worse than that in the other two cases. This is a consequence of the fact that the performance specifications used in the μ -design and \mathcal{H}_∞ design are not explicitly adopted in the design of the \mathcal{H}_∞ LSDP controller. The larger magnitude over the low frequencies leads to an expectation of larger steady-state errors.

The robust performance of the closed-loop system is studied also with the aid of the μ -analysis using the function `mussv`. In this case the uncertainty block Δ_P , which is used in the computation of the structured singular value, has dimension 18×17 and consists of the matrix Δ plus 3×2 complex block that corresponds to a fictitious uncertainty and connects the inputs and outputs of the closed-loop

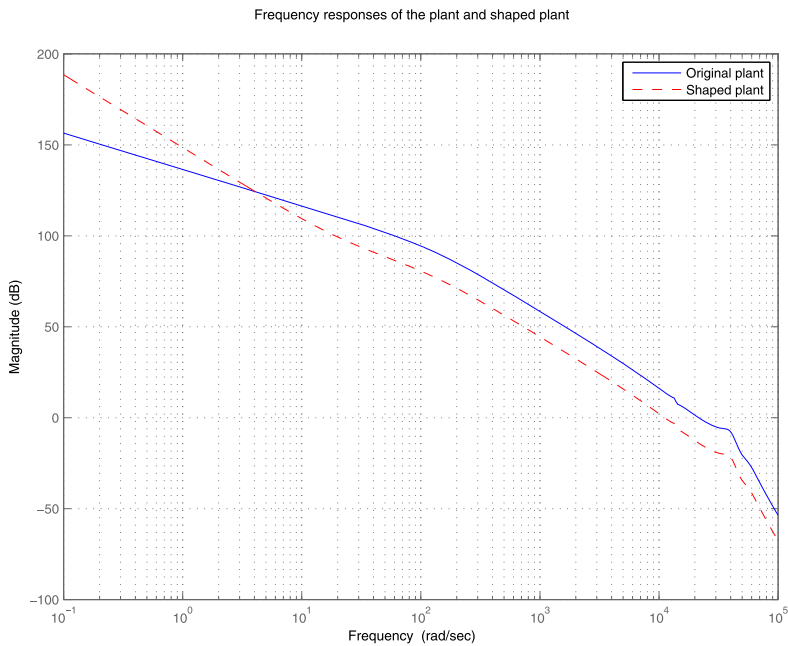


Fig. 14.22 Magnitudes plots of the original and shaped systems

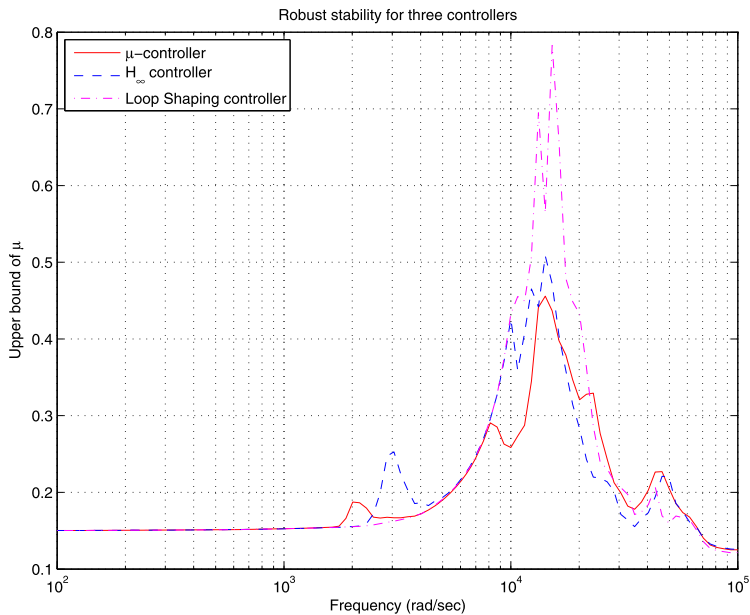


Fig. 14.23 Robust stability of the closed-loop systems

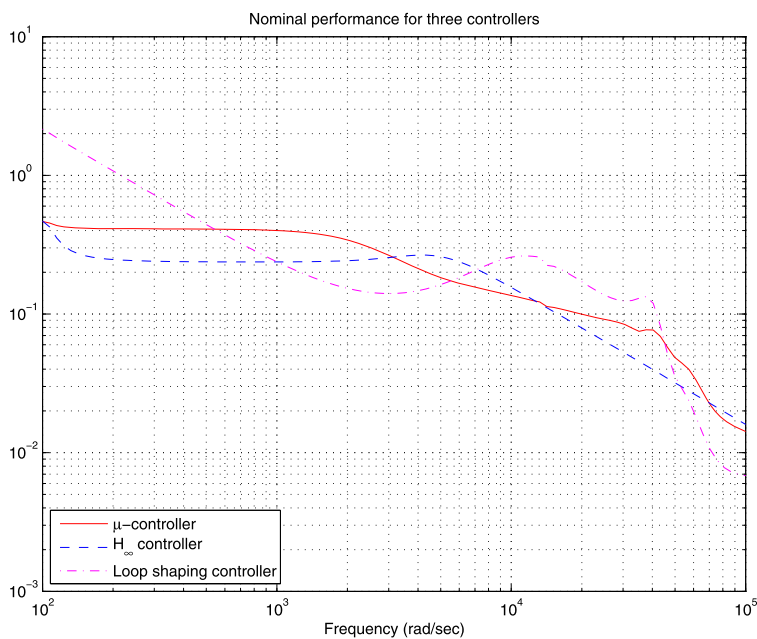


Fig. 14.24 Nominal performance of the closed-loop systems

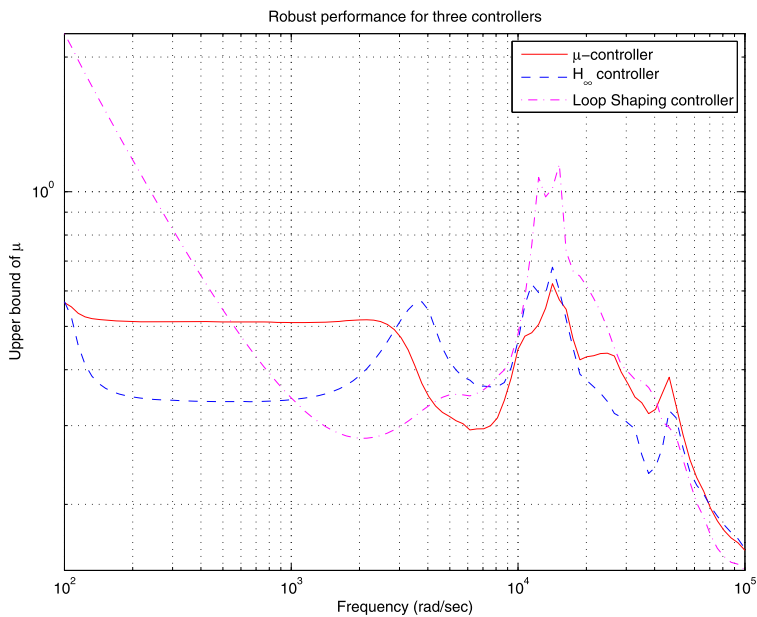


Fig. 14.25 Robust performance of the closed-loop systems

system (see Fig. 14.15). The robust performance (with respect to the uncertainty and performance weighting functions) is achieved if and only if for each frequency $\mu_{\Delta_P}(\cdot)$, computed for the closed-loop frequency response, is less than 1. The robust performance test is done by the file `rbp_hdd`.

The robust performance test for all controllers is shown in Fig. 14.25. Again, the \mathcal{H}_∞ LSDP controller shows large μ -values over the low-frequency range.

The robust stability and robust performance analysis also shows that the worse results may possibly occur over frequencies around the resonant frequencies.

The Bode plots of the closed-loop systems with three controllers are shown in Fig. 14.26. It is seen that the system with the \mathcal{H}_∞ LSDP controller has the largest bandwidth, which may lead to the fast transient response. A larger bandwidth, however, may also lead to a larger effect of noises and resonances.

The plot of the output sensitivity to disturbances (Fig. 14.27) shows that in the low-frequency range the influence of the disturbance on the system output in the case of the \mathcal{H}_∞ LSDP controller is the largest. Better disturbance attenuation for this controller may be achieved by choosing higher gain in the prefilter. This will lead, however, to greater overshoot in the transient response. The sensitivity to disturbance in the cases of μ and \mathcal{H}_∞ controllers reaches a maximum value in the frequency range from 10^3 rad/s to 10^4 rad/s, which is inside of the closed-loop bandwidth. This means that the closed-loop system will be susceptible to disturbances over that frequency range. It is interesting to notice that the sensitivity of the \mathcal{H}_∞ LSDP controller over the same range is 8 dB lower.

The output sensitivity to noise is shown in Fig. 14.28. (Note that the sensitivity is with respect to the unit-bounded noise, the input of the noise shaping filter.) The lowest sensitivity to noise has been achieved by the system with the μ -controller and the largest sensitivity by the system with the \mathcal{H}_∞ LSDP controller.

The Bode plots of the three controllers are compared in Fig. 14.29. It is seen that the \mathcal{H}_∞ LSDP controller has a very low gain in the range from 40 rad/s to 2000 rad/s, which is the reason for the weak attenuation of the disturbances over that range.

The transient responses of the closed-loop systems for the nominal parameter values are obtained by using the file `clp_hdd`. In Fig. 14.30 we show the transient responses of the closed-loop systems to a reference signal, equivalent to one track. While for the \mathcal{H}_∞ and \mathcal{H}_∞ LSDP controllers the undershoot is about 60 %, it is about 38 % for the μ -controller. The settling time for the \mathcal{H}_∞ LSDP, μ and \mathcal{H}_∞ controllers is 0.8 ms, 1 ms and 1.5 ms, respectively.

The control actions of the three controllers are shown in Fig. 14.31. For all controllers the control signal amplitude does not exceed 1.2 V, as required.

In Fig. 14.32 we show the system response to a step torque disturbance $t_d = 0.0005$ N m (equivalent to a force of 9.8×10^{-3} N applied to the disk head assembly). The transient error for the \mathcal{H}_∞ LSDP controller has the smallest undershoot (10.5 %) but has a nonzero steady-state error. This reveals that in the given case the LSDP controller does not have integrating action with respect to the disturbance. The transient error for μ and \mathcal{H}_∞ controllers is less than 18 % of the track width and the steady-state error is practically equal to zero.

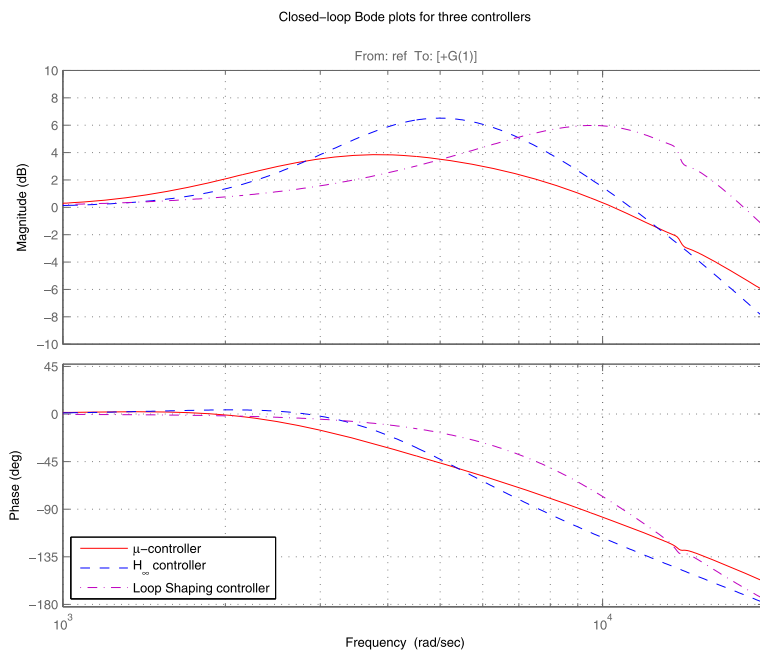


Fig. 14.26 Bode plots of the closed-loop systems

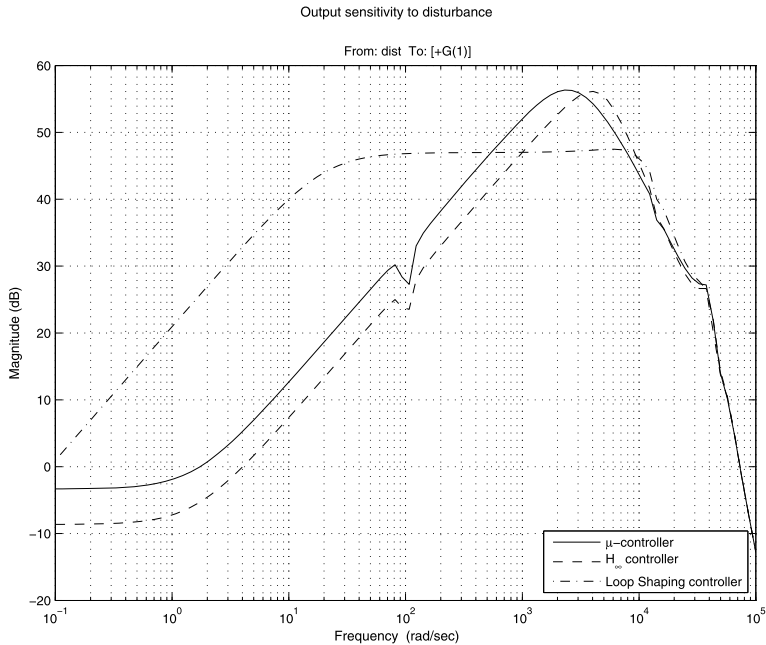


Fig. 14.27 Output sensitivity to disturbance

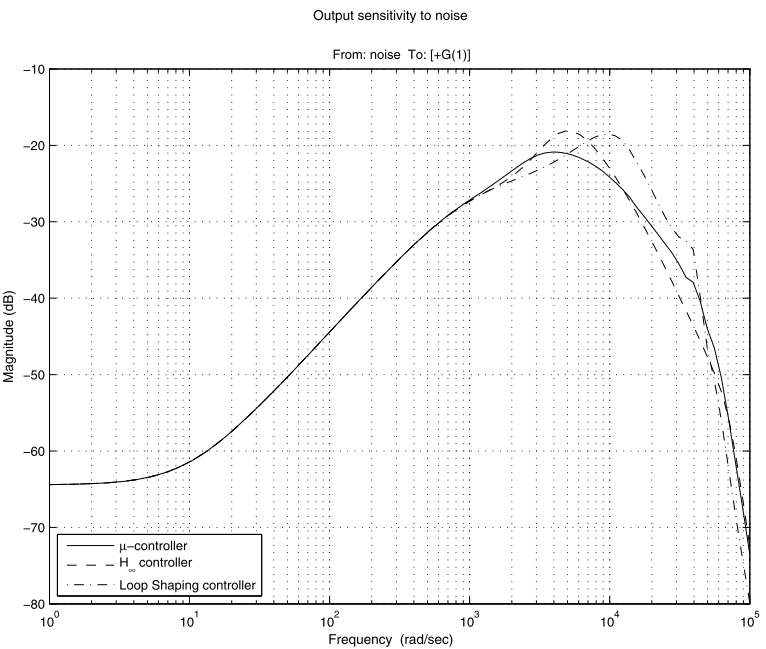


Fig. 14.28 Output sensitivity to noise

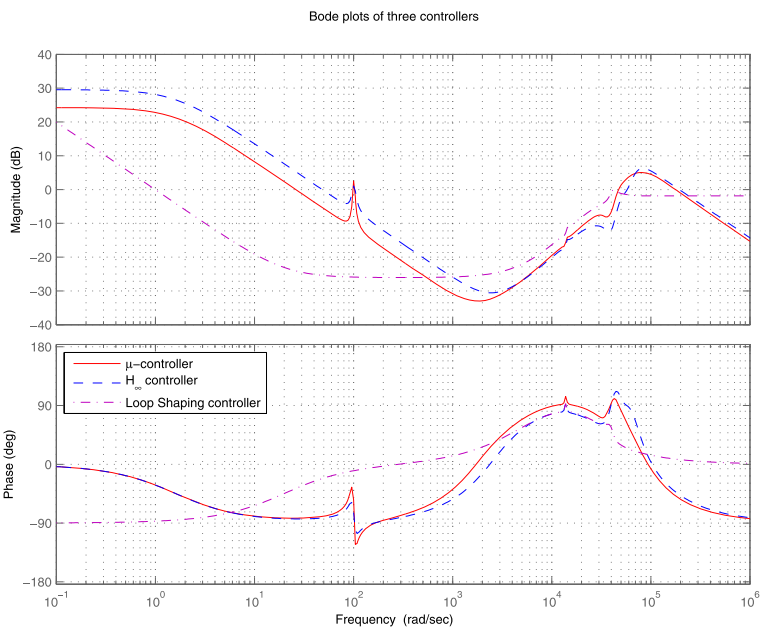


Fig. 14.29 Bode plots of three controllers

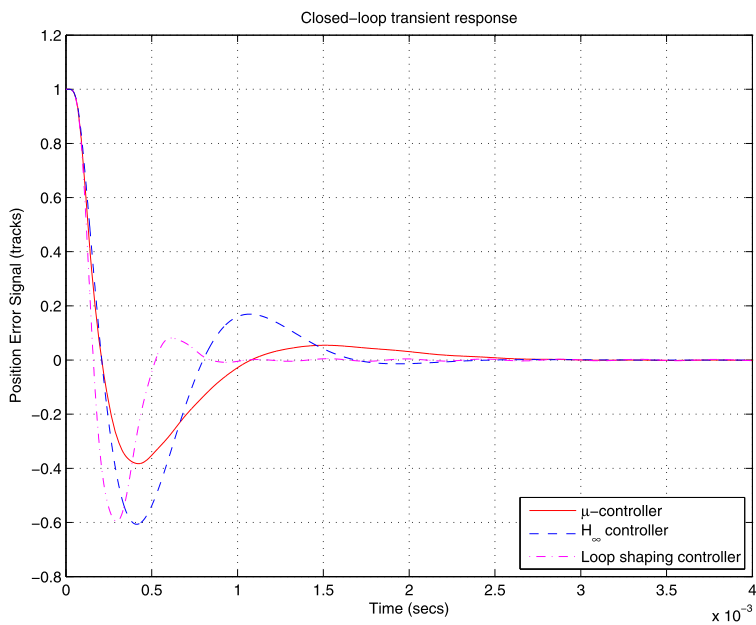


Fig. 14.30 Transient responses with three controllers

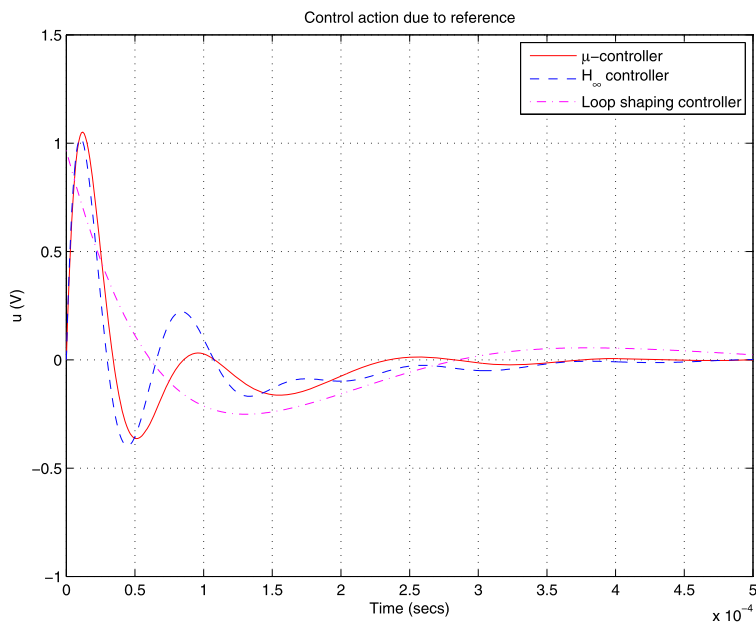


Fig. 14.31 Control actions of three controllers

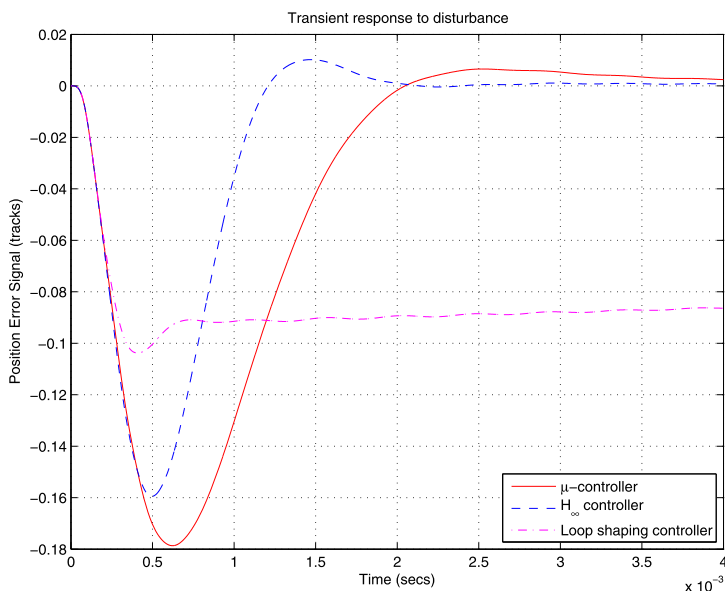


Fig. 14.32 Transient disturbance responses of three controllers

The output response to the position-sensing noise is simulated for a noise signal with an amplitude that does not exceed 60 mV (5 % of the track width). This signal is obtained at the output of the noise shaping filter whose input is chosen as a sequence of uniformly distributed random numbers in the interval $[-1, 1]$. In the case of the μ -controller the output due to noise is less than 0.6 % of the track width. The largest output to the noise (1 % of the track width) is seen in the \mathcal{H}_∞ LSDP controller case due to the largest closed-loop bandwidth of this controller.

The comparison of the robust stability and robust performance for the three controllers, as well as the comparison of the corresponding frequency and transient responses shows that it is reasonable to conclude that the μ controller is preferable.

Consider now the properties of the closed-loop system with μ -controller for variations of the plant parameters.

In Fig. 14.33 we show the magnitude closed-loop plots for random plant parameters. The magnitude plot given with continuous line corresponds to the worse gain when the parameter change (this gain is determined by the function `wcgain`).

In Fig. 14.34 we show the output sensitivity and in Fig. 14.35 the sensitivity to noise for 20 random combinations of the uncertain plant parameters.

14.7 Controller-Order Reduction

The controller obtained by the μ -synthesis is initially of 34th order. It is useful to reduce as much as possible the controller order, which will simplify the implementation and increase the reliability. To do this we use the function `reduce`. The

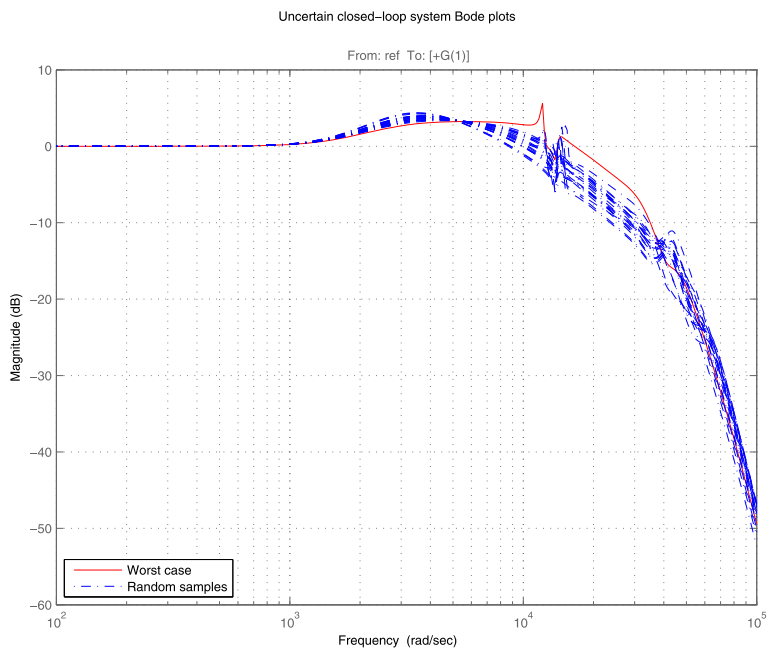


Fig. 14.33 Magnitude closed-loop plots for random plant parameters

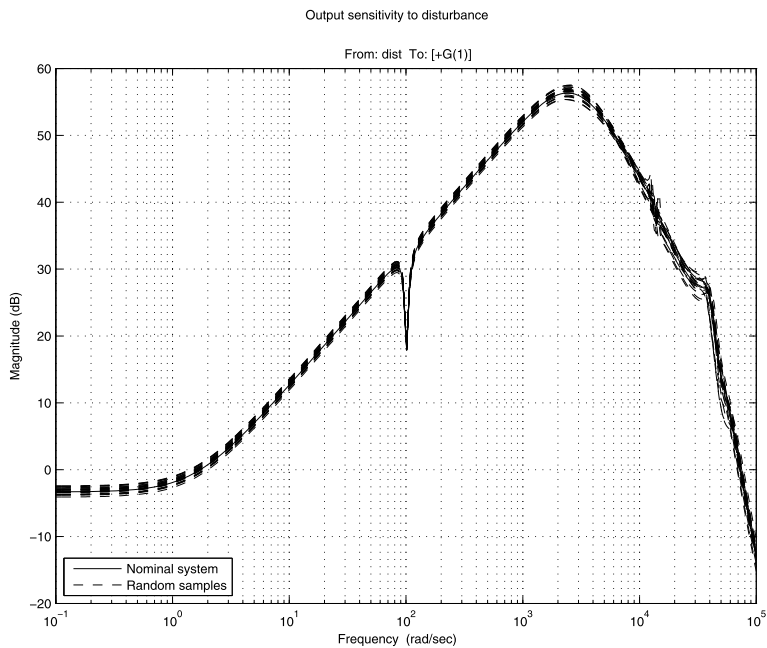


Fig. 14.34 Output disturbance sensitivity for random plant parameters

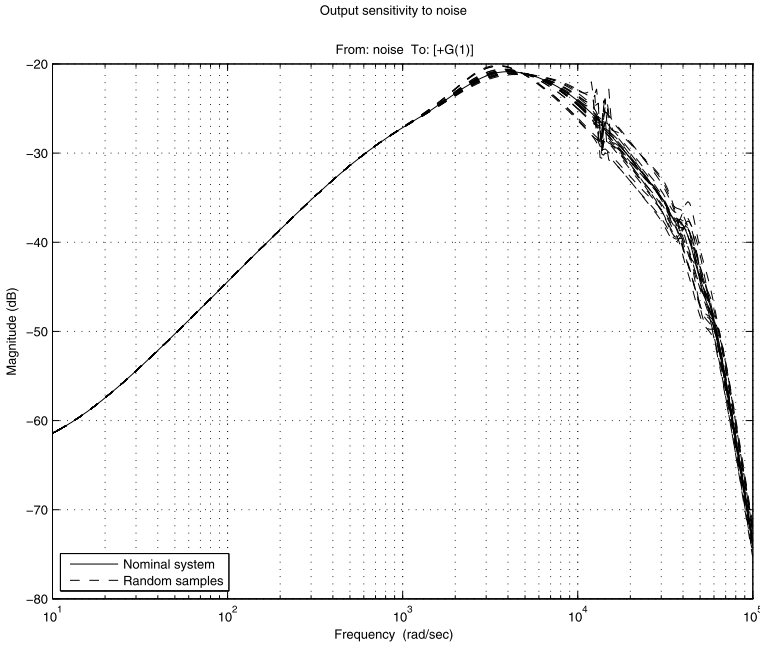


Fig. 14.35 Output sensitivity to noise for random plant parameters

experience with different controllers shows that the order may be reduced to 12. Further reduction of the controller order leads to deterioration of the closed-loop performance.

In Fig. 14.36 we compare the Bode plots of the full order and reduced-order controllers. The corresponding plots practically coincide with each other, which implies similar performances of the closed-loop systems.

In Fig. 14.37 we show the transient responses of the closed-loop system with the reduced order μ -controller for 30 random combinations of the uncertain parameters and in Figs. 14.38 and 14.39 we show the control action and the transient disturbance responses, respectively, for the same values of the uncertain parameters. The transient responses are obtained by the M-file `mcs_hdd`.

14.8 Design of Discrete-Time Controller

In general, there are two approaches to designing a discrete-time servo controller.

The first approach is to sample the already designed continuous-time controller at a given sampling frequency $f_s = 1/T_s$. This may be accomplished by the M-file `dcl_hdd.m` that utilizes the function `c2d`. The resultant sampled-data, closed-loop system is simulated by using the function `sdlsim`. This approach gives satisfactory results for a sufficiently high sampling frequency (say, 100 kHz in the given case).

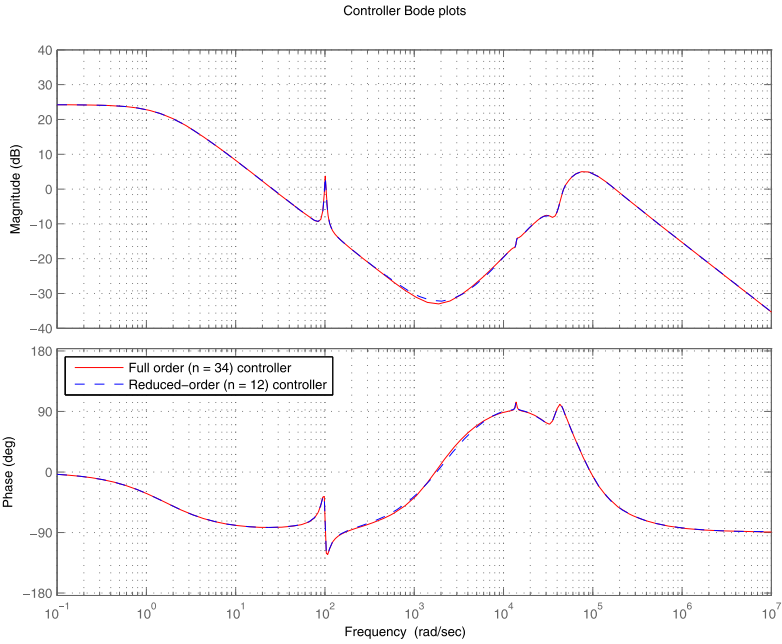


Fig. 14.36 Bode plots of full- and reduced-order μ -controllers

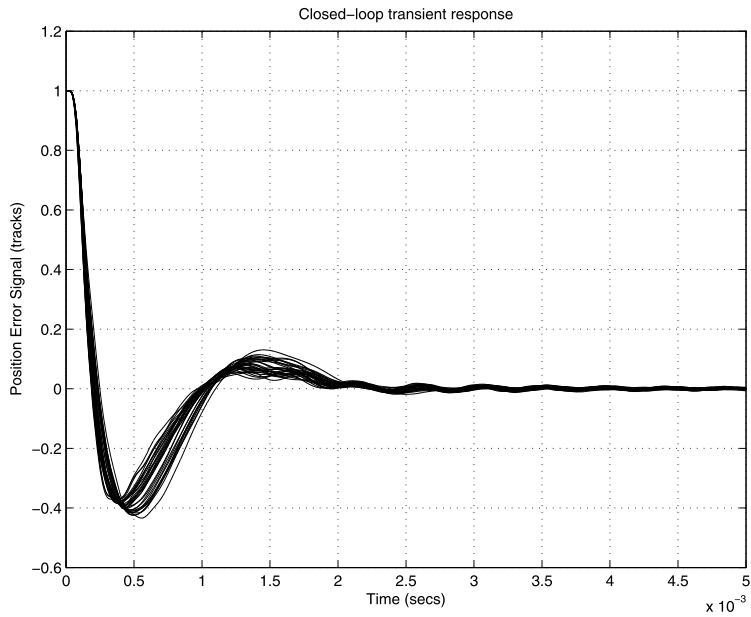


Fig. 14.37 Transient responses with μ -controller for random plant parameters

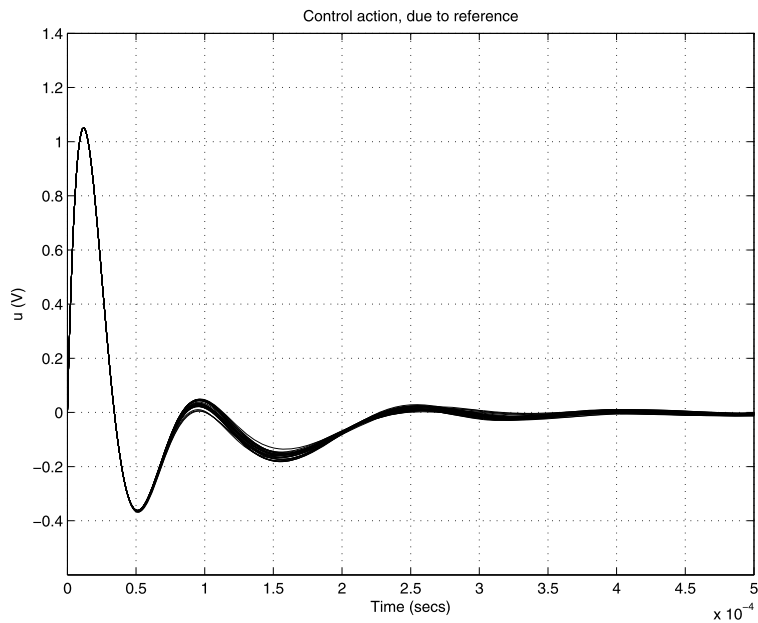


Fig. 14.38 Control action of the μ -controller for random plant parameters

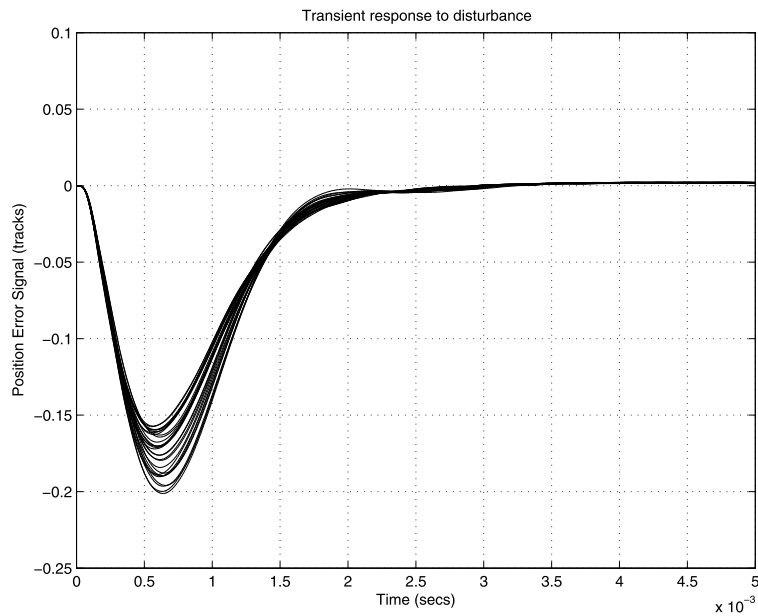


Fig. 14.39 Transient disturbance responses with μ -controller for random plant parameters

The second method is to sample the continuous-time, open-loop system (including the weighting filters) and to design directly a discrete-time controller by using \mathcal{H}_∞ optimization (implementing the function `hinfsyn`) or μ -synthesis (implementing the function `dksyn`).

The choice of the sampling frequency in the discrete-time case has a strong influence on the closed-loop system performance. A low sampling frequency limits the system bandwidth and would deteriorate the transient performance such as the disturbance rejection. On the other hand, the increase of the sampling frequency would complicate the controller implementation and raise the cost of the HDD.

Later we consider the μ -synthesis of the discrete-time controller for two sampling rates: 24 kHz and 36 kHz. In both cases we use the same performance weighting function

$$W_{p2}(s) = 10^{-4} \frac{s^2 + 4 \times 10^5 s + 2.5 \times 10^9}{s^2 + 3.9 \times 10^5 s + 6.25 \times 10^5}$$

utilized already in the continuous-time design. Depending on the sample rate we use two different control weighting functions

$$W_{u1}(s) = 10^{-7} \frac{4s^2 + 2s + 1}{2 \times 10^{-3}s^2 + 2 \times 10^{-3}s + 1}$$

(for $f_s = 24$ kHz) and

$$W_{u2}(s) = 10^{-7} \frac{1.04s^2 + 2s + 1}{7.5 \times 10^{-5}s^2 + 2 \times 10^{-3}s + 1}$$

(for $f_s = 36$ kHz). This allows, in both cases, to obtain control signals that do not exceed 1.2 V.

The noise shaping filter is the same as in the continuous-time case.

The sampling of the extended open-loop system for the given sampling rate is conducted by the M-file `d1p_hdd.m`.

In Fig. 14.40 we show the Bode plots of the continuous time plant and the sampled with $f_s = 36$ kHz plant.

The discrete-time μ -synthesis is accomplished by the file `dms_hdd.m`. As in the continuous-time μ -synthesis, only the rigid-body uncertain parameters are taken into account.

The results from the μ -synthesis for $f_s = 24$ kHz show that for the chosen weighting functions the closed-loop system almost achieves robust performance at the sixth D - K iteration ($\mu_{\max} = 1.125$) but the closed-loop response is relatively slow and the undershoot is large (60 %). To obtain better results it is necessary to increase the sampling frequency.

We now present in more detail the results from the μ -synthesis at $f_s = 36$ kHz. In this case an appropriate controller is obtained after six D - K iterations and the maximum robust performance μ_{\max} achieved is almost equal to 1. In Figs. 14.41 and 14.42, we show the μ -plots, obtained by the M-file `dmu_hdd.m`, for robust

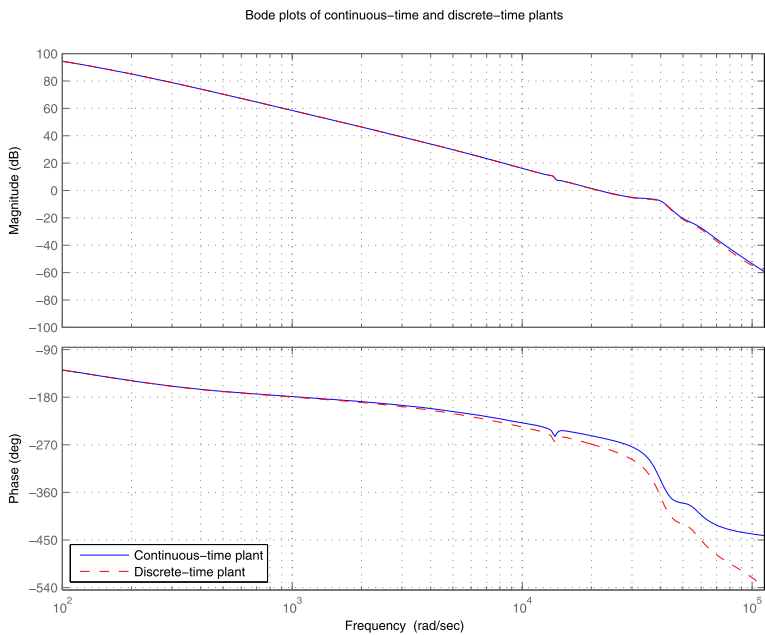


Fig. 14.40 Bode plots of the continuous time plant and the sampled with $f_s = 36$ kHz plant

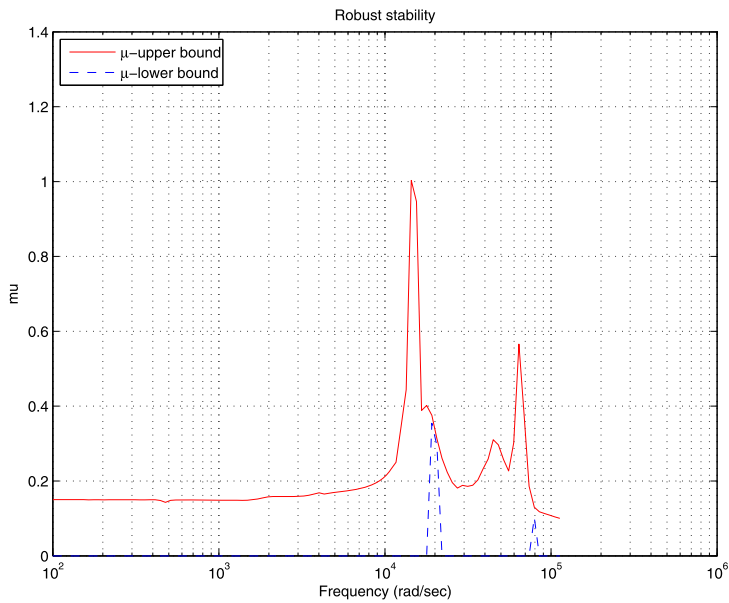


Fig. 14.41 Robust stability of $f_s = 36$ kHz design

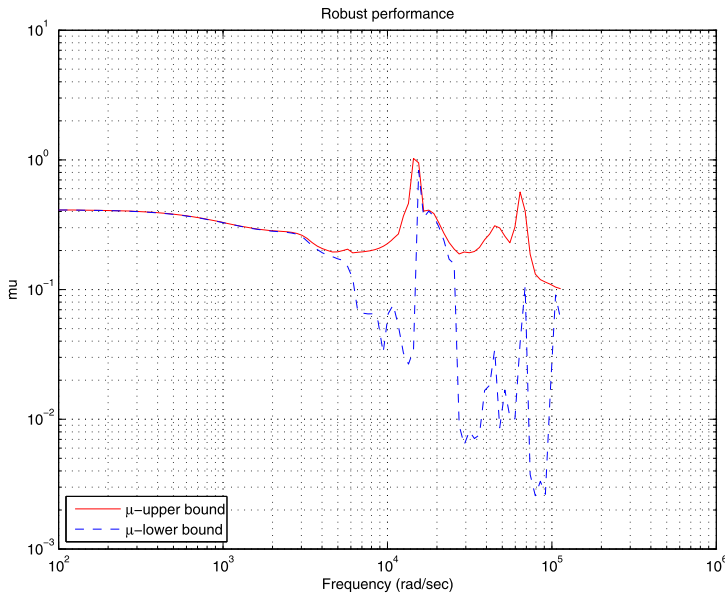


Fig. 14.42 Robust performance of $f_s = 36$ kHz design

stability analysis and robust performance analysis, respectively. In both plots the worst results are seen around the second resonant frequency of 2.2 kHz.

The transient responses of the closed-loop system for 30 random combinations of the uncertain parameters are obtained by the file `dsl_hdd.m`, which implements the function `sdl_sim`. The function `sdl_sim` also computes the control action signal obtained at the output of the digital-to-analog converter. The closed-loop transient response is shown in Fig. 14.43 and the corresponding control action in Fig. 14.44. The undershoot for all parameter values is less than 50 % and the maximum magnitude of the control signal is 1 V.

The transient response to disturbance is shown in Fig. 14.45. Overall, the results obtained are almost as good as the results obtained with the continuous-time, μ -controller.

14.9 Nonlinear System Simulation

In order to obtain a realistic idea about the behavior of the designed system, the nonlinear, closed-loop servo system is simulated by using Simulink®. For this aim, two models are developed, namely `c_hdd.mdl` for the continuous-time system and `d_hdd.mdl` for the sampled-data system. In the simulation we take into account the amplifier saturation, which was neglected in the design so far. Both models allow us to simulate the closed-loop system for different reference, disturbance and noise signals and for different values of the uncertain parameters.

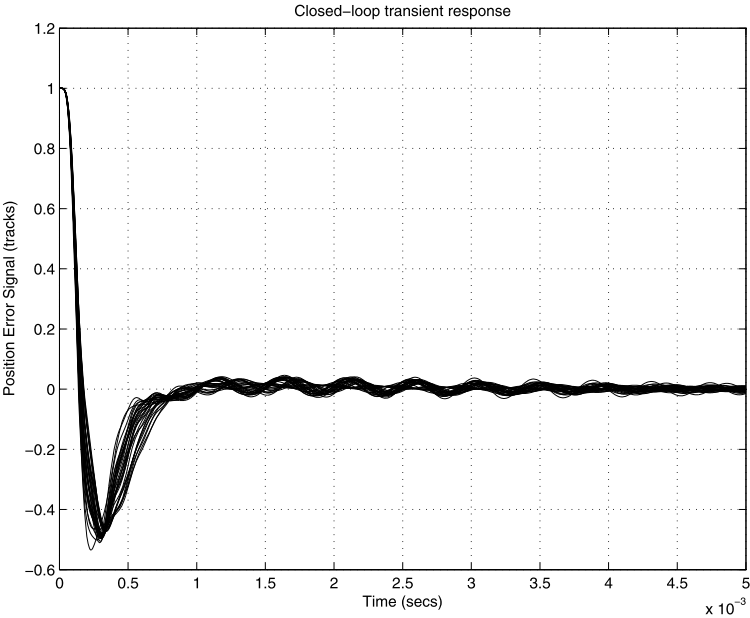


Fig. 14.43 Transient response of $f_s = 36$ kHz design

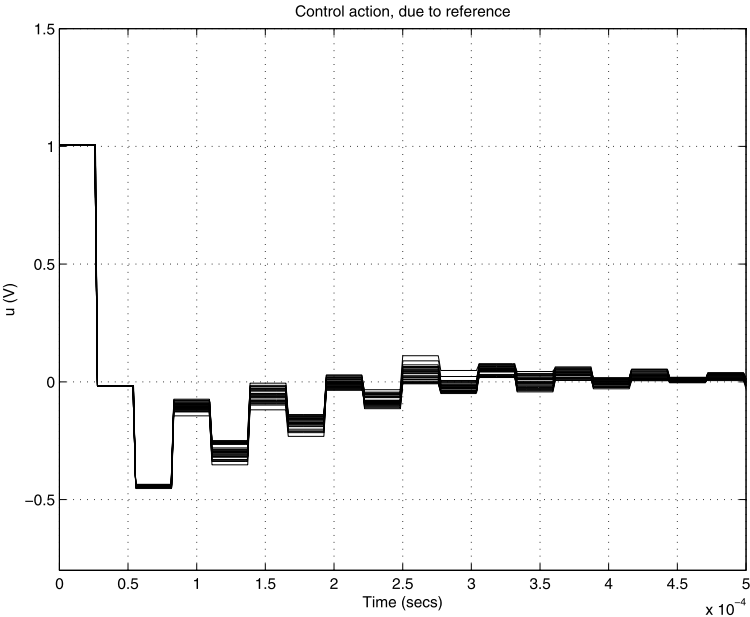


Fig. 14.44 Control action of $f_s = 36$ kHz design

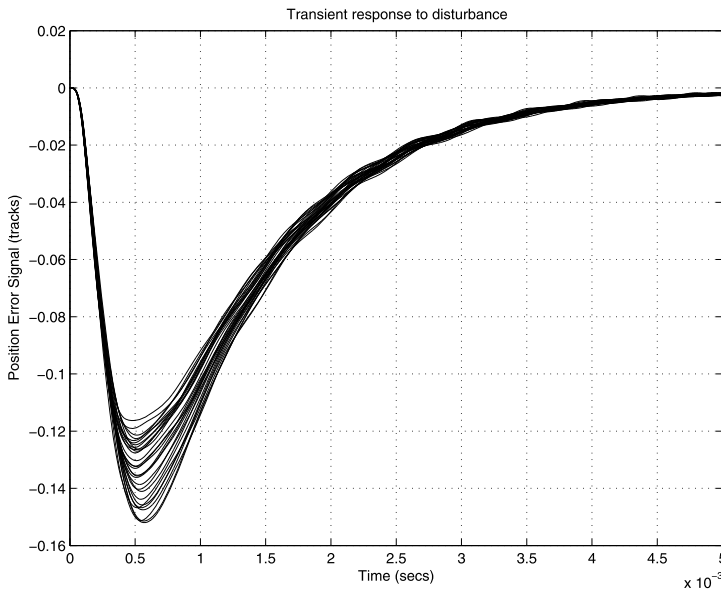


Fig. 14.45 Transient response to disturbance of $f_s = 36$ kHz design

Before simulating the continuous-time system it is necessary to assign the model parameters by using the M-file `init_c_hdd.m`.

The sampled-data model involves the discrete-time controller, 16-bit analogue-to-digital converter with maximum input voltage 2.5 V and 16-bit digital-to-analogue converter with maximum output voltage 10 V. It is assumed that the discrete-time controller is implemented on a DSP with word length of 64 bits. These parameters are set prior to the simulation by using the M-file `init_d_hdd.m`. It is assumed that the control action calculation requires one sampling period T_s .

In Fig. 14.46 we show the Simulink[®] model `d_hdd.mdl` of the nonlinear, sampled-data, closed-loop system.

As in the linear case, the transient responses of the nonlinear closed-loop system are obtained for a simulated runout of one track width (1 μm) and torque disturbance $t_d = 0.0005$ N m.

In Fig. 14.47 and in Fig. 14.48 we compare the results from the simulation of the continuous-time and discrete-time nonlinear systems. The continuous-time controller is the reduced-order μ -controller in Sect. 14.7 and the discrete-time controller is the controller designed at the sampling frequency of 36 kHz.

The transient responses of the nonlinear system are close to the corresponding responses of the linear system due to the small input signals (amplitude less than 1.2 V).

It should be mentioned that the controllers designed are appropriate for small reference signals (equivalent to one track). For larger references the amplifier saturates and it is necessary to implement an appropriate seeking algorithm.

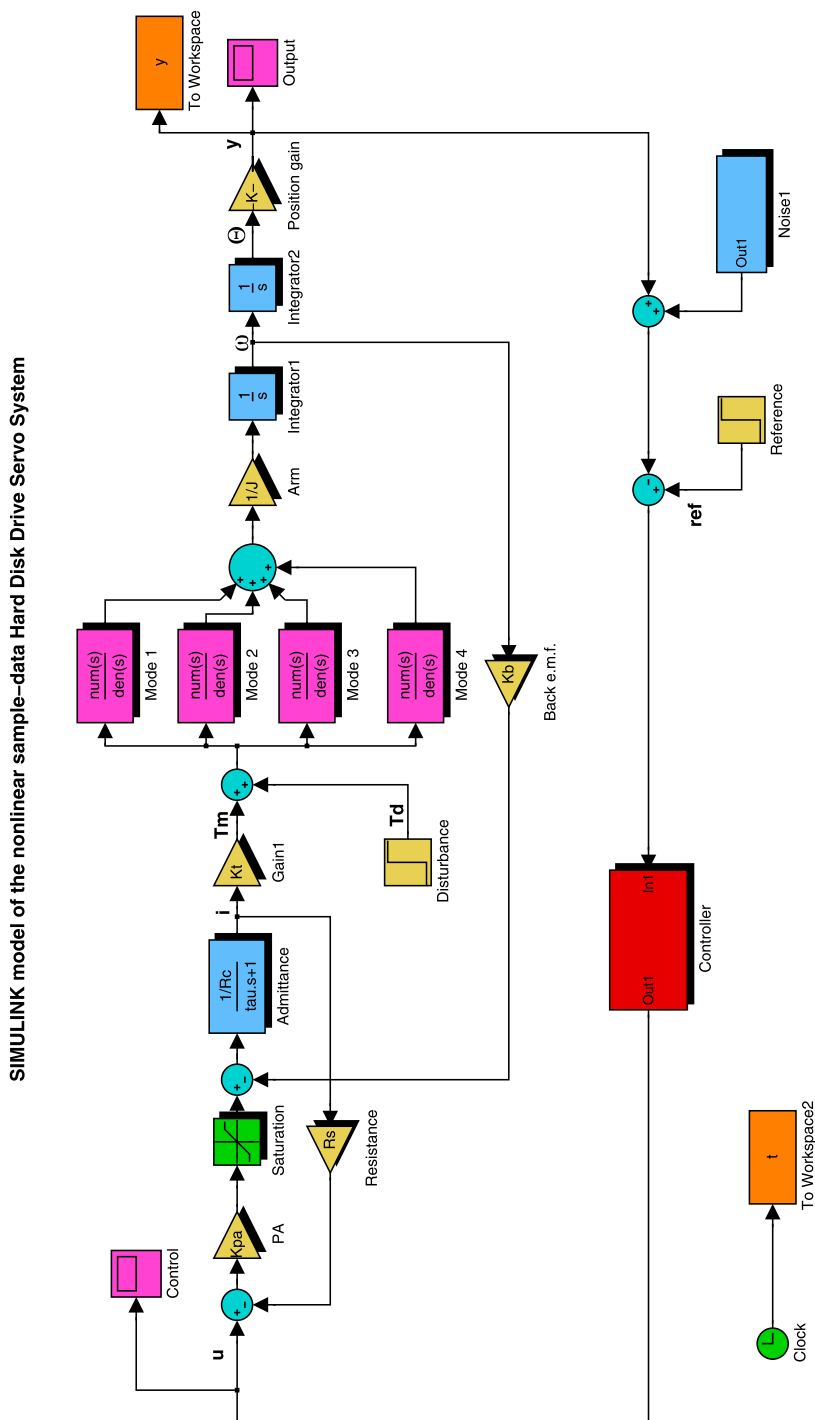


Fig. 14.46 Simulation model of the nonlinear sampled-data system

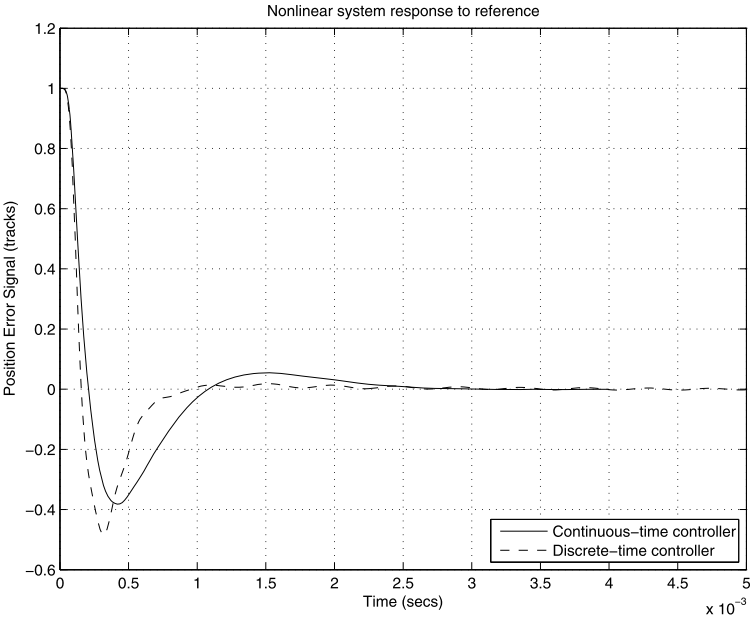


Fig. 14.47 Transient responses of the nonlinear systems

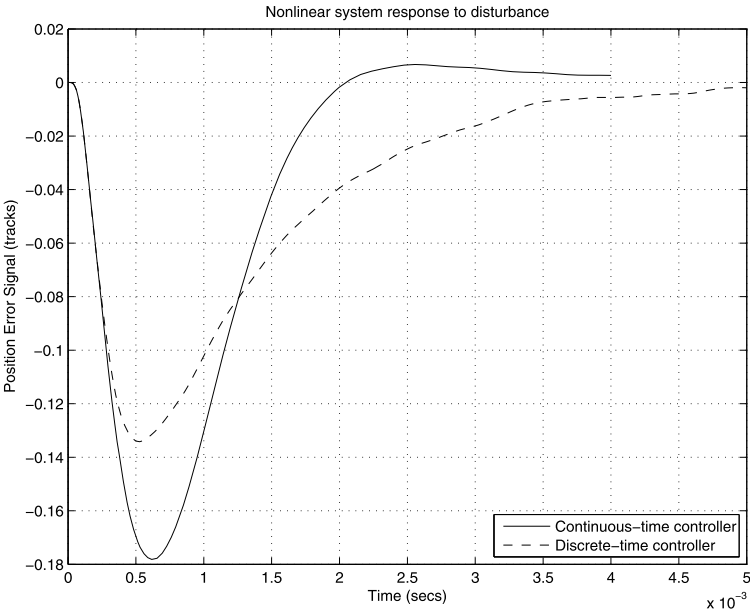


Fig. 14.48 Disturbance responses of the nonlinear systems

14.10 Conclusions

The experience gained in the design of the HDD servo controllers makes it possible to derive the following conclusions:

- The implementation of designed μ , \mathcal{H}_∞ , and \mathcal{H}_∞ LSDP controllers in the HDD servo system gives satisfactory results with respect to robustness and performance. All three controllers ensure robust stability of the closed-loop system. The best robust performance is achieved by using the μ and \mathcal{H}_∞ controllers. The implementation of the \mathcal{H}_∞ LSDP controller gives the fastest transient response and the corresponding design is less complicated. This controller, however, leads to the worst performance in the low-frequency range, which results in a large steady-state error. In the given case the best trade-off between the robustness and transient response requirements is achieved by using the μ -controller that is, to some extent, due to the specially chosen weighting functions.
- The number of original uncertain parameters is very large (more than 25 in the given case). This complicates the design and produces heavy computation demands. This is why it is necessary to investigate the parameter importance with respect to the robustness and performance in order to reduce their number to an acceptable value. However, in the evaluation of the design, it is better to take into account all the possible uncertainties to ensure a satisfactory design in a real case.
- In the μ -synthesis, the order of the resultant controller depends on the order of the plant, of the weighting functions, and of the scaling diagonal elements approximations. The designed controllers are usually of high orders, which complicates the implementation of the controller. Hence, an order reduction should usually be considered right after the controller design. Most controllers used in the HDD designs are of order between 8 and 15.
- Good disturbance attenuation requires sufficiently large closed-loop bandwidth. This may, however, lead to difficulties in achieving robust stability and robust performance in the presence of resonant modes. Some resonances whose frequencies are much higher than the closed-loop bandwidth and thus seem innocent may even actually destroy the robust stability of the system. In such cases, it is necessary to increase the damping of these modes by using techniques of passive/active damping.
- The presence of resonant modes may require sufficiently high sampling rates in the case of using a discrete-time controller.
- It is important to stress that better results, with respect to the transient response (overshoot and settling time), are difficult to obtain for the current plant parameters. If higher performance demands are required, it is necessary to change the HDD parameters, for instance to increase the VCM torque constant.

14.11 Notes and References

The history of the Hard Disk Drive control is presented in the fascinating papers of Abramovitch and Franklin [2, 3]. An excellent survey on similarities and contrasts

in the magnetic and optical disk controls is given in [1]. In [87] one may find an attractive description of the HDD construction and functioning.

The book of Chen et al. [20] is entirely devoted to HDD servo systems and contains rich information related to the design of such systems. A tutorial on HDD control can be found in [117].

A detailed model of the HDD servo system, which has very much influenced the model used in this chapter, is presented in the book of Franklin et al. [46, Chap. 14]. Below 100 Hz the rotary actuator dynamics is affected by pivot bearing nonlinearity, which is known under the name “stick–slip”. It has a strong effect particularly in the case of small disk drives with lower actuator inertia. Analysis and simulation of this phenomenon are presented in [4, 174].

An important step in the design of HDD servo systems is the reliable estimation of the various disturbances and noises acting on the system. Methods for such estimations are described in [5, 35, 69].

The harmonic compensation used to reduce RRO is considered in [21, 83, 163].

The track-seeking and track-following modes require different control algorithms. Track-seeking algorithms are described in [46, 135]. The switching of control mode from track seeking to track following should be smooth so that the residual vibration of the read/write head suspension is minimal. There are several control algorithms that work for both track-seeking and following, see for instance [71, 79]. Such algorithms utilize 2-degree-of-freedom (2DOF) controllers in which case the track seeking is accomplished by using a feedforward controller along with a reference trajectory.

Apart from the track seeking and track following the HDD also contains a spindle velocity control loop. The purpose of this loop is to control the air flow over the disk in order to guarantee the appropriate flying height of the read/write head. This is a low-frequency control loop and its design does not represent a serious difficulty.

Further expansion of the closed-loop bandwidth of the HDD control system may be achieved by using the so-called dual-stage servos that consist of a low-bandwidth coarse actuator (the usual VCM) and a high bandwidth fine actuator. The fine actuator has a small stroke and may be implemented as a *piezoelectric transducer* (PZT) [40]. The design of dual-stage servos is considered in [29, 74, 86].

Other important aspects of the analysis and design of HDD servo systems are presented in [59, 77, 94, 95, 110, 183], to name a few.

Chapter 15

A Triple Inverted Pendulum Control System Design

Robust design of a triple inverted pendulum control system is discussed in this chapter.

The triple inverted pendulum is an interesting control system that resembles many features found in, for instance, walking robots and flexible space structures, and other industrial applications. This kind of pendulum system is difficult to control due to the inherent instability and nonlinear behavior. Some of the pendulum parameters may not be known exactly in practice, which influences significantly the system dynamics.

In the design of a robust control system for triple inverted pendulums it is conventionally assumed that the system is affected by unstructured uncertainties and thus the robust properties of the closed-loop system could be achieved by using an \mathcal{H}_∞ controller. In real cases, however, the uncertainties of such a pendulum system would be more reasonably considered to have some structures. For instance, because the moments of inertia and the friction coefficients are difficult to estimate precisely, it makes sense to assume unknown deviations in those parameters. Also, we would like to design the closed-loop control system to be more “robust” against those parameters that have a “larger” or more serious influence upon the system behavior. For instance, the viscous friction in the joints may destroy the controllability of the linearized model. Hence, it would be important to treat uncertainties in such parameters individually rather than aggregate them in an overall, unstructured uncertainty of the system dynamics. Consequently, it may be more suitable to apply the μ -synthesis technique that may lead to a less-conservative design to meet tighter design specifications.

In the pendulum control-system design we first model the uncertainties as a mixed type that consists of complex uncertainties in the actuators, real uncertainties in the moments of inertia and in the viscous friction coefficients. A 2-degree-of-freedom (2DOF) design framework is adopted. Both \mathcal{H}_∞ suboptimal and μ -controllers are designed. The \mathcal{H}_∞ controller shows better transient and disturbance responses but does not ensure robust stability nor robust performance. The μ -controller achieves both robust stability and robust performance, however, at the price of poorer time responses. The μ -controller designed is initially of quite

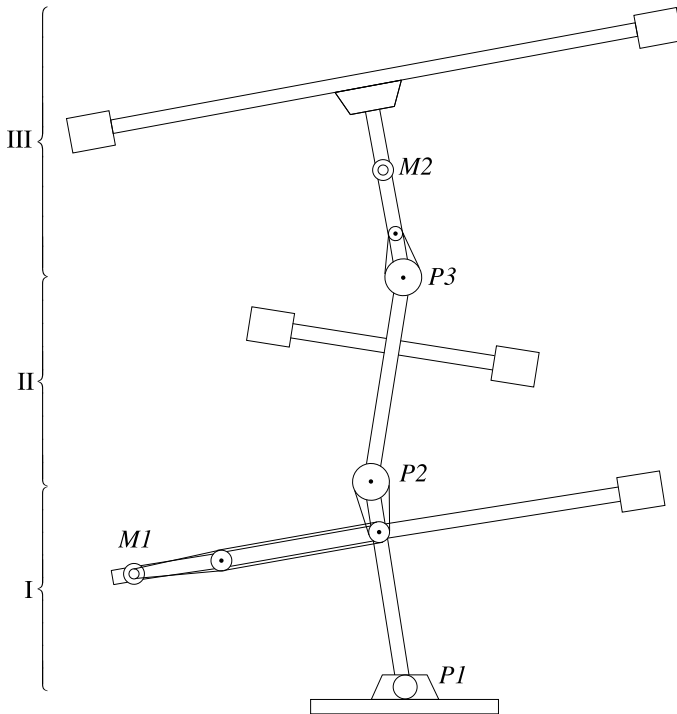


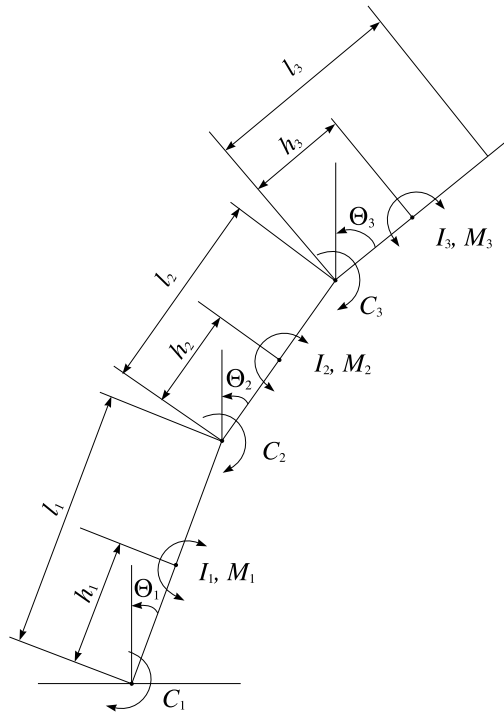
Fig. 15.1 Triple inverted pendulum

high order, which makes it unsuitable for implementation in practice. A model reduction is then conducted that leads to a reduced-order controller maintaining the required robust stability and robust performance of the closed-loop system.

15.1 System Description

The triple inverted pendulum considered is the experimental setup realized by Furuta et al. [48] (Fig. 15.1). The pendulum consists of three arms that are hinged by ball bearings and can rotate in the vertical plane. The torques of the two upper hinges are controlled by motors, with the lowest hinge made free for rotation. By controlling the angles of the two upper arms around specified values, the pendulum can be stabilized inversely with the desired angle attitudes. A horizontal bar is fixed to each of the arms to ease the control by increasing the moment of inertia. Two dc motors, M_1 and M_2 , are mounted on the first and third arm, respectively, acting as actuators that provide torques to the two upper hinges through timing belts. The potentiometers P_1 , P_2 , and P_3 are fixed to the hinges to measure the corresponding angles.

Fig. 15.2 Geometric relationship of potentiometers



Let Θ_i denote the angle of the i th arm. The first potentiometer measures the angle Θ_1 , and the second and third potentiometers measure the angles $\Theta_2 - \Theta_1$ and $\Theta_3 - \Theta_2$, respectively (Fig. 15.2).

The mathematical description of the triple inverted pendulum is derived under the following assumptions:

- (a) each arm is a rigid body;
- (b) the lengths of the belts remain constant during the operation of the system;
- (c) the friction force in the bottom hinge is proportional to the velocity of the bottom arm and the friction forces in the upper hinges are proportional to the differences of the respective velocities of two neighboring arms.

We shall first consider the mathematical model of the pendulum itself, without the actuators. The pendulum model is constructed using the Lagrange differential equations [48], which yield the following nonlinear vector–matrix differential equation:

$$M(\Theta) \begin{bmatrix} \ddot{\Theta}_1 \\ \ddot{\Theta}_2 \\ \ddot{\Theta}_3 \end{bmatrix} + N \begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \\ \dot{\Theta}_3 \end{bmatrix} + \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} + G \begin{bmatrix} t_{m1} \\ t_{m2} \end{bmatrix} = T \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad \text{where } \Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \end{bmatrix} \quad (15.1)$$

and we have

Table 15.1 System nomenclature

Symbol	Description
u_j	input voltage to the j th motor
t_{mj}	control torque of the j th motor
τ_i	disturbance torque to the i th arm
l_i	length of the i th arm
h_i	the distance from the bottom to the center of gravity of the i th arm
m_i	mass of the i th arm
I_i	moment of inertia of the i th arm around the center of gravity
C_i	coefficient of viscous friction of the i th hinge
Θ_i	angle of the i th arm from vertical line
C_{m_i}	viscous friction coefficient of the i th motor
I_{m_i}	moment of inertia of the i th motor
K_i	ratio of teeth of belt–pulley system of the i th hinge
$C_{p'_i}$	viscous friction coefficient of the belt–pulley system of the i th hinge
$I_{p'_i}$	moment of inertia of the belt–pulley system of the i th hinge
α_i	gain of the i th potentiometer
g	acceleration of gravity

 $M(\Theta)$

$$= \begin{bmatrix} J_1 + I_{p_1} & l_1 M_2 \cos(\Theta_1 - \Theta_2) - I_{p_1} & l_1 M_3 \cos(\Theta_1 - \Theta_3) \\ l_1 M_2 \cos(\Theta_1 - \Theta_2) - I_{p_1} & J_2 + I_{p_1} + I_{p_2} & l_2 M_3 \cos(\Theta_2 - \Theta_3) - I_{p_2} \\ l_1 M_3 \cos(\Theta_1 - \Theta_3) & l_2 M_3 \cos(\Theta_2 - \Theta_3) - I_{p_2} & J_3 + I_{p_2} \end{bmatrix}$$

$$N = \begin{bmatrix} C_1 + C_2 + C_{p_1} & -C_2 - C_{p_1} & 0 \\ -C_2 - C_{p_1} & C_{p_1} + C_{p_2} + C_2 + C_3 & -C_3 - C_{p_2} \\ 0 & -C_3 - C_{p_2} & C_3 + C_{p_2} \end{bmatrix}$$

$$q_1 = l_1 M_2 \sin(\Theta_1 - \Theta_2) \dot{\Theta}_2^2 + l_1 M_3 \sin(\Theta_1 - \Theta_3) \dot{\Theta}_3^2 - M_1 g \sin(\Theta_1)$$

$$q_2 = l_1 M_2 \sin(\Theta_1 - \Theta_2) \dot{\Theta}_1^2 + l_2 M_3 \sin(\Theta_2 - \Theta_3) \dot{\Theta}_3^2 - M_2 g \sin(\Theta_2)$$

$$q_3 = l_1 M_3 \sin(\Theta_1 - \Theta_3) (\dot{\Theta}_1^2 - 2\dot{\Theta}_1 \dot{\Theta}_3) + l_2 M_3 \sin(\Theta_2 - \Theta_3) (\dot{\Theta}_2^2 - 2\dot{\Theta}_2 \dot{\Theta}_3) - M_3 g \sin(\Theta_3)$$

$$G = \begin{bmatrix} K_1 & 0 \\ -K_1 & K_2 \\ 0 & -K_2 \end{bmatrix}, \quad T = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_{p_i} = C_{p'_i} + K_i^2 C_{m_i}, \quad I_{p_i} = I_{p'_i} + K_i^2 I_{m_i}$$

$$M_1 = m_1 h_1 + m_2 l_1 + m_3 l_1, \quad M_2 = m_2 h_2 + m_3 l_2, \quad M_3 = m_3 h_3$$

$$J_1 = I_1 + m_1 h_1^2 + m_2 l_1^2 + m_3 l_1^2, \quad J_2 = I_2 + m_2 h_2^2 + m_3 l_2^2$$

$$J_3 = I_3 + m_3 h_3^2$$

and all other parameters and variables are defined in Table 15.1.

After linearization of (15.1) under the assumptions of small deviations of the pendulum from the vertical position and of small velocities, one obtains the following equation:

$$M \begin{bmatrix} \ddot{\Theta}_1 \\ \ddot{\Theta}_2 \\ \ddot{\Theta}_3 \end{bmatrix} + N \begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \\ \dot{\Theta}_3 \end{bmatrix} + P \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \end{bmatrix} + G \begin{bmatrix} t_{m1} \\ t_{m2} \end{bmatrix} = T \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (15.2)$$

where

$$M = \begin{bmatrix} J_1 + I_{p1} & l_1 M_2 - I_{p1} & l_1 M_3 \\ l_1 M_2 - I_{p1} & J_2 + I_{p1} + I_{p2} & l_2 M_3 - I_{p2} \\ l_1 M_3 & l_2 M_3 - I_{p2} & J_3 + I_{p2} \end{bmatrix}$$

$$P = \begin{bmatrix} M_1 g & 0 & 0 \\ 0 & -M_2 g & 0 \\ 0 & 0 & -M_3 g \end{bmatrix}$$

By introducing the control torques vector $t_m = [t_{m1} \ t_{m2}]^T$ and the vector of disturbance torques $d = [\tau_1, \tau_2, \tau_3]^T$, (15.2) can be rewritten in the form

$$M\ddot{\Theta} + N\dot{\Theta} + P\Theta + Gt_m = Td$$

i.e.,

$$\ddot{\Theta} = M^{-1}(-N\dot{\Theta} - P\Theta - Gt_m + Td)$$

As for the output vector, we define

$$y = [\Theta_1 \ \Theta_2 \ \Theta_3]^T$$

The outputs are measured by linear potentiometers, whose voltages are given by

$$y_{p1} = \alpha_1 \Theta_1, \quad y_{p2} = \alpha_2 (\Theta_2 - \Theta_1), \quad y_{p3} = \alpha_3 (\Theta_3 - \Theta_2)$$

By introducing the vector of the measured outputs

$$y_p = [y_{p1} \ y_{p2} \ y_{p3}]^T$$

we obtain

$$y_p = C_p \Theta, \quad C_p = \begin{bmatrix} \alpha_1 & 0 & 0 \\ -\alpha_2 & \alpha_2 & 0 \\ 0 & -\alpha_3 & \alpha_3 \end{bmatrix}$$

The block-diagram of the pendulum system is shown in Fig. 15.3 and the nominal values of the parameters are given in Table 15.2.

15.2 Modeling of Uncertainties

Based on practical considerations, we in particular consider the variations of the moments of inertia I_1 , I_2 , and I_3 of the three arms and the variations of the viscous

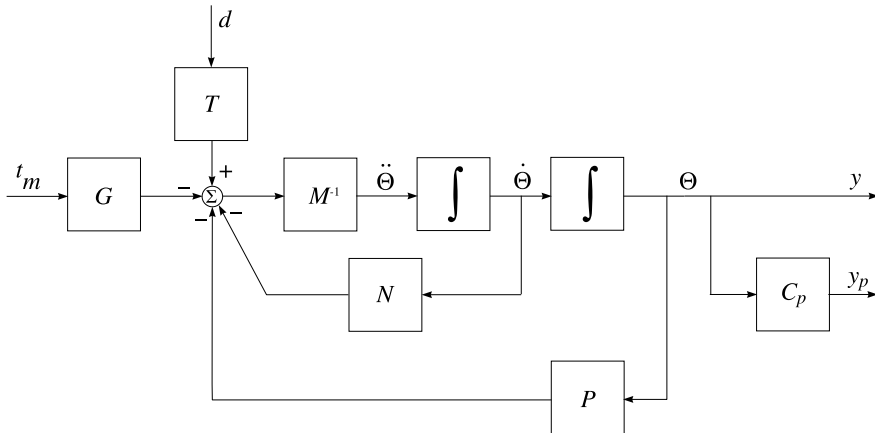


Fig. 15.3 Block-diagram of the triple inverted pendulum system

Table 15.2 Nominal values of the parameters

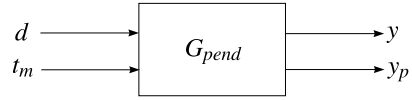
Symbol (unit)	Value	Symbol (unit)	Value
l_1 (m)	0.5	α_1 (V/rad)	1.146
l_2 (m)	0.4	α_2 (V/rad)	1.146
h_1 (m)	0.35	α_3 (V/rad)	0.9964
h_2 (m)	0.181	C_{m_1} (N m s)	2.19×10^{-3}
h_3 (m)	0.245	C_{m_2} (N m s)	7.17×10^{-4}
m_1 (kg)	3.25	I_{m_1} (kg m ²)	2.40×10^{-5}
m_2 (kg)	1.90	I_{m_2} (kg m ²)	4.90×10^{-6}
m_3 (kg)	2.23	$C_{p'_1}$ (N m s)	0
I_1 (kg m ²)	0.654	$C_{p'_2}$ (N m s)	0
I_2 (kg m ²)	0.117	$I_{p'_1}$ (kg m ²)	7.95×10^{-3}
I_3 (kg m ²)	0.535	$I_{p'_2}$ (kg m ²)	3.97×10^{-3}
C_1 (N m s)	6.54×10^{-2}	K_1	30.72
C_2 (N m s)	2.32×10^{-2}	K_2	27.00
C_3 (N m s)	8.80×10^{-3}		

friction coefficients C_1 , C_2 , C_3 , and C_{m_1} , C_{m_2} . It is assumed that the moments of inertia are constants but with possible relative error of 10 % around the nominal values; similarly, the friction coefficients may have with 15 % relative errors.

The eight real uncertain parameters I_1 , I_2 , I_3 , C_1 , C_2 , C_3 , C_{m_1} , C_{m_2} are set by using the following command lines:

```
i1 = ureal('i1',0.654,'Percentage',10);
i2 = ureal('i2',0.117,'Percentage',10);
i3 = ureal('i3',0.535,'Percentage',10);
```

Fig. 15.4 Uncertain model of the triple inverted pendulum



```

c1 = ureal('c1', 0.0654, 'Percentage', 15);
c2 = ureal('c2', 0.0232, 'Percentage', 15);
c3 = ureal('c3', 0.0088, 'Percentage', 15);
cd1 = ureal('cd1', 0.00219, 'Percentage', 15);
cd2 = ureal('cd2', 0.000717, 'Percentage', 15);
  
```

Then it is possible to compute the uncertain matrices M , N , Q which appear in the linearized pendulum model.

The uncertain pendulum model is obtained on the basis of the block-diagram shown in Fig. 15.3 by using the function `sysic`. This pendulum model is implemented by the M-file `mod_pend.m`.

The input/output relation of the uncertain pendulum model is described by

$$\begin{bmatrix} y \\ y_p \end{bmatrix} = G_{\text{pend}} \begin{bmatrix} d \\ t_m \end{bmatrix}$$

and is depicted in Fig. 15.4.

The singular value plot of the uncertain parameters triple, inverted pendulum is shown in Fig. 15.5.

It should be pointed out that the uncertain pendulum model can be obtained in several ways. For instance, it is possible to use the state space representation, derived on the basis of (15.2),

$$G_{\text{pend}} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]$$

where

$$A = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ -M^{-1}P & -M^{-1}N \end{bmatrix}, \quad B = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 2} \\ M^{-1}T & -M^{-1}G \end{bmatrix}$$

$$C = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \\ C_p & 0_{3 \times 3} \end{bmatrix}, \quad D = 0_{6 \times 5}$$

In such a case, however, the uncertain parameters I_1 , I_2 , and I_3 are repeated at least twice in the model G_{pend} such that the total number of uncertain parameters increase to 11. In the derivation, based on the block-diagram shown in Fig. 15.3, the parameters I_1 , I_2 , and I_3 appear only once thus keeping the total number of uncertain parameters equal to eight.

We now consider the models of the actuators. The nominal transfer functions of the actuators are taken as first-order, phase-lag models of

$$\bar{G}_{m_1} = \frac{\bar{K}_{m_1}}{\bar{T}_{m_1}s + 1}, \quad \bar{G}_{m_2} = \frac{\bar{K}_{m_2}}{\bar{T}_{m_2}s + 1}$$

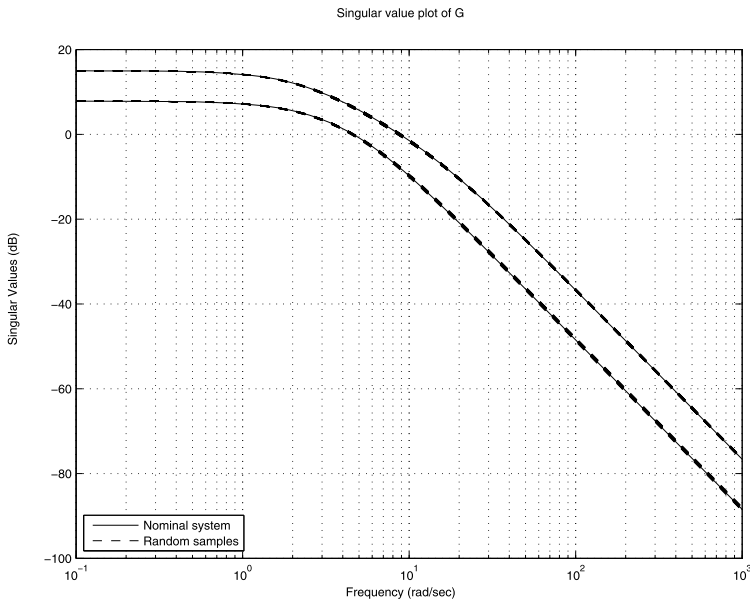


Fig. 15.5 Singular values of the pendulum system

with parameters

$$\bar{K}_{m_1} = 1.08, \quad \bar{T}_{m_1} = 0.005, \quad \bar{K}_{m_2} = 0.335, \quad \bar{T}_{m_2} = 0.002$$

It is assumed that the actual gain coefficients (K_{m_1}, K_{m_2}) are constants with relative error 10 % around their nominal values and the time constants (T_{m_1}, T_{m_2}) with relative error 20 %.

The uncertain frequency responses of the actuators are shown in Fig. 15.6.

In order to account for unmodeled dynamics and nonlinear effects, the uncertainties in the actuator models are approximated by input multiplicative uncertainties that give rise to the perturbed transfer functions

$$G_{m_1} = (1 + W_{m_1}\delta_{m_1})\bar{G}_{m_1}, \quad G_{m_2} = (1 + W_{m_2}\delta_{m_2})\bar{G}_{m_2}$$

where

$$|\delta_{m_1}| \leq 1, \quad |\delta_{m_2}| \leq 1$$

and the uncertainty weights W_{m_1}, W_{m_2} are so chosen that

$$\frac{|G_{m_1}(j\omega) - \bar{G}_{m_1}(j\omega)|}{|\bar{G}_{m_1}(j\omega)|} < |W_{m_1}(j\omega)|, \quad \frac{|G_{m_2}(j\omega) - \bar{G}_{m_2}(j\omega)|}{|\bar{G}_{m_2}(j\omega)|} < |W_{m_2}(j\omega)|.$$

Note that $\delta_{m_1}, \delta_{m_2}$ are complex uncertain parameters corresponding to uncertain LTI dynamics.

The frequency responses of W_{m_1}, W_{m_2} are found graphically as shown in Fig. 15.7 and then approximated by first-order transfer functions using the file `wfit.m`. As a result, we obtain

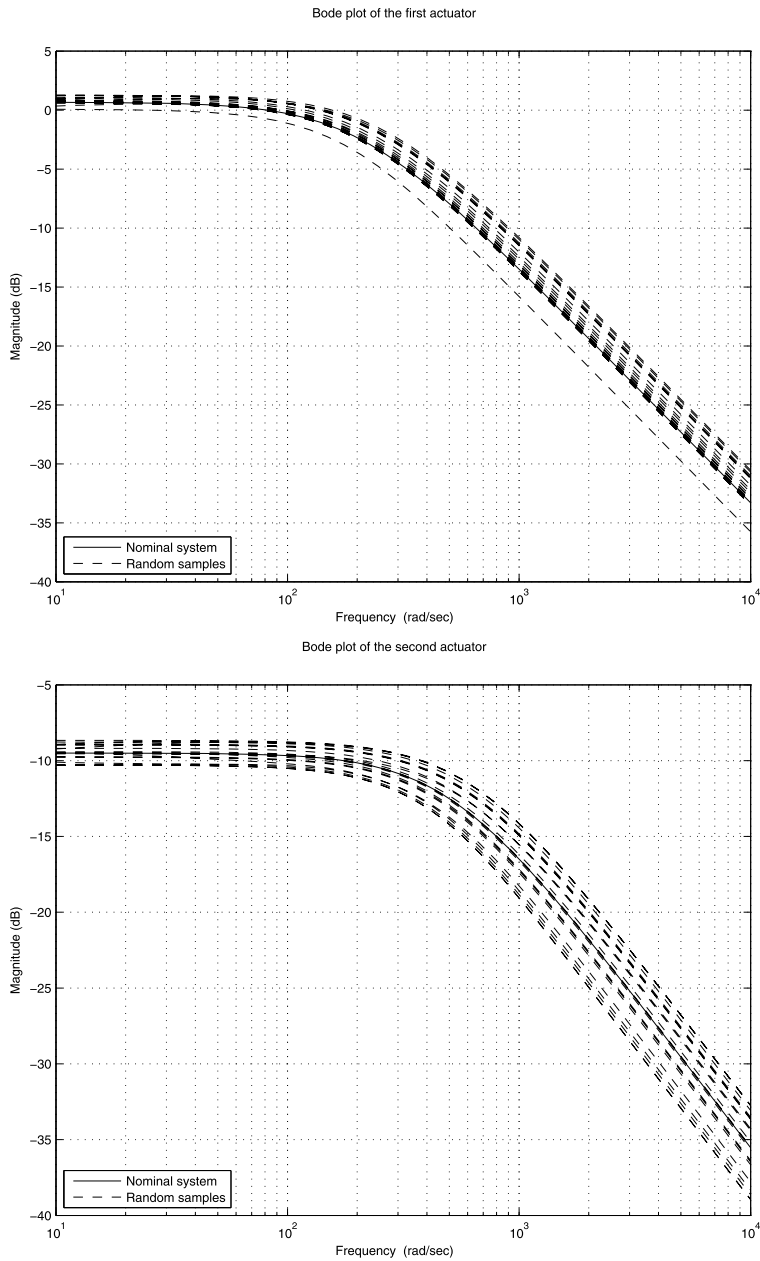


Fig. 15.6 Uncertain frequency responses of the actuators

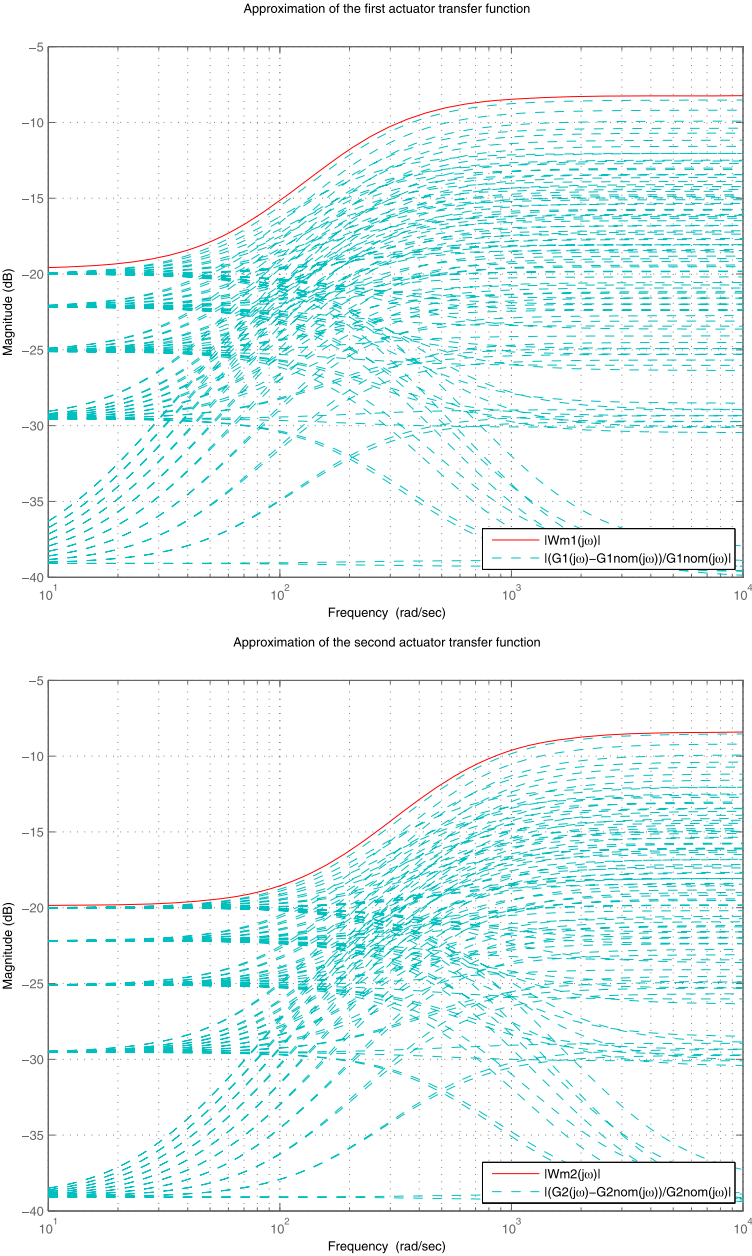


Fig. 15.7 Actuators uncertainty approximations

Fig. 15.8 Block-diagram of the uncertain actuators model

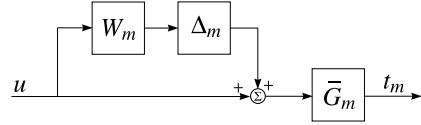
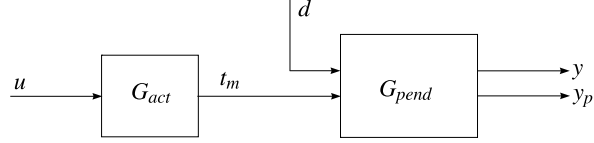


Fig. 15.9 Triple inverted pendulum system with uncertainties



$$W_{m1} = \frac{0.3877s + 25.6011}{1.0000s + 246.3606}, \quad W_{m2} = \frac{0.3803s + 60.8973}{1.0000s + 599.5829}$$

By introducing the vector $u = [u_1, u_2]^T$, the equations of the actuators are rewritten as

$$t_m = G_{act}u$$

where

$$G_{act} = \bar{G}_m(I_2 + W_m \Delta_m),$$

$$\bar{G}_m = \begin{bmatrix} \bar{G}_{m1} & 0 \\ 0 & \bar{G}_{m2} \end{bmatrix}, \quad W_m = \begin{bmatrix} W_{m1} & 0 \\ 0 & W_{m2} \end{bmatrix}, \quad \Delta_m = \begin{bmatrix} \delta_{m1} & 0 \\ 0 & \delta_{m2} \end{bmatrix}$$

The block-diagram of the actuators with the input multiplicative uncertainty is shown in Fig. 15.8.

The uncertain actuators model is obtained by using the file `act_pend.m` that contains the following command lines:

```
% First actuator
gd1 = 1.080; Td1 = 0.005;
G1 = tf([gd1],[Td1 1]);
Wm1 = tf([0.3877 25.6011],[1.0000 246.3606]);
Delta_act1 = ultidyn('Delta_act1',[1 1]);
Act1 = G1*(1 + Wm1*Delta_act1);
%
% Second actuator
gd2 = 0.335; Td2 = 0.002;
G2 = tf([gd2],[Td2 1]);
Wm2 = tf([0.3803 60.8973],[1.0000 599.5829]);
Delta_act2 = ultidyn('Delta_act2',[1 1]);
Act2 = G2*(1 + Wm2*Delta_act2);
```

Note that the uncertain LTI dynamics δ_1 , δ_2 are set implementing the function `ultidyn`.

Having modeled both the pendulum and the actuators with consideration of possible perturbations, the block-diagram of the whole system is seen as in Fig. 15.9. Note that we have a case of a mixed, real, and complex, uncertainty configuration.

Fig. 15.10 Uncertain model of the triple inverted pendulum system

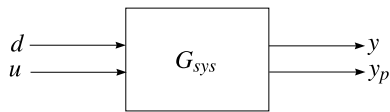
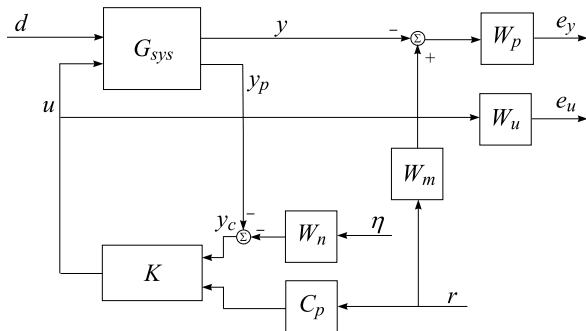


Fig. 15.11 Interconnection structure of the closed-loop system



The whole, perturbed, triple inverted pendulum system can be described by the equation

$$\begin{bmatrix} y \\ y_p \end{bmatrix} = G_{\text{sys}} \begin{bmatrix} d \\ u \end{bmatrix}$$

where the uncertain state-space system G_{sys} can be easily obtained by using the function `sysic`. The system is shown in Fig. 15.10 and contains 10 uncertain parameters (eight real and two LTI dynamics).

15.3 Design Specifications

The block-diagram of the closed-loop system, which includes the triple inverted pendulum model, the feedback structure and the controller, as well as the elements reflecting the model uncertainty and the performance objectives, is depicted in Fig. 15.11. In order to achieve better performance, we shall make use of the configuration of a 2-degree-of-freedom (2DOF) controller.

The system has a reference input (r), an input disturbance (d), a measurement noise (η) and two output costs (e_y and e_u). The system W_m is an ideal dynamics model to which the designed closed-loop system has to match. The model of the triple inverted pendulum system is the uncertain state-space system G_{sys} . The feedback signal is generated on the basis of the potentiometers output $y_p = C_p y$, which requires the reference r to be multiplied by the matrix C_p (the other way to deal with the situation is to multiply y_p by C_p^{-1}). The measurement of the arm angles is accompanied by introduction of frequency-dependent noises that are inevitable in practice and are thus added to the corresponding measurements. This is why at the outputs of the potentiometers one obtains the signal $y_c = -(y_p + W_n \eta)$, where W_n is a weighting function (shaping filter on the measurement noise) and $\eta = [\eta_1 \ \eta_2 \ \eta_3]^T$

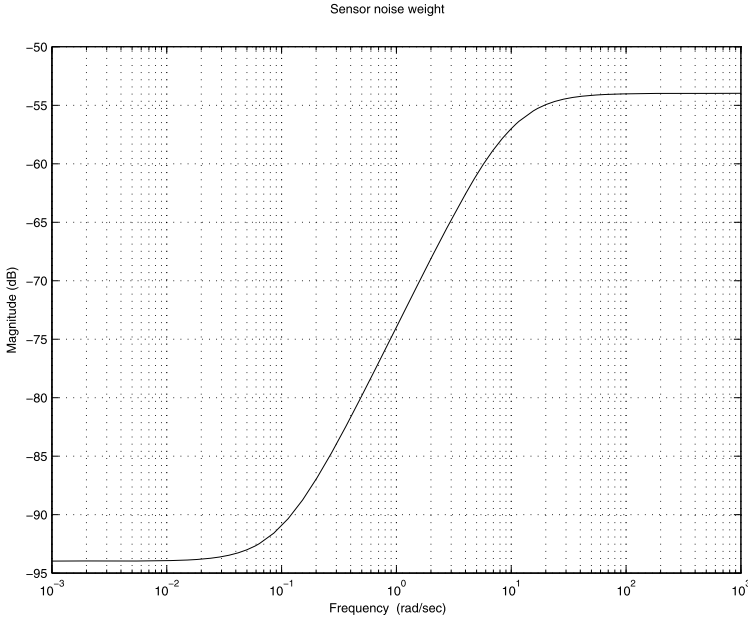


Fig. 15.12 Measurement noise weighting function

is an arbitrary noise signal satisfying $\|\eta\|_2 \leq 1$. By choosing an appropriate W_n it is possible to form the desired spectral contents of the actual noise signal (in the form of $W_n\eta$).

In the given case the matrix W_n is chosen as

$$W_n(s) = \begin{bmatrix} w_n(s) & 0 & 0 \\ 0 & w_n(s) & 0 \\ 0 & 0 & w_n(s) \end{bmatrix}$$

where the weighting transfer function $w_n = 2 \times 10^{-5} \frac{10s+1}{0.1s+1}$ is a high pass filter that shapes the noise spectral density for the type of potentiometers under consideration. The magnitude plot of this filter is shown in Fig. 15.12. This transfer function means that in the low-frequency range the magnitude of the measurement error is about 2×10^{-5} V, and in the high-frequency range, about 2×10^{-3} V.

As for feedback signals, we shall use $y_c = -(y_p + W_n\eta)$ and $C_p r$.

The weighted closed-loop outputs e_y and e_u satisfy the equation

$$\begin{bmatrix} e_y \\ e_u \end{bmatrix} = \Phi(s) \begin{bmatrix} r \\ d \\ \eta \end{bmatrix}$$

where the matrix $\Phi = \Phi(G_{\text{sys}})$ is the uncertain closed-loop transfer function matrix.

The performance objective requires the transfer function matrix $\Phi(s)$ from r , d and η to e_y and e_u to be small in the sense of $\|\cdot\|_\infty$, for all possible (stable)

uncertainties. The appropriately chosen weighting functions matrices W_p and W_u are used to reflect the relative significance over different frequency ranges for which the performance is required.

The design problem for the triple inverted pendulum system is to find a linear, output controller $K(s)$ to generate the output feedback

$$u(s) = K(s) \begin{bmatrix} y_c(s) \\ C_p r(s) \end{bmatrix}$$

to ensure the following properties of the closed-loop system.

15.3.1 Robust Stability

The closed-loop system achieves robust stability if the closed-loop system is internally stable for each possible, perturbed plant dynamics G_{sys} .

15.3.2 Nominal Performance

The closed-loop system achieves nominal performance if the following performance objective is satisfied for the nominal plant model $G_{\text{sys nom}}$:

$$\|\Phi(G_{\text{sys nom}})\|_{\infty} < 1 \quad (15.3)$$

This objective is similar to the mixed S/KS sensitivity optimization.

15.3.3 Robust Performance

The closed-loop system must maintain, for each G_{sys} , the performance objective

$$\|\Phi(G_{\text{sys}})\|_{\infty} < 1 \quad (15.4)$$

In addition to the above requirements, it is desirable that the controller designed would have acceptable complexity, i.e. it is of reasonably low order.

The ideal system model to be matched with, is chosen as

$$W_m(s) = \begin{bmatrix} w_{m1}(s) & 0 & 0 \\ 0 & w_{m2}(s) & 0 \\ 0 & 0 & w_{m3}(s) \end{bmatrix}$$

where

$$\begin{aligned} w_{m1}(s) &= \frac{1}{100s^2 + 14s + 1} \\ w_{m2}(s) &= \frac{1}{25s^2 + 7s + 1} \\ w_{m3}(s) &= \frac{1}{9s^2 + 4.2s + 1} \end{aligned}$$

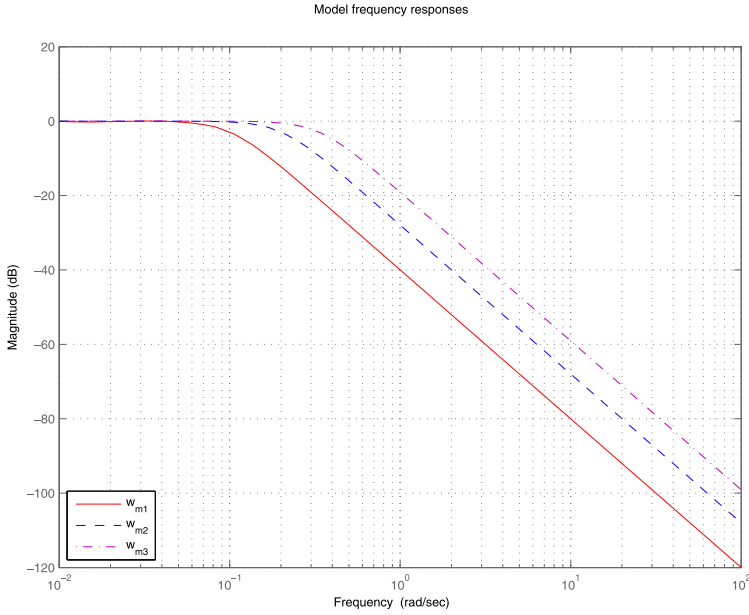


Fig. 15.13 Model frequency responses

This model takes into account that the first arm has the slowest dynamics and the third arm—the fastest one.

The magnitude responses of the model are shown in Fig. 15.13.

In the given case the weighting performance functions are chosen in the forms of

$$W_p(s) = \begin{bmatrix} w_{p1}(s) & 0 & 0 \\ 0 & w_{p2}(s) & 0 \\ 0 & 0 & w_{p3}(s) \end{bmatrix}, \quad W_u(s) = \begin{bmatrix} w_u(s) & 0 \\ 0 & w_u(s) \end{bmatrix}$$

where

$$\begin{aligned} w_{p1}(s) &= 1 \times 10^{-2} \frac{s^2 + 2s + 4}{s^2 + 1.5s + 0.1} \\ w_{p2}(s) &= 2 \times 10^{-2} \frac{s^2 + 2s + 4}{s^2 + 1.5s + 0.01} \\ w_{p3}(s) &= 4 \times 10^{-2} \frac{s^2 + 2s + 4}{s^2 + 1.5s + 0.01} \end{aligned}$$

and $w_u(s) = 10^{-6} \frac{s+1}{0.001s+1}$.

The weighting functions are set in the M-file `wt_s_pend.m`.

The magnitude responses of the inverse weighting function W_p^{-1} are shown in Fig. 15.14. It is seen from the figure that e_{y1} is the least penalized output, while e_{y3} is the most penalized one. Such a choice of the weighting functions is justified by

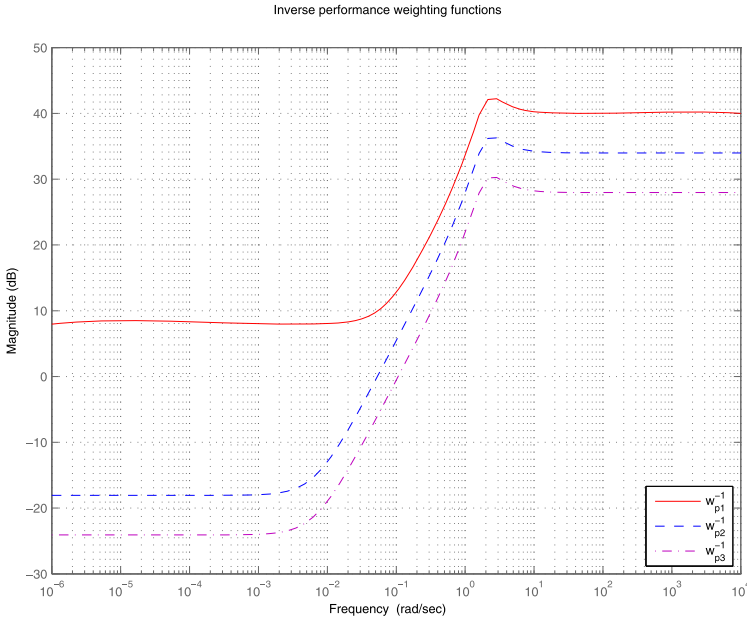


Fig. 15.14 Frequency responses of the inverse weighting function

the fact that the variable $y_1 = \Theta_1$ is not controlled directly and that y_3 has the fastest dynamics.

15.4 System Interconnections

The internal structure of the 11-input, 11-output 30th order open-loop system, which is saved as the variable `pend_ic`, is shown in Fig. 15.15. The reference, the disturbance and the noise are saved in the variables `ref`, `dist`, `noise`, respectively. The control signal is saved in the variable `control`.

The variables `ref`, `dist`, `noise`, `y`, `y_p`, `y_c`, `e_y` have all three elements and the variables `control`, `e_u` have two elements. The open-loop connection is assigned by the M-file `olp_pend` using the function `sysic`.

The schematic diagram showing the specific input/output ordering for the variable `pend_ic` is shown in Fig. 15.16.

The block-diagram used in the closed-loop system simulation is shown in Fig. 15.17. The corresponding closed-loop interconnection, which is saved in the variable `pend_sm`, is obtained by the M-file `sim_pend`.

Figure 15.18 shows the schematic diagram of the specific input/output ordering for the variable `pend_sm`.

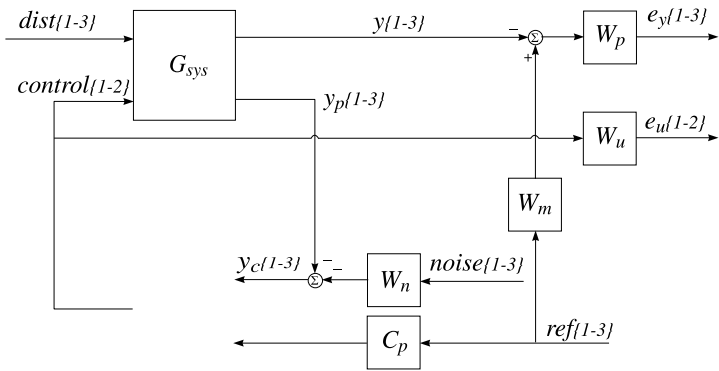


Fig. 15.15 Open-loop interconnection of the triple inverted pendulum system

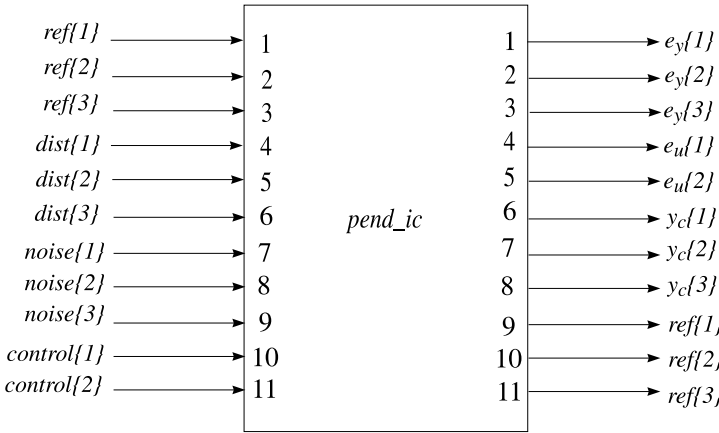
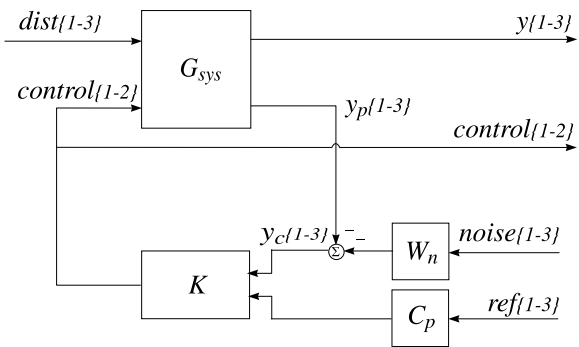


Fig. 15.16 Schematic diagram of the open-loop system

Fig. 15.17 Closed-loop interconnection structure of the pendulum system



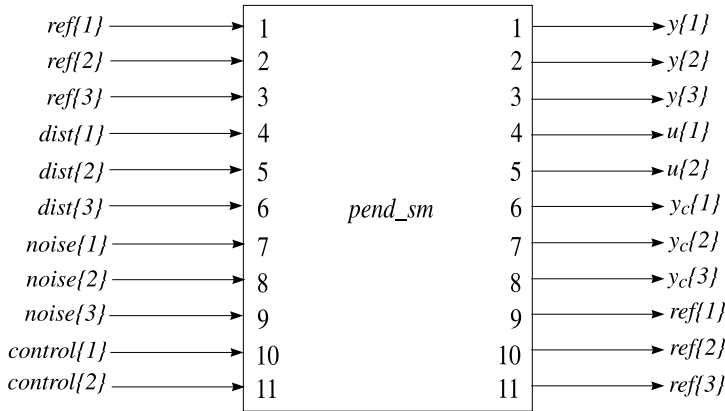
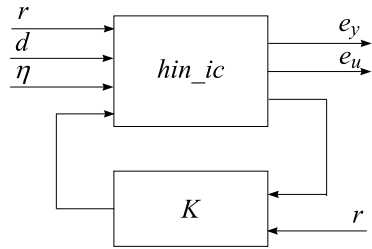


Fig. 15.18 Schematic diagram of the closed-loop system

Fig. 15.19 Block diagram for \mathcal{H}_∞ design



15.5 \mathcal{H}_∞ Design

The design goal in this case is to find an \mathcal{H}_∞ (sub)optimal control law for the interconnection shown in Fig. 15.19, in which we make use of the nominal system model. The variable `hin_ic`, which corresponds to the nominal transfer function matrix P_{nom} of the augmented system, is obtained by the command line

```
hin_ic = pvget(pend_ic, 'NominalValue');
```

The \mathcal{H}_∞ optimal control minimizes the $\|\cdot\|_\infty$ norm of $F_L(P_{\text{nom}}, K)$ over the stabilizing controller transfer matrix K . In the given case $F_L(P_{\text{nom}}, K)$ is the nominal closed-loop system transfer matrix from the references, disturbances and noises (the signals r , d , and η) to the weighted outputs e_y and e_u (Fig. 15.19).

The \mathcal{H}_∞ design is conducted by using the M-file `hinf_pend.m`. It utilizes the function `hinfsyn`, which determines a (sub)optimal \mathcal{H}_∞ control law, based on the prescribed open-loop interconnection. The interval for γ iteration is chosen between 0 and 10 with a tolerance $\text{tol} = 0.001$. The controller obtained is of 30th order and for this controller the closed-loop system achieves \mathcal{H}_∞ norm equal to 0.4346. An undesired property of the controller is that it has a pole at 68.71, i.e. this controller is unstable, which makes it less favorable in practice.

The singular value plot of the \mathcal{H}_∞ controller transfer matrix is shown in Fig. 15.20.

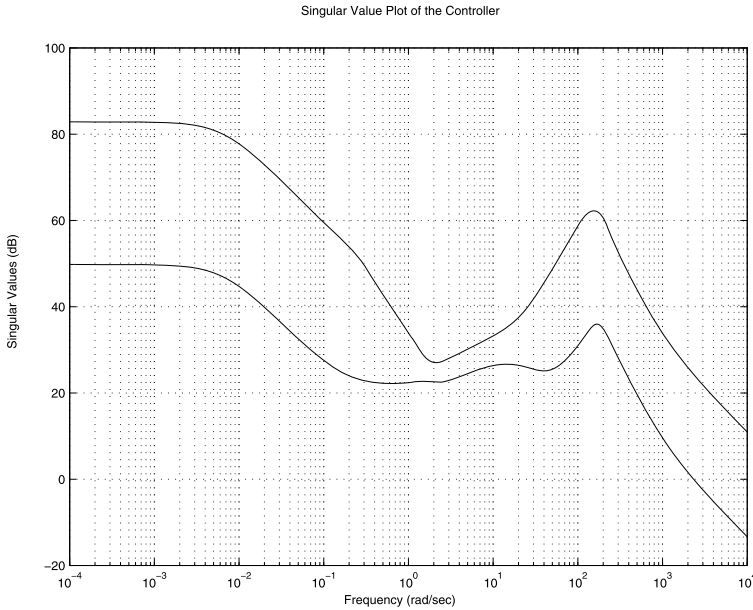


Fig. 15.20 Singular values of the \mathcal{H}_∞ controller

Since the closed-loop system does not achieve robust stability, it is meaningful to obtain the system transient responses only for the nominal parameters. The closed-loop, transient responses (reference tracking) for the nominal model are obtained by using the file `clpPend`. The transient response of the closed-loop system with \mathcal{H}_∞ controller for the reference vector

$$r = \begin{bmatrix} 0 \\ -0.1 \\ 0.2 \end{bmatrix}$$

(the references are measured in radians) is shown in Fig. 15.21. The response is characterized by small overshoots of the output variables. The steady-state errors are small, except for a small error on the position of the first arm.

The disturbance rejection of the closed-loop system with the \mathcal{H}_∞ controller is shown in Fig. 15.22. The disturbance vector is set to

$$d = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$$

and measured in N m.

In Fig. 15.23 we show the singular values plot of the uncertain closed-loop transfer matrix with the \mathcal{H}_∞ controller. It is seen from the figure that for some values of the uncertain parameters the magnitude responses have very high picks that may indicate potential instability of the closed loop system.

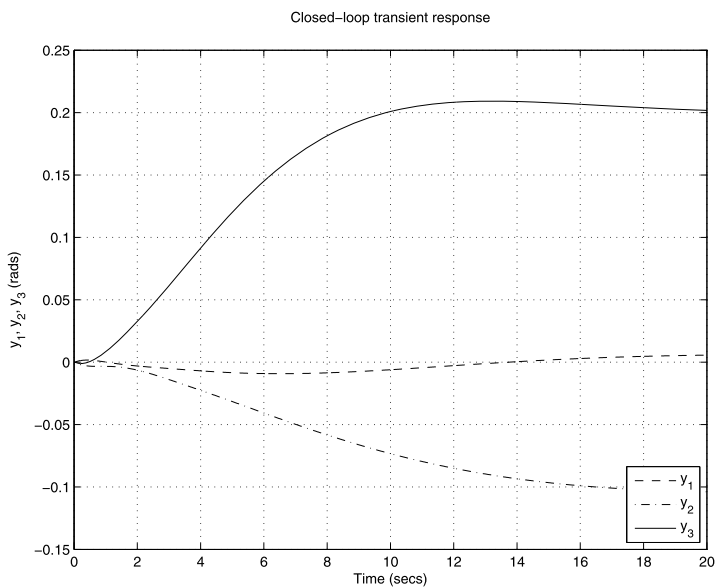


Fig. 15.21 Closed-loop transient response of \mathcal{H}_∞ controller

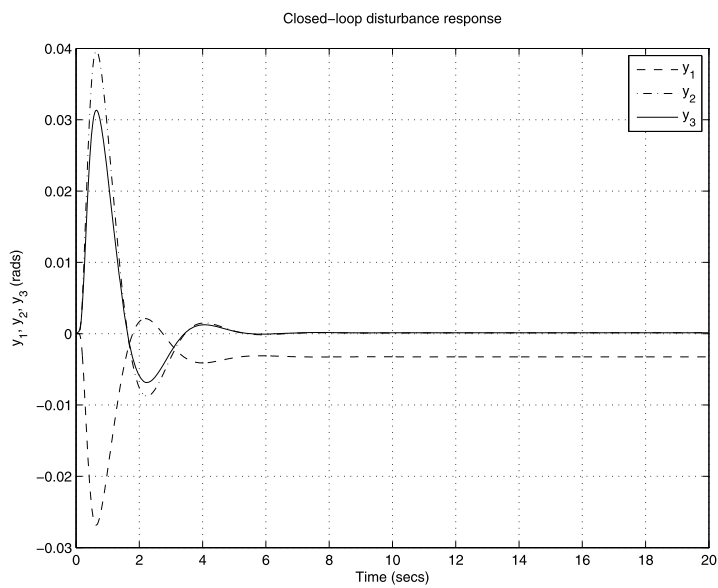


Fig. 15.22 Disturbance rejection of \mathcal{H}_∞ controller

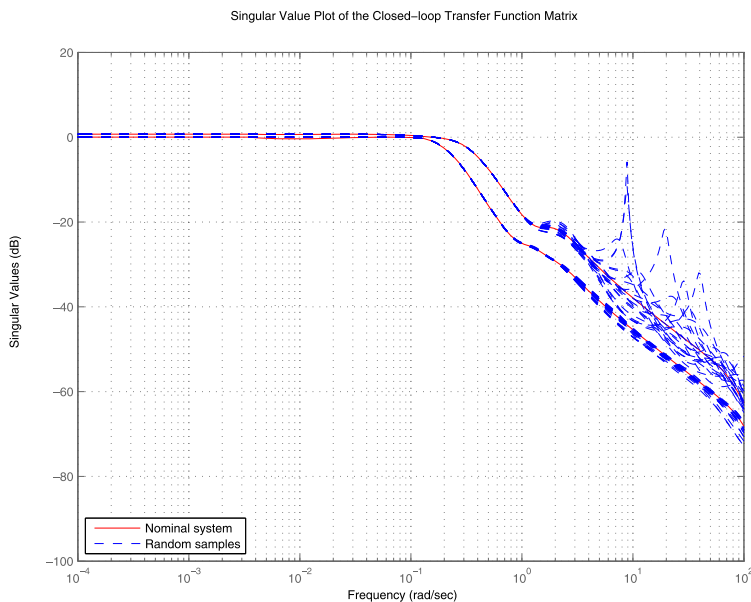


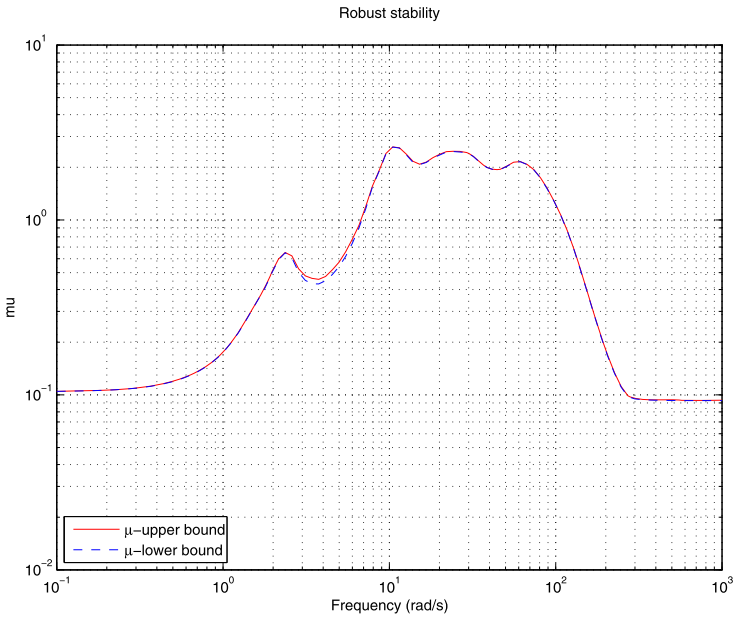
Fig. 15.23 Closed-loop singular values with \mathcal{H}_∞ controller

The robust stability analysis of the closed-loop system is done by the file `mu_pend` implementing the function `robuststab`. As a result one obtains the following report:

```
report =
```

```
Uncertain System is NOT robustly stable to modeled
uncertainty.
```

- ```
-- It can tolerate up to 38.2% of the modeled uncertainty.
-- A destabilizing combination of 38.3% of the modeled
 uncertainty exists, causing an instability
 at 10.5 rad/s.
-- Sensitivity with respect to uncertain element ...
 'Delta_act1' is 54%. Increasing 'Delta_act1' by 25%
 leads to a 14% decrease in the margin.
 'Delta_act2' is 21%. Increasing 'Delta_act2' by 25%
 leads to a 5% decrease in the margin.
 'c1' is 0%. Increasing 'c1' by 25% leads
 to a 0% decrease in the margin.
 'c2' is 0%. Increasing 'c2' by 25% leads
 to a 0% decrease in the margin.
 'c3' is 0%. Increasing 'c3' by 25% leads
 to a 0% decrease in the margin.
 'cm1' is 15%. Increasing 'cm1' by 25% leads
 to a 4% decrease in the margin.
```



**Fig. 15.24** Robust stability test of  $\mathcal{H}_\infty$  controller

|       |                                                                      |
|-------|----------------------------------------------------------------------|
| 'cm2' | is 0%. Increasing 'cm2' by 25% leads to a 0% decrease in the margin. |
| 'i1'  | is 23%. Increasing 'i1' by 25% leads to a 6% decrease in the margin. |
| 'i2'  | is 7%. Increasing 'i2' by 25% leads to a 2% decrease in the margin.  |
| 'i3'  | is 31%. Increasing 'i3' by 25% leads to a 8% decrease in the margin. |

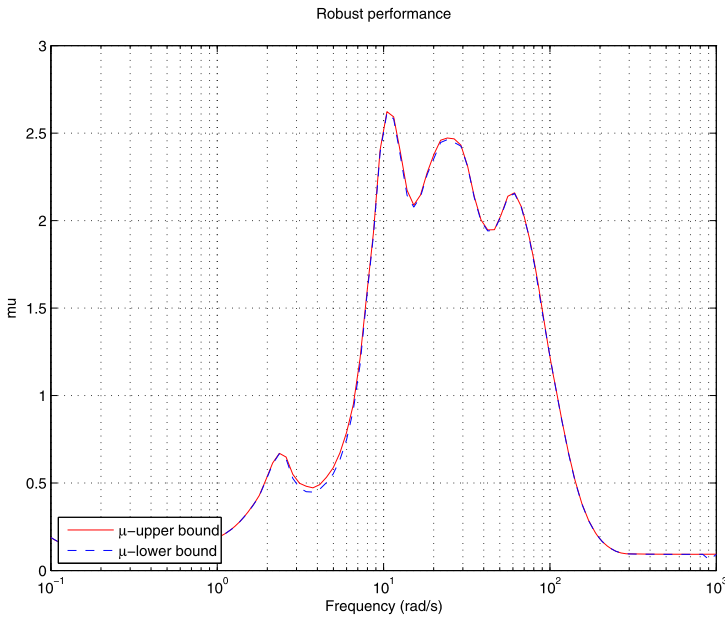
The upper and lower bounds of the structured singular value  $\mu$  (over the frequency range) are shown in Fig. 15.24. The closed-loop system with  $\mathcal{H}_\infty$  controller does not achieve robust stability, the maximum value of  $\mu$  exceeding 2.6. It follows from the report that the uncertain parameters  $C_1$ ,  $C_2$ ,  $C_3$ , and  $C_{m_2}$  practically do not affect the robust stability.

The robust performance of the closed-loop system with  $\mathcal{H}_\infty$  controller is checked with the function `robustperf`. As a result we obtain the following report:

report =

Uncertain System achieves a robust performance margin of 0.3822.

- A model uncertainty exists of size 38.1% resulting in a performance margin of 2.62 at 10.5 rad/s.
- Sensitivity with respect to uncertain element ...



**Fig. 15.25** Robust performance of  $\mathcal{H}_\infty$  controller

The  $\mu$  values corresponding to the case of robust performance analysis are shown in Fig. 15.25. Again, the closed-loop system does not achieve robust performance either.

Hence, it is concluded that the designed  $\mathcal{H}_\infty$  controller leads to good closed-loop transient responses, but does not ensure the necessary robustness of the closed-loop system as required.

## 15.6 $\mu$ -Synthesis

Let us denote by  $P(s)$  the transfer function matrix of the 11-input, 11-output, open-loop system consisting of the pendulum system model plus the weighting functions, and let the block structure  $\Delta_P$  of uncertainties be defined by

$$\Delta_P := \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_F \end{bmatrix} : \Delta \in \mathcal{R}^{10 \times 10}, \Delta_F \in \mathcal{C}^{9 \times 5} \right\}$$

The first block of the matrix  $\Delta_P$ , the uncertainty block  $\Delta$ , corresponds to the parametric and LTI uncertainties modeled in the triple inverted pendulum system. The second block,  $\Delta_F$ , is a fictitious uncertainty block, introduced to include the performance objectives in the framework of the  $\mu$ -approach. The inputs to this block are the weighted error signals  $e_y$  and  $e_u$  the outputs being the exogenous inputs  $r$ ,  $d$ , and  $\eta$ .

**Table 15.3**  $D$ - $K$  iterations results in  $\mu$ -synthesis

| Iteration | Controller order | Maximum value of $\mu$ |
|-----------|------------------|------------------------|
| 1         | 30               | 2.400                  |
| 2         | 68               | 0.810                  |
| 3         | 82               | 0.525                  |
| 3         | 80               | 0.458                  |

To meet the design objectives a stabilizing controller  $K$  is to be found such that, at each frequency  $\omega \in [0, \infty]$ , the structured singular value  $\mu$  satisfies the condition

$$\mu_{\Delta_P} [F_L(P, K)(j\omega)] < 1$$

The fulfillment of this condition guarantees robust performance of the closed-loop system, i.e.,

$$\|\Phi(G_{\text{sys}})\|_{\infty} < 1$$

for all stable perturbations  $\Delta$  with  $\|\Delta\|_{\infty} < 1$ .

The  $\mu$ -synthesis is conducted by using the M-file `ms_pend.m` implementing the function `dksyn`.

The  $\mu$ -synthesis is done for the same performance and control action weighting functions as the  $\mathcal{H}_{\infty}$  design. The progress of the  $D$ - $K$  iteration is shown in Table 15.3.

Four iterations are done reducing the maximum value of  $\mu$  to 0.458. The final controller obtained is of 80th order. Note that this controller is stable.

The singular value plot of the  $\mu$ -controller is shown in Fig. 15.26.

The robust stability and robust performance analysis of the closed-loop system with  $\mu$ -controller yields the following results:

```
report =
```

```
Uncertain System is robustly stable to modeled uncertainty.
```

```
-- It can tolerate up to 266% of the modeled uncertainty.
```

```
-- A destabilizing combination of 273% of the modeled
 uncertainty exists, causing an instability
 at 3.43 rad/s.
```

```
-- Sensitivity with respect to uncertain element ...
```

```
report =
```

```
Uncertain System achieves a robust performance margin
of 2.238.
```

```
-- A model uncertainty exists of size 223% resulting in a
 performance margin of 0.448 at 0.1 rad/s.
```

```
-- Sensitivity with respect to uncertain element ...
```

The structured singular values for the robust stability study (over the frequency range) are shown in Fig. 15.27.

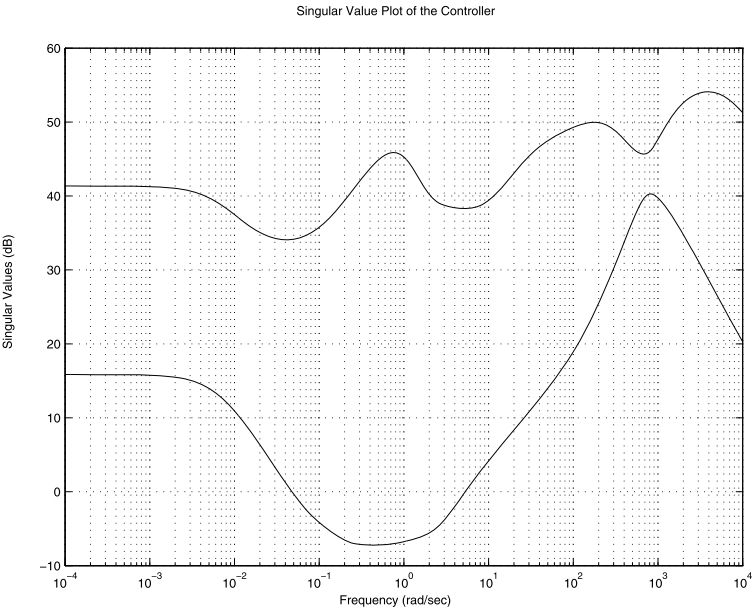


Fig. 15.26 Singular values of the  $\mu$ -controller

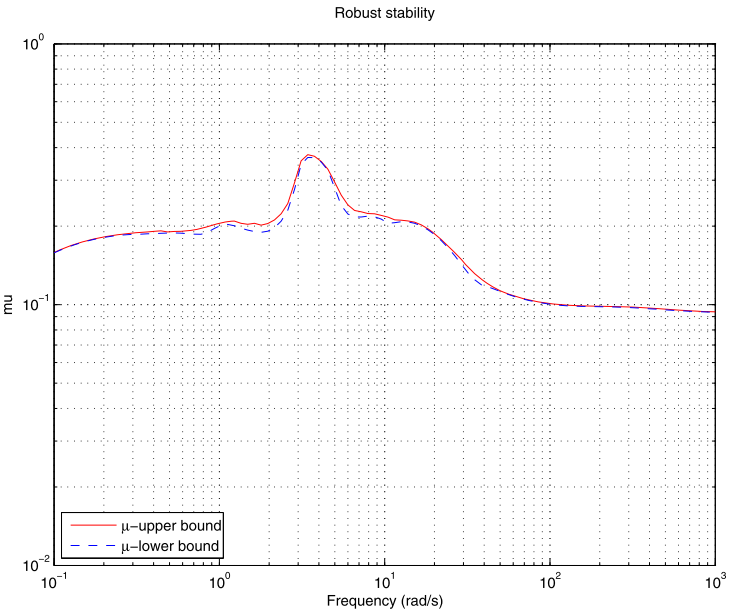
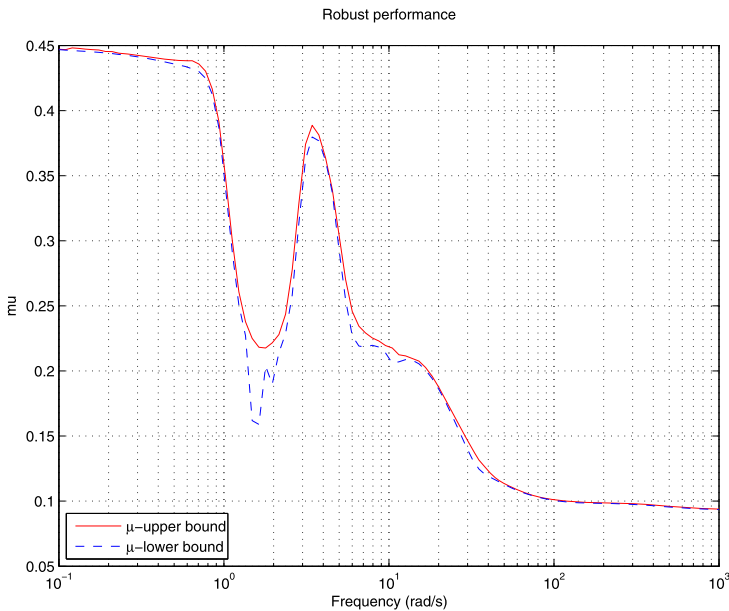


Fig. 15.27 Robust stability of  $\mu$  controller



**Fig. 15.28** Robust performance of  $\mu$  controller

The  $\mu$  values over the frequency range for the case of robust performance analysis are shown in Fig. 15.28.

Consider now the closed-loop transient responses. Since the closed-loop system achieves robust stability and performance, it is possible to obtain the transient responses for different values of the uncertain parameters. This is done by using the M-file `mcs_pend.m`. The same reference signal as used in the  $\mathcal{H}_\infty$  simulation is used here. That is,

$$r = \begin{bmatrix} 0 \\ -0.1 \\ 0.2 \end{bmatrix}$$

The transient response of the uncertain closed-loop system is shown in Fig. 15.29. The response is slightly slower than in the case of  $\mathcal{H}_\infty$  controller.

The control action in the case of a  $\mu$ -controller is shown in Fig. 15.30. The motor voltages are kept within 0.14 V for all possible perturbations.

The disturbance rejection of the uncertain closed-loop system with the  $\mu$ -controller is shown in Fig. 15.31. We see that the overshoot of the third output is almost 5 times larger than in the case of the  $\mathcal{H}_\infty$  controller.

Hence, it is clear that the (nominal) transient responses in terms of reference tracking and disturbance attenuation are worse in the case of the  $\mu$ -controller. This is the price that has to be paid to ensure the robust stability and robust performance of the closed-loop system.

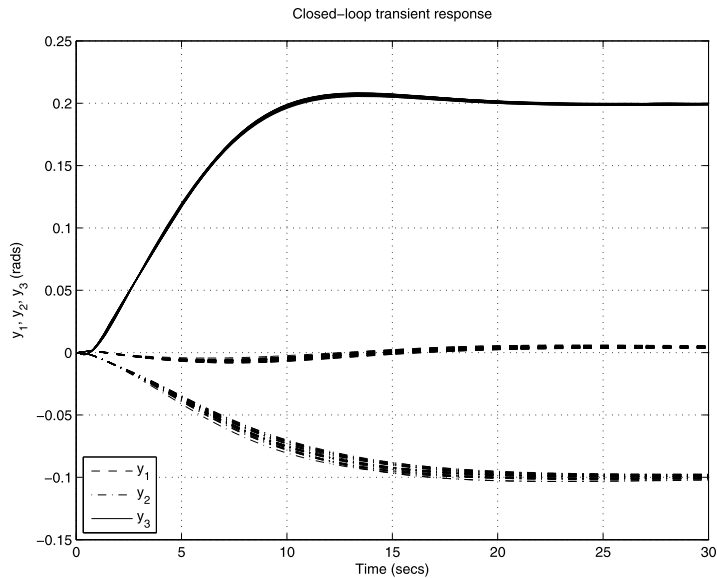


Fig. 15.29 Closed-loop transient response of  $\mu$ -controller

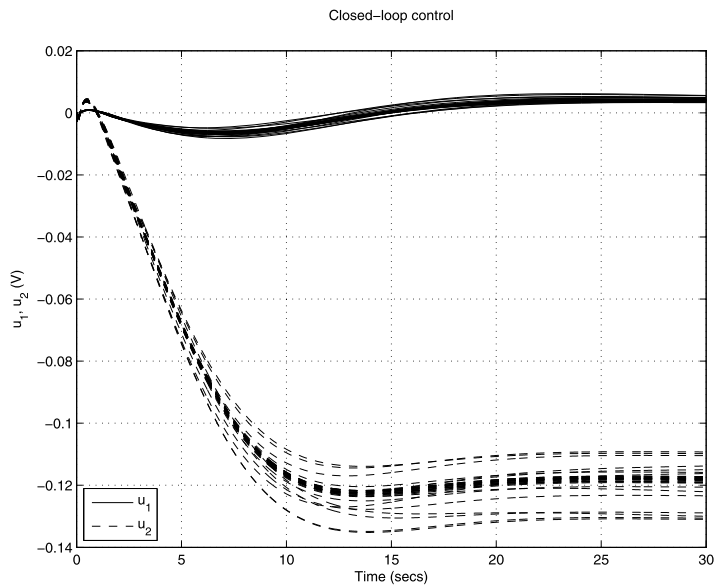
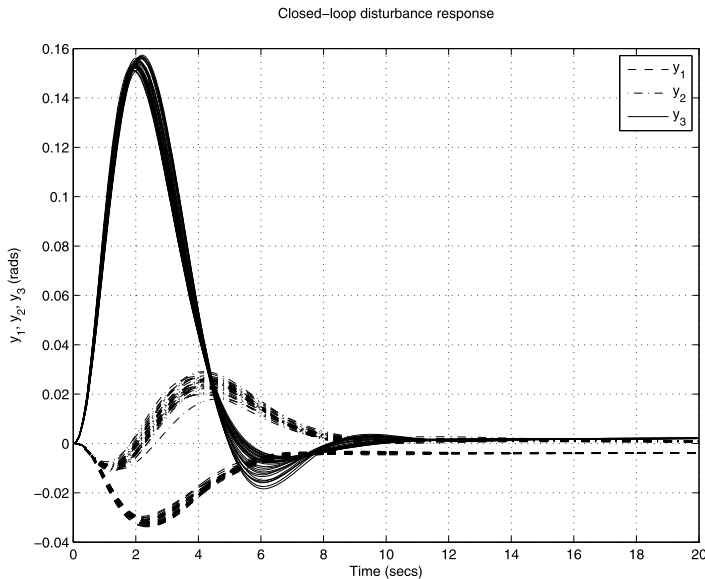


Fig. 15.30 Control action of  $\mu$ -controller

In Fig. 15.32, we show the noise  $w_n\eta_3$ , which acts at the output of the third potentiometer, and the corresponding response of the output  $y_3$ . From the magnitudes of  $w_n\eta_3$  and  $y_3$ , one may conclude that the closed-loop system is susceptible to noises.





**Fig. 15.31** Disturbance rejection of  $\mu$ -controller

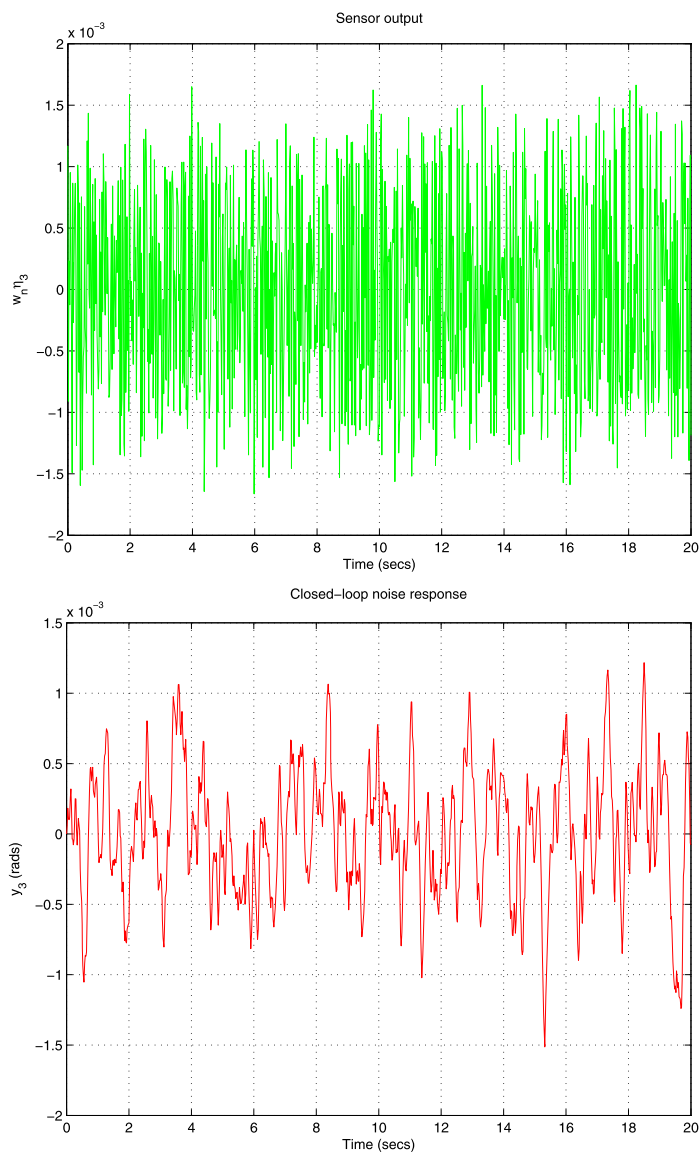
The closed-loop frequency responses of the uncertain closed-loop system are obtained by the M-file `frs_pend.m`. The singular values of the closed-loop transfer matrix with the  $\mu$ -controller are plotted in Fig. 15.33. The comparison with the plot shown in Fig. 15.23 reveals that the closed-loop bandwidth in the case of a  $\mu$ -controller is slightly smaller, which leads to a slower response and worse disturbance attenuation. In contrast to the case of  $\mathcal{H}_\infty$  controller, the closed-loop frequency responses do not have peaks.

The singular values of the transfer function matrix concerning disturbance rejection are shown in Fig. 15.34. We see that the disturbance attenuation is worst for frequencies around 1 rad/s.

From the singular values plot of the transfer function matrix concerning noise attenuation, shown in Fig. 15.35, we see that the influence of noises on the system output would be maximal for frequencies between 1 rad/s and 10 rad/s.

As mentioned earlier, the controller obtained by  $\mu$ -synthesis is initially of 80th order, which makes its implementation in practice very difficult. Therefore, it is necessary to reduce the controller order. For this purpose, we implement the M-file `red_pend.m` implementing the function `reduce`. This function allows us to reduce the controller order to 25. Further reduction of the controller order leads to deterioration of the closed-loop transient responses and would even cause the instability of the closed-loop system.

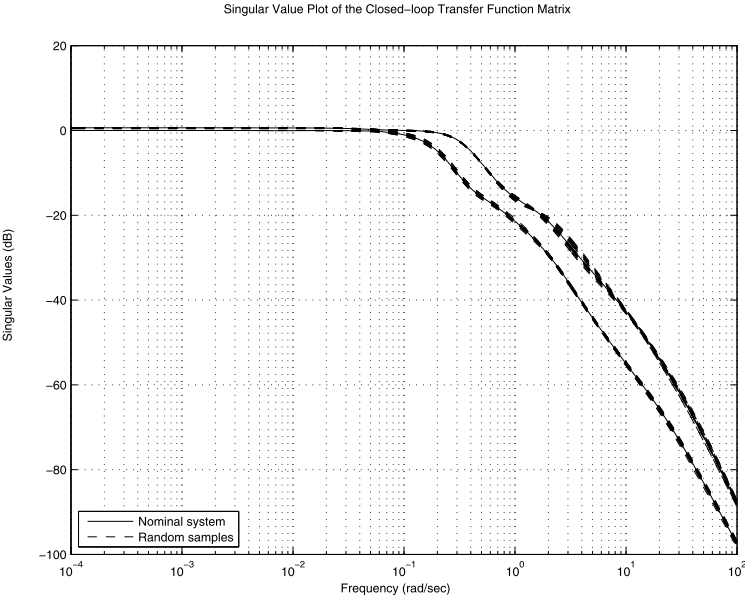
In Fig. 15.36 we compare the singular values frequency responses of the full-order and reduced-order  $\mu$ -controllers. Up to the frequency  $10^4$  rad/s the frequency plots of both controllers coincide with each other, which implies very similar performance of the closed-loop systems. In fact, the closed-loop transient responses with



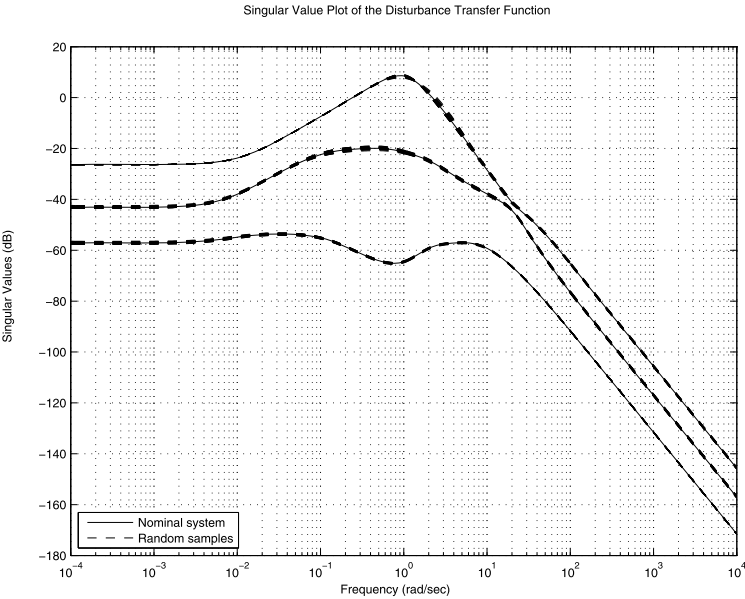
**Fig. 15.32** Sensor noise and closed-loop noise response

the full-order controller and those with the reduced-order controller (not included in the book) are practically undistinguishable.

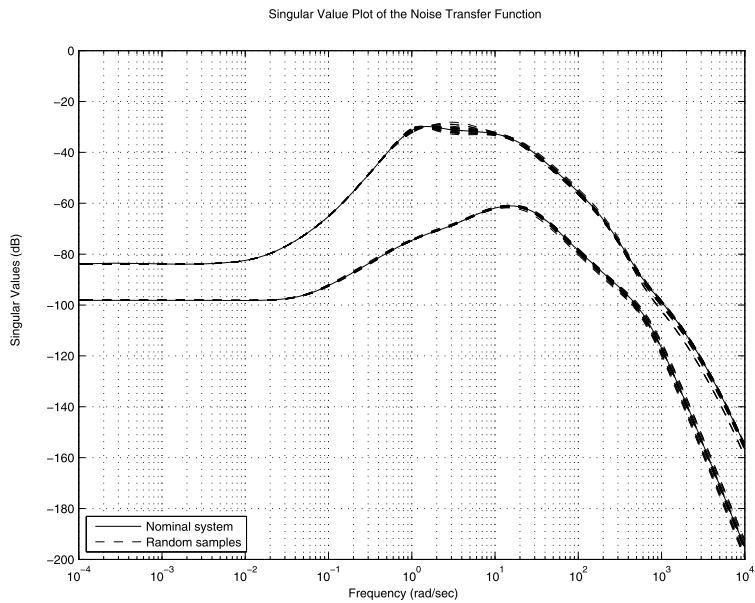
A discrete-time controller may be obtained by sampling the already designed continuous-time controller for a sampling frequency  $f_s = 1/T_s$ . This can be conducted by the M-file `dclPend.m`, which utilizes the function `c2d`. The resulting, sampled-data closed-loop system can be simulated by using the function `sdlSim`.



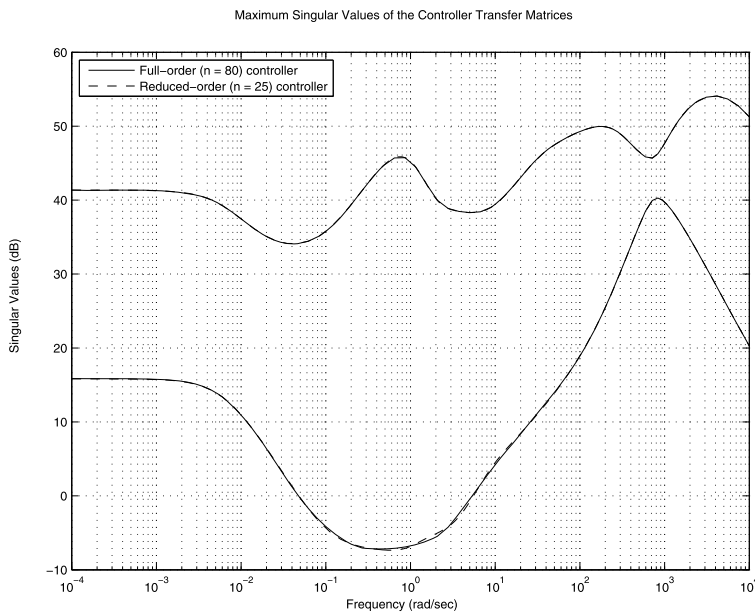
**Fig. 15.33** Closed-loop singular value plot



**Fig. 15.34** Singular values of the disturbance rejection transfer function matrix



**Fig. 15.35** Singular values of the noise-attenuation transfer function matrix



**Fig. 15.36** Frequency responses of the full- and reduced-order controllers

Since the inverted pendulum is a relatively slow system, the sampling frequency may be chosen to be small. For instance, if this frequency is 100 Hz then the transient responses of the sampled-data system are close to the corresponding responses of the continuous-time system.

## 15.7 Nonlinear System Simulation

The nonlinear, closed-loop system of the triple inverted pendulum is simulated by using the Simulink® models `c_pend.mdl` (for the continuous-time system) and `d_pend.mdl` (for the sampled-data system). Both models allow us to simulate the closed-loop system for different references, disturbances and noise signals. Both models utilize the M-file S-function `s_pend.m` that includes the nonlinear differential equations (15.1) of the triple inverted pendulum. The initial conditions of the pendulum are assigned in the M-file `inc_pend.m`.

The sampled-data model consists of the discretized controller, a 14-bit analogue-to-digital converter with maximum input voltage 5 V, and a 14-bit digital-to-analogue converter with maximum output voltage 5 V. It is assumed that the calculation of the control action requires no longer than one sampling period  $T_s$ .

Before simulating the system it is necessary to set the model parameters by using the M-file `init_c_pend.m` (in the continuous-time case) or the M-file `init_d_pend.m` (in the discrete-time case).

The simulation of the nonlinear system for small references and disturbances produces results that are very close to the results obtained for the linearized model.

The Simulink® model `c_pend.mdl` of the continuous-time, nonlinear, pendulum system is shown in Fig. 15.37.

In Fig. 15.38 we show the transient response of the continuous-time, nonlinear system for a reference  $r = [0 \ -0.1 \ 0.2]^T$ . The transient response is close to that of the linear system (Fig. 15.29).

Since the triple inverted pendulum is essentially nonlinear, the simulation results may differ in the case of large references and disturbances. To illustrate this point, in Fig. 15.39 we show the transient response of the continuous-time, nonlinear system for a reference  $r = [0 \ 0 \ 0.6]^T$ .

## 15.8 Conclusions

The experience gained in the design of the triple inverted pendulum control system makes it possible to make the following comments.

- The uncertainty model of the triple inverted pendulum system involves structured (parametric) as well as unstructured uncertainty. Some care is necessary in deriving the pendulum uncertainty model in order to minimize the number of the uncertain parameters.

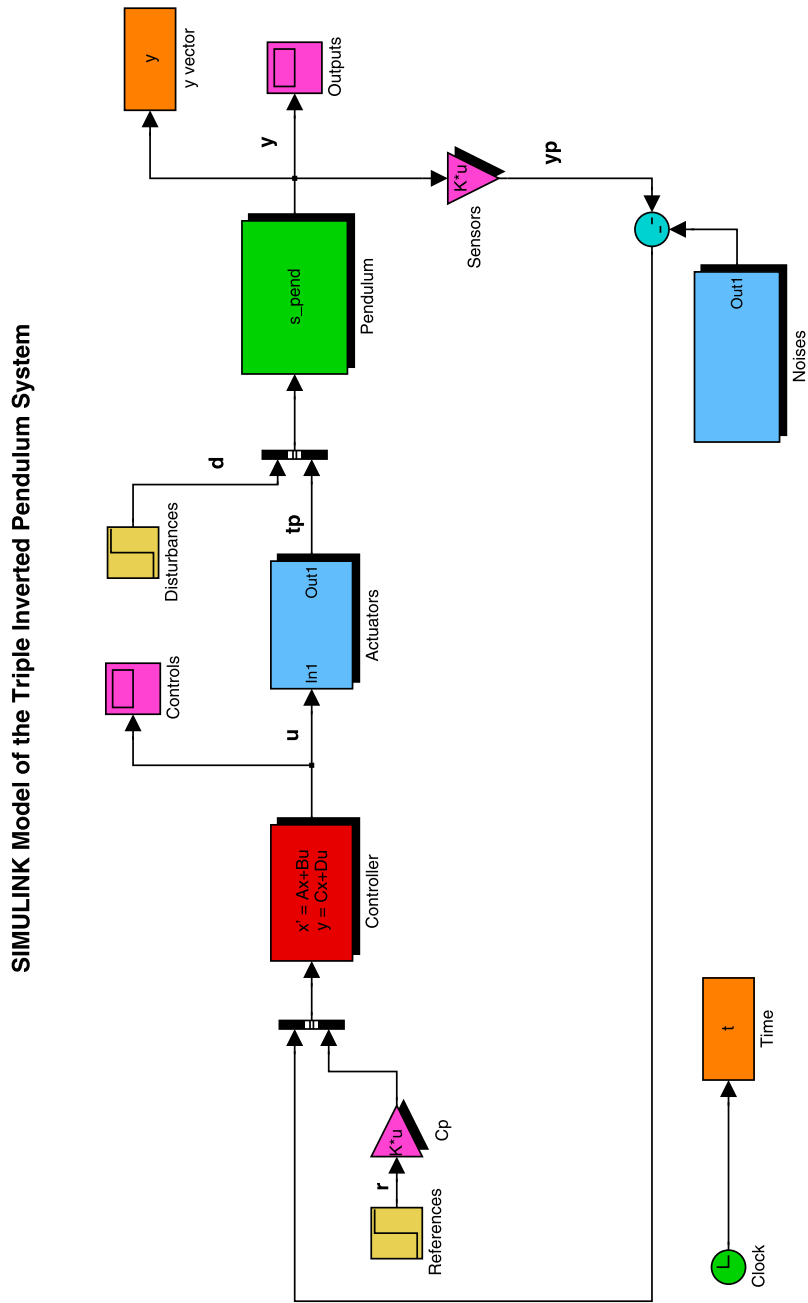


Fig. 15.37 Simulation model of the nonlinear system

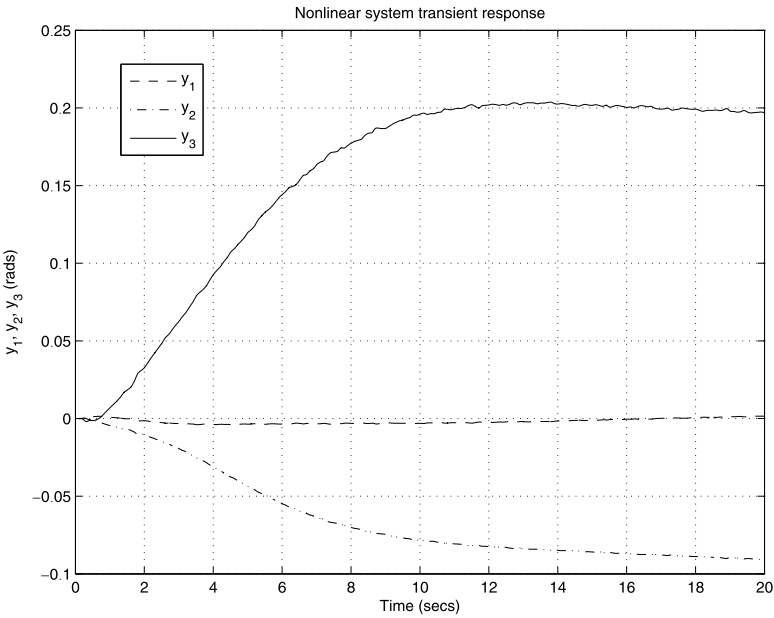


Fig. 15.38 Transient response of the nonlinear system—small reference

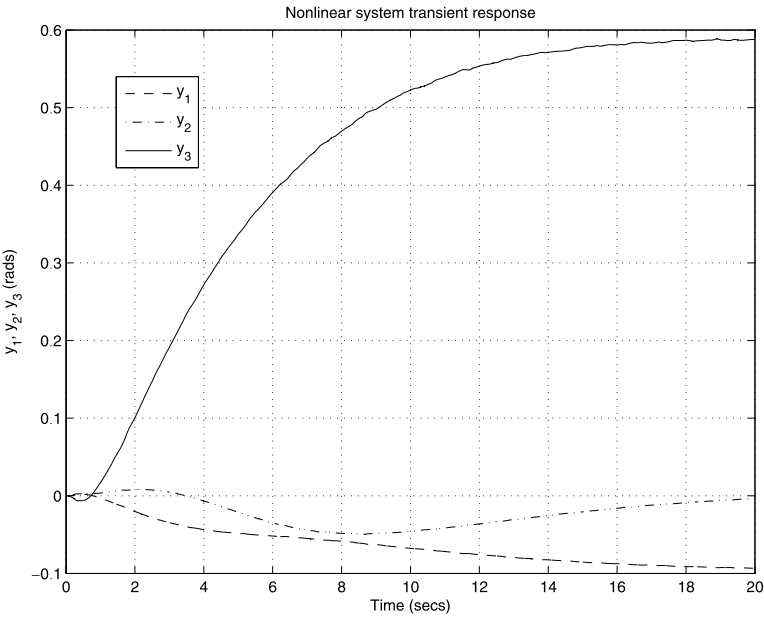


Fig. 15.39 Transient response of the nonlinear system—large reference

- The use of a 2-degree-of-freedom control structure allows us to obtain acceptable performance as well as robustness of the closed-loop system.
- The necessity to achieve robust stability and robust performance of the closed-loop system leads to decrease of the closed-loop bandwidth that in turn leads to slower response and worse disturbance attenuation. Thus, there is a trade-off between nominal performance and robustness of the closed-loop system. In the given case an acceptable trade-off is achieved by using a  $\mu$ -controller.
- The order of the  $\mu$ -controller is very high that thus requires controller order reduction to preserve the closed-loop performance and robustness while making implementation of the controller easier and more reliable.
- The triple inverted pendulum is an essentially nonlinear system and the results obtained by using the linearized model are valid only for sufficiently small arm angles and arm velocities.

## 15.9 Notes and References

The inverted pendulum is introduced into Control Engineering by the Japanese scientist Katsuhisa Furuta and his colleagues from Tokyo Institute of Technology [47] and is frequently called *Furuta pendulum*. As noted in [187] the inverted pendulum is frequently used as a good example to show the power of modern control theory. The design of inverted pendulum control systems is initially done by using state-space methods [122].

The single and double inverted pendulums are relatively easy to stabilize by using linear controllers [47, 91, 182]. The triple inverted pendulum is more difficult to control and it is very sensitive to torque disturbances, joint frictions and measurement noises [164]. The design of analogue and digital controllers for triple inverted pendulum systems has been considered in several papers (see, for instance, [48, 115]). References [37, 113] address the challenging problem of stabilizing a triple inverted pendulum, a cart system that uses only one actuator.

The robust control of different types of inverted pendulum is considered in several publications, see, for instance, [81, 114, 164, 167].



This page intentionally left blank

## Chapter 16

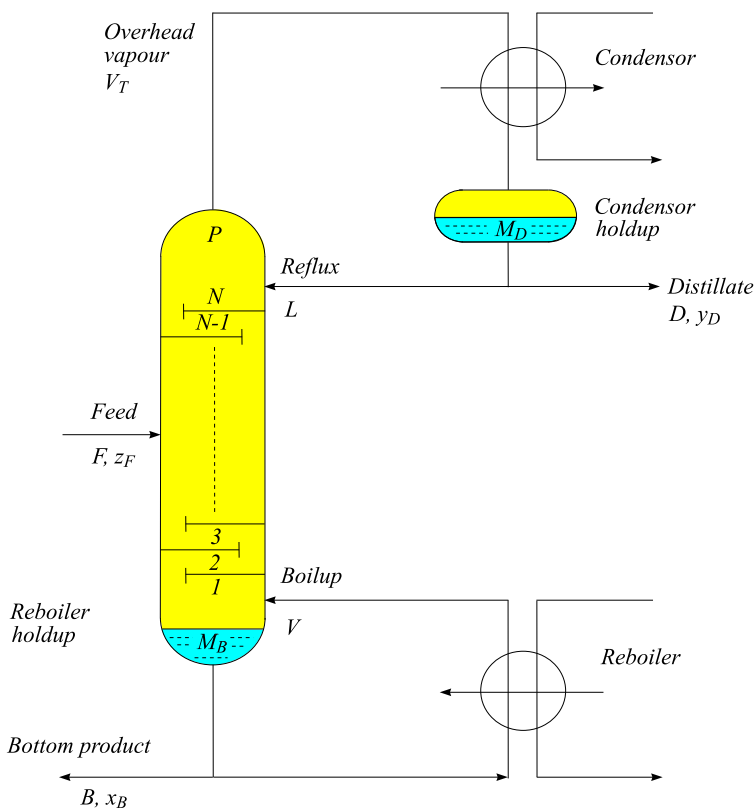
# Robust Control of a Distillation Column

In this chapter we present the design of a robust control system for a high-purity distillation column. The original nonlinear model of the column is of high order and it includes parametric gain and time-delay uncertainty. A low-order linearized distillation column model is used in the design of a 1-degree-of-freedom  $\mathcal{H}_\infty$  loop-shaping controller and a 2-degree-of-freedom (2DOF)  $\mu$ -controller. Both controllers ensure robust stability of the closed-loop system and fulfillment of a mixture of time-domain and frequency-domain specifications. A reduced order  $\mu$ -controller is then found that preserves the robust stability and robust performance of the closed-loop system. The simulation of the closed-loop system with the nonlinear distillation column model shows very good performance for different reference and disturbance signals as well as for different values of the uncertain parameters.

### 16.1 Introduction

Distillation is an important process in the separation and purification of chemicals. The process exploits the difference at boiling points of multicomponent liquids. The control of distillation columns is difficult, because the distillation process is highly nonlinear and the corresponding linearized models are often ill-conditioned around the operating point.

The aim of the design, presented in this chapter, is to find a controller that achieves robust stability and robust performance of the closed-loop control system of a high-purity distillation column. The original nonlinear model of the column is of 82nd order and it includes uncertainties in the form of parametric gains and time delay. The uncertainty model is considered in the form of an input multiplicative complex uncertainty. In our design exercises we try to achieve the desired performance of the closed-loop system using 1-degree-of-freedom  $\mathcal{H}_\infty$  loop-shaping design procedure and 2-degree-of-freedom  $\mu$ -synthesis/analysis method. The designs are based on a sixth-order linearized distillation column model. Both designed controllers ensure robust stability of the closed-loop system and achieve a mixed



**Fig. 16.1** The distillation column system

set of time-domain and frequency-domain specifications. We present several time-domain and frequency-domain characteristics of the corresponding closed-loop systems that makes possible the comparison of controllers efficiency. A 12th-order reduced-order  $\mu$ -controller is found that preserves the stability and performance of the closed-loop system in the presence of uncertainties. The simulation of the closed-loop system with this  $\mu$ -controller and with the nonlinear distillation column model is conducted in Simulink<sup>®</sup> and shows very good performance for different reference and disturbance signals as well as for different values of the uncertain parameters.

## 16.2 Dynamic Model of the Distillation Column

A typical two-product distillation column is shown in Fig. 16.1. The objective of the distillation column is to split the feed  $F$ , which is a mixture of a light and a heavy component with composition  $z_F$ , into a distillate product  $D$  with composition  $y_D$ ,

**Table 16.1** Column nomenclature

| Symbol                   | Description                                                                            |
|--------------------------|----------------------------------------------------------------------------------------|
| $F$                      | Feed rate [kmol/min]                                                                   |
| $z_F$                    | feed composition [mole fraction]                                                       |
| $q_F$                    | fraction of liquid in feed                                                             |
| $D$ and $B$              | distillate (top) and bottom product flowrate [kmol/min]                                |
| $y_D$ and $x_B$          | distillate and bottom product composition (usually of light component) [mole fraction] |
| $L$                      | reflux flow [kmol/min]                                                                 |
| $V$                      | boilup flow [kmol/min]                                                                 |
| $N$                      | number of stages (including reboiler)                                                  |
| $N_{\text{tot}} = N + 1$ | total number of stages (including condensor)                                           |
| $i$                      | stage number (1—bottom, $N_F$ —feed stage, $N_T$ —total condensor)                     |
| $L_i$ and $V_i$          | liquid and vapor flow from stage $i$ [kmol/min]                                        |
| $x_i$ and $y_i$          | liquid and vapor composition of light component on stage $i$                           |
| $M_i$                    | liquid holdup on stage $i$ [kmol] ( $M_B$ —reboiler, $M_D$ —condensor holdup)          |
| $\alpha$                 | relative volatility between light and heavy component                                  |
| $\tau_L$                 | time constant for liquid flow dynamics on each stage [min]                             |

**Table 16.2** Column data

| $N$ | $N_{\text{tot}}$ | $N_F$ | $F$ | $z_F$ | $q_F$ | $D$ | $B$ | $L$      | $V$      | $y_D$ | $x_B$ | $M_i$ | $\tau_L$ |
|-----|------------------|-------|-----|-------|-------|-----|-----|----------|----------|-------|-------|-------|----------|
| 40  | 41               | 21    | 1   | 0.5   | 1     | 0.5 | 0.5 | 2.706 29 | 3.206 29 | 0.99  | 0.01  | 0.5   | 0.063    |

which contains most of the light component, and a bottom product  $B$  with composition  $z_B$ , which contains most of the heavy component. For this aim, the column contains a series of trays that are located along its height. The liquid in the column flows through the trays from top to bottom, while the vapor in the column rises from bottom to top. The constant contact between the vapor and liquid leads to increasing concentration of the more-volatile component in the vapor, while simultaneously increasing concentration of the less volatile component in the liquid. The operation of the column requires that some of the bottom product is reboiled at a rate  $V$  to ensure the continuity of the vapor flow and some of the distillate is refluxed to the top tray at a rate  $L$  to ensure the continuity of the liquid flow.

The notations used in the derivation of the column model are summarized in Table 16.1 and the column data are given in Table 16.2.

The index  $i$  denotes the stages numbered from the bottom ( $i = 1$ ) to the top ( $i = N_{\text{tot}}$ ) of the column. Index  $B$  denotes the bottom product and  $D$  the distillate product. A particular high-purity distillation column with 40 stages (39 trays and a reboiler) plus a total condensor is considered.

The nonlinear model equations are:

1. Total material balance on stage  $i$

$$dM_i/dt = L_{i+1} - L_i + V_{i-1} - V_i$$

2. Material balance for the light component on each stage  $i$

$$d(M_i x_i)/dt = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i$$

This equation leads to the following expression for the derivative of the liquid mole fraction:

$$dx_i/dt = (d(M_i x_i)/dt - x_i(dM_i/dt))/M_i$$

3. Algebraic equations

The vapor composition  $y_i$  is related to the liquid composition  $x_i$  on the same stage through the algebraic vapor–liquid equilibrium

$$y_i = \alpha x_i / (1 + (\alpha - 1)x_i)$$

From the assumption of constant molar flows and no vapor dynamics, one obtains the following expression for the vapor flows:

$$V_i = V_{i-1}$$

The liquid flows depend on the liquid holdup on the stage above and the vapor flow as follows:

$$L_i = L0_i + (M_i - M0_i)/\tau_L + \lambda(V_{i-1} - V0_{i-1})$$

where  $L0_i$  [kmol/min] and  $M0_i$  [kmol] are the nominal values for the liquid flow and holdup on stage  $i$  and  $V0_i$  is the nominal boilup flow. If the vapor flow into the stage effects the holdup then the parameter  $\lambda$  is different from zero. For the column under investigation  $\lambda = 0$ .

The above equations apply at all stages except in the top (condensor), feed stage and bottom (reboiler).

1. For the feed stage,  $i = N_F$  (it is assumed that the feed is mixed directly into the liquid at this stage)

$$dM_i/dt = L_{i+1} - L_i + V_{i-1} - V_i + F$$

$$d(M_i x_i)/dt = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} - L_i x_i - V_i y_i + F z_F$$

2. For the total condensor,  $i = N_{\text{tot}}$  ( $M_{N_{\text{tot}}} = M_D$ ,  $L_{N_{\text{tot}}} = L_T$ )

$$dM_i/dt = V_{i-1} - L_i - D$$

$$d(M_i x_i)/dt = V_{i-1} - L_i x_i - D x_i$$

3. For the reboiler,  $i = 1$  ( $M_i = M_B$ ,  $V_i = V_B = V$ )

$$d(M_i x_i)/dt = L_{i+1} x_{i+1} - V_i y_i - B x_i$$

As a result, we obtain a nonlinear model of the distillation column of 82nd order. There are two states per tray, one representing the liquid composition and the other representing the liquid holdup. The model has four manipulated inputs ( $L_T$ ,  $V_B$ ,  $D$  and  $B$ ) and three disturbances ( $F$ ,  $z_F$  and  $q_F$ ).

In order to find a linear model of the distillation column it is necessary to have a steady-state operating point around which the column dynamics is to be linearized. However, the model contains two integrators, because the condenser and reboiler levels are not under control. To stabilize the column, we make use of the so called *LV-configuration of the distillation column* where we use  $D$  to control  $M_D$  and  $B$  to control  $M_B$ . This is done by two proportional controllers with both gains equal to 10.

The nonlinear model is linearized at the operating point given in Table 16.2 (the values of  $F$ ,  $L$ ,  $V$ ,  $D$ ,  $B$ ,  $y_D$ ,  $x_B$  and  $z_F$ ). These steady-state values correspond to an initial state where all liquid compositions are equal to 0.5 and the tray holdups are also equal to 0.5 [kmol]. The steady-state vector is obtained for  $t = 5000$  min by numerical integration of the nonlinear model equations of the LV-configuration given in the M-file `cola_lv.m`. The linearization is carried out by implementing the M-file `cola_lin`, which makes use of the equations given in the file `cola_lv_lin.m`. The 82nd-order, linear model is stored in the variable  $G_{4u}$  and has four inputs (the latter two are actually disturbances),

$$[L_T \quad V_B \quad F \quad z_F]$$

and two outputs,

$$[y_D \quad x_B]$$

Before reducing the model order, the model  $G_{4u}$  is scaled in order to make all inputs/disturbances and all outputs at about the same magnitude. This is done by dividing each variable by its maximum change, i.e.

$$u = U/U_{\max}; \quad y = Y/Y_{\max}$$

where  $U$ ,  $Y$  are the input and output of the model  $G_{4u}$  in original units,  $U_{\max}$ ,  $Y_{\max}$  are the corresponding maximum values allowed, and  $u$ ,  $y$  are the scaled variables. The scaling is achieved by using the input scaling matrix

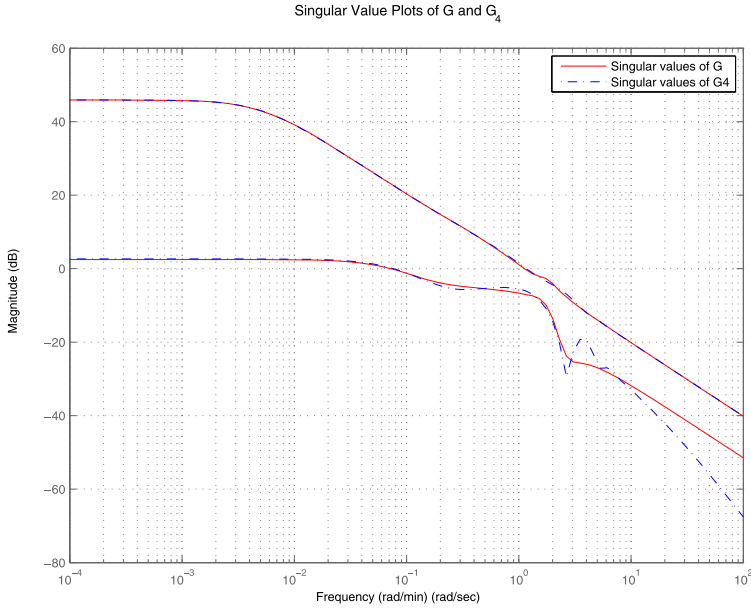
$$S_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$

and output scaling matrix

$$S_o = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$$

The scaled model is then found as  $G_4 = S_o G_{4u} S_i$ .

The final stage in selecting the column model is the order reduction of the scaled model  $G_4$ . This is done by using the function `reduce`. As a result, we obtain a sixth-order model saved in the variable  $G$ .



**Fig. 16.2** Singular values of  $G$  and  $G_4$

All commands for finding the sixth-order linear model of the distillation column are contained in the file `mod_col.m`.

The frequency responses of the singular values of  $G$  are compared with the singular values of the 82nd order linearized model  $G_4$  in Fig. 16.2. It is seen that the behavior of both models is close until the frequency becomes 2 rad/min.

### 16.3 Uncertainty Modeling

The uncertainties considered in the distillation column control systems are a gain uncertainty of  $\pm 20\%$  and a time delay of up to 1 min in each input channel. Thus, the uncertainty may be represented by the transfer matrix

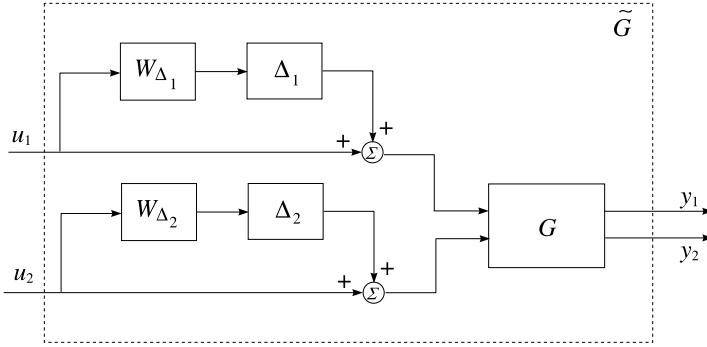
$$W_u = \begin{bmatrix} k_1 e^{-\theta_1 s} & 0 \\ 0 & k_2 e^{-\theta_2 s} \end{bmatrix}$$

where  $k_i \in [0.8 \ 1.2]$ ;  $\theta_i \in [0.0 \ 1.0]$ ;  $i = 1, 2$ . It is convenient to represent this uncertainty by an input multiplicative uncertainty, as shown in Fig. 16.3, with

$$\Delta = \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix}$$

where  $|\Delta_1| \leq 1$ ,  $|\Delta_2| \leq 1$ . The uncertainty weighting function

$$W_\Delta = \begin{bmatrix} W_{\Delta_1} & 0 \\ 0 & W_{\Delta_2} \end{bmatrix}$$



**Fig. 16.3** Distillation column with input multiplicative uncertainty

is determined in the following way.

Denote by  $\overline{W}_{u_i} = 1$  the nominal transfer function in the  $i$ th channel for  $k_i = 1$  and  $\Theta_i = 0$ ;  $i = 1, 2$ .

According to Fig. 16.3 we have

$$W_{u_i} = (1 + W_{\Delta_i} \Delta_i) \overline{W}_{u_i}, \quad i = 1, 2$$

Taking into account that  $|\Delta_i| \leq 1$  it follows that the relative uncertainty should satisfy

$$\frac{|W_{u_i}(j\omega) - \overline{W}_{u_i}(j\omega)|}{|\overline{W}_{u_i}(j\omega)|} \leq |W_{\Delta_i}(j\omega)|, \quad i = 1, 2$$

where  $W_{u_i}(j\omega) = k_i e^{-j\omega\Theta_i} = k_i(\cos(-\omega\Theta_i) + j \sin(-\omega\Theta_i))$ . In this way, to choose the uncertainty weight  $W_{\Delta_i}$  is equivalent to determining an upper bound of the frequency response of the relative uncertainty

$$\frac{|W_{u_i}(j\omega) - \overline{W}_{u_i}(j\omega)|}{|\overline{W}_{u_i}(j\omega)|} = \sqrt{(k_i \cos(\omega\Theta_i) - 1)^2 + (k_i \sin(\omega\Theta_i))^2}.$$

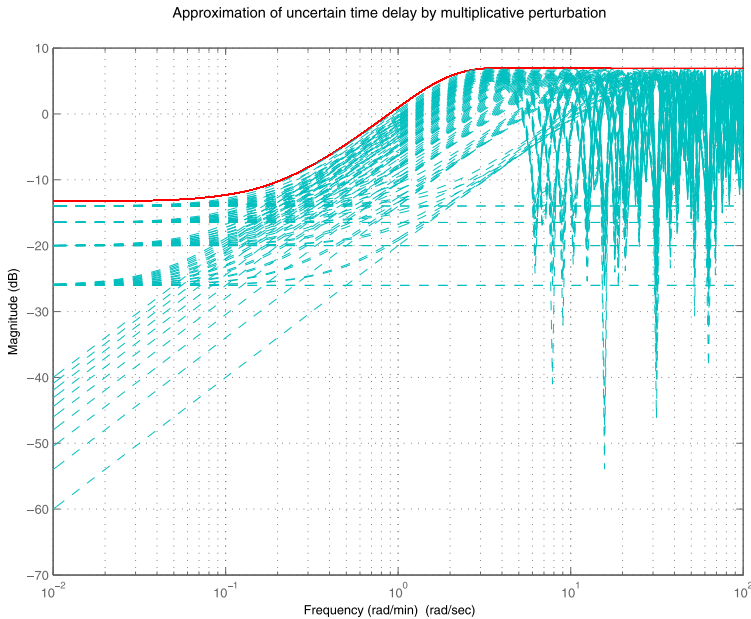
The frequency responses of the relative uncertainty

$$\frac{|W_{u_i}(j\omega) - \overline{W}_{u_i}(j\omega)|}{|\overline{W}_{u_i}(j\omega)|}$$

are computed by the file `unc_col.m` and shown in Fig. 16.4. These responses are then approximated by third-order transfer functions using the file `wfit.m`. As a result, one obtains

$$W_{\Delta_i} = \frac{2.2138s^3 + 15.9537s^2 + 27.6702s + 4.9050}{1.0000s^3 + 8.3412s^2 + 21.2393s + 22.6705}, \quad i = 1, 2$$





**Fig. 16.4** Approximation of the uncertain time delay

## 16.4 Closed-Loop System Performance Specifications

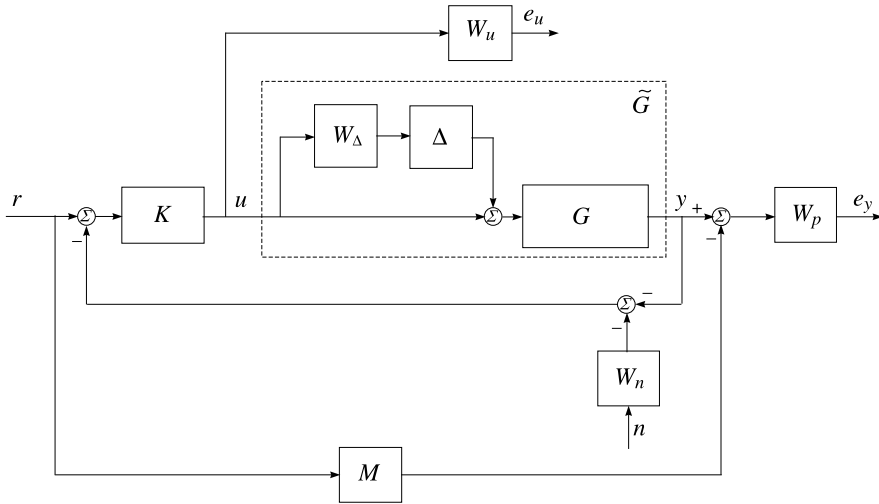
The aim of the distillation column control-system design is to determine a controller that meets robust stability and robust performance specifications for the LV configuration. We try to satisfy these requirements by using a 1-degree-of-freedom  $\mathcal{H}_\infty$  loop-shaping controller and a 2-degree-of-freedom  $\mu$ -controller. In the given case, the robust stability means guaranteed closed-loop stability for all  $0.8 \leq k_1, k_2 \leq 1.2$  and  $0 \leq \Theta_1, \Theta_2 \leq 1$  min. The time-domain specifications are given in terms of step-response requirements, which must be met for all values of  $k_1, k_2, \Theta_1$  and  $\Theta_2$ . Specifically, for a unit step command to the first input channel at  $t = 0$ , the scaled plant outputs  $y_1$  (tracking) and  $y_2$  (interaction) should satisfy:

- $y_1(t) \geq 0.9$  for all  $t \geq 30$  min;
- $y_1(t) \leq 1.1$  for all  $t$ ;
- $0.99 \leq y_1(\infty) \leq 1.01$ ;
- $y_2(t) \leq 0.5$  for all  $t$ ;
- $-0.01 \leq y_2(\infty) \leq 0.01$ .

Correspondingly, similar requirements should be met for a unit step command at the second input channel.

In addition, the following frequency-domain specification should be met:

- $\bar{\sigma}(\hat{K}_y \hat{S})(j\omega) < 316$ , for each  $\omega$ , where  $\hat{K}_y$  denotes the feedback part of the unscaled controller. (Here and latter, a variable with a hat refers to the case of un-



**Fig. 16.5** Closed-loop interconnection structure of the distillation column system with 1DOF controller

scaled plant.) This specification is included mainly to avoid saturation of the plant inputs.

- $\bar{\sigma}(\hat{G}\hat{K}_y)(j\omega) < 1$ , for  $\omega \geq 150$ ; or  $\bar{\sigma}(\hat{K}_y\hat{S})(j\omega) \leq 1$ , for  $\omega \geq 150$ .

In the above,  $\bar{\sigma}$  denotes the largest singular value, and  $\hat{S} = (I + \hat{G}\hat{K}_y)^{-1} < 1$  is the sensitivity function for  $\hat{G}$ .

The block diagrams of the closed-loop systems with 1-degree-of-freedom and 2-degree-of-freedom controllers, incorporating the design requirements consideration represented by weights, are shown in Figs. 16.5 and 16.6, respectively. The plant  $\tilde{G}$ , enclosed by the dashed rectangle, consists of the nominal scaled model  $G$  plus the input multiplicative uncertainty. The controller  $K$  implements a feedback from outputs  $y_D$  and  $x_B$  and a feedforward from the reference signal  $r$ . The measurement of the distillate and bottom products composition is corrupted by the noise  $n$ . The desired dynamics of the closed-loop system is sought by implementation of a suitably chosen model  $M$ . The model  $M$  represents the desired dynamic behavior of the closed-loop system from the reference signal to the outputs. The usage of a model of the desired dynamics allows us to take easily into account the design specifications.

The transfer function matrix of the model  $M$  is selected as

$$M = \begin{bmatrix} \frac{1}{Ts^2 + 2\xi Ts + 1} & 0 \\ 0 & \frac{1}{Ts^2 + 2\xi Ts + 1} \end{bmatrix}$$

The coefficients of the transfer functions ( $T = 6, \xi = 0.8$ ) in both channels of the model are chosen such as to ensure an overdamped response with a settling time of about 30 min. The off-diagonal elements of the transfer matrix are set as zeros in order to minimize the interaction between the channels.



The frequency response of the model  $M$  is shown in Fig. 16.7.

Consider first the closed-loop system with 1-degree-of-freedom controller. It is easy to show that

$$\begin{bmatrix} e_p \\ e_u \end{bmatrix} = \begin{bmatrix} W_p(T - M) & -W_p T W_n \\ W_u K S & -W_u K S W_n \end{bmatrix} \begin{bmatrix} r \\ n \end{bmatrix}$$

where  $S = (I + \tilde{G}K)^{-1}$  is the sensitivity function for the scaled plant,  $T = (I + \tilde{G}K)^{-1}\tilde{G}K$  is the complementary sensitivity function and  $\tilde{G} = G(I + W_\Delta\Delta)$  is the uncertain, scaled plant model.

The performance objective is to satisfy

$$\left\| \begin{bmatrix} W_p(T - M) & -W_p T W_n \\ W_u K S & -W_u K S W_n \end{bmatrix} \right\|_\infty < 1 \quad (16.1)$$

for each uncertain  $\tilde{G}$ .

Let the scaled, 2-degree-of-freedom controller be partitioned as

$$K(s) = \begin{bmatrix} K_y(s) & K_r(s) \end{bmatrix}$$

where  $K_y$  is the feedback part of the controller and  $K_r$  is the pre-filter part. Then it is possible to show that

$$\begin{bmatrix} e_p \\ e_u \end{bmatrix} = \begin{bmatrix} W_p(S\tilde{G}K_r - M) & -W_p T W_n \\ W_u(I + K_y\tilde{G})^{-1}K_r & -W_u K_y S W_n \end{bmatrix} \begin{bmatrix} r \\ n \end{bmatrix}$$

where  $S = (I + \tilde{G}K_y)^{-1}$  is the sensitivity function and  $T = (I + \tilde{G}K_y)^{-1}\tilde{G}K_y$  is the complementary sensitivity function.

The performance objective in the given case is to satisfy

$$\left\| \begin{bmatrix} W_p(S\tilde{G}K_r - M) & -W_p T W_n \\ W_u(I + K_y\tilde{G})^{-1}K_r & -W_u K_y S W_n \end{bmatrix} \right\|_\infty < 1 \quad (16.2)$$

for each uncertain  $\tilde{G}$ .

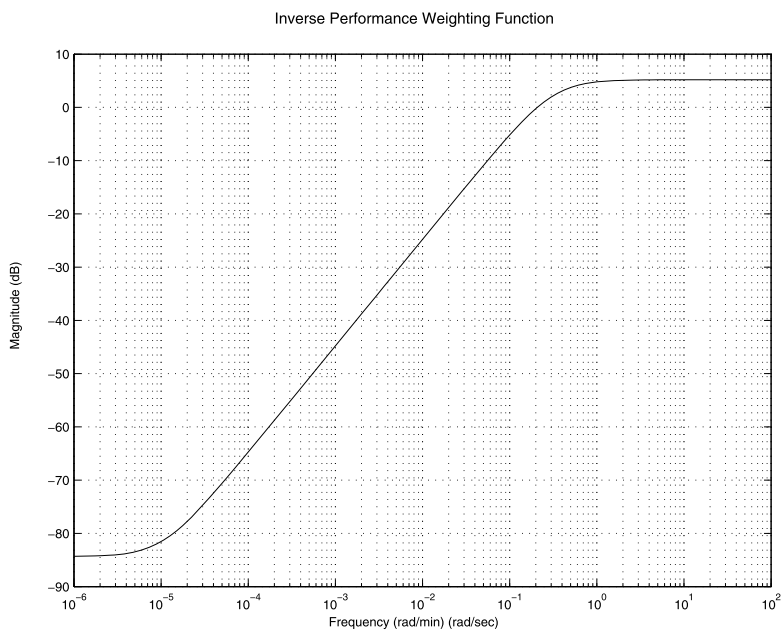
The performance and control action weighting functions for both type of the controller are chosen as

$$W_p = \begin{bmatrix} 0.55 \frac{9.5s+3}{9.5s+10^{-4}} & 0.3 \\ 0.3 & 0.55 \frac{9.5s+3}{9.5s+10^{-4}} \end{bmatrix}, \quad W_u = \begin{bmatrix} 0.87 \frac{s+1}{0.01s+1} & 0 \\ 0 & 0.87 \frac{s+1}{0.01s+1} \end{bmatrix}$$

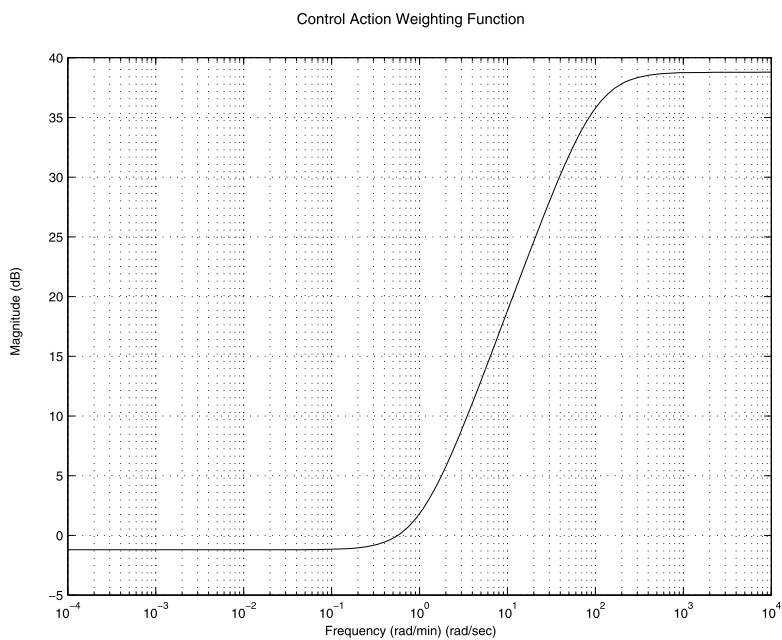
The implementation of the performance weighting function  $W_p$  aims to ensure closeness of the system dynamics to the model over the low-frequency range. Note that this function contains nonzero off-diagonal elements, which make it easier to meet the time-domain specifications. A small constant equal to  $10^{-4}$  is added in the denominator in each channel to make the design problem regular.

The usage of the control weighting function  $W_u$  allows us to limit the magnitude of control actions over the specified frequency range ( $\omega \geq 150$ ).

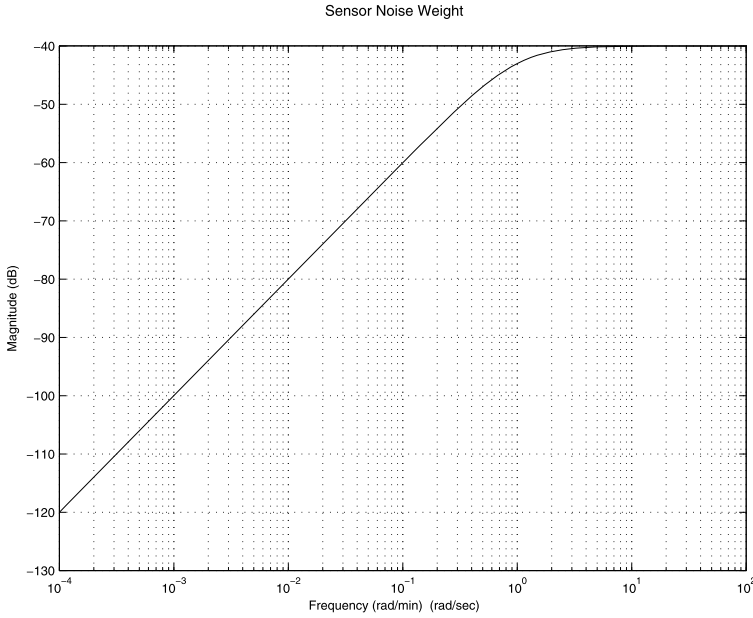
The magnitude plot of the inverse of the performance weighting function  $W_p$  is shown in Fig. 16.8 and the magnitude plot of the control weighting function is shown in Fig. 16.9.



**Fig. 16.8** Inverse of performance weighting function



**Fig. 16.9** Control-action weighting function



**Fig. 16.10** Noise weighting function

The noise shaping filter

$$W_n = \begin{bmatrix} 10^{-2} \frac{s}{s+1} & 0 \\ 0 & 10^{-2} \frac{s}{s+1} \end{bmatrix}$$

is determined according to the spectral contents of the sensor noises accompanying the measurement of the distillate and bottom product composition.

The magnitude plot of the noise shaping filter is shown in Fig. 16.10.

The model transfer function, the performance and control weighting functions as well as the noise shaping filter are all set in the file `wts_col.m`.

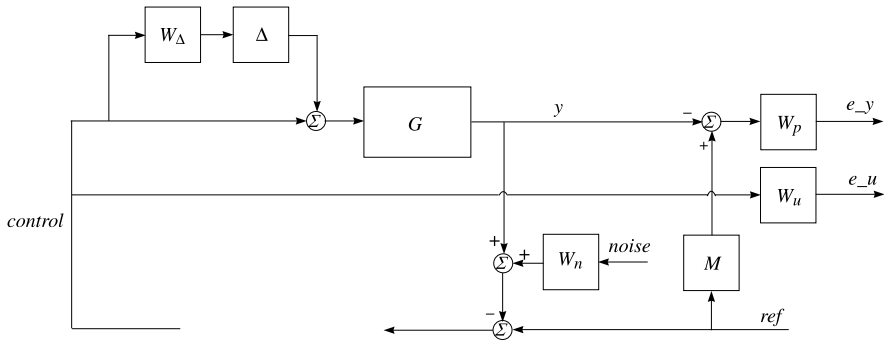
## 16.5 Open-Loop and Closed-Loop System Interconnections

The open-loop system interconnections for both types of controller are obtained by the M-file `olp_col`.

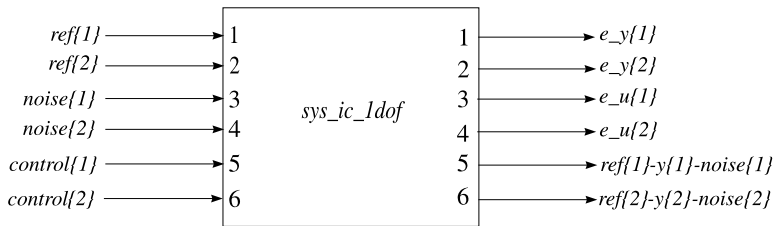
The internal structure of the six-input, six-output 22nd-order open-loop system with 1-degree-of-freedom controller, which is saved as the variable `sys_ic_1dof`, is shown in Fig. 16.11. The references and the noises are saved as the variables `ref` and `noise`, respectively, and the controls as the variable `control`.

All variables have two elements (i.e. they are 2-dimensional vectors).

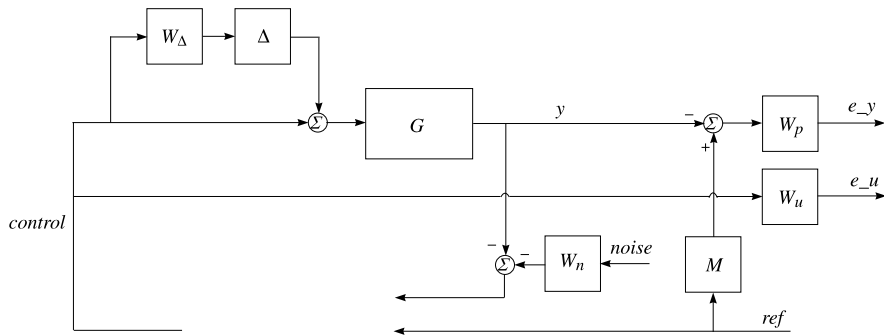
The schematic diagram showing the specific input/output ordering for the variable `sys_ic_1dof` is given in Fig. 16.12.



**Fig. 16.11** Open-loop interconnection structure of the distillation column system for 1DOF controller



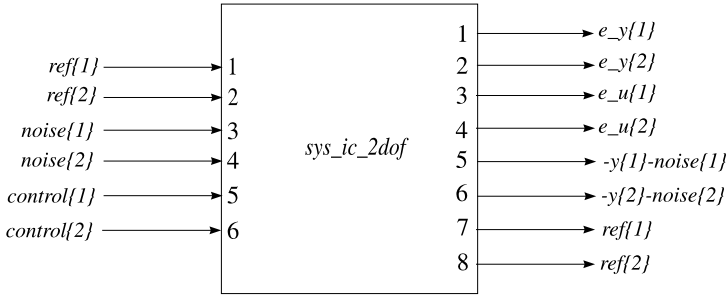
**Fig. 16.12** Schematic diagram of the open-loop interconnection for 1DOF controller



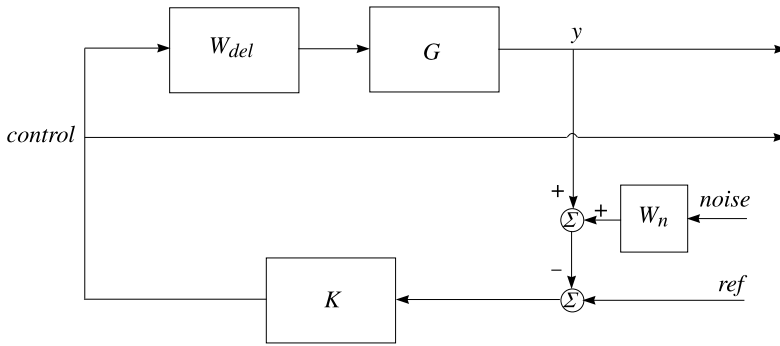
**Fig. 16.13** Open-loop interconnection structure of the distillation column system for 2DOF controller

The internal structure of the six-input, eight-output 22nd-order open-loop system with 2-degree-of-freedom controller, which is saved as the variable `sys_ic_2dof`, is shown in Fig. 16.13.

The schematic diagram showing the specific input/output ordering for the variable `sys_ic_2dof` is given in Fig. 16.14.



**Fig. 16.14** Schematic diagram of the open-loop interconnection for 2DOF controller



**Fig. 16.15** Closed-loop interconnection structure of the distillation column system for 1DOF controller

The block-diagram used in the simulation of the closed-loop system with 1-degree-of-freedom controller is shown in Fig. 16.15. The corresponding closed-loop interconnection is saved in the variable `clp_ic`. The uncertain delays are represented by the transfer function matrix

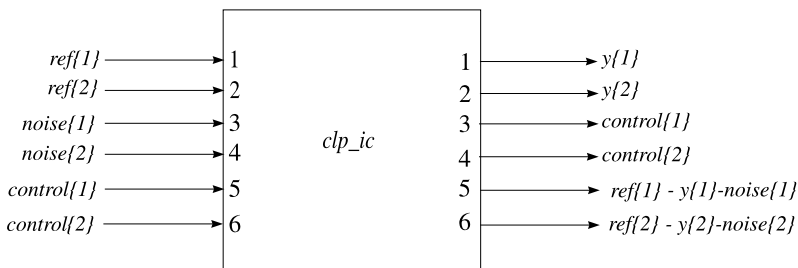
$$W_{\text{del}} = \begin{bmatrix} k_1 \text{del}_1 & 0 \\ 0 & k_2 \text{del}_2 \end{bmatrix}$$

where  $k_i \in [0.8 \ 1.2]$   $i = 1, 2$  and  $\text{del}_i$  is a six order Padé approximation of the time delay  $e^{-\Theta_i s}$ ,  $\Theta_i \in [0.0 \ 1.0]$ ;  $i = 1, 2$ .

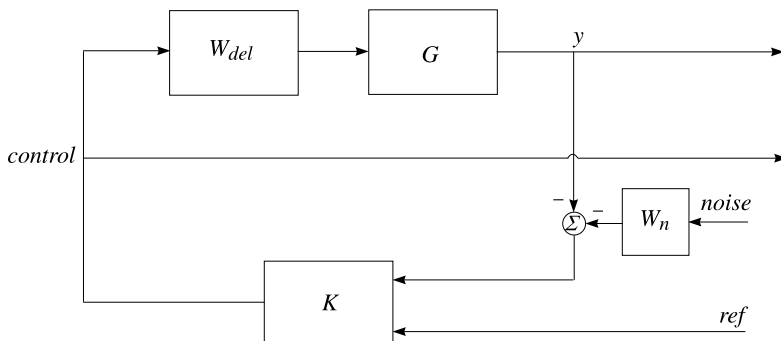
The schematic diagram showing the specific input/output ordering for the variable `clp_ic` is shown in Fig. 16.16.

The block-diagram used in the simulation of the closed-loop system with 2-degree-of-freedom controller, is shown in Fig. 16.17, and the schematic diagram showing the specific input/output ordering for the variable `clp_ic` is shown in Fig. 16.18.

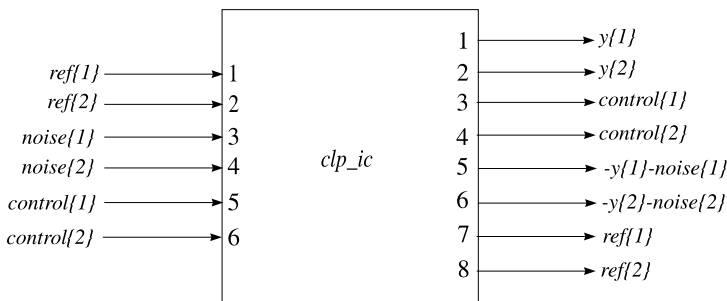




**Fig. 16.16** Schematic diagram of the closed-loop interconnection for 1DOF controller



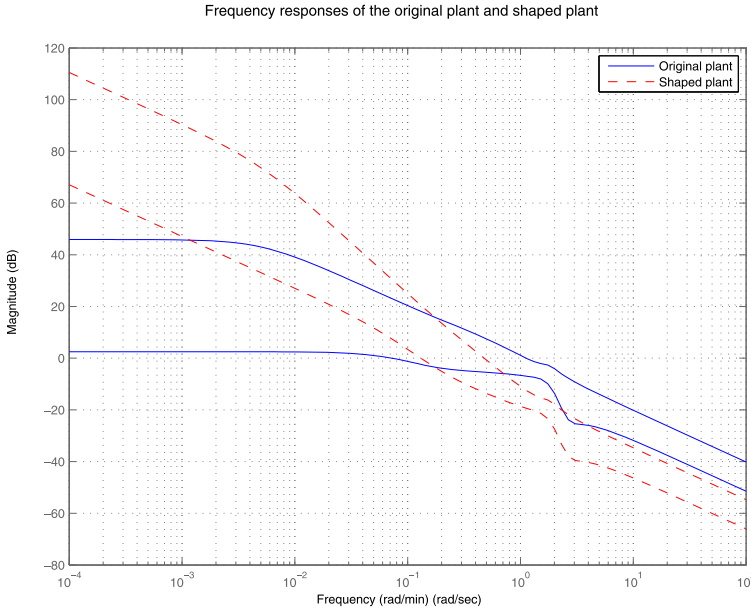
**Fig. 16.17** Closed-loop interconnection structure of the distillation column system for 2DOF controller



**Fig. 16.18** Schematic diagram of the closed-loop interconnection for 2DOF controller

## 16.6 Controller Design

Successful design of the distillation column control system may be obtained by using the  $\mathcal{H}_\infty$  loop-shaping design procedure (LSDP) and the  $\mu$ -synthesis. Note that in the case of LSDP we do not use the performance specifications implemented in the case of  $\mu$ -synthesis. Instead of these specifications we use a pre-filter  $W_1$



**Fig. 16.19** Singular values of the original system and shaped system

and a post-filter  $W_2$  in order to shape appropriately the open-loop transfer function  $W_1 G W_2$ .

### 16.6.1 Loop-Shaping Design

In the present case, we choose a pre-filter with transfer function

$$W_1 = \begin{bmatrix} 1.7 \frac{1.1s+1}{10s} & 0 \\ 0 & 1.7 \frac{1.1s+1}{10s} \end{bmatrix}.$$

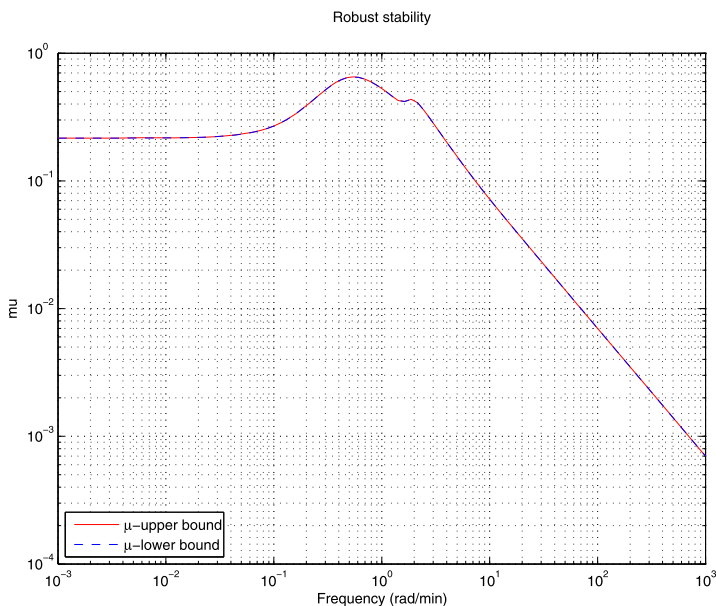
The choice of the gain equal to 1.7 is done to ensure a sufficiently small steady-state error. Larger gain leads to smaller steady-state errors but worse transient response.

The post-filter is taken simply as  $W_2 = I_2$ .

The singular value plots of the original and shaped systems are shown in Fig. 16.19.

The design of 1-degree-of-freedom LSDP controller is done by using the M-file `lsh_col.m`, which implements the function `ncfsyn`. The controller obtained is of order nine.

The robust stability analysis of the closed-loop system is done by the file `mu_col`. As a result one obtains the following report:



**Fig. 16.20** Robust stability for loop-shaping controller

```
report =
```

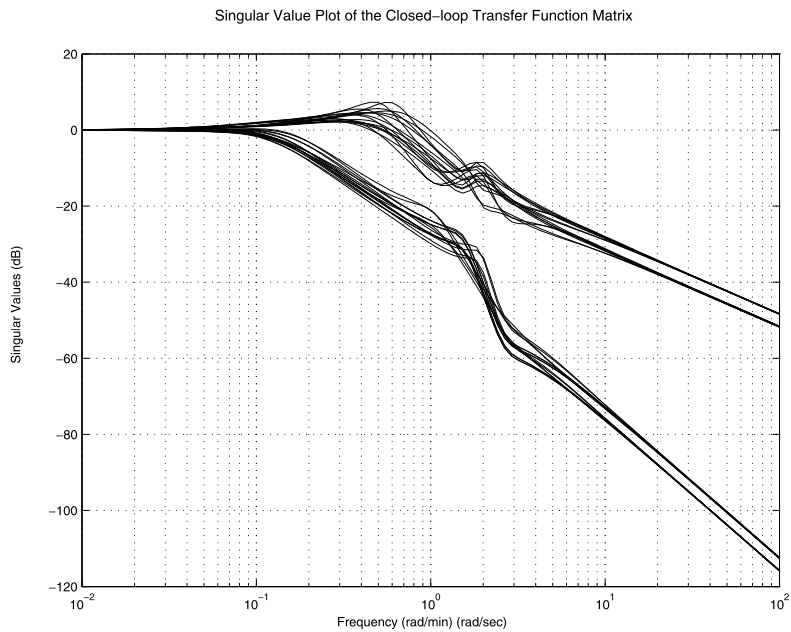
```
Uncertain System is robustly stable to modeled uncertainty.
-- It can tolerate up to 154% of the modeled uncertainty.
-- A destabilizing combination of 154% of the modeled
 uncertainty exists,
 causing an instability at 0.534 rad/s.
-- Sensitivity with respect to uncertain element ...
'Delta_1' is 42%. Increasing 'Delta_1' by 25% leads
 to a 11% decrease in the margin.
'Delta_2' is 67%. Increasing 'Delta_2' by 25% leads
 to a 17% decrease in the margin.
```

The frequency response plot of the structured value  $\mu$  with respect to the robust stability is shown in Fig. 16.20. The closed-loop system preserves stability for all perturbations with norm less than 1.54. As usual, the requirements for nominal performance and robust performance are not fulfilled with this controller.

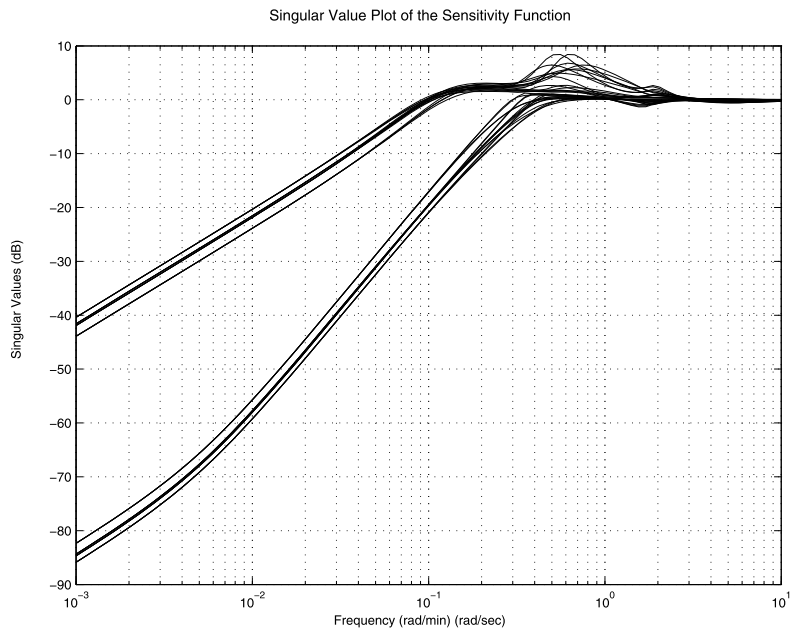
The closed-loop frequency responses are obtained by using the file `frs_col.m`.

The singular value plot of the unscaled closed-loop system transfer function is shown in Fig. 16.21. Both low-frequency gains are equal to 1 that ensures zero steady-state errors in both channels.

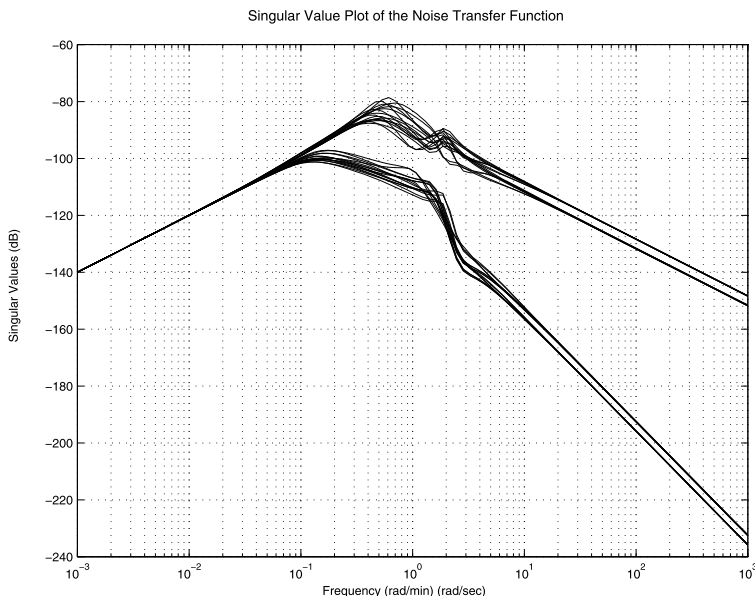
In Fig. 16.22 we show the singular-value plot of the unscaled sensitivity function  $\hat{S}$ .



**Fig. 16.21** Frequency response of the closed-loop system with loop-shaping controller



**Fig. 16.22** Frequency responses of the sensitivity function



**Fig. 16.23** Frequency response to the noises

The singular value plots of the transfer function matrix with respect to the noises (Fig. 16.23) show that the noises are attenuated by at least a factor of  $10^4$  times at the system output.

The singular-value plots of the transfer function matrices  $\hat{G}\hat{K}_y$  and  $\hat{K}_y\hat{S}$  are shown in Figs. 16.24 and 16.25, respectively. The maximum of the largest singular value of  $\hat{G}\hat{K}_y$  is far less than 1 for  $\omega \geq 150$  and the maximum of the largest singular value of  $\hat{K}_y\hat{S}$  is less than 200 so that the corresponding frequency-domain specification is met.

In Figs. 16.26, 16.27, 16.28 and 16.29 we show the transient responses of the scaled closed-loop system obtained by the file `mcs_col.m` for different values of the uncertain gain and time delay. The time-domain specification is met and the closed-loop system transient response has a small settling time.

The control action in the closed-loop system for the same variations of the uncertain parameters is shown in Figs. 16.30, 16.31, 16.32 and 16.33.

## 16.6.2 $\mu$ -Synthesis

Let us denote by  $P(s)$  the transfer function matrix of the six-input, eight-output open-loop system consisting of the distillation column model plus the weighting functions and let the block structure  $\Delta_P$  is defined as

$$\Delta_P := \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_F \end{bmatrix} : \Delta \in \mathcal{C}^{2 \times 2}, \Delta_F \in \mathcal{C}^{4 \times 4} \right\}$$

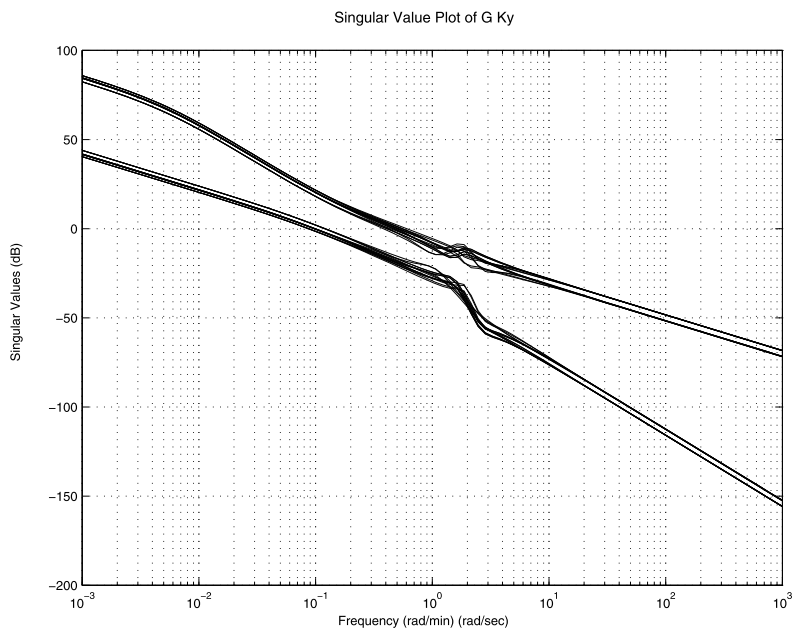


Fig. 16.24 Singular-value plot of  $\hat{G} \hat{K}_y$

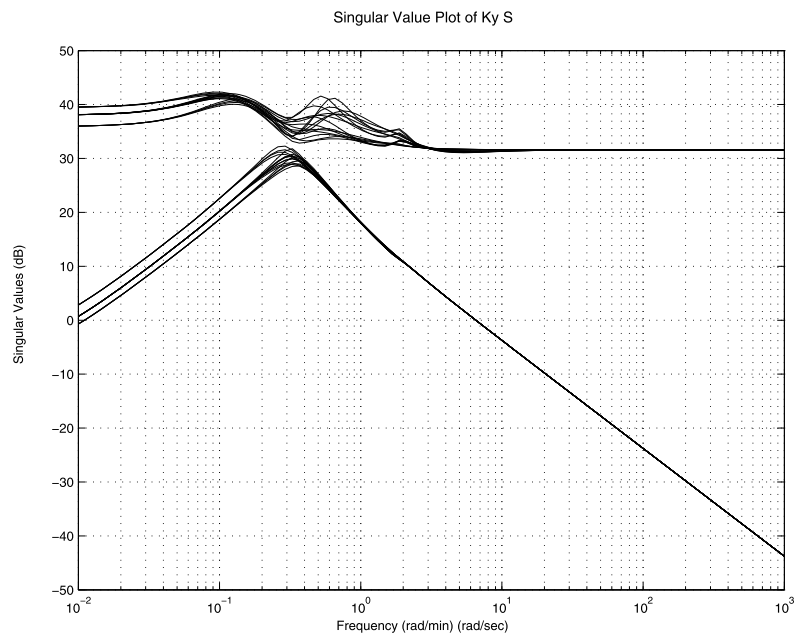


Fig. 16.25 Singular-value plot of  $\hat{K}_y \hat{S}$

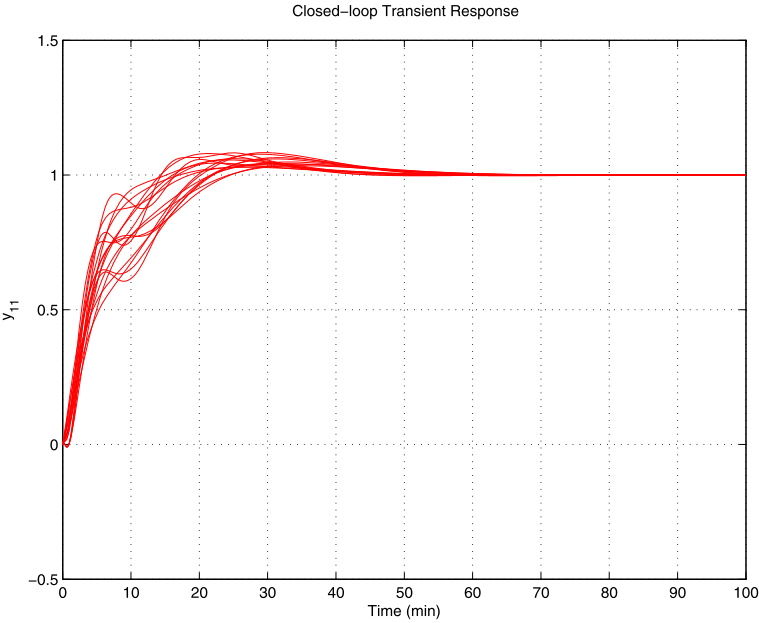


Fig. 16.26 Perturbed transient response  $y_{11}$  for loop-shaping controller

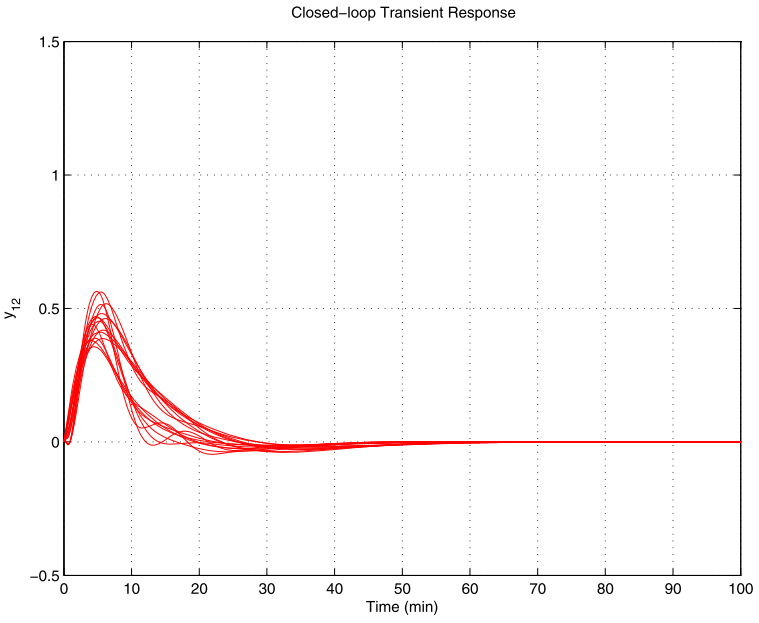


Fig. 16.27 Perturbed transient response  $y_{12}$  for loop-shaping controller

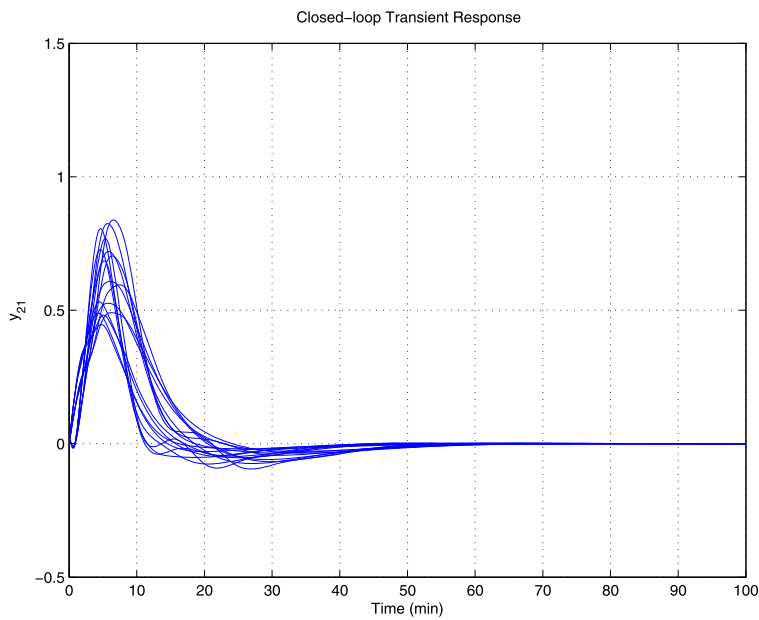


Fig. 16.28 Perturbed transient response  $y_{21}$  for loop-shaping controller

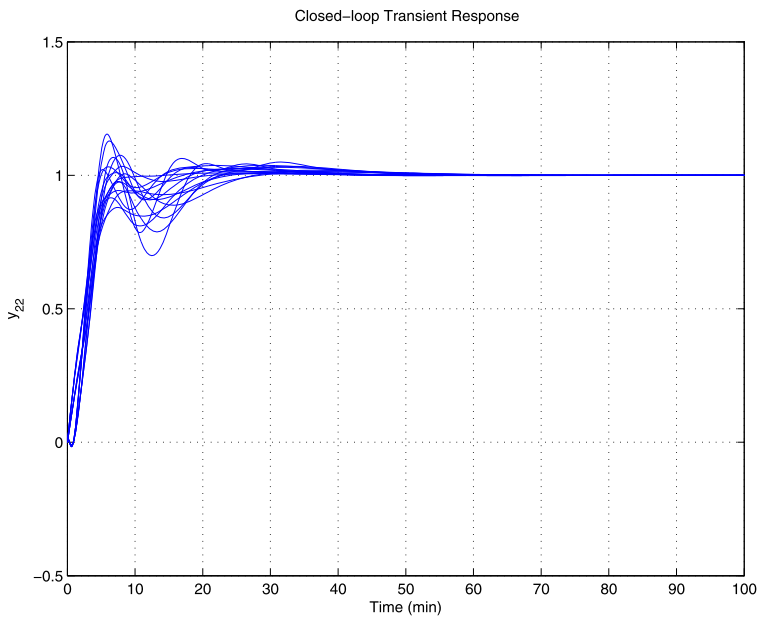
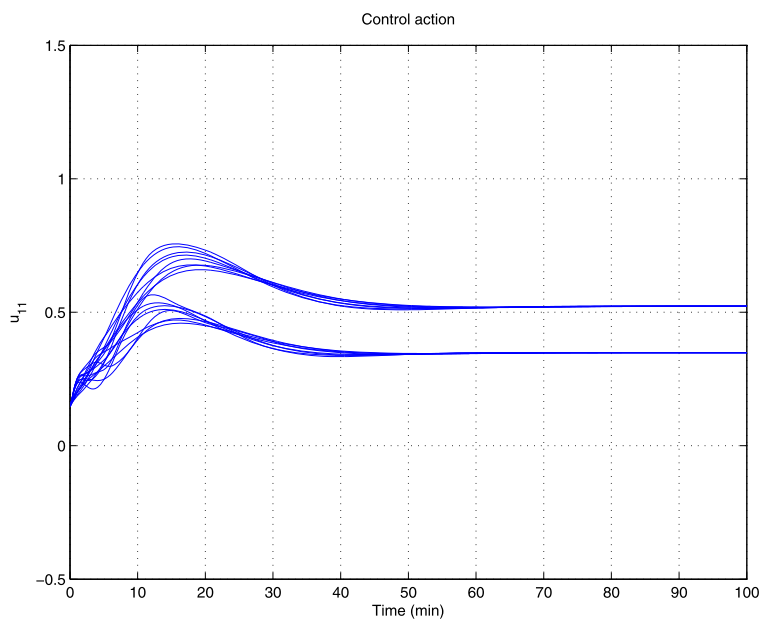
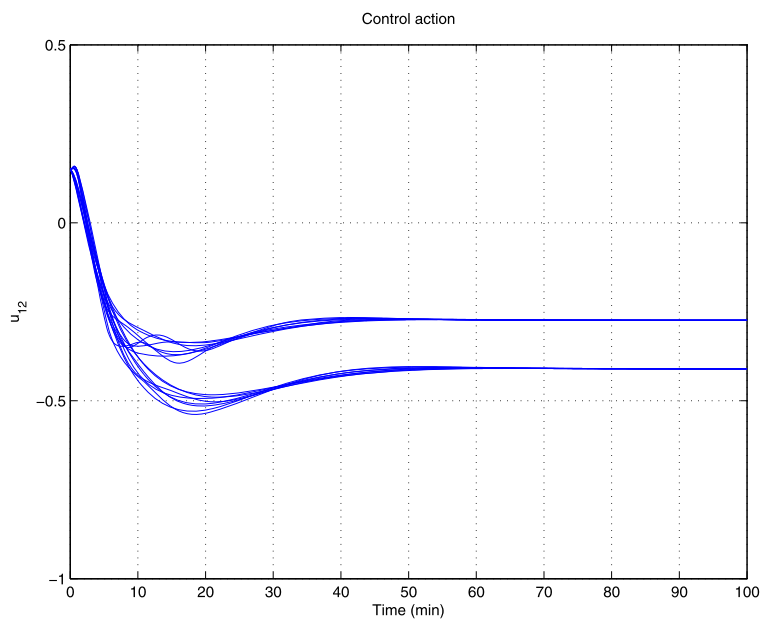


Fig. 16.29 Perturbed transient response  $y_{22}$  for loop-shaping controller





**Fig. 16.30** Perturbed control action  $u_{11}$  for loop-shaping controller



**Fig. 16.31** Perturbed control action  $u_{12}$  for loop-shaping controller

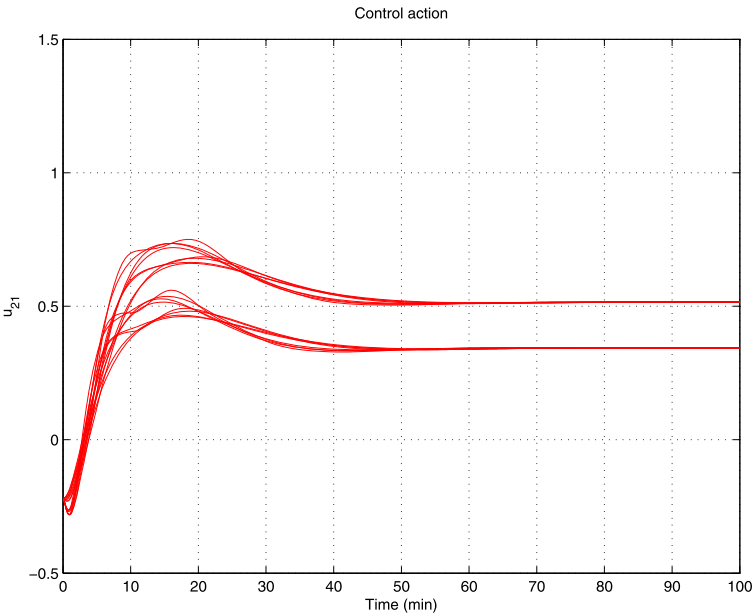


Fig. 16.32 Perturbed control action  $u_{21}$  for loop-shaping controller

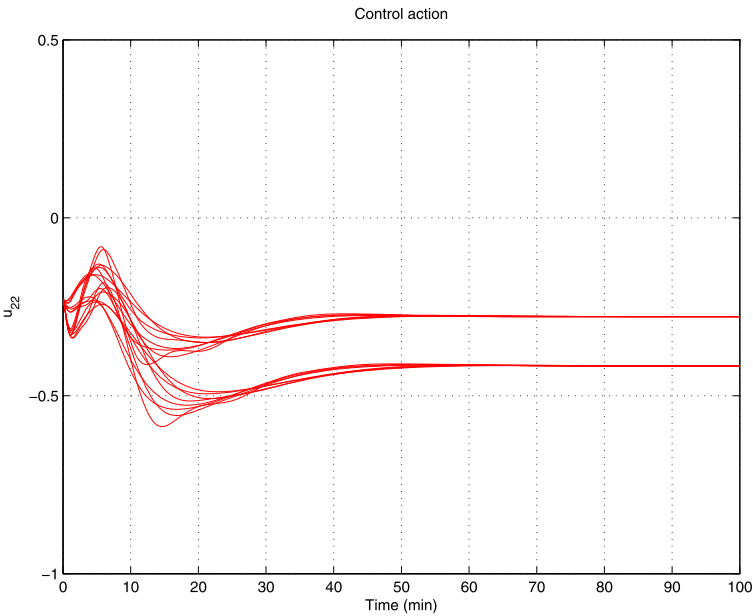


Fig. 16.33 Perturbed control action  $u_{22}$  for loop-shaping controller

**Table 16.3** Results of the  $\mu$ -synthesis

| Iteration | Controller order | Maximum value of $\mu$ |
|-----------|------------------|------------------------|
| 1         | 22               | 1.091                  |
| 2         | 28               | 0.991                  |
| 3         | 28               | 0.978                  |

The first block of this matrix corresponds to the uncertainty block  $\Delta$ , used in modeling the uncertainty of the distillation column. The second block  $\Delta_F$  is a fictitious uncertainty  $4 \times 4$  block, introduced to include the performance objectives in the framework of the  $\mu$ -approach. The inputs to this block are the weighted error signals  $e_p$  and  $e_u$ , the outputs being the exogenous inputs  $r$  and  $n$ .

To meet the design objectives a 2-degree-of-freedom stabilizing controller  $K(s) = [K_y(s) \ K_r(s)]$  is to be found such that, at each frequency  $\omega \in [0, \infty]$ , the structured singular value satisfies the condition

$$\mu_{\Delta_P}[F_L(P, K)(j\omega)] < 1.$$

The fulfillment of this condition guarantees robust performance of the closed-loop system, i.e.,

$$\left\| \begin{bmatrix} W_p(S\tilde{G}K_r - M) & -W_pTW_n \\ W_u(I + K_y\tilde{G})^{-1}K_r & -W_uK_ySW_n \end{bmatrix} \right\|_{\infty} < 1 \quad (16.3)$$

The  $\mu$ -synthesis is done by using the M-file `ms_col.m`, which implements the function `dkitopt`.

The progress of the  $D$ - $K$  iteration is shown in Table 16.3.

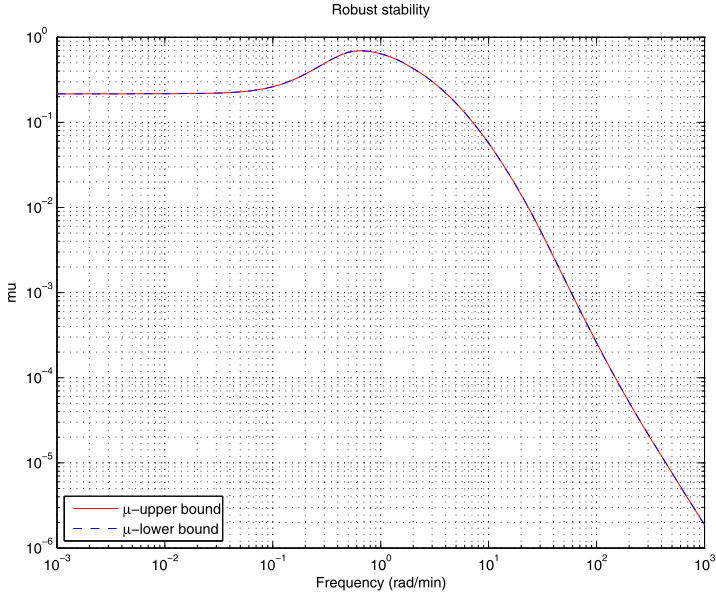
In the given case an appropriate controller is obtained after the third  $D$ - $K$  iteration. The controller is stable and its order is equal to 28.

It can be seen from Table 16.3 that after the third iteration the maximum value of  $\mu$  is equal to 0.978.

The  $\mu$ -analysis of the closed-loop system is done by the file `mu_col`. According to the reports, given below, the system achieve robust stability and robust performance.

```
report =
```

```
Uncertain System is robustly stable to modeled uncertainty.
-- It can tolerate up to 145% of the modeled uncertainty.
-- A destabilizing combination of 145% of the modeled
 uncertainty exists,
 causing an instability at 0.614 rad/s.
-- Sensitivity with respect to uncertain element ...
'Delta_1' is 54%. Increasing 'Delta_1' by 25% leads
 to a 14% decrease in the margin.
'Delta_2' is 68%. Increasing 'Delta_2' by 25% leads
 to a 17% decrease in the margin.
```



**Fig. 16.34** Robust stability for  $\mu$ -controller

```
report =
```

```
Uncertain System achieves a robust performance margin
of 1.023.
```

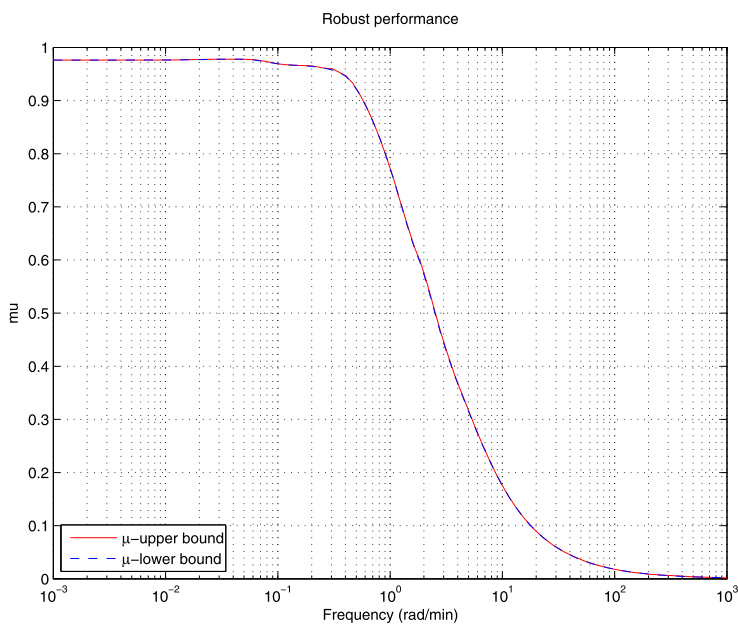
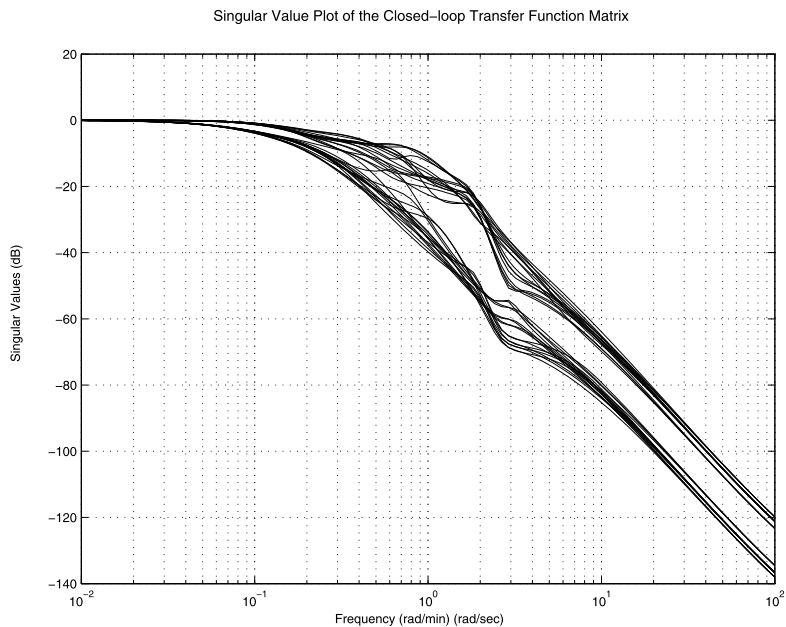
- ```
-- A model uncertainty exists of size 102% resulting
    in a performance margin
    of 0.978 at 0.0433 rad/s.
-- Sensitivity with respect to uncertain element ...
'Delta_1' is 52%. Increasing 'Delta_1' by 25% leads
    to a 13% decrease in the margin.
'Delta_2' is 46%. Increasing 'Delta_2' by 25% leads
    to a 12% decrease in the margin.
```

The frequency-response plot of the structured singular value for the case of robust stability is shown in Fig. 16.34. The stability of the system is preserved under perturbations that satisfy $\|\Delta\|_\infty < 1.45$.

The frequency response of μ for the case of robust performance analysis is shown in Fig. 16.35. The closed-loop system achieves robust performance, the maximum value of μ being equal to 0.978.

The unscaled closed-loop system singular-value plot is shown in Fig. 16.36. The closed-loop bandwidth is about 0.1 rad/min.

In Fig. 16.37 we show the singular-value plot of the unscaled sensitivity function \hat{S} .

**Fig. 16.35** Robust performance for μ -controller**Fig. 16.36** Closed-loop singular-value plots

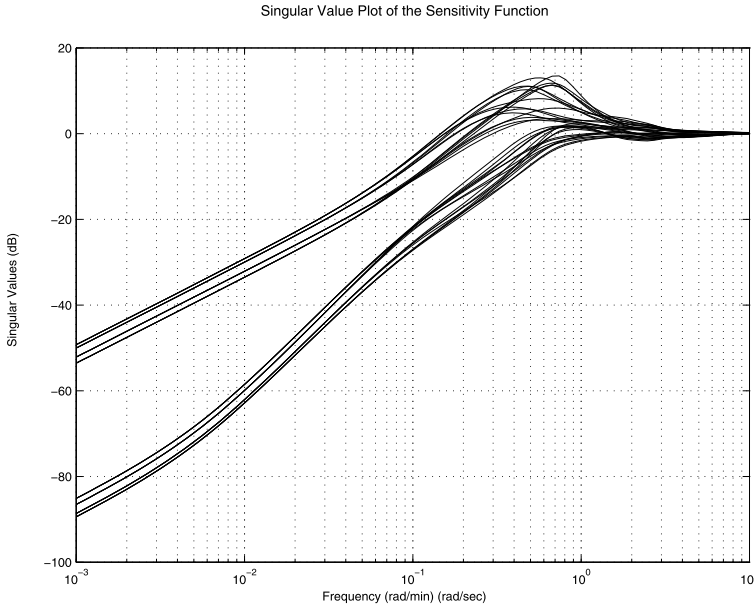


Fig. 16.37 Frequency responses of the sensitivity function

The frequency responses with respect to the noise are shown in Fig. 16.38. It is seen from the figure that the noises in measuring the distillate and bottom-product composition have a very small effect on the system output.

The singular-value plots of the unscaled μ -controller are shown in Fig. 16.39.

The singular-value plots of $\hat{G}\hat{K}_y$ and $\hat{K}_y\hat{S}$ are shown in Figs. 16.40 and 16.41, respectively. The maximum of the largest singular value of $\hat{G}\hat{K}_y$ is less than 1 for $\omega \geq 150$ and the maximum of the largest singular value of $\hat{K}_y\hat{S}$ is less than 300, thus the frequency-domain specification is met.

The perturbed transient responses of the scaled closed-loop system with a μ -controller are shown in Figs. 16.42, 16.43, 16.44 and 16.45. The responses to the corresponding references have no overshoots and the interaction of channels is weaker than in the case of using loop-shaping controller.

The control action in the case of perturbed system with the μ -controller is shown in Figs. 16.46, 16.47, 16.48 and 16.49.

Consider now the reduction of controller order. For this aim we implement the M-file `red_col.m`. Using the function `reduce` the controller order is reduced to 12.

In Fig. 16.50 we compare the frequency responses of the maximum singular values of the scaled full-order and reduced-order controllers. The frequency responses of both full-order and reduced-order controllers coincide up to 23 rad/min, which is much more than the closed-loop bandwidth of the system. This is why the tran-

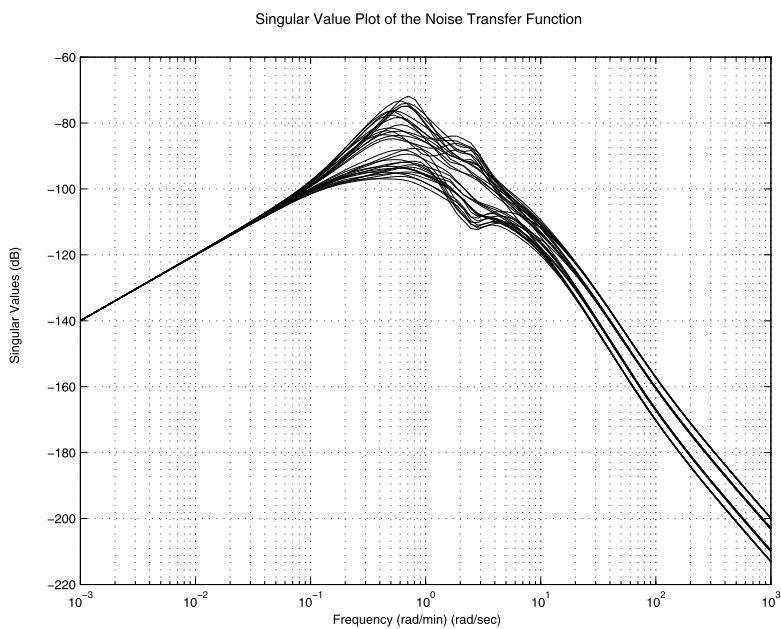


Fig. 16.38 Frequency responses with respect to noises

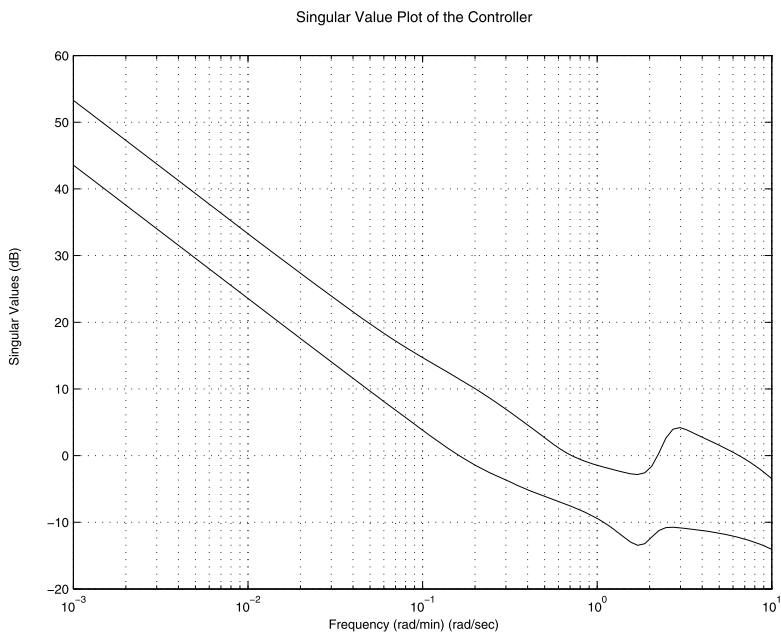


Fig. 16.39 Singular values of the controller

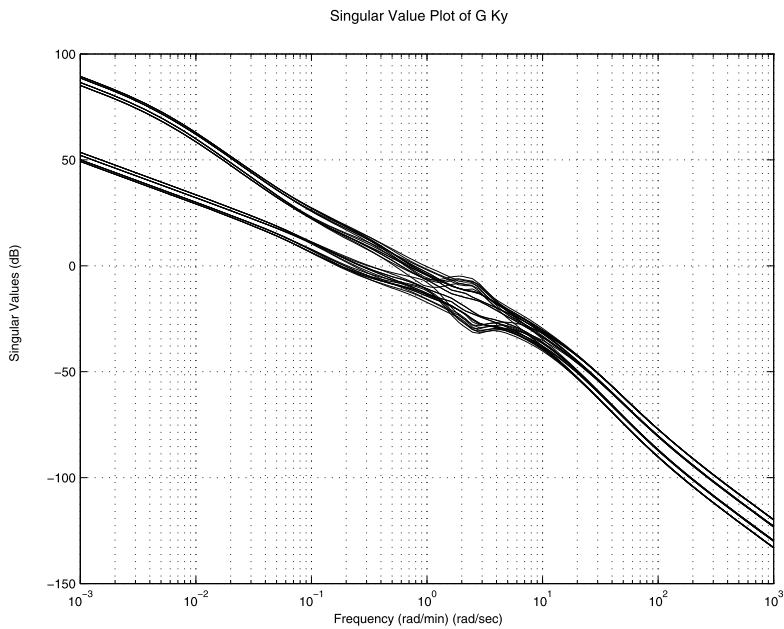


Fig. 16.40 Frequency responses of $\hat{G} \hat{K}_y$

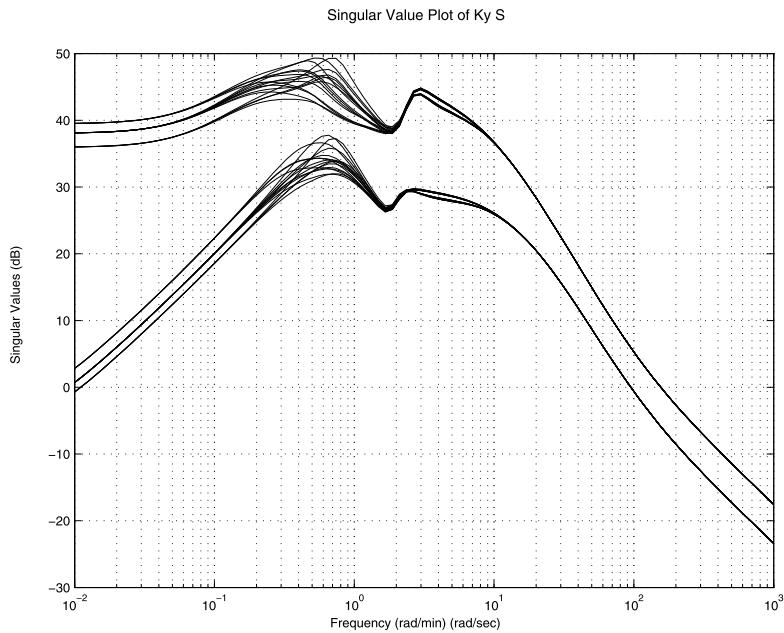


Fig. 16.41 Frequency responses of $\hat{K}_y \hat{S}$

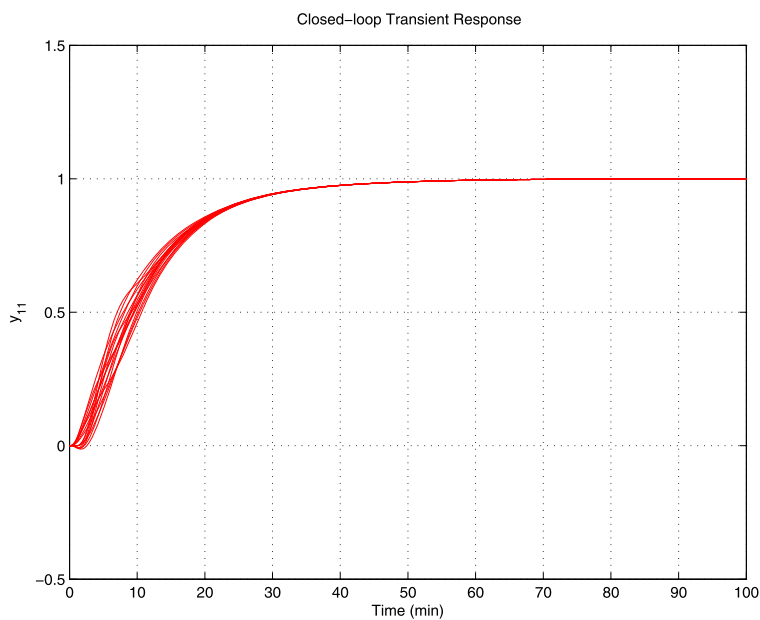


Fig. 16.42 Perturbed transient response y_{11} for μ -controller

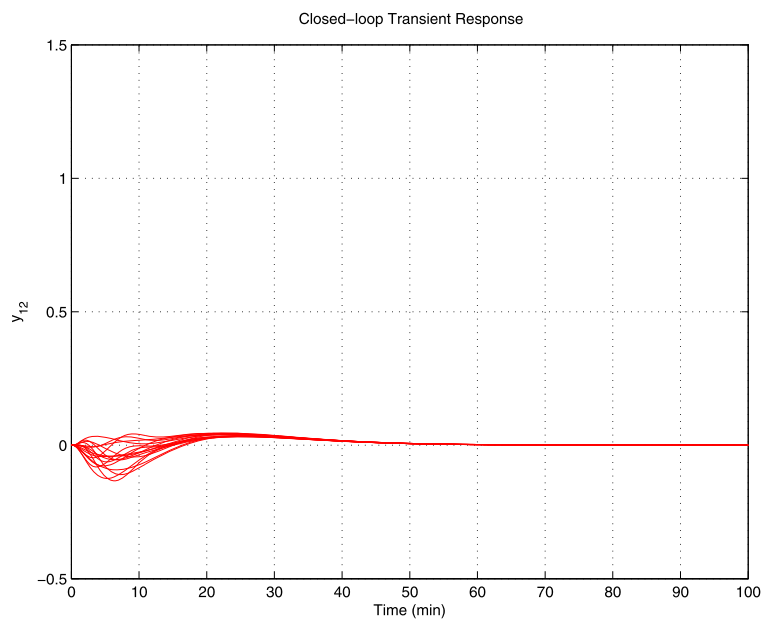


Fig. 16.43 Perturbed transient response y_{12} for μ -controller

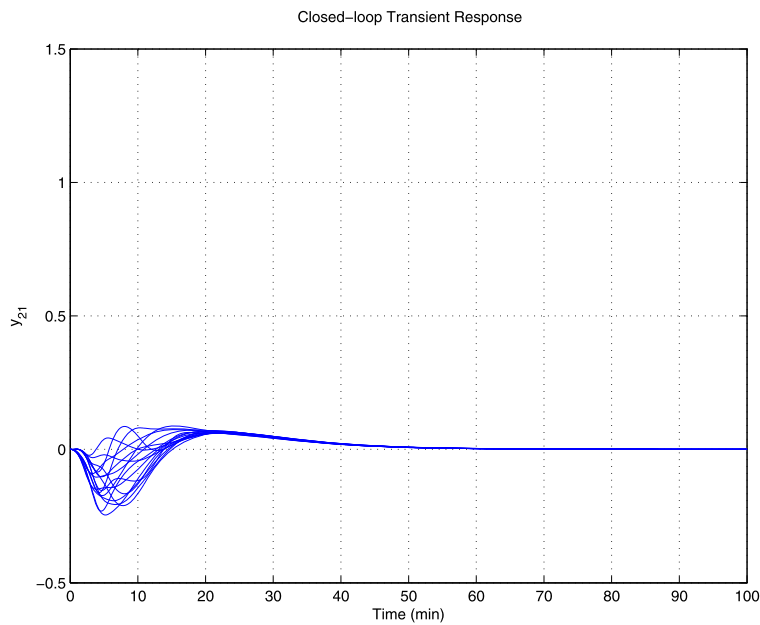


Fig. 16.44 Perturbed transient response y_{21} for μ -controller

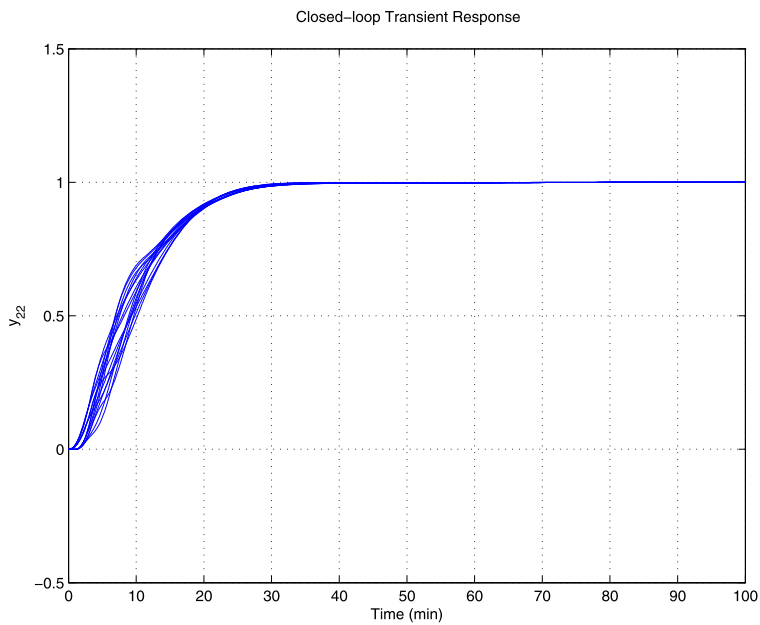


Fig. 16.45 Perturbed transient response y_{22} for μ -controller

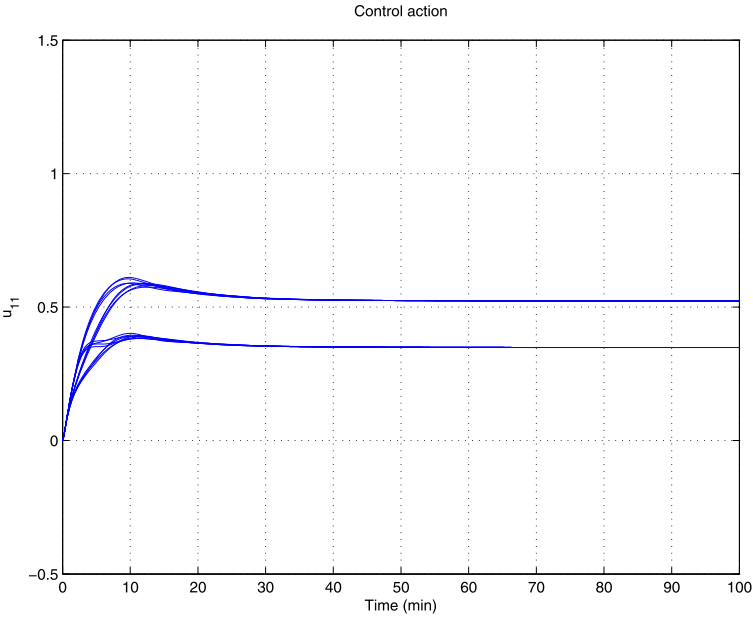


Fig. 16.46 Perturbed control action u_{11} for μ -controller

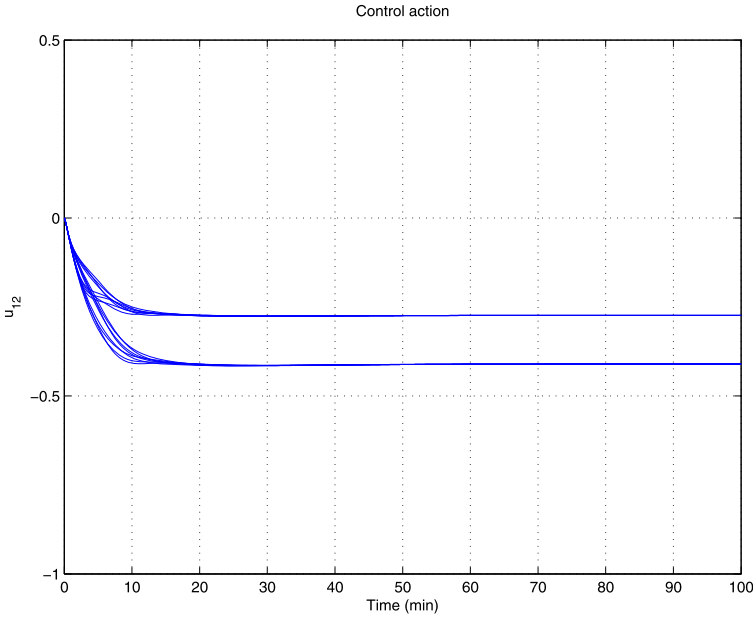


Fig. 16.47 Perturbed control action u_{12} for μ -controller

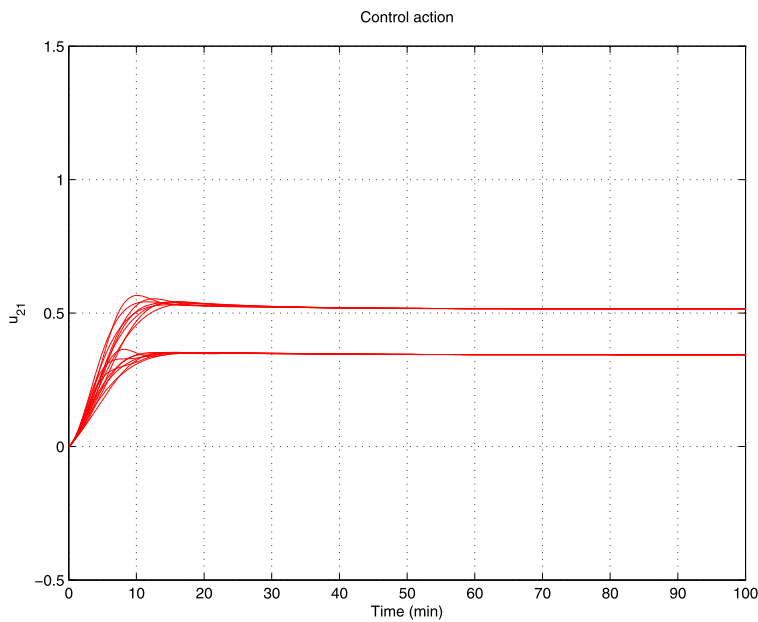


Fig. 16.48 Perturbed control action u_{21} for μ -controller

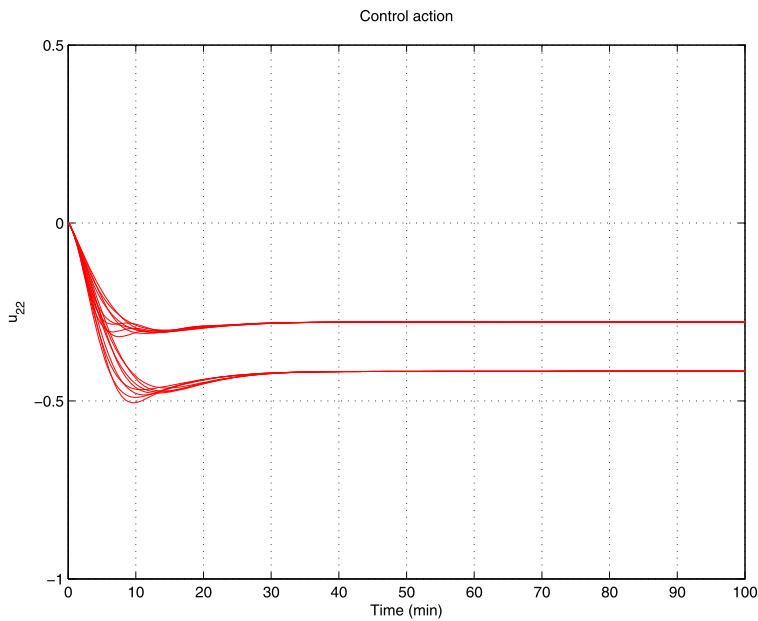


Fig. 16.49 Perturbed control action u_{22} for μ -controller

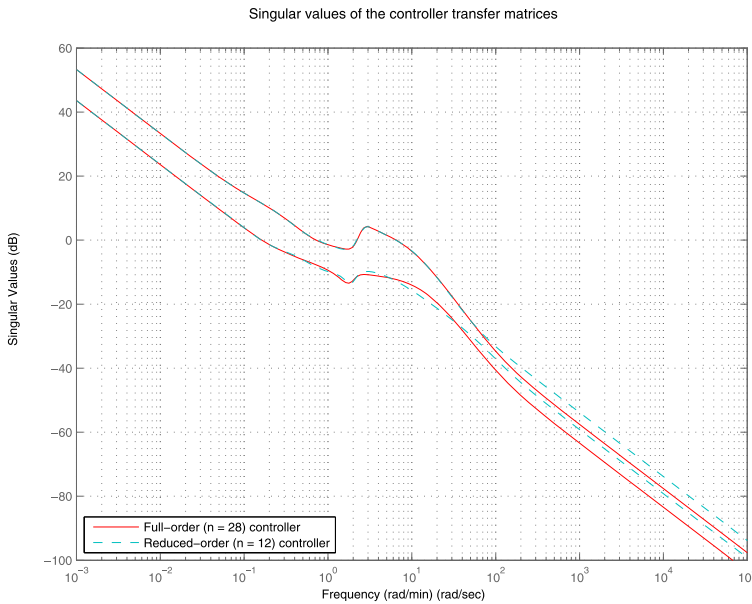


Fig. 16.50 Frequency responses of the full-order and reduced-order controllers

sient responses of the closed-loop system with full-order and with reduced-order controllers are practically undistinguishable.

16.7 Nonlinear System Simulation

The μ -controller designed is investigated by simulation of the corresponding nonlinear closed-loop system. The simulation is carried out by the Simulink[®] model `nls_col.mdl`, which implements the nonlinear plant model given in Sect. 16.2. To simulate the nonlinear plant we use the M-files `colamod` and `colas` by kind permission of the author, Sigurd Skogestad.

The Simulink[®] model of the distillation column control system shown in Fig. 16.51 allows us to carry out a number of simulations for different set points and disturbances. Note that the inputs to the controller are formed as differences between the values of the corresponding variables and their nominal (steady-state) values used in the linearization. In contrast, the controller outputs are added to the corresponding nominal inputs in order to obtain the full inputs to the nonlinear model of the column.

Before simulation of the system it is necessary to set the model parameters by using the M-file `init_col.m`. Also, the controller is rescaled so as to implement the unscaled input/output variables.

The nonlinear system simulation is done for the following reference and disturbance signals. At $t = 10$ min the feed rate F increases from 1 to 1.2, at $t = 100$ min

SIMULINK Model of the Distillation Column System

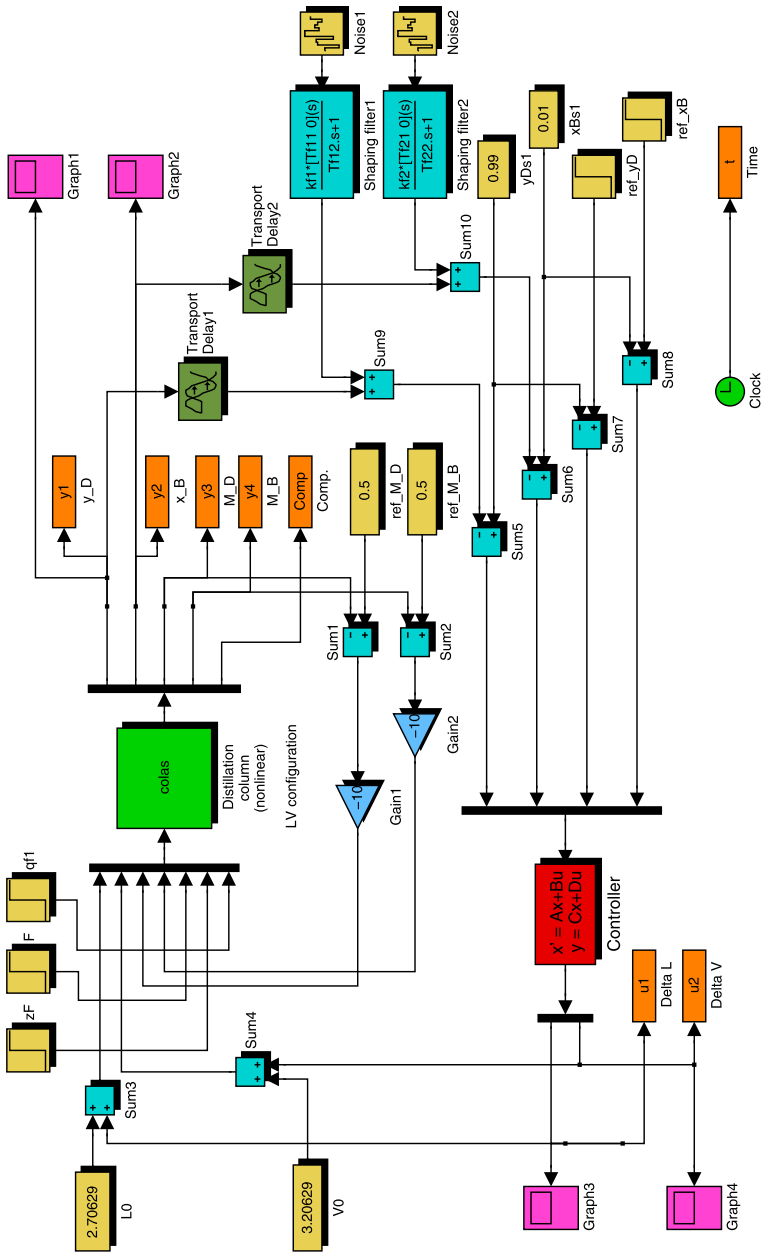


Fig. 16.51 Simulation model of the nonlinear system

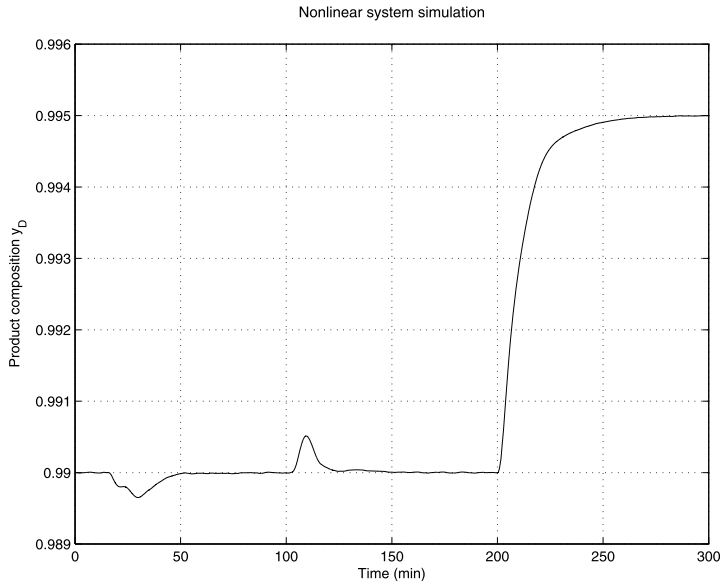


Fig. 16.52 Transient response of the nonlinear system— y_D

the feed composition z_F increases from 0.5 to 0.6 and at $t = 200$ min the set point in y_D increases from 0.99 to 0.995.

The time response of the distillate y_D for the case of the reduced-order μ -controller is given in Fig. 16.52. It is seen from the figure that the disturbances are attenuated well and the desired set point is achieved exactly.

The time response of the bottom-product composition x_B for the same controller is given in Fig. 16.53.

The simulation results show that the robust design method is appropriately chosen and confirms the validity of the uncertain model used.

16.8 Conclusions

The results from the analysis and design of a distillation column control system may be summarized as follows.

- It is possible to use a sufficiently low-order linearized model of the given nonlinear plant, so that the designed linear controllers allow to be achieved satisfactory dynamics of the nonlinear closed-loop system. The linearized model is scaled in order to avoid very small or very large signals.
- Two controllers are designed—one by using a 1-degree-of-freedom \mathcal{H}_∞ loop-shaping design method and the other by using a 2-degree-of-freedom μ -synthesis method. Both controllers satisfy the time-domain and frequency-domain specifications and ensure robust stability of the corresponding closed-loop systems. It

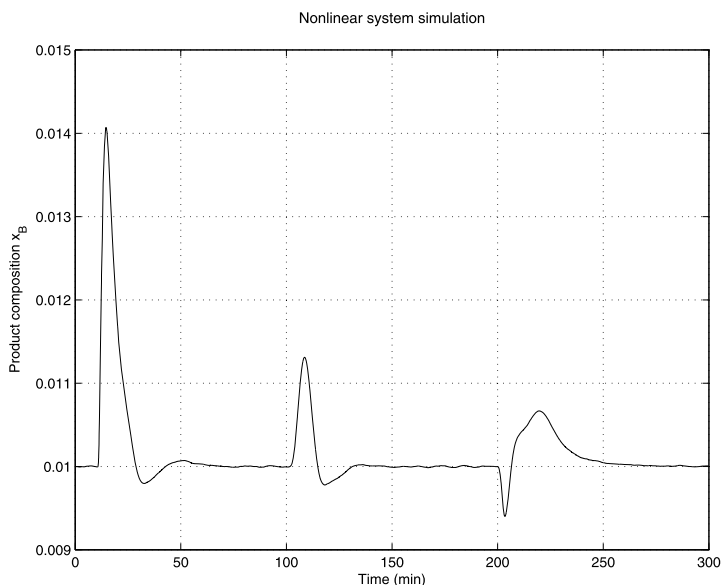


Fig. 16.53 Transient response of the nonlinear system— x_B

is impressive how the low-order, easily designed loop-shaping controller allows us to obtain practically the same characteristics of the closed-loop systems as the μ -controller, while the latter requires much more experiments for tuning the weighting functions.

- The nonlinear system simulation results confirm the ability of the controller to achieve disturbance attenuation and good responses to reference signals. The simulation confirms the validity of the uncertain model used.

16.9 Notes and References

The distillation column control problem presented in this chapter was introduced by Limebeer [97] as a benchmark problem at the 1991 *Conference on Decision and Control*. In [97] the uncertainty is defined in terms of parametric gain and delay uncertainty and the control objectives are a mixture of time-domain and frequency-domain specifications. The problem originates from Skogestad et al. [156] where a simple model of a high-purity distillation column was used and uncertainty and performance specifications were given as frequency-dependent weighting functions. A tutorial introduction to the dynamics of the distillation column is presented in [154].

A design of a 2-degree-of-freedom loop-shaping controller for the distillation column is presented in [62] where an eighth-order model of the column is used. A 2-degree-of-freedom controller for the distillation column system is proposed in

[106] with a reference model and using μ -synthesis. In that paper, one may find a selection procedure for the weighting functions described in detail. Our design differs from the design in [106] in several respects. First, instead of a second-order model with time delay we use a sixth-order model that is justified by the results from nonlinear system simulation. Second, we use modified weighting functions in order to obtain better results. In particular, we use a performance weighting transfer function matrix with nonzero off-diagonal elements that meets the time-domain specifications much better. Also, the control weighting functions are taken as first-order, low-pass filters.

Various design methods have been reported, in addition to the above, to tackle this distillation column problem [126, 139, 155, 160, 178]. In [178], the design problem is formulated as a mixed optimization problem. It is well known that control-system design problems can be formulated as constrained optimization problems. Design specifications in both the time and frequency domains as well as stability can be naturally formulated as constraints. Numerical optimization approaches can be used directly and a solution obtained, if there is one, will characterize an acceptable design. However, the optimization problems so derived are usually very complicated with many unknowns, many nonlinearities, many constraints, and in most cases, they are multiobjective with several conflicting design aims that need to be simultaneously achieved. Furthermore, a direct parameterization of the controller will increase the complexity of the optimization problem. In [178], the \mathcal{H}_∞ loop-shaping design procedure is followed. Instead of direct parameterization of controllers, the pre- and post-weighting functions used to shape the open-loop, augmented system are chosen as design (optimization) parameters. The low order and simple structure of such weighting functions make the numerical optimization much more efficient. The \mathcal{H}_∞ norm requirement is also included in the cost/constraint set. The stability of the closed-loop system is naturally met by such designed controllers. Satisfactory designs are reported in that paper. Reference [160] further extends the optimization approach in [178] by using a Genetic Algorithm to choose the weighting function parameters.

Chapter 17

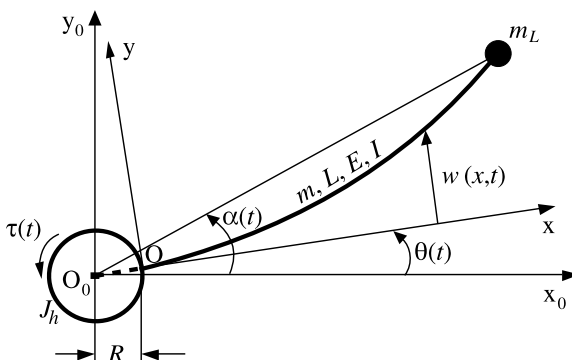
Robust Control of a Flexible-Link Manipulator

In this chapter we discuss the robust control system design of a flexible-link manipulator that moves in the horizontal plane.

Lightweight manipulators possess many advantages over the traditional bulky manipulators. The most important benefits include high payload-to-arm weight ratio, faster motion, safer operation, improved mobility, low cost, longer reach and better energy efficiency, etc. However, the reduction of weight leads to the increase of the link elasticity that significantly complicates the control of the manipulator. The difficulty in control is caused by the fact that the link model is a distributed parameter plant. In this case, several elastic modes are required to achieve sufficiently high accuracy. Also, the plant has several uncertain parameters (payload mass, hub and structural damping factors, etc.) that influence significantly the system performance. The inherent, non-minimum phase behavior of the flexible manipulator is another obstacle to achieving simultaneously a high-level performance as well as good robustness.

The aim of the present case study is to design a control system for a single-link flexible manipulator. A two-mode dynamic model of the manipulator is first obtained by using the Lagrangian-assumed modes method. This is followed by the modeling of uncertainties involved in the manipulator. The uncertainties include the real parametric uncertainties in the payload mass as well as in the hub and structural damping factors. These parameters are the basic uncertainty source in the dynamic behaviors of the flexible-link manipulators. The μ -synthesis method is then applied to design a robust, noncollocated controller on the feedback signals of joint angle and tip acceleration. In the design, in order to obtain a feasible solution, a simplified uncertainty description is considered in the D - K iterations. Appropriate weighting functions are chosen in the design to ensure robust stability and robust performance. It is shown in this chapter that good robust performance has been achieved in the design. The closed-loop system exhibits excellent tip-motion performance for a wide range of payload mass and the system efficiently suppresses the elastic vibrations during the fast motion of the manipulator tip. For the sake of implementation and reliability in practice, a reduced-order controller is found that maintains the robust

Fig. 17.1 Schematic diagram of the flexible-link manipulator



stability and robust performance of the closed-loop system. Finally, the advantages of the μ -controller over a conventional, collocated PD controller are demonstrated.

17.1 Dynamic Model of the Flexible Manipulator

Figure 17.1 shows the schematic model used to derive the equations of motion for the flexible-link manipulator. The manipulator moves in the horizontal plane. Frame $x_0-O_0-y_0$ is the fixed-base frame. Frame $x-O-y$ is the local frame rotating with the hub. The x -axis coincides with the undeformed longitudinal axis of the link. The rotating inertia of the servomotor, the gear box, and the clamping hub are modeled as a single hub inertia J_h . The distance between the hub center and the root of the link is denoted by R . The flexible link is assumed to be a homogeneous rod with a constant cross-sectional area. L is the length of the link, m is the mass per unit length of the link, I is the link cross-sectional moment of inertia and E is Young's modulus of elasticity for the material of the link. The payload is modeled as a point mass m_L . The variables $\tau(t)$ and $\theta(t)$ are the driving torque and the joint angle, respectively. The elastic deflection of a point located at a distance x from O along the link is denoted by $w(x, t)$. It is assumed that the elastic deflections of the link lie in the horizontal plane, and are perpendicular to the x -axis and small in magnitude compared to the link length.

The motion equations of the flexible manipulator are to be derived by using the Lagrangian approach combined with the assumed-modes method [25]. The flexible link is modeled as an Euler–Bernoulli beam. The free vibration of the link is described by the partial differential equation [116]

$$EI \frac{\partial^4 w(x, t)}{\partial x^4} + m \frac{\partial^2 w(x, t)}{\partial t^2} = 0$$

with boundary conditions

$$w(0, t) = 0, \quad \frac{\partial w(0, t)}{\partial x} = 0$$

$$\begin{aligned}\frac{\partial^2 w(L, t)}{\partial x^2} &= 0 \\ \frac{\partial^3 w(L, t)}{\partial x^3} - \frac{m_L}{EI} \frac{\partial^2 w(L, t)}{\partial t^2} &= 0\end{aligned}$$

According to the assumed-modes method the elastic deflection can be expressed as

$$w(x, t) = \sum_{i=1}^n \varphi_i(x) \eta_i(t) \quad (17.1)$$

where $\eta_i(t)$ is the generalized coordinate of the i th mode, $\varphi_i(x)$ is the space eigenfunction of the i th mode, and n is the number of the modes that describe the link deflection. The mode angular frequencies ω_i , $i = 1, \dots, n$, of the flexible link are given by

$$\omega_i = \beta_i^2 \sqrt{\frac{EI}{m}} \quad (17.2)$$

where β_i , $i = 1, \dots, n$, are the first n positive roots of the transcendental equation

$$1 + \cosh(\beta L) \cos(\beta L) + \frac{m_L}{mL} (\beta L) [\sinh(\beta L) \cos(\beta L) - \cosh(\beta L) \sin(\beta L)] = 0 \quad (17.3)$$

The shape functions $\varphi_i(x)$, $i = 1, \dots, n$, satisfy the orthogonality condition

$$m \int_0^L \varphi_i(x) \varphi_j(x) dx + m_L \varphi_i(L) \varphi_j(L) = 0, \quad i \neq j$$

and can be written in the form

$$\begin{aligned}\varphi_i(x) = \lambda_i \bigg\{ & [\cosh(\beta_i x) - \cos(\beta_i x)] \\ & - \frac{\cosh(\beta_i L) + \cos(\beta_i L)}{\sinh(\beta_i L) + \sin(\beta_i L)} [\sinh(\beta_i x) - \sin(\beta_i x)] \bigg\} \quad (17.4)\end{aligned}$$

A normalization of the shape functions convenient for the uncertainty modeling is accomplished by determining the coefficients λ_i , $i = 1, \dots, n$, in (17.4) on the basis of the relation

$$m \int_0^L \varphi_i^2(x) dx + m_L \varphi_i^2(L) = 1$$

The joint angle θ and the deflection variables η_i , $i = 1, \dots, n$, are used as generalized coordinates in the derivation of the equation of motion. As a result of applying the Lagrangian procedure, the following nonlinear dynamic model of the flexible manipulator is obtained:

$$\begin{bmatrix} m_r(\eta) & m_{rf}^T \\ m_{rf} & I_n \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\eta} \end{bmatrix} + \begin{bmatrix} d_r & 0_n^T \\ 0_n & D_f \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\eta} \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0_n^T \\ 0_n & C_f \end{bmatrix} \begin{bmatrix} \theta \\ \eta \end{bmatrix} + \begin{bmatrix} h_r(\dot{\theta}, \eta, \dot{\eta}) \\ h_f(\dot{\theta}, \eta) \end{bmatrix} = \begin{bmatrix} 1 \\ 0_n \end{bmatrix} \tau \quad (17.5)$$

where

$$\begin{aligned} \eta &= [\eta_1 \quad \dots \quad \eta_n]^T \\ m_r(\eta) &= a_0 + \sum_{i=1}^n \eta_i^2 \\ a_0 &= J_a + \frac{1}{3}m[(L+R)^3 - R^3] + m_L(L+R)^2 \\ m_{rf} &= [a_1 \quad \dots \quad a_n]^T \\ a_i &= m \int_0^L (x+R)\varphi_i(x) dx + m_L(L+R)\varphi_i(L), \quad i = 1, \dots, n \quad (17.6) \\ C_f &= \text{diag}(\omega_1^2, \dots, \omega_n^2) \\ D_f &= \text{diag}(d_1, \dots, d_n) \\ h_r(\dot{\theta}, \eta, \dot{\eta}) &= \sum_{i=1}^n 2\dot{\theta}\dot{\eta}_i\eta_i \\ h_f(\dot{\theta}, \eta) &= [-\dot{\theta}^2\eta_1 \quad \dots \quad -\dot{\theta}^2\eta_n]^T \end{aligned}$$

I_n denotes the $n \times n$ identity matrix, 0_n is the n -dimensional null vector, and d_r, d_1, \dots, d_n are damping coefficients. The terms $d_r\dot{\theta}$ and $D_f\dot{\eta}$ have been included to account for the viscous friction at the hub and for the structural damping of the flexible link, respectively.

The angle

$$\alpha = \theta + \arctan \frac{w(L, t)}{L + R} \quad (17.7)$$

is chosen as the coordinate that determines the position of the manipulator tip.

The following numerical values of the manipulator parameters are used: $L = 1$ m, $R = 0.4$ m, $J_h = 0.1$ kg m², $m = 0.54$ kg/m, flexural rigidity of the flexible link $EI = 18.4$ N m². The values of m and EI correspond to an aluminum link with $E = 6.9 \times 10^{10}$ N m², density $\rho = 2700$ kg/m³, and a rectangular cross-section with dimensions $s_1 = 0.05$ m and $s_2 = 0.004$ m, s_2 being in the motion plane.

It is assumed that in performing a given motion the payload mass has a constant but unknown value in the range from 0.15 kg to 0.35 kg.

The first two natural frequencies of the flexible link, calculated for the average value of the payload according to (17.2) and (17.3), are $\omega_1 = 12.1$ rad/s and $\omega_2 = 99.2$ rad/s. Since the rest natural frequencies are very large ($\omega_3 = 302.5$ rad/s and so on), a two-mode model of the flexible manipulator is used in the controller design.

Usually, the friction coefficients cannot be determined exactly and also changes in their values are possible with the time. It is assumed that the coefficient of the

friction at the hub has a nominal value $d_r = 0.15 \text{ kg m}^2/\text{s}$ and a relative uncertainty 20 %, and the coefficients of the structural damping of the first two generalized coordinates of the flexible link have nominal values $d_1 = 0.4 \text{ kg m}^2/\text{s}$ and $d_2 = 10 \text{ kg m}^2/\text{s}$ and a relative uncertainty 40 %.

After neglecting in (17.5) the nonlinear terms $\sum_{i=1}^n \eta_i^2$, $h_r(\dot{\theta}, \eta, \dot{\eta})$, and $h_f(\dot{\theta}, \eta)$, whose effect is relatively small, and after setting $\arctan(z) = z$ in (17.7), since $w(L, t) \ll L$, one obtains a linear model of the manipulator. The transfer functions of the manipulator with input the driving torque τ and outputs the joint angle θ and the coordinate angle α , determined for $n = 2$, $m_L = 0.25 \text{ kg}$, $d_r = 0.12 \text{ kg m}^2/\text{s}$, $d_1 = 0.24 \text{ kg m}^2/\text{s}$ and $d_2 = 6 \text{ kg m}^2/\text{s}$, may be written, respectively, in the form

$$W_{\theta\tau} = \frac{9.793(s^2 + 0.24s + 145.8)(s^2 + 6s + 9839)}{s(s + 0.2095)(s^2 + 2.067s + 741.9)(s^2 + 6.799s + 10850)}$$

and

$$W_{\alpha\tau} = \frac{0.1239(s - 172.1)(s - 66.2)(s + 173.1)(s + 57.5)}{s(s + 0.2095)(s^2 + 2.067s + 741.9)(s^2 + 6.799s + 10850)}$$

The transfer function $W_{\alpha\tau}$ has positive zeros and hence the manipulator, considered with the tip position as output, represents a non-minimum phase plant. It is known that to ensure high performance for such plants is a difficult problem.

The second order polynomials in the transfer functions $W_{\theta\tau}$ and $W_{\alpha\tau}$ have very small damping factors (between 0.01 and 0.04), which causes intensive vibrations of the flexible link when performing fast motions. The undamped frequencies of the numerator polynomial of $W_{\theta\tau}$ are equal to the natural frequencies ω_1 and ω_2 of the flexible link. The nonzero natural frequencies of the manipulator (including the flexible link and the driving part) are determined from the second order polynomial in the denominators of the transfer functions and for the given numerical values of the parameters are equal to 27.2 rad/s and 104.2 rad/s.

To illustrate the effect of uncertainty, in Figs. 17.2 and 17.3 we show the Bode plots of the flexible manipulator with input the driving torque τ and outputs the joint angle θ and the coordinate angle α , respectively, determined for several values of the payload mass in the interval from 0.15 kg to 0.35 kg.

17.2 A Linear Model of the Uncertain System

In this section we first consider how to model the uncertainties of the flexible-link manipulator and then develop a linear dynamic model of the uncertain system. As mentioned earlier, the uncertainties considered are related to the payload mass, hub damping factors and the damping levels of the first two modes. It is important to note that these parameters are the basic source of uncertainty dynamic behavior of flexible-link manipulators.

The plant input is the driving torque τ . The controlled variable is the tip position α and the measured variables are the joint angle θ and the tip acceleration $\ddot{\alpha}$.

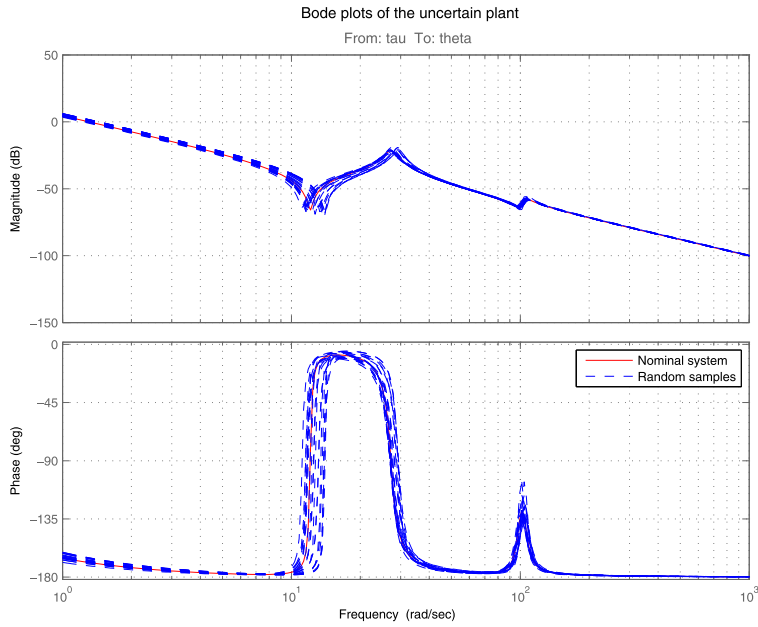


Fig. 17.2 Bode plot of the flexible-link manipulator with input τ and output θ

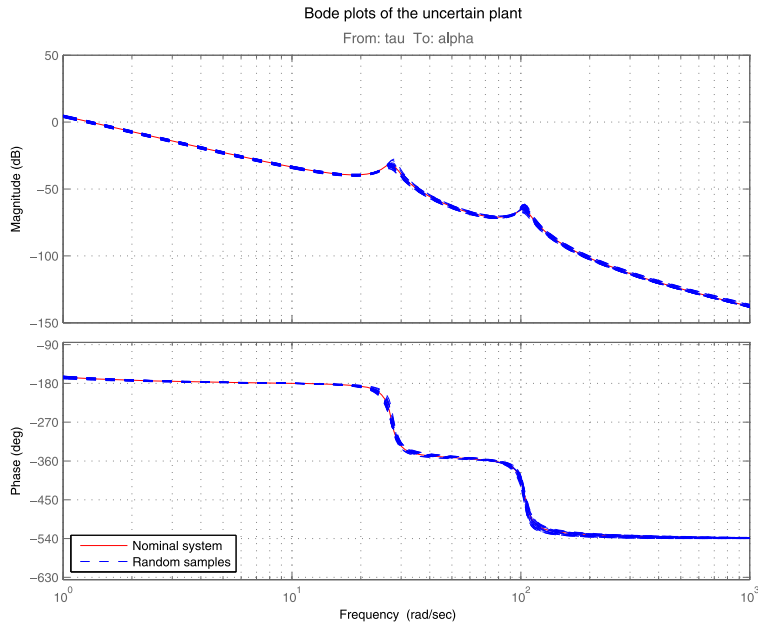


Fig. 17.3 Bode plot of the flexible-link manipulator with input τ and output α

The building of the uncertainty model is done on the basis of the linear model of the flexible manipulator, obtained after neglecting the nonlinear terms in (17.5) and after setting $\arctan(z) = z$ in (17.7). In this way one obtains the equations

$$\begin{aligned}
 a_0\ddot{\theta} + d_r\dot{\theta} + a_1\ddot{\eta}_1 + a_2\ddot{\eta}_2 &= \tau \\
 \ddot{\eta}_i + d_i\dot{\eta}_i + c_i\eta_i + a_i\ddot{\theta} &= 0, \quad i = 1, 2 \\
 \alpha &= \theta + b_1\eta_1 + b_2\eta_2 \\
 \ddot{\alpha} &= \ddot{\theta} + b_1\ddot{\eta}_1 + b_2\ddot{\eta}_2 \\
 w_L &= (L + R)b_1\eta_1 + (L + R)b_2\eta_2
 \end{aligned} \tag{17.8}$$

where the following notation is used:

$$\begin{aligned}
 w_L &= w(L, t) \\
 b_i &= \frac{\varphi_i(L)}{L + R}, \quad i = 1, 2
 \end{aligned}$$

In the given case the coefficients $a_0, a_1, a_2, b_1, b_2, c_1$, and c_2 in (17.8) are functions of the uncertain real parameter m_L . According to (17.6), the coefficient a_0 depends in an affine way on m_L and in building the model it may be represented as

$$a_0 = a_{01} + a_{02}m_L \tag{17.9}$$

with

$$a_{01} = J_a + \frac{1}{3}m[(L + R)^3 - R^3], \quad a_{02} = (L + R)^2$$

In contrast to a_0 , the rest coefficients depend nonlinearly and implicitly on m_L according to complicated relationships that cannot be used to obtain the uncertainty model. However, the investigation of these relationships shows that with sufficient accuracy the coefficients a_1, a_2, b_1, b_2, c_1 , and c_2 are approximated by the following rational functions of m_L :

$$\begin{aligned}
 a_i &\approx a_{i1} + a_{i2}m_L, \\
 b_i &\approx \frac{1}{b_{i1} + b_{i2}m_L} \\
 c_i &\approx \frac{1}{c_{i1} + c_{i2}m_L}, \quad i = 1, 2
 \end{aligned} \tag{17.10}$$

The constants $a_{i1}, a_{i2}, b_{i1}, b_{i2}, c_{i1}, c_{i2}, i = 1, 2$, in (17.10) are determined by least squares approximation, done so that for $m_L = 0.25$ kg the corresponding relationships are fulfilled exactly. Hence, for the nominal value of the payload mass the manipulator model will be exact. The plots of the exact and the determined in this way approximate dependencies of the coefficients a_1, b_1 , and c_1 on the payload

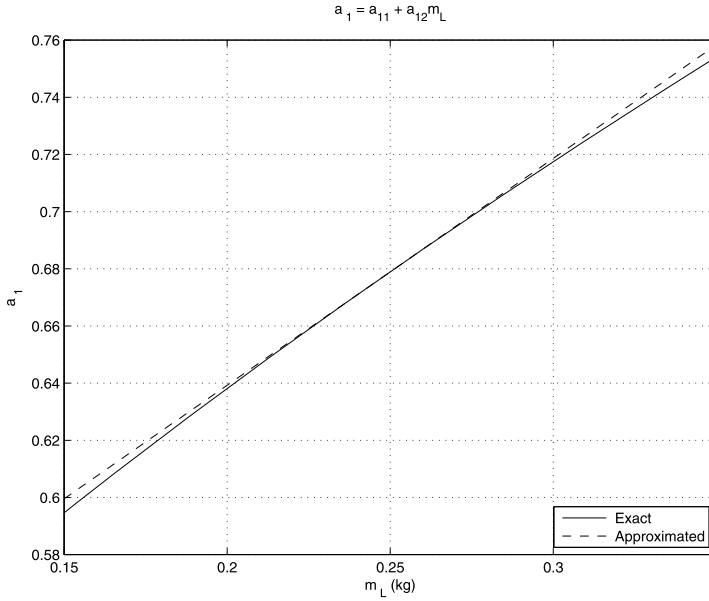


Fig. 17.4 Approximation of a_1

mass are shown in Figs. 17.4, 17.5, and 17.6. Similar results are obtained also for a_2 , b_2 , and c_2 . The relative error in approximating each dependence is biggest for $m_L = 0.15$ kg.

In the expressions (17.9) and (17.10) the uncertain parameter m_L appears once. Since in (17.8) the coefficients a_1 and a_2 are repeated twice, and b_1 and b_2 three times, it appears that as a whole the parameter m_L is repeated in (17.8) 13 times. However, in the manipulator model, built on the basis of the state space representation of (17.8), the uncertain parameter m_L is repeated 32 times, which makes the robust analysis and design of the system difficult. This number may be reduced significantly, building the manipulator model by using the function `sysic` representing appropriately (17.8).

For this aim the manipulator equations (17.8) are written in the form

$$\begin{aligned}
 M\ddot{y} + N_1\dot{y} + N_2y &= T\tau, \\
 z &= P_1y, \\
 \ddot{\alpha} &= P_2\ddot{y}
 \end{aligned} \tag{17.11}$$

where

$$y = [\theta \quad \eta_1 \quad \eta_2]^T, \quad z = [\alpha \quad w_L]^T$$

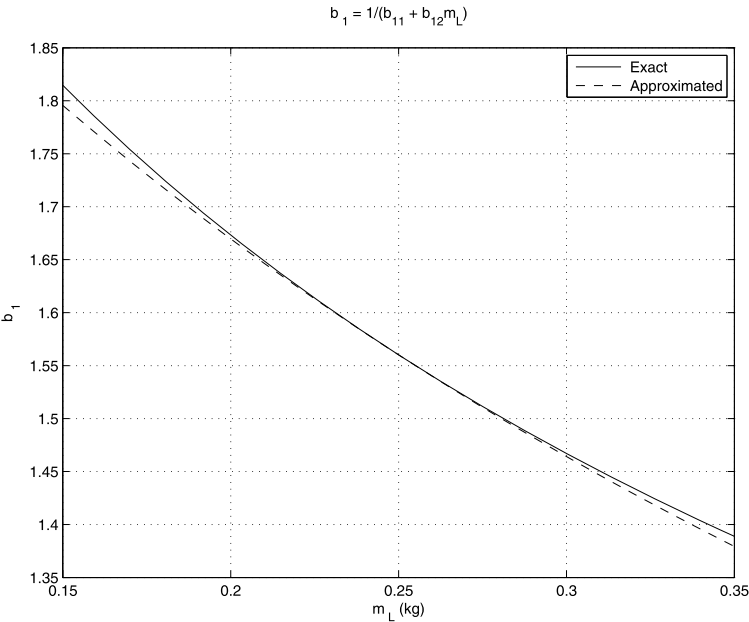


Fig. 17.5 Approximation of b_1

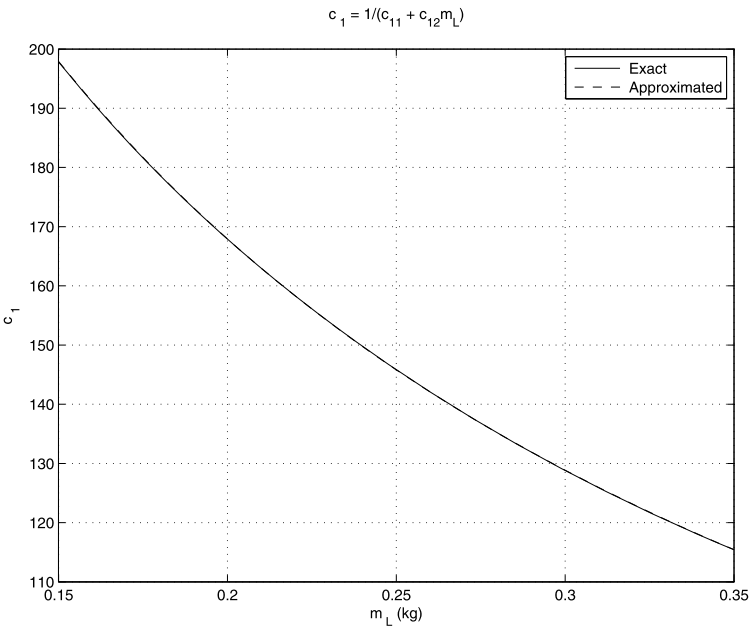


Fig. 17.6 Approximation of c_1

Fig. 17.7 Block diagram of the manipulator model

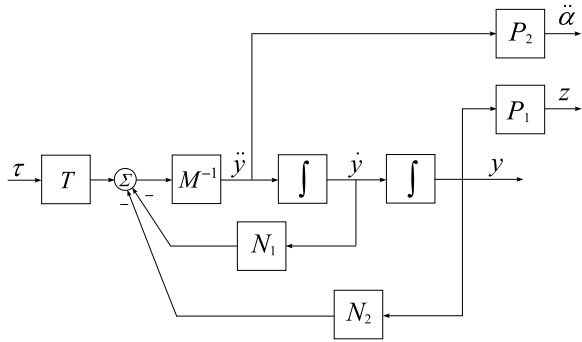
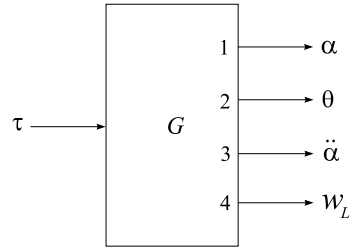


Fig. 17.8 Input–output representation of the manipulator model



$$M = \begin{bmatrix} a_0 & a_1 & a_2 \\ a_1 & 1 & 0 \\ a_2 & 0 & 1 \end{bmatrix}, \quad N_1 = \begin{bmatrix} d_r & 0 & 0 \\ 0 & d_1 & 0 \\ 0 & 0 & d_2 \end{bmatrix}, \quad N_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & c_1 & 0 \\ 0 & 0 & c_2 \end{bmatrix},$$

$$T = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad P_1 = \begin{bmatrix} 1 & b_1 & b_2 \\ 0 & (L+R)b_1 & (L+R)b_2 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 1 & b_1 & b_2 \end{bmatrix}$$

A block diagram corresponding to the matrix–vector equations (17.11) is shown in Fig. 17.7.

The model with uncertain parameters is obtained on the basis of the given block diagram by using the M-file `mod_flm.m` and is saved in the variable `G`. The constants a_{01} , a_{02} , a_{i1} , a_{i2} , b_{i1} , b_{i2} , c_{i1} , c_{i2} , $i = 1, 2$, are determined by the M-file `par_flm.m`.

A schematic diagram for the output ordering is shown in Fig. 17.8.

As a result from the execution of the file `mod_flm.m` one obtains

`G`

```
USS: 6 States, 4 Outputs, 1 Input, Continuous System
d_1: real, nominal = 0.4, variability = [-40 40]%,
    1 occurrence
d_2: real, nominal = 10, variability = [-40 40]%,
    1 occurrence
d_r: real, nominal = 0.15, variability = [-20 20]%,
    1 occurrence
```

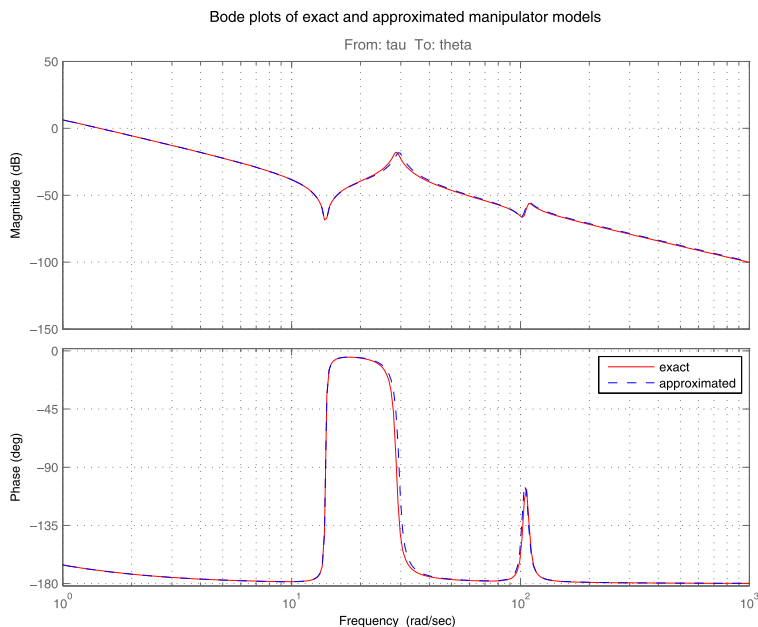


Fig. 17.9 Bode plots of exact and approximated manipulator models with input τ and output Θ for $m_L = 0.125$ kg

```
m_L: real, nominal = 0.25, variability = [-40  40]%,
      10 occurrences
```

G is an uncertain system of the class `uss`, depending on four uncertain real parameters, m_L being repeated ten times. The accuracy of the model derived is confirmed by the closeness of the “exact” and “approximated” Bode plots of the manipulator with input the driving torque τ and output the joint angle θ and angle position α shown in Figs. 17.9 and 17.10, respectively. These plots are obtained for the most unfavorable case $m_L = 0.15$ kg. The exact plots are computed from (17.8) using the exact values of the coefficients and the approximated—from the derived uncertainty model. It can be seen that the match between those two models is very good.

In the design and analysis of the noncollocated controller, presented latter on, we take three of the five outputs of the derived manipulator model. That is why the parameter m_L will occur 8 times instead of 10 times as in the full model. Thus, the total number of the uncertain parameters is equal to 11.

It is interesting to note that in using the option ‘full’ instead of ‘basic’ in `ureal` or in using the command `simplify` it is not possible to achieve positive result in building the model. For some versions of the model the multiplicity of the uncertain parameter m_L is reduced but the behavior of the model is completely changed.

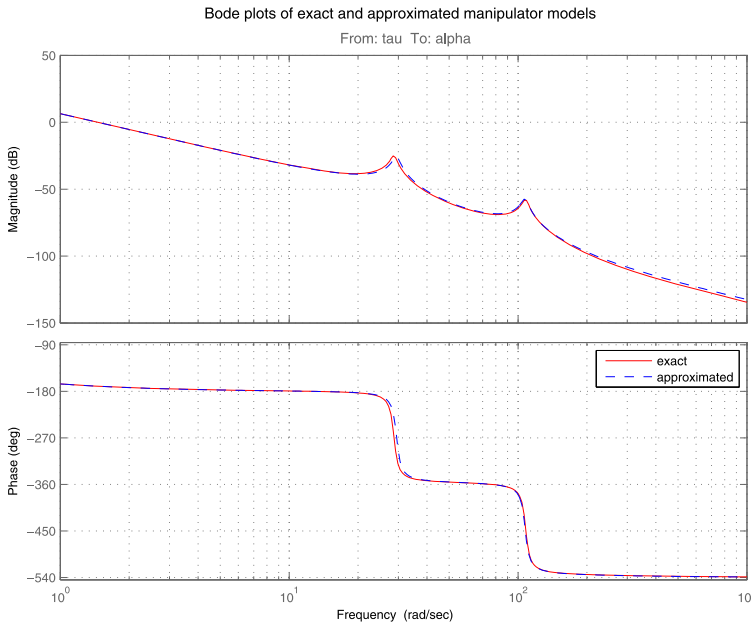


Fig. 17.10 Bode plots of exact and approximated manipulator models with input τ and output α for $m_L = 0.125$ kg

17.3 System Performance Specifications

In this flexible-link manipulator control-system design exercise, the purpose is to find a controller that suppresses efficiently the elastic vibrations of the flexible link in fast motions and moves the tip to a desired position in the presence of uncertainties in the payload mass, hub and structural damping factors. Since the uncertainties considered are real and structured, the most appropriate robust control design method to be applied in the present case is μ -synthesis.

The block diagram of the closed-loop system incorporating the design requirements is shown in Fig. 17.11. The controlled variable is the angular position α of the manipulator tip. The reference variable, the control variable, and the motor torque are denoted, respectively, by r , u , and τ_a . The controller K works on the reference r and feedback signals of the joint angle θ and the tip acceleration $\ddot{\alpha}$. The inclusion of the tip acceleration in the control scheme aims to achieve better tip-motion performance and leads to a noncollocated controller structure. Input disturbance is the resistant torque d . By the use of the signals n_1 and n_2 and the shaping filters W_{n1} and W_{n2} it is possible to take into account the influence of the noise, accompanying the measurement of θ and $\ddot{\alpha}$, respectively. Furthermore, in the given design case, we select a suitable dynamic model and target the dynamics of the designed closed-loop

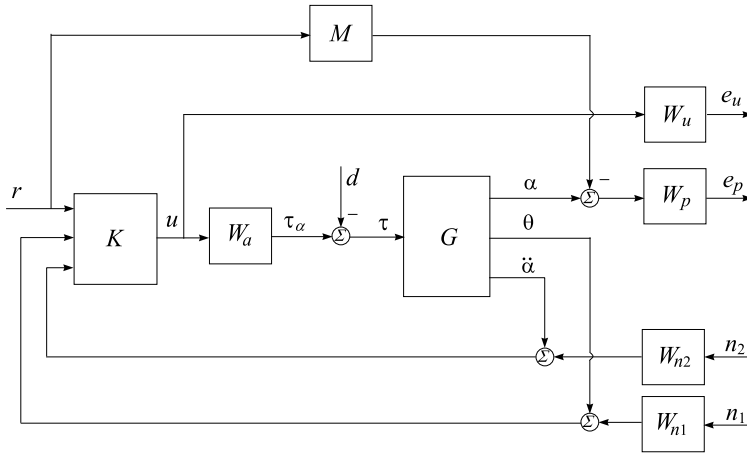


Fig. 17.11 Closed-loop interconnection structure of the flexible-link manipulator system

system to be close to that model. The use of such a model to represent the desired dynamics allows us to take into account the requirements on system performance more easily and directly. In other words, such a model (named M in Fig. 17.11) prescribes the desired dynamic behavior of the closed-loop system from the reference signal to the tip position. The uncertainty plant is denoted by G in Fig. 17.11. The internal, current control loop of the servo drive W_a is modeled as a first order lag with the time constant 0.004 s and gain equal to 1.

Let the 3×1 transfer matrix G be partitioned as

$$G(s) = \begin{bmatrix} G_{\alpha\tau}(s) \\ G_{\theta\tau}(s) \\ G_{\ddot{\alpha}\tau}(s) \end{bmatrix}$$

where $G_{\alpha\tau}$, $G_{\theta\tau}$, $G_{\ddot{\alpha}\tau}$ are the transfer functions from the control torque τ to the outputs α , θ , and $\ddot{\alpha}$, respectively, and let the controller is represented as

$$K(s) = [K_r(s) \quad K_\theta(s) \quad K_{\ddot{\alpha}}(s)]$$

where $K_r(s)$, $K_\theta(s)$, and $K_{\ddot{\alpha}}(s)$ are the transfer functions of the controller with respect to the corresponding inputs. It can be shown by direct manipulations that

$$\begin{bmatrix} e_p \\ e_u \end{bmatrix} = \Phi \begin{bmatrix} r \\ d \\ n_1 \\ n_2 \end{bmatrix}$$

where

$$\Phi = \begin{bmatrix} W_p(SG_{\alpha\tau}W_aK_r - M) & -W_pSG_{\alpha\tau} \\ W_uSK_r & -W_uS(G_{\theta\tau}K_\theta + G_{\ddot{\alpha}\tau}K_{\ddot{\alpha}}) \end{bmatrix} \begin{bmatrix} W_pSG_{\alpha\tau}W_aK_\theta W_{n1} & W_pSG_{\alpha\tau}W_aK_{\ddot{\alpha}} W_{n2} \\ W_uSK_\theta W_{n1} & W_uSK_{\ddot{\alpha}} W_{n2} \end{bmatrix} \quad (17.12)$$

and

$$S = \frac{1}{1 - G_{\theta\tau}W_aK_\theta - G_{\ddot{\alpha}\tau}W_aK_{\ddot{\alpha}}}$$

The design objective for the controller K is thus to be set as

$$\|\Phi\|_\infty < 1 \quad \text{for all perturbed} \quad \begin{bmatrix} G_{\alpha\tau}(s) \\ G_{\theta\tau}(s) \\ G_{\ddot{\alpha}\tau}(s) \end{bmatrix} \quad (17.13)$$

It is clear that, with appropriately chosen weighting functions, a controller K satisfying the above equation (17.13) makes the closed-loop system robustly stable, robustly achieving good matching to the dynamic model M (in terms of e_p), and with restricted control effort (in terms of e_u).

The model transfer function to be matched is taken in this design as

$$M = \frac{625}{s^2 + 50s + 625}$$

The coefficients of this transfer function are chosen to ensure overdamped response with a settling time of about 0.19 s. The performance weighting functions are chosen as

$$W_p = \frac{s^2 + 25s + 150}{s^2 + 22s + 0.15}, \quad W_u = 0.001$$

The criterion for the performance weighting function W_p aims to ensure the closeness of the system dynamics to that of the model M over the low-frequency range. The use of the control weighting function W_u allows us to bound the magnitude of the control action in the frequency range containing the natural frequencies of the flexible link. The magnitude plot of the inverse of the performance weighting function is shown in Fig. 17.12.

The noise shaping filters

$$W_{n1} = 10^{-5} \frac{0.5s + 1}{0.005s + 1}, \quad W_{n2} = 10^{-3} \frac{s + 1}{0.01s + 1}$$

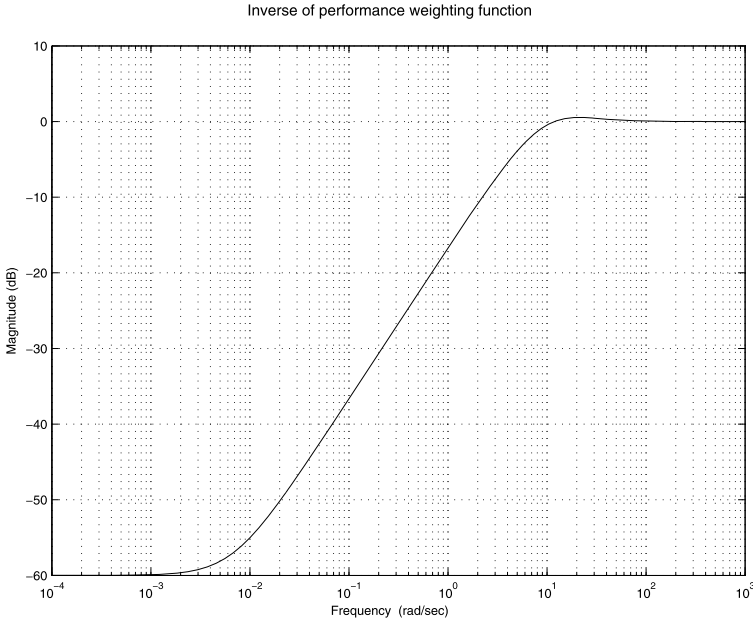


Fig. 17.12 Inverse of performance weighting function

are determined according to the spectral contents of the sensor noises n_1 and n_2 at the measurements of joint angle and the tip acceleration signals, respectively. The magnitude plots of the noise shaping filters are shown in Fig. 17.13.

The model transfer function, the performance and control weighting functions as well as the noise shaping filters are assigned in the file `wt_s_flm.m`.

17.4 System Interconnections

The open-loop interconnection is obtained by the M-file `olp_flm`. The internal structure of the 13 states, five-input/five-output open-loop system, which is saved in the variable `sys_ic`, is shown in Fig. 17.14. The reference and the noises are saved in the variables `ref`, `noise1`, and `noise2`, the disturbance is saved in the variable `dist` and the control signal in the variable `control`.

A schematic diagram of the specific input/output ordering for the variable `sys_ic` is shown in Fig. 17.15.

The block-diagram used in the simulation of the closed-loop system is shown in Fig. 17.16. The corresponding closed-loop interconnection, which is saved in the variable `sim_ic`, is obtained by the M-file `sim_flm`.

A schematic diagram of the specific input/output ordering for the variable `sim_ic` is shown in Fig. 17.17.

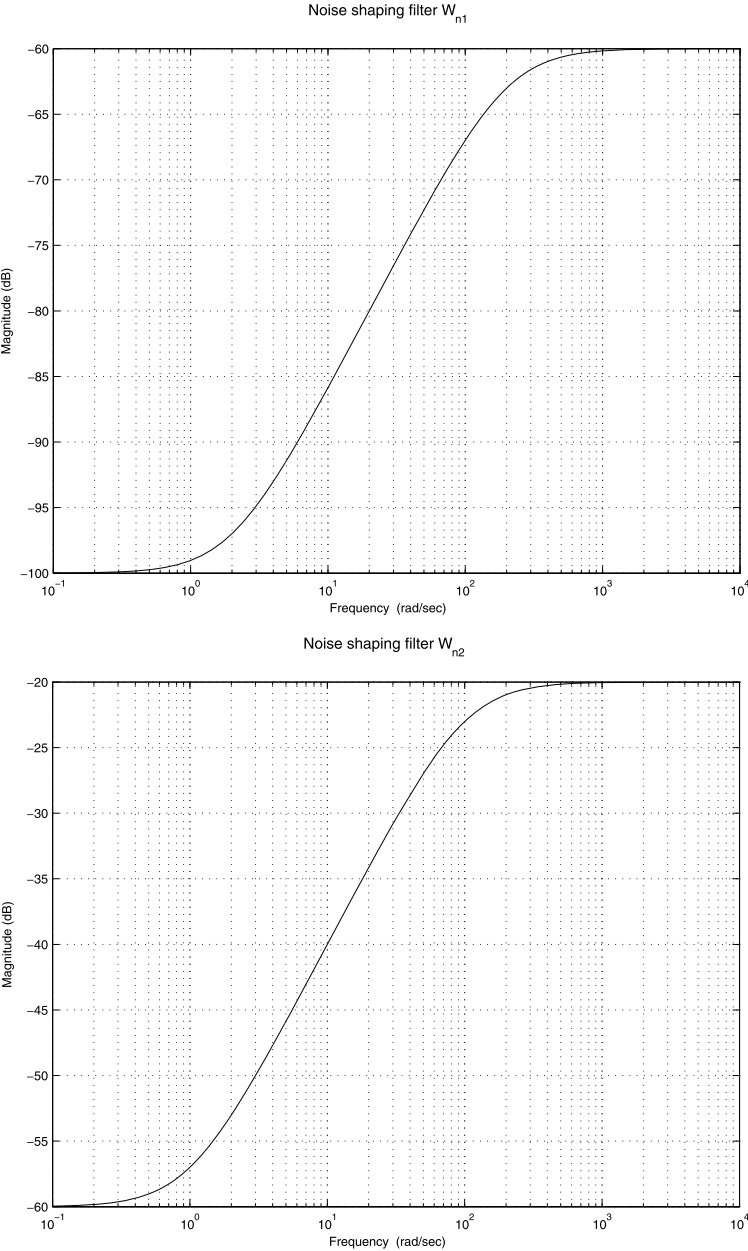


Fig. 17.13 Noise weighting functions

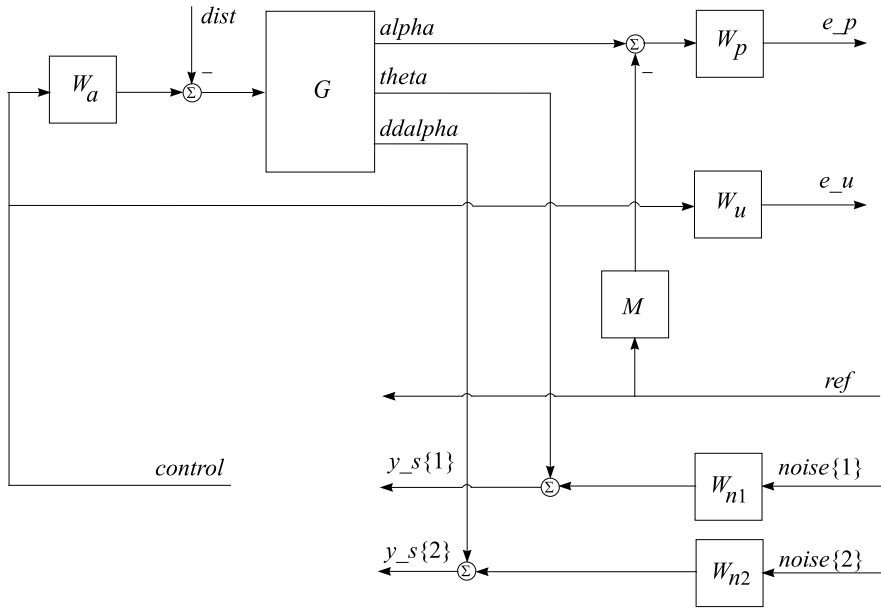
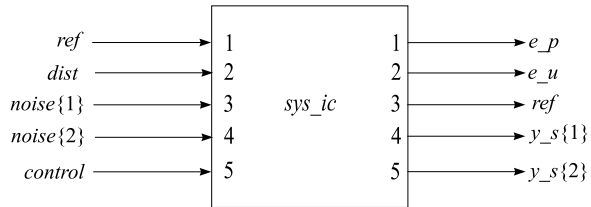


Fig. 17.14 Open-loop interconnection structure of the flexible-link manipulator system

Fig. 17.15 Schematic diagram of the open-loop interconnection



17.5 Controller Design and Analysis

Let us denote by $P(s)$ the transfer function matrix of the five-input, five-output open-loop system consisting of the flexible-link manipulator model and the actuator and weighting functions (Fig. 17.14). We may represent this transfer function as

$$P(s) = F_U(P_{\text{nom}}, \Delta_P)$$

where P_{nom} is the nominal transfer function and the uncertainty Δ_P is given by

$$\Delta_P := \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_F \end{bmatrix} : \Delta \in \mathcal{R}^{11 \times 11}, \Delta_F \in \mathcal{C}^{3 \times 2} \right\}$$

The first part of this matrix corresponds to the 11×11 uncertain block Δ that consists of the uncertainties in the flexible manipulator. The second block Δ_F is a fictitious uncertainty 3×2 block and is introduced to represent the robust performance

Fig. 17.16 Closed-loop interconnection structure of the flexible-link manipulator system

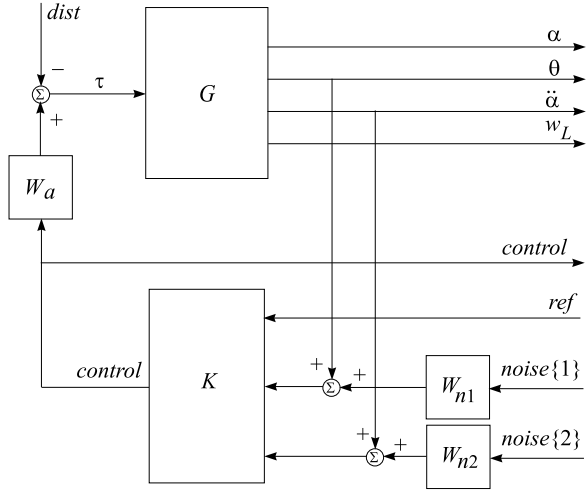
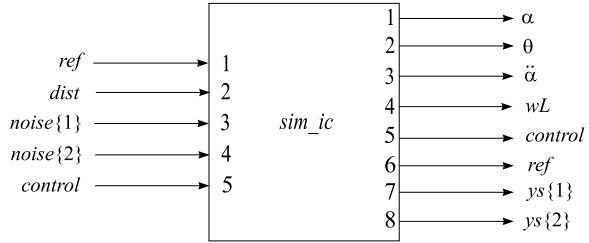


Fig. 17.17 Schematic diagram of the closed-loop interconnection



objective in the framework of the μ -approach. The inputs to the block Δ_F are the weighted error signals e_p and e_u and the outputs from Δ_F are the exogenous signals r , n_1 , and n_2 (inputs to the manipulator closed-loop system).

As discussed in previous sections, in order to meet the design objectives a stabilizing controller $K = [K_r(s) \ K_\theta(s) \ K_{\ddot{\alpha}}(s)]$ is to be found such that, at each frequency $\omega \in [0, \infty]$, the structured singular value satisfies the condition

$$\mu_{\Delta_P} [F_L(P, K)(j\omega)] < 1$$

The fulfillment of the above condition guarantees the robust performance of the closed-loop system, i.e.

$$\|\Phi\|_\infty < 1 \quad (17.14)$$

In the computation of a μ -controller, there is, however, a numerical problem. That is, with the inclusion of the multiple 8×8 real uncertainty block (corresponding to the uncertainty in m_L) the D - K iteration algorithm does not converge. In particular, it is difficult to obtain the approximation of a 8×8 scaling function matrix in the D -step. Hence, in our computation that multiple 8×8 real uncertainty block was removed in the uncertainty matrix during the D - K iteration. It should be

Table 17.1 Results of the μ -synthesis

Iteration	Controller order	Maximum value of μ
1	13	2.035
2	15	0.970
3	17	0.585

stressed that the robust stability and robust performance analysis of the closed-loop system of the designed controller, which will be presented next, is tested with regard to the whole uncertainty structure, i.e. with the inclusion of that multiple 8×8 real uncertainty block.

The μ -synthesis is carried out by using the M-file `ms_f1m.m` implementing the function `dksyn`. In the synthesis, the uncertainty parameter m_L is replaced by its nominal value, as described above.

The progress of the D – K iteration is shown in Table 17.1.

In the design exercise, an appropriate controller is obtained after the third D – K iteration. The controller is stable and has an order of 17. The Bode plots of the μ -controller are shown in Fig. 17.18.

It can be seen from Table 17.1 that after the third iteration the maximum value of μ is equal to 0.585. Note that this, however, does not necessarily mean that the robust performance has been achieved since we neglected the multiple 8×8 real uncertainty block in the computation. Hence, additional robust performance analysis is needed as below.

The robust stability analysis of the closed-loop system is conducted by the file `rbs_f1m`, which takes into account all uncertainty blocks discussed in Sect. 17.2. The result from the execution of this file is

```
stabmarg =
```

```

        UpperBound: 3.4178
        LowerBound: 1.9081
DestabilizingFrequency: 6.2803e+003
```

```
report =
```

```

Uncertain System is robustly stable to modeled uncertainty.
-- It can tolerate up to 191% of the modeled uncertainty.
-- A destabilizing combination of 342% of the modeled
    uncertainty exists, causing an instability at
    6.28e+003 rad/s.
```

This result confirms that the closed-loop system achieves robust stability.

The frequency response plot of the structured singular value corresponding to robust stability analysis is shown in Fig. 17.19.

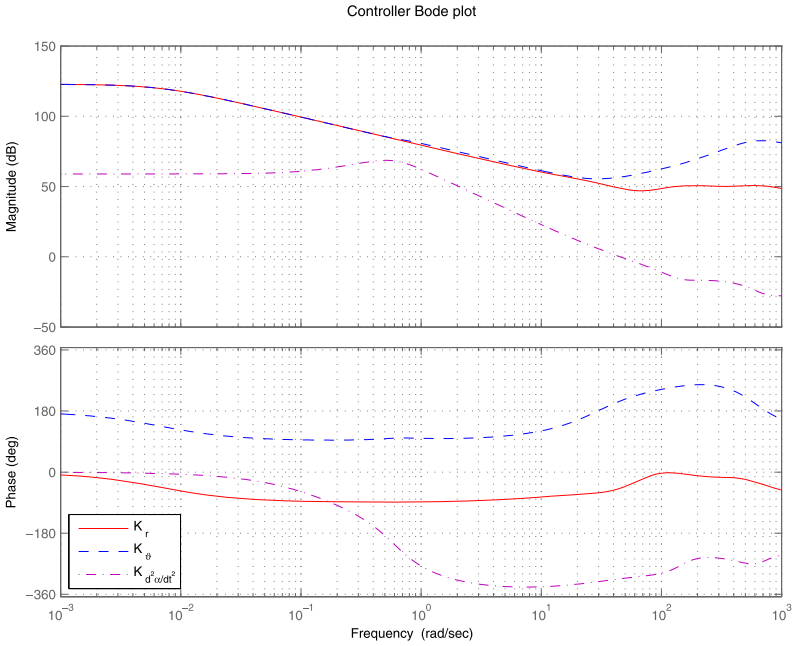


Fig. 17.18 Controller Bode plots

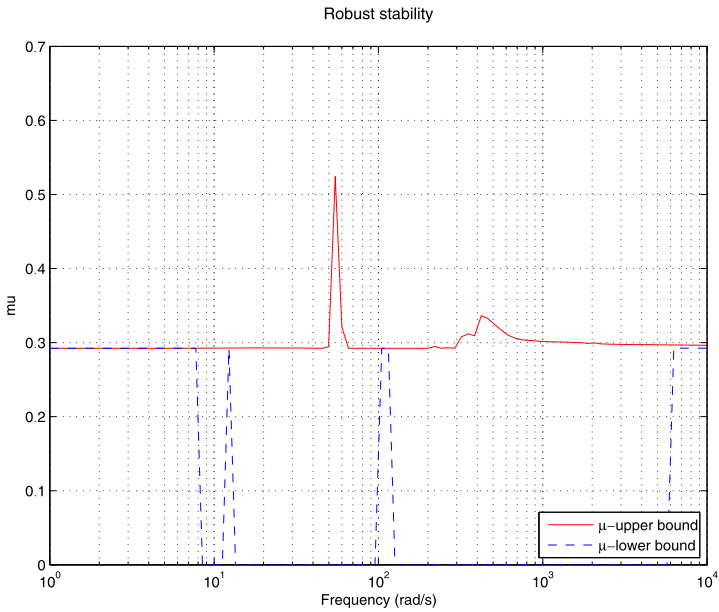


Fig. 17.19 Robust stability for μ -controller

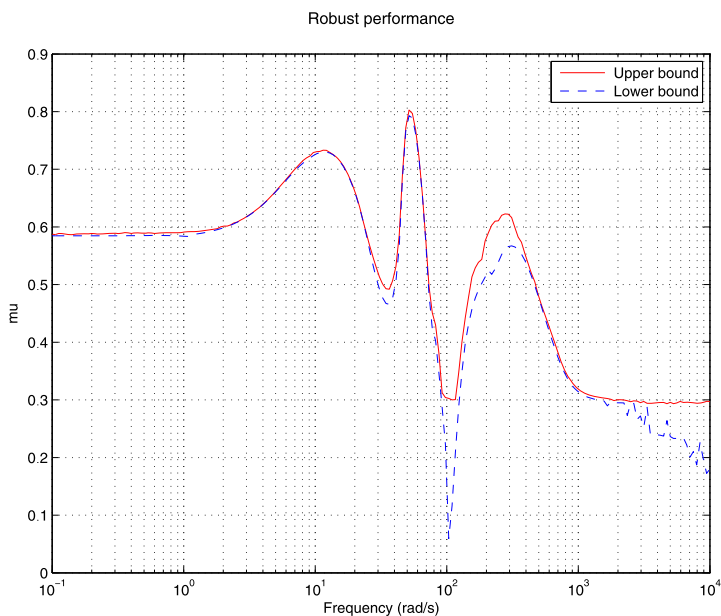


Fig. 17.20 Robust performance for μ -controller

The robust performance analysis is done by using the file `rbp_f1m`. As a result one obtains

```
perfmargin =
```

```
    UpperBound: 1.2612
    LowerBound: 1.2456
    CriticalFrequency: 51.7092
```

```
report =
```

```
Uncertain System achieves a robust performance margin of 1.261.
- A model uncertainty exists of size 125% resulting in a
  performance margin of 0.803 at 51.7 rad/s.
```

The frequency response of μ for the robust performance analysis is shown in Fig. 17.20.

Consider now the closed-loop transient responses that are computed by using the M-file `mcs_f1m`.

The reference trajectory for the manipulator tip movement in the simulation is chosen in the form

$$r = \begin{cases} at - (a/\psi) \sin(\psi t) + r_0, & 0 \leq t \leq t_m \\ r(t_m), & t_m < t \leq t_f \end{cases}$$

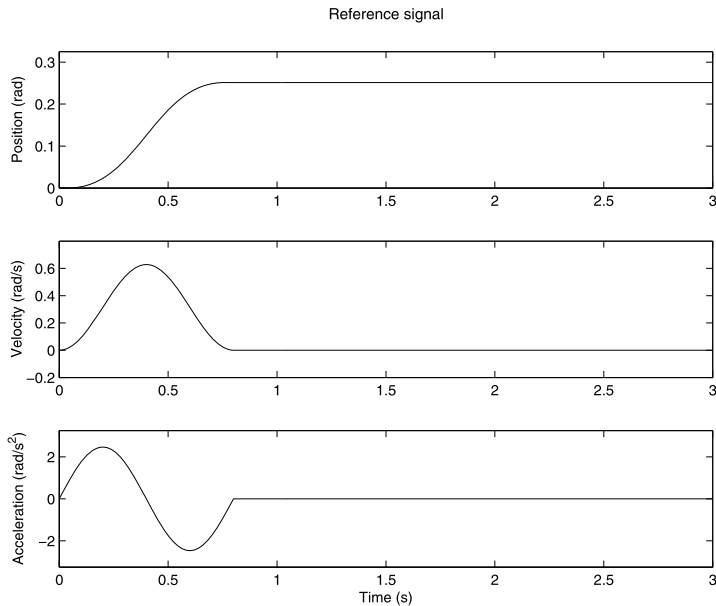


Fig. 17.21 Reference signal $r(t)$, velocity and acceleration

This trajectory allows the tip to be moved smoothly from an arbitrary initial position r_0 to a desired final position $r(t_m) = at_m$, with an appropriate ψ .

In the simulation, the following numerical values of the parameters are taken: $a_r = 0.1\pi$ rad/s; $\omega_r = 2.5\pi$ s⁻¹; $r_0 = 0$ rad; $t_m = 0.8$ s; $t_f = 3$ s. They are chosen such that to obtain a fast changing reference, that is, settling in the desired final position $\pi/12.5$ rad for 0.8 s.

The plots of the reference $r(t)$, velocity, and acceleration for 30 random combinations of the uncertain parameters are shown in Fig. 17.21.

In Fig. 17.22 we show the transient response of the tip position α along with the joint angle θ and the reference r , computed for 30 random combinations of the uncertain plant parameters.

The transient response corresponding to the tip deflection $w(L, t)$ is shown in Fig. 17.23.

The control action generated by the designed μ -controller is shown in Fig. 17.24.

The closed-loop frequency responses of the uncertain closed-loop system are obtained by the M-file `frs_flm.m`.

The Bode plot of the closed-loop system is shown in Fig. 17.25. The closed-loop bandwidth is about 10 rad/s. Note that a good match in magnitude between the closed-loop system and the dynamic model M is achieved for frequencies up to 100 rad/s.

The Bode plots of the tip-deflection transfer function are shown in Fig. 17.26. The maximum amplitude of the tip deflection is observed for the input signal with frequency 50 rad/s.

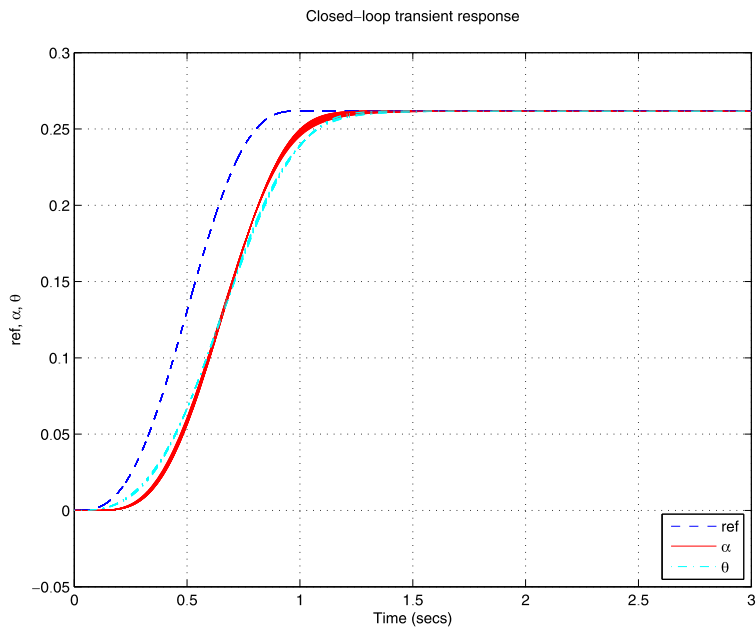


Fig. 17.22 Closed-loop transient response for μ -controller

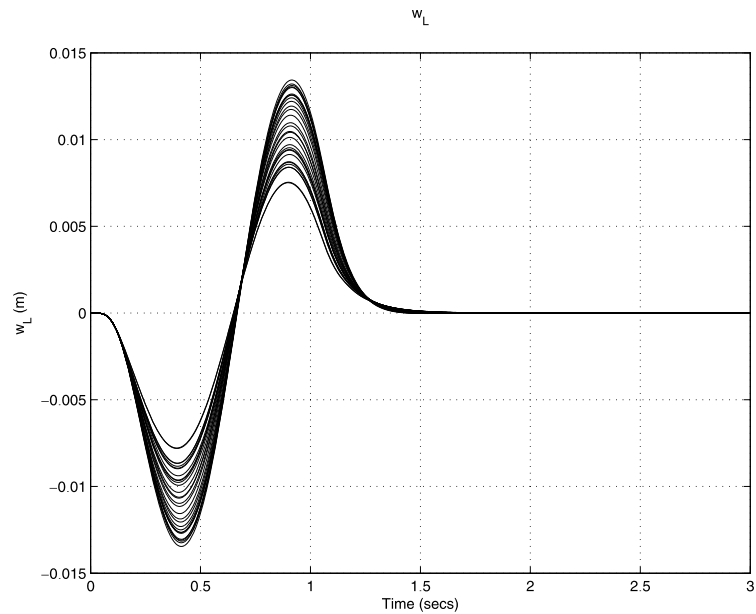


Fig. 17.23 Tip-deflection transient response for μ -controller

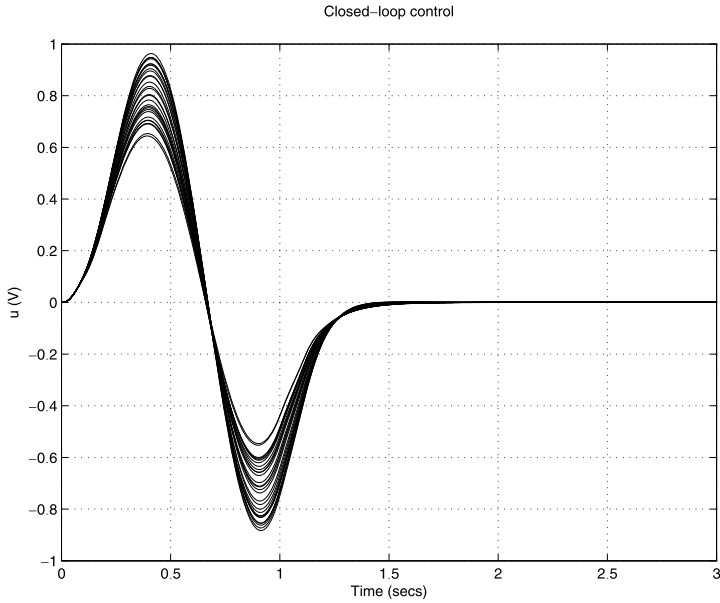


Fig. 17.24 Control action for μ -controller

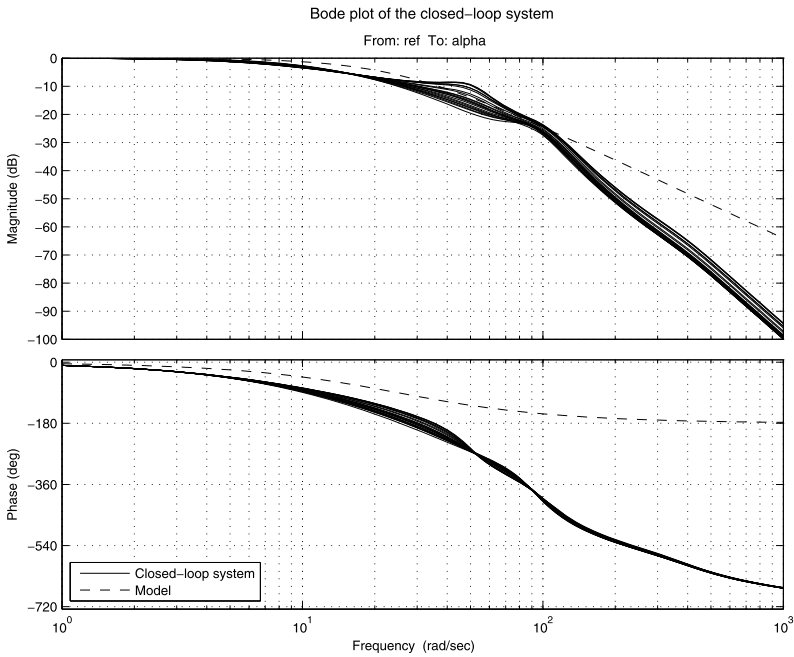


Fig. 17.25 Bode plot of the uncertain system

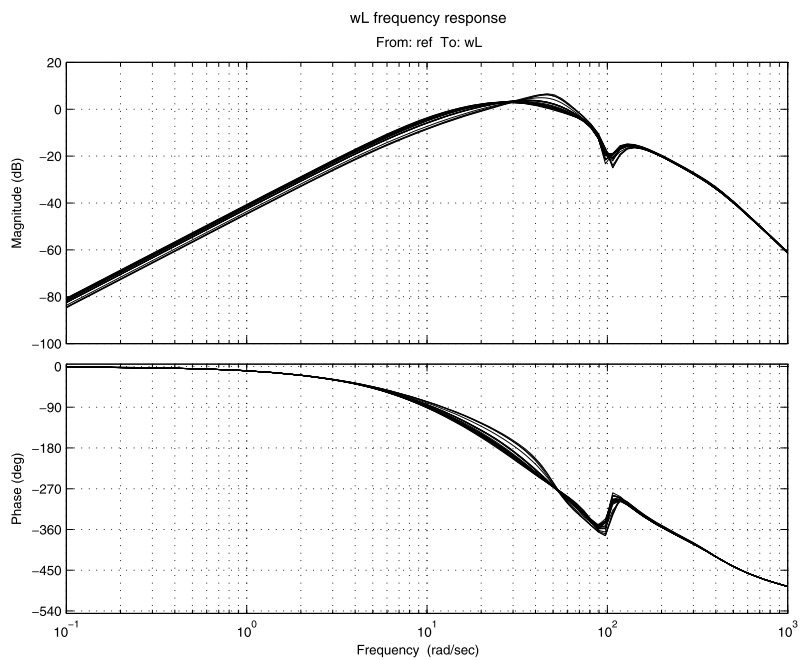


Fig. 17.26 Bode plots for the tip deflection

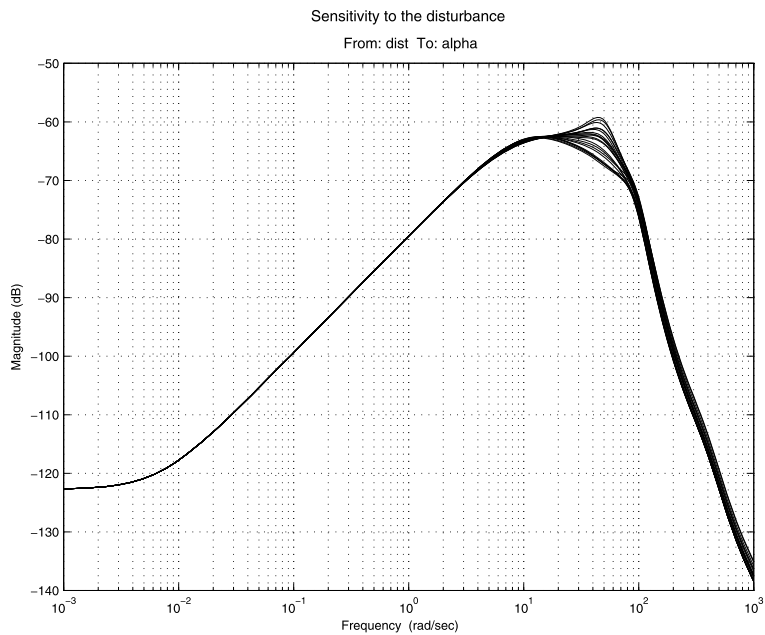


Fig. 17.27 Bode plots for the disturbance

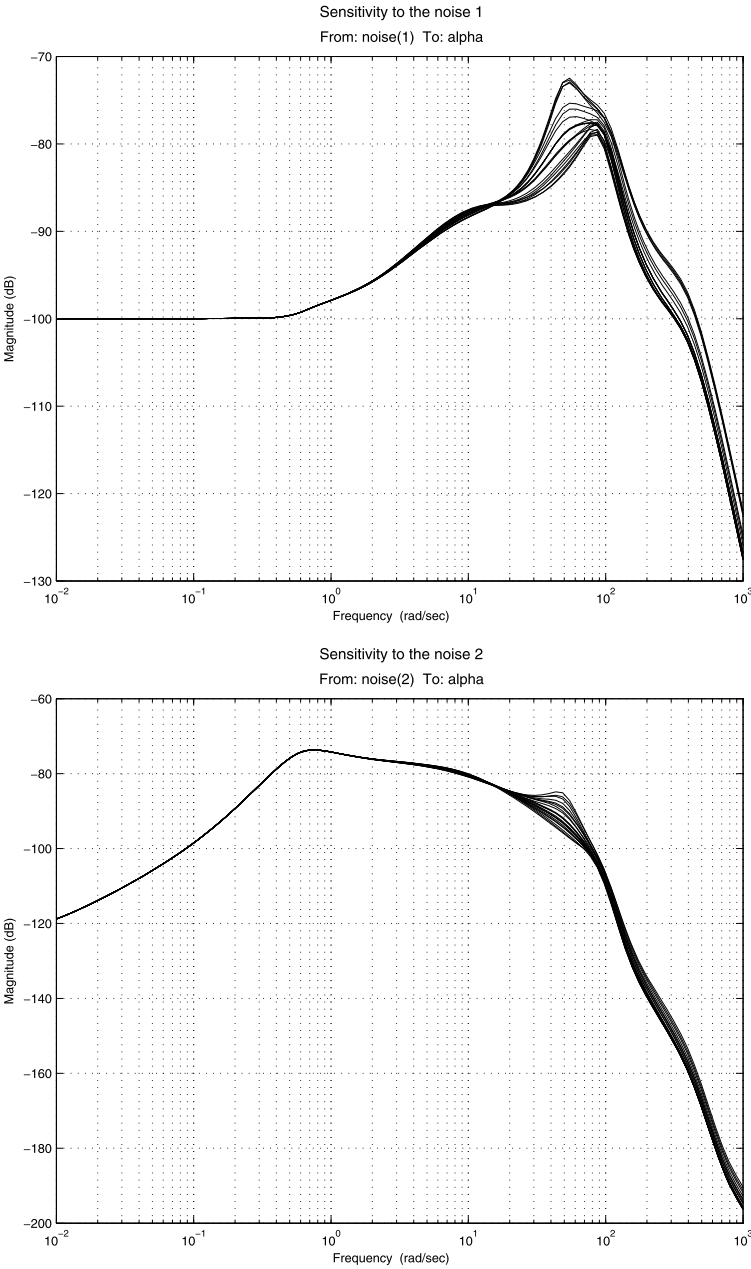


Fig. 17.28 Magnitude plots for the first and second noise

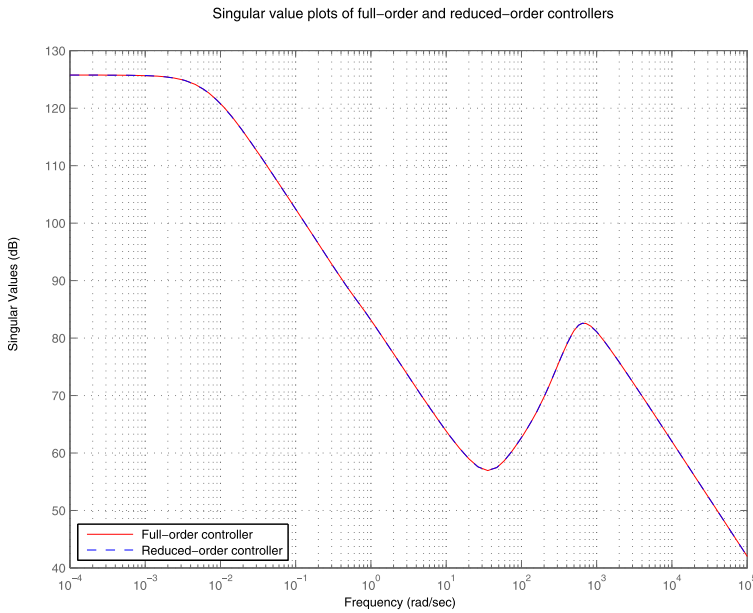


Fig. 17.29 Frequency responses of the full-order and reduced-order controllers

In Fig. 17.27 we show the Bode plots of the uncertain system obtained from the transfer function with respect to the disturbance.

Finally, in Fig. 17.28 we show the magnitude plots with respect to the first and second noise. It is seen from the figure that the noise in measuring the joint angle has a negligible effect on the system output.

We consider now the order reduction of the designed controller. As indicated in Table 17.1, the order of the μ -controller is 17. It would be good for implementation if the order could be reduced while essentially keeping the achieved performance. For this aim, we use the M-file `red_f1m.m`. After the balanced realization transformation of the controller and by neglecting the small Hankel singular values, the order of the controller can be reduced to 12 without losing too much performance.

In Fig. 17.29 we compare the frequency responses of the maximum singular values of the full-order and reduced-order controllers. The frequency responses of both full-order and reduced-order controllers practically coincide. The transient responses of the closed-loop system with full-order and with reduced-order controller are also practically indistinguishable. (Figures are not included here.)

It is interesting to compare the results obtained with the μ -controller with those from the conventional collocated PD controller in the form of

$$u = k_P(r - \theta) - k_D\ddot{\theta}$$

The proportional and derivative coefficients are chosen as $k_P = 358$ N m/rad and $k_D = 28.5$ N m/(rad/s). The values of k_P and k_D are selected such that after neglect-

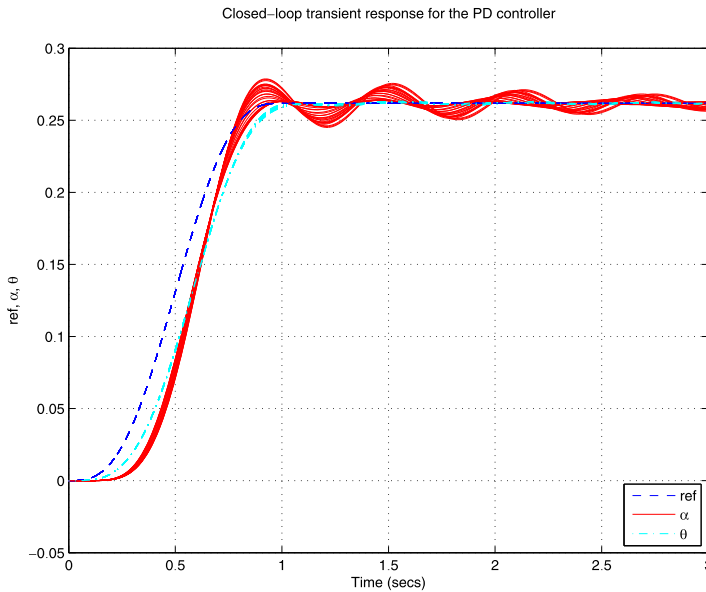


Fig. 17.30 Closed-loop transient response for the PD controller

ing the link flexibility the closed-loop transfer function coincides with the transfer function of the model. The results by using the μ -analysis method in this case (i.e. with the PD controller) are 3.4212 and 0.1398 for the stability margin and performance margin, respectively. Therefore, the PD controller leads to poor robust performance in comparison to the μ -controller designed. This can be seen by comparing Figs. 17.30 and 17.31 with Figs. 17.22 and 17.23, respectively.

It has to be noticed that good results in the design may also be obtained by using a collocated controller on the feedback from the joint angle θ and the velocity $\dot{\theta}$. The use of the tip acceleration \ddot{a} , however, allows better results with respect to the robust performance to be obtained.

17.6 Nonlinear System Simulations

The performance of the μ -controller designed in the previous section is further investigated by simulations of the nonlinear closed-loop system with this controller. The simulation is carried out by the Simulink® model `nls_flm.mdl` using the nonlinear plant model (17.5). A number of simulations may be performed for several values of the payload mass and of the damping coefficients. The Simulink® model `nls_flm.mdl` is shown in Fig. 17.32.

Before running the simulation, it is necessary to set the model parameters by using the M-file `init_flm.m`. The values of the damping coefficients correspond to the case of light damping of the mechanical structure. In particular, the value of

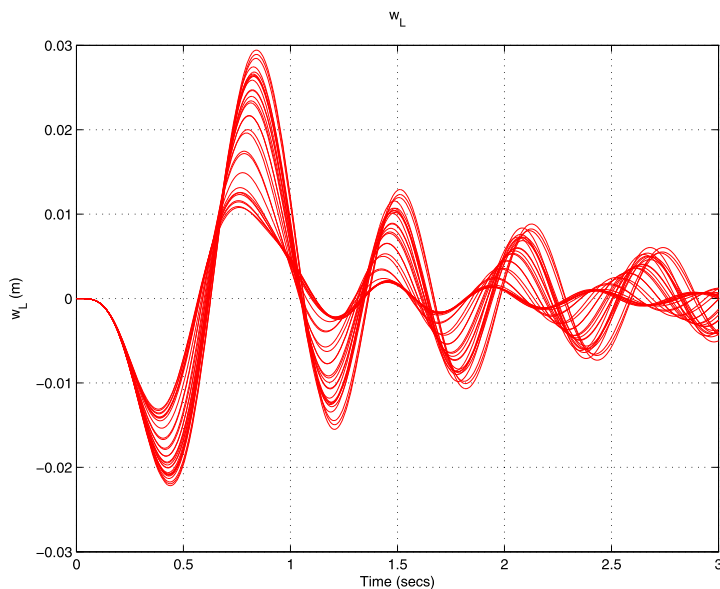


Fig. 17.31 Tip-deflection transient response for the PD controller

the hub damping coefficient d_r corresponds to a relative uncertainty of -20% . The damping coefficients d_1 and d_2 are taken so that the respective relative perturbations in d_1 and d_2 for the nominal payload are equal to -40% .

The time response of the tip position $\alpha(t)$, along with the joint angle θ and the reference r , for the case of the reduced-order μ -controller and nominal payload mass is given in Fig. 17.33.

Apart from the Simulink[®] model, in this section we also present the M-files `s3d_f1m` and `p3d_f1m`, which allow to animate the motion of the flexible manipulator with μ -controller and PD controller, respectively. These files also utilize the nonlinear manipulator model.

In Figs. 17.34 and 17.35 we show the elastic deflection $w(x, t)$ as a function of x and t , obtained for the μ -controller and PD controller by the files `s3d_f1m` and `p3d_f1m`, respectively.

In the nonlinear system simulations, it is shown that the μ -controller efficiently suppresses the elastic vibrations during the fast motion of the manipulator tip. It thus justifies that the μ -synthesis is an appropriate robust design method in this exercise. It also confirms the validity of the uncertain model derived.

17.7 Conclusions

A few conclusions may be drawn as the following, based on the analysis and design of the flexible manipulator control system:

SIMULINK Model of the Flexible Manipulator System

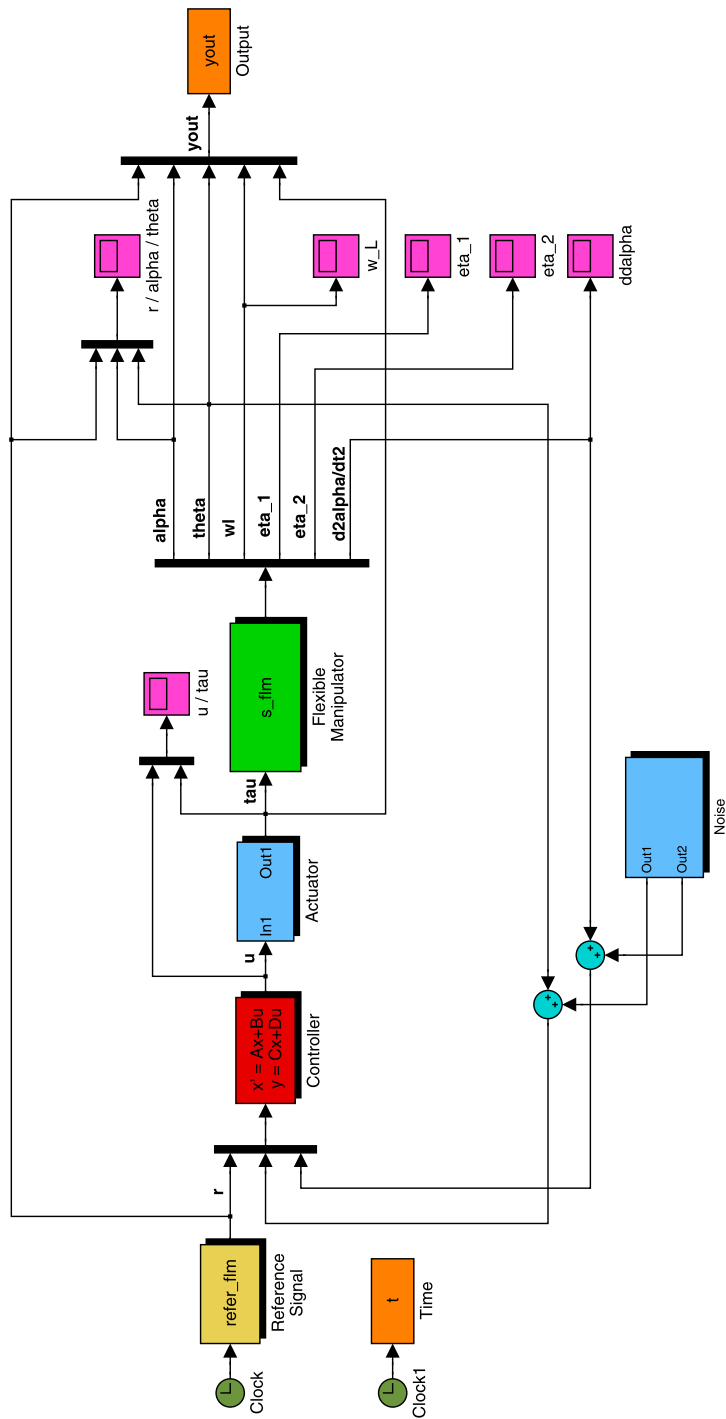


Fig. 17.32 Simulation model of the nonlinear system

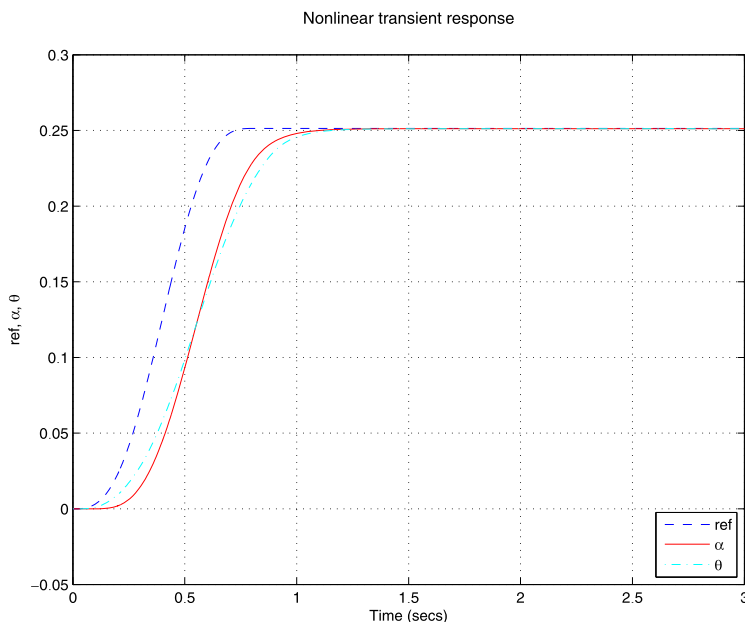


Fig. 17.33 Transient response of the nonlinear system

- In applying linear robust control system design techniques for a nonlinear plant it is usually unavoidable to derive a complicated uncertainty model, because of the requirement of a sufficiently accurate linear approximation. That would, however, adversely affect the controller design and analysis. It is important, therefore, to simplify the model of uncertainty. Methods such as the numerical approximation used in this study can be considered.
- The uncertainty model derived in this study for the flexible manipulator system contains real parametric uncertainties in a highly structured form. Such a model appeals naturally to the application of μ -synthesis and analysis method, which greatly reduces the conservativeness in the controller design.
- A robust noncollocated controller on the feedback signals of joint angle and tip acceleration is designed in this study on the basis of the uncertainty model derived and by using the μ -synthesis. The μ -controller shows very good robust performance on the tip motion for a wide range of payload mass. The controller efficiently suppresses the elastic vibrations during the fast motion of the manipulator tip.
- The nonlinear system simulation results confirm the high performance of the controller designed and also verify the validity of the uncertain model used.
- It is also possible to investigate various noncollocated and collocated controller structures on different output feedback signals, with the uncertainty model and linearized plant derived in this study.

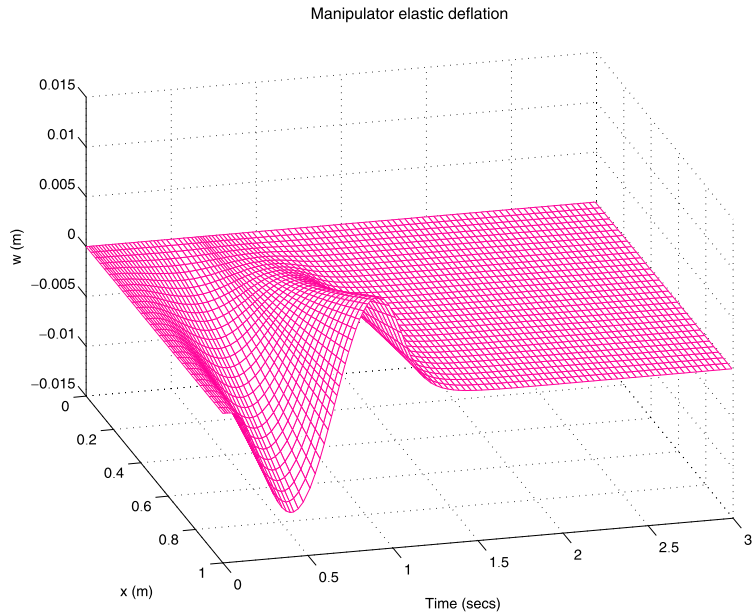


Fig. 17.34 Elastic deflection $w(x, t)$ for the case of μ -controller

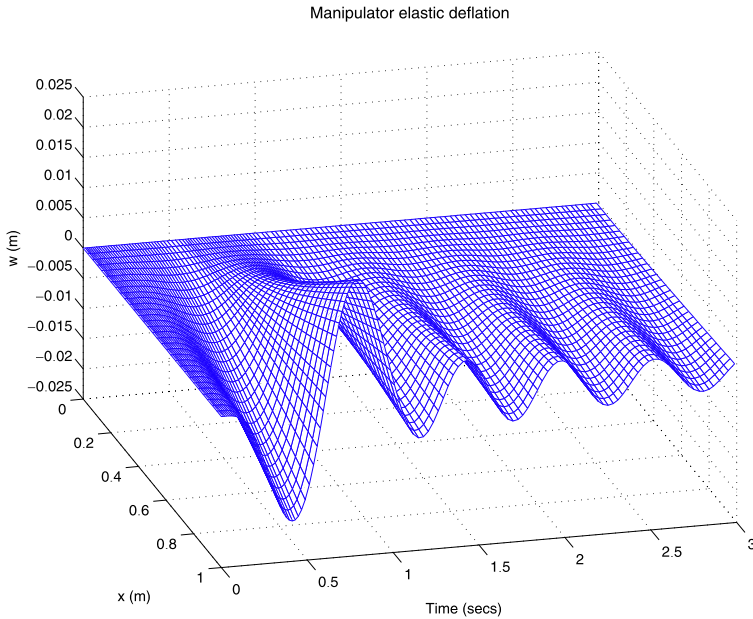


Fig. 17.35 Elastic deflection $w(x, t)$ for the case of PD-controller

17.8 Notes and References

The control of flexible manipulators has been an area of intensive research in recent years. An efficient approach to improve the manipulator performance is to use a feedback from the manipulator tip position [53], tip acceleration [51] or base-strain [52]. The usage of such feedbacks leads to a noncollocated control scheme that may increase the closed-loop system sensitivity to modeling errors or to parameter uncertainties [137].

The necessity to achieve robustness of the manipulator control system in the presence of uncertainties makes it appropriate to apply the robust control design methods. In a few recent papers the authors develop different \mathcal{H}_∞ controllers [54, 96, 159] and μ -synthesis controllers [82] for flexible-link manipulators. A common disadvantage in the previous robust designs for flexible manipulators is the use of an unstructured uncertainty model that leads to potentially very conservative results.

This page intentionally left blank

Chapter 18

Robust Control of a Twin-Rotor Aerodynamic System

This chapter presents the design and experimental evaluation of a 2-degree-of-freedom, discrete-time, μ -controller for a laboratory Twin-Rotor Aerodynamic System with ten uncertain parameters. The controller implemented is of 24th order and ensures robust stability and robust performance of the closed-loop, sampled-data system. This controller is realized on a PC by using Simulink Coder[®] with sampling frequency of 100 Hz. The experimental results are close to the results predicted by using the linearized model of the system and highlight some of the difficulties associated with implementation of robust control laws.

18.1 Twin-Rotor Aerodynamic System

The Twin-Rotor Aerodynamic System (TRAS) is a laboratory set-up designed for control experiments whose behavior resembles that of helicopters. The set-up is shown in Fig. 18.1. There are two propellers at either end of a beam, driven by DC motors, joined to the base with an articulation. The articulation allows the beam to be rotated such that its ends move on a spherical surface. The main propeller (the propeller with larger diameter) controls the beam position in the vertical plane, while the tail propeller controls the beam position in the horizontal plane. There are two counter-weights fixed to the beam that determine the stable equilibrium position. The system is balanced in such a way that, when the motors are switched off, the main rotor end of beam is lowered. The control actions are the motor supply voltages. The measured system outputs are the two angles of beam deviation in the horizontal plane (azimuth angle) and in the vertical plane (pitch angle). The motor control is realized in Pulse-Width Modulation (PWM) mode.

The TRAS control aims at stabilization of the beam in an arbitrary (within practical limits) desired position (pitch and azimuth) or tracking of a desired trajectory. In the given case we consider as measurable outputs the pitch and azimuth angles only. In the general case it is possible also to use the information of propeller angular velocities.

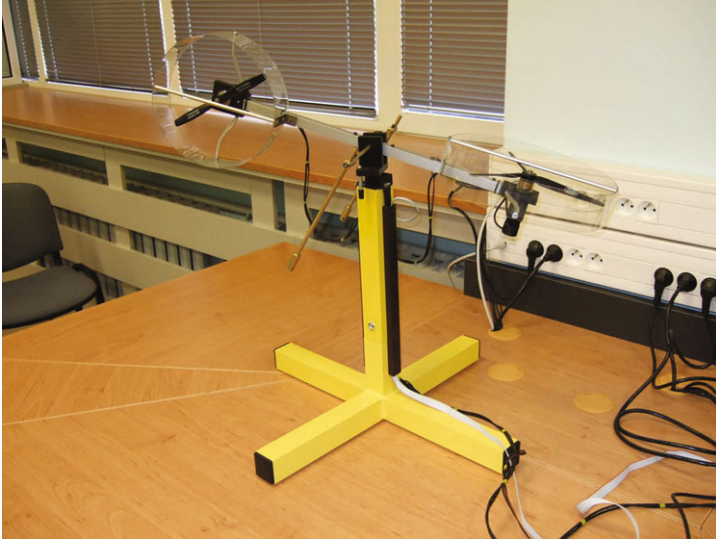


Fig. 18.1 Twin-Rotor Aerodynamic System

18.2 Nonlinear System Model

The mathematical model of TRAS is derived under some simplifying assumptions. First of all, it is supposed that the dynamics of the propeller subsystem may be described by first order differential equations. Next, it is assumed that the friction in the system is of viscous type (i.e., proportional to the angular velocity). Furthermore, it is supposed that the “propeller–air” subsystem may be described according to the principles of flow theory.

Consider first the rotation of the beam in the vertical plane, i.e., around the horizontal axis. Taking into account that the control torques are produced by the propellers, the beam rotation motion is described as

$$J_v \frac{d^2 \alpha_v}{dt^2} = M_v \quad (18.1)$$

where M_v is the total moment of forces in the vertical plane, J_v is the sum of moments of inertia relative to the horizontal axis, and α_v is the pitch angle of the beam (the beam rotation angle in the vertical plane).

The total moment of forces may be represented as the sum $M_v = M_{v1} + M_{v2} + M_{v3} + M_{v4} + M_{v5} + M_{vd}$, in which the individual terms are described as follows.

1. Return torque corresponding to the gravitational forces

$$M_{v1} = -k_1 \cos(\alpha_v) - k_2 \sin(\alpha_v)$$

where the coefficients k_1 and k_2 are determined by the mass and geometrical sizes of the beam and the devices mounted on it.

2. Moment of the propulsive force produced by the main propeller

$$M_{v2} = l_m F_v(\omega_v)$$

where l_m is the length of the main part of the beam, ω_v is the angular velocity of the main propeller, and $F_v(\omega_v)$ is the dependence of the propulsive force on the angular velocity of the main rotor.

3. Moment of centrifugal forces corresponding to the beam motion around the vertical axis

$$M_{v3} = -k_3 \Omega_h^2 \sin(\alpha_v) \cos(\alpha_v)$$

where $\Omega_h = d\alpha_h/dt$ is the angular velocity of the beam around the vertical axis, α_h is the beam azimuth angle (the beam rotation angle in the horizontal plane), and k_3 is a coefficient, dependent on the mass and geometrical sizes of the beam and devices mounted on it.

4. Friction moment depending on the angular velocity of the beam around the horizontal axis

$$M_{v4} = -k_{fv} \Omega_v$$

where $\Omega_v = d\alpha_v/dt$ is the angular velocity around the horizontal axis and k_{fv} is a constant.

5. Cross moment from the control force in the horizontal plane

$$M_{v5} = k_{hv} u_h$$

where u_h is the azimuth control action and k_{hv} is a constant.

6. Disturbance torque M_{vd} in the vertical plane.

The dependence of the main propeller propulsive force $F_v(\omega_v)$ on the rotor angular velocity, obtained experimentally, is shown in Fig. 18.2. This dependence is approximated using the M-file `approx_char.m` by the fifth order polynomial

$$\begin{aligned} \tilde{F}_v = & -7.31 \times 10^{-19} \omega_v^5 - 3.79 \times 10^{-16} \omega_v^4 + 2.41 \times 10^{-11} \omega_v^3 \\ & + 1.87 \times 10^{-8} \omega_v^2 + 2.89 \times 10^{-5} \omega_v - 0.0142 \end{aligned}$$

In the same way, it is possible to describe the beam rotation around the vertical axis. Taking into account that the control torque is produced by the tail rotor and that the moment of inertia depends on the pitch angle, the rotation in the horizontal plane is described by

$$J_h \frac{d^2 \alpha_h}{dt^2} = M_h \quad (18.2)$$

where M_h is the total moment of forces in the horizontal plane, J_h is the sum of moment of inertia around the vertical axis, $J_h = k_4 \cos^2(\alpha_v) + k_5$, where the coefficients k_4 and k_5 are determined by the mass and geometrical sizes of the beam and devices mounted on it.

The total moment of forces may be represented by the sum $M_h = M_{h1} + M_{h2} + M_{h3} + M_{hd}$, in which the individual terms are described in the following way.

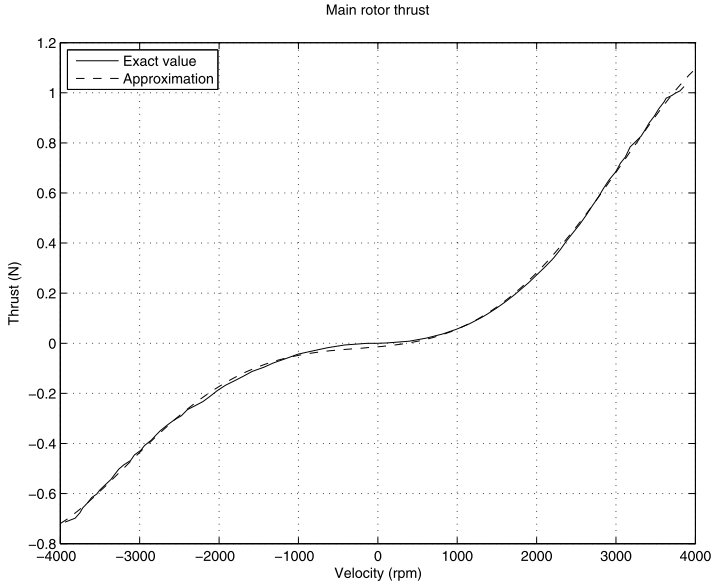


Fig. 18.2 Main propeller thrust

1. Moment of the propulsive force produced by the tail rotor

$$M_{h1} = l_t F_h(\omega_h) \cos(\alpha_v)$$

where l_t is the length of the tail part of the beam, ω_h is the angular velocity of the tail rotor, and $F_h(\omega_h)$ is the dependence of the tail propulsive force on the rotor angular velocity.

2. Friction moment, depending on the beam angular velocity around the vertical axis

$$M_{h2} = -k_{f_h} \Omega_h$$

where k_{f_h} is a constant.

3. Cross moment from the control action in the horizontal plane

$$M_{h3} = k_{v_h} \cos(\alpha_v) u_v$$

where u_v is the pitch control action and k_{v_h} is a constant.

4. Disturbance torque in the horizontal plane M_{hd} .

The dependence of the propulsive force $F_h(\omega_h)$ of the tail propeller on the rotor angular velocity, obtained experimentally, is shown in Fig. 18.3. This dependence is approximated by the fifth order polynomial

$$\begin{aligned} \tilde{F}_h = & -2.56 \times 10^{-20} \omega_h^5 - 4.10 \times 10^{-17} \omega_h^4 + 3.17 \times 10^{-12} \omega_h^3 \\ & - 7.34 \times 10^{-9} \omega_h^2 + 2.13 \times 10^{-5} \omega_h - 9.14 \end{aligned}$$

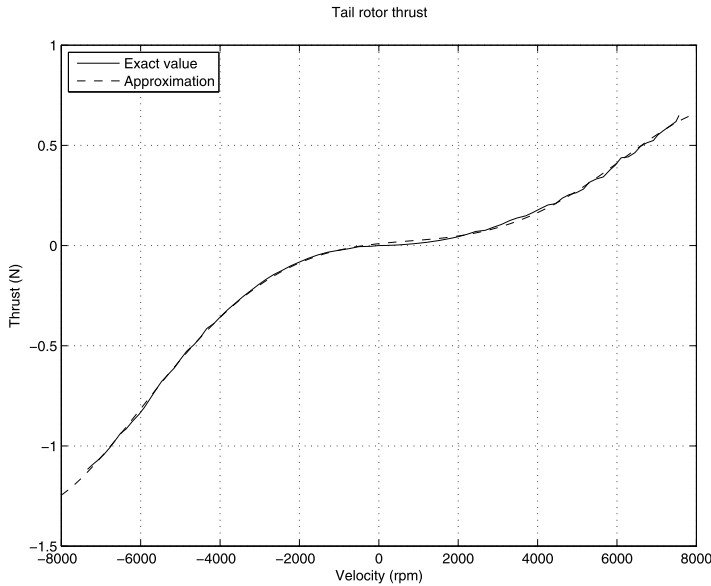


Fig. 18.3 Tail propeller thrust

Consider now the propeller motion equations. Since the drive motors are controlled by Pulse Width Modulation, the control actions are normalized so that they vary in the range $[-1, +1]$, which corresponds to the variation of the input voltage in the interval $[-24 \text{ V}, +24 \text{ V}]$. The equation describing the motion of the main propeller is written in the form

$$I_v \frac{d\omega_v}{dt} = u_v - H_v^{-1}(\omega_v) \quad (18.3)$$

where I_v is the moment of inertia of the main propeller and $\omega_v = H_v(u_v)$ is the static velocity characteristic of this propeller.

The experimentally obtained characteristic $\omega_v = H_v(u_v)$ is shown in Fig. 18.4. This dependence is approximated by the seventh order polynomial

$$\begin{aligned} \tilde{\omega}_v = & -6.17 \times 10^3 u_v^7 - 1.30 \times 10^2 u_v^6 + 1.37 \times 10^4 u_v^5 + 1.50 \times 10^2 u_v^4 \\ & - 1.10 \times 10^4 u_v^3 - 3.76 \times 10^1 u_v^2 + 7.33 \times 10^3 u_v - 5.36 \end{aligned}$$

Similarly, the equation describing the motion of the tail propeller is written as

$$I_h \frac{d\omega_h}{dt} = u_h - H_h^{-1}(\omega_h) \quad (18.4)$$

where I_h is the moment of inertia of tail rotor and $\omega_h = H_h(u_h)$ is the static velocity characteristic of this propeller.

The experimentally obtained characteristic $\omega_h = H_h(u_h)$ is shown in Fig. 18.5. This characteristic is approximated by the fifth order polynomial

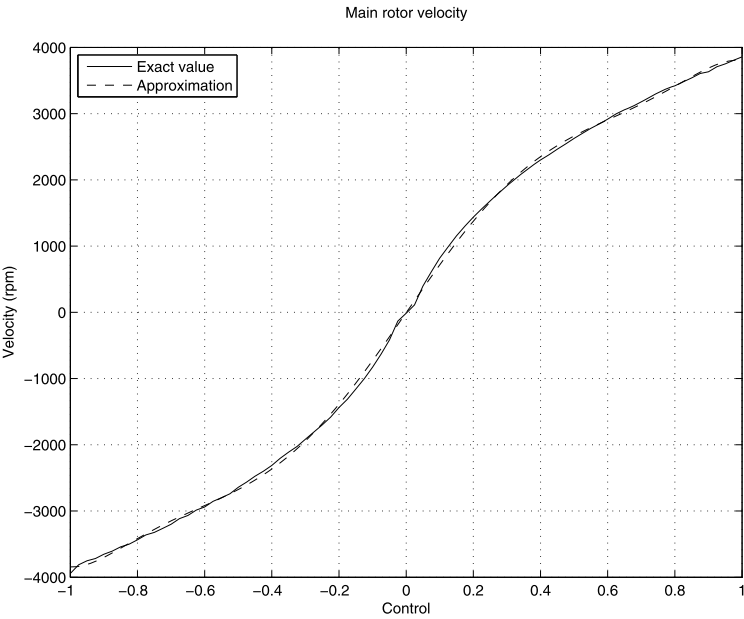


Fig. 18.4 Main rotor velocity characteristic

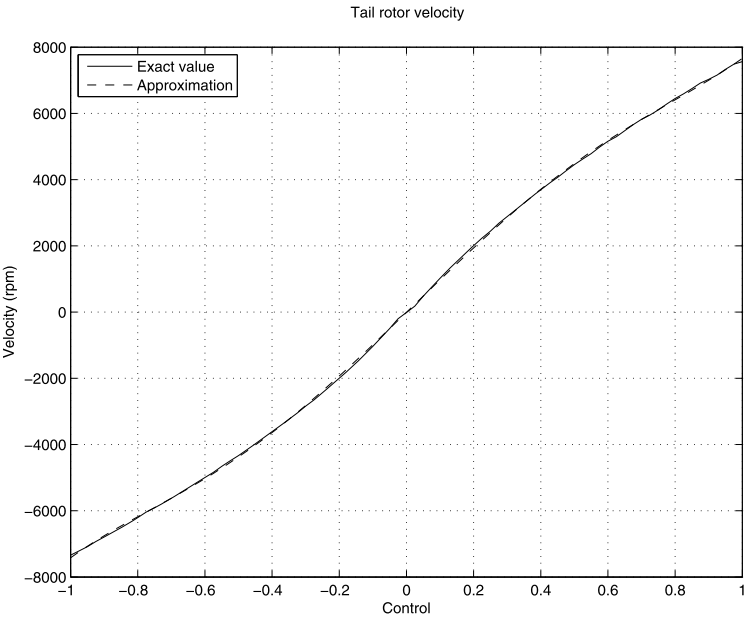


Fig. 18.5 Tail rotor velocity characteristic

Table 18.1 Parameters of the nonlinear model

Parameter	Value	Units
I_h	1/37000	kg m ²
I_v	1/6100	kg m ²
J_v	3.00581×10^{-2}	kg m ²
k_1	5.00576×10^{-2}	Nm
k_2	9.36008×10^{-2}	Nm
k_3	2.12485×10^{-2}	Nm s ² /rad ²
k_4	2.37904×10^{-2}	kg m ²
k_5	3.00962×10^{-3}	kg m ²
k_{fh}	5.88996×10^{-3}	Nm s/rad
k_{fv}	1.27095×10^{-2}	Nm s/rad
k_{hv}	4.17495×10^{-3}	Nm
k_{vh}	-1.78200×10^{-2}	Nm
l_m	0.202	m
l_t	0.216	m

$$\begin{aligned}\tilde{\omega}_h = & -6.17 \times 10^3 u_h^5 - 1.30 \times 10^2 u_h^4 + 1.37 \times 10^4 u_h^3 \\ & + 1.50 \times 10^2 u_h^2 - 1.10 \times 10^4 u_h - 37.6\end{aligned}$$

Equations (18.1)–(18.4) together with the equations

$$\frac{d\alpha_v}{dt} = \Omega_v \quad (18.5)$$

$$\frac{d\alpha_h}{dt} = \Omega_h \quad (18.6)$$

constitute the sixth order nonlinear model of the Twin-Rotor Aerodynamic System. The input variables are the voltages u_h and u_v of the tail rotor and main rotor motors, respectively, and output variables are the azimuth angle α_h and pitch angle α_v . The plant is two-channel and there is an interaction between the two channels. In order to reveal in full the dynamic behavior of the plant it should be considered as multivariable, i.e., the two channels cannot be considered as independent.

The nominal values of the coefficients participating in the model are given in Table 18.1.

18.3 Linearized System Model

The model of the Twin-Rotor Aerodynamic System is linearized analytically under the usual assumption for small deviations of the variables, describing the system behavior, around their trim values. The variables describing the beam angular position are represented in the form

Table 18.2 Propeller coefficients

Parameter	Value	Units
k_{H_h}	9.83891×10^3	rad/s
k_{F_h}	2.12932×10^{-5}	Ns/rad
k_{H_v}	4.87457×10^3	rad/s
k_{F_v}	3.07723×10^{-4}	Ns/rad

$$\begin{aligned}\alpha_v &= \alpha_{v,\text{nom}} + \delta\alpha_v, & \omega_v &= \omega_{v,\text{nom}} + \delta\omega_v, & \Omega_v &= \Omega_{v,\text{nom}} + \delta\Omega_v \\ \alpha_h &= \alpha_{h,\text{nom}} + \delta\alpha_h, & \omega_h &= \omega_{h,\text{nom}} + \delta\omega_h, & \Omega_h &= \Omega_{h,\text{nom}} + \delta\Omega_h\end{aligned}$$

where $\alpha_{v,\text{nom}}$, $\omega_{v,\text{nom}}$, $\Omega_{v,\text{nom}}$, $\alpha_{h,\text{nom}}$, $\omega_{h,\text{nom}}$, $\Omega_{h,\text{nom}}$ denote the nominal values of the corresponding parameters and $\delta\alpha_v$, $\delta\omega_v$, $\delta\Omega_v$, $\delta\alpha_h$, $\delta\omega_h$, $\delta\Omega_h$ denote the deviations from their nominal values. In a similar way the motor control voltages are represented as

$$u_v = u_{v,\text{nom}} + \delta u_v, \quad u_h = u_{h,\text{nom}} + \delta u_h$$

The linearization is done around steady operation point assuming that

$$\Omega_{v,\text{nom}} = 0, \quad \Omega_{h,\text{nom}} = 0$$

It is supposed that the position of the counter-weights is chosen so that for an input signal $u_{v,\text{nom}} = 0.3$ the beam is in strictly horizontal position. In this equilibrium position of the beam the main rotor thrust is balanced by the gravitational forces acting on the beam and the devices mounted on it.

Consider first the linearization of the static characteristics of two propellers. It is done by using the M-file `approx_char.m`.

The operation point of the main propeller is determined for motor voltage $u_{v,\text{nom}} = 0.3$. The velocity characteristic of this propeller is approximated by the linear relationship $\omega_v = k_{H_v} u_v$. The coefficient k_{H_v} is found by differentiating the approximating polynomial $\tilde{\omega}_v$ in u_v and substituting the motor voltage by the value $u_{v,\text{nom}} = 0.3$ in the result obtained. Since the linear relationship obtained may be written as $u_v = \frac{1}{k_{H_v}} \omega_v$, then it follows that the inverse function H_v^{-1} is approximated by the coefficient $1/k_{H_v}$.

From the polynomial $\tilde{\omega}_v$, approximating the velocity characteristic of the main propeller, it is found that for $u_v = 0.3$ one has $\omega_v = 1926.4$ r.p.m. The static characteristic $F_v(\omega_v)$ of the main thrust is approximated by the linear relationship $k_{H_v} \omega_v$. The coefficient k_{H_v} is determined differentiating the approximating polynomial \tilde{F}_v in ω_v and substituting the rotor angular velocity by the value $\omega_v = 1926.4$ in the result obtained.

The static characteristics of the tail rotor are linearized in a similar way determining the propeller operation point for $u_h = 0$.

The coefficients obtained from the linearization of the propeller static characteristics are shown in Table 18.2.

Consider next the linearization of plant differential equations. There are some trigonometric functions of the pitch angle in these equations that have to be approximated by linear relationships. We have

$$\sin(\alpha_{v,\text{nom}} + \delta\alpha_v) \approx \sin(\alpha_{v,\text{nom}}) + \cos(\alpha_{v,\text{nom}})\delta\alpha_v$$

and

$$\cos(\alpha_{v,\text{nom}} + \delta\alpha_v) \approx \cos(\alpha_{v,\text{nom}}) - \sin(\alpha_{v,\text{nom}})\delta\alpha_v$$

Replacing these approximations in the expression for the torque M_{v1} , participating in (18.1), one obtains

$$M_{v1} = M_{v1,\text{nom}} + R_v \delta\alpha_v$$

where $M_{v1,\text{nom}} = -k_1 \cos(\alpha_{v,\text{nom}}) - k_2 \sin(\alpha_{v,\text{nom}})$, $R_v = k_1 \sin(\alpha_{v,\text{nom}}) - k_2 \cos(\alpha_{v,\text{nom}})$.

The torque M_{v2} is represented in the form

$$M_{v2} = M_{v2,\text{nom}} + l_m k_{F_v} \delta\omega_v$$

where $M_{v2,\text{nom}} = l_m k_{F_v} (\omega_{v,\text{nom}})$.

The torque M_{v3} is represented as

$$\begin{aligned} M_{v3} = & M_{v3,\text{nom}} - 2k_3 \Omega_{h,\text{nom}} \sin(\alpha_{v,\text{nom}}) \cos(\alpha_{v,\text{nom}}) \delta\Omega_h \\ & - k_3 \Omega_{h,\text{nom}}^2 (\cos^2(\alpha_{v,\text{nom}}) - \sin^2(\alpha_{v,\text{nom}})) \delta\alpha_v \end{aligned}$$

where $M_{v3,\text{nom}} = k_3 \Omega_{h,\text{nom}} \sin(\alpha_{v,\text{nom}}) \cos(\alpha_{v,\text{nom}})$. Taking into account that $\Omega_{h,\text{nom}} = 0$, further on the dependence of the torque M_{v3} on $\delta\Omega_h$ and on $\delta\alpha_v$ is neglected.

The torques M_{v4} and M_{v5} are given in the form of linear relationships and their linearization is obvious.

Consider now the linearization of (18.2). The torque M_{h1} is represented as

$$M_{h1} = M_{h1,\text{nom}} + l_t k_{F_h} \cos(\alpha_{v,\text{nom}}) \delta\omega_h - l_t k_{F_h} \sin(\alpha_{v,\text{nom}}) \delta\alpha_v$$

where $M_{h1,\text{nom}} = l_t k_{F_h} \omega_{h,\text{nom}} \cos(\alpha_{v,\text{nom}})$.

The torque M_{h3} is represented as

$$M_{h3} = M_{h3,\text{nom}} - k_{vh} \sin(\alpha_{v,\text{nom}}) u_{v,\text{nom}} \delta\alpha_v + k_{vh} \cos(\alpha_{v,\text{nom}}) \delta u_v$$

where $M_{h3,\text{nom}} = k_{vh} \cos(\alpha_{v,\text{nom}}) u_{v,\text{nom}}$.

Equations (18.3) and (18.4) are linearized, respectively, as

$$\begin{aligned} I_v \frac{d\delta\omega_v}{dt} &= \delta u_v - (1/k_{H_v}) \delta\omega_v \\ I_h \frac{d\delta\omega_h}{dt} &= \delta u_h - (1/k_{H_h}) \delta\omega_h \end{aligned}$$

The obtained linearized model of the Twin-Rotor Aerodynamic System is shown in Fig. 18.6. (Here and next the variable deviations are denoted by the same names as the corresponding variables.) The plant has two channels and there is an interaction between these channels. In order to reveal in full the plant behavior the system should be considered as multivariable, i.e., the two channels cannot be considered independently.

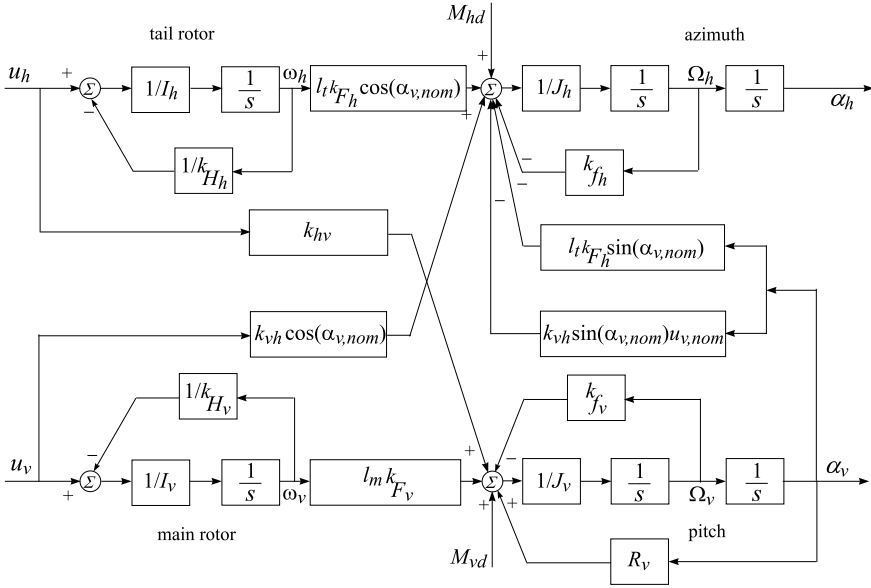


Fig. 18.6 Block-diagram of the linearized model

18.4 Uncertainty Modeling

For uncertain parameters in the mathematical description of the aerodynamical system we consider the moment of inertia J_h with respect to the vertical axis, thrust coefficients k_{F_h}, k_{F_v} of both rotors, velocity gains k_{H_h}, k_{H_v} of the two rotors, friction coefficients k_{f_h}, k_{f_v} , cross moment coefficients k_{v_h}, k_{h_v} , as well as the coefficient R_v of the return torque, altogether 10 parameters. The uncertainties in the moment of inertia J_h and in the coefficient R_v are due to their dependence on the pitch angle α_v , the uncertainties in the coefficients $k_{F_h}, k_{F_v}, k_{H_h}$, and k_{H_v} are introduced as a result of the measuring and approximation of the static characteristics of the rotors, the uncertainties in the coefficients k_{f_h} and k_{f_v} are due to the errors in determination of the friction moments, and the uncertainties in the coefficients k_{v_h} and k_{h_v} result from simplification of the aerodynamic interaction between the two channels. Furthermore we assume that the moment of inertia J_h and the coefficients $k_{F_h}, k_{F_v}, k_{H_h}$, and k_{H_v} are known with errors up to 10 % while the rest coefficients come with errors up to 5 %.

The ten real uncertain parameters $J_h, k_{H_h}, k_{H_v}, k_{F_h}, k_{F_v}, k_{f_h}, k_{f_v}, k_{h_v}, k_{v_h}$, and R_v are set by using the following command lines:

```
J_h_nom = 0.023790*(cos(alpha_v))^2 + 0.0030096;
J_h      = ureal('J_h', J_h_nom, 'Percentage', 10);
k_H_h    = ureal('k_H_h', 9.8389*10^3, 'Percentage', 10);
k_F_h    = ureal('k_F_h', 2.1293*10^(-5), 'Percentage', 10);
k_f_h    = ureal('k_f_h', 0.00589, 'Percentage', 5);
k_v_h    = ureal('k_v_h', -0.01782, 'Percentage', 5);
```

Fig. 18.7 Uncertain model of the twin-rotor aerodynamic system

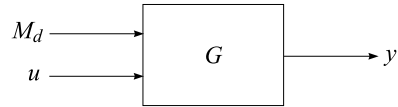
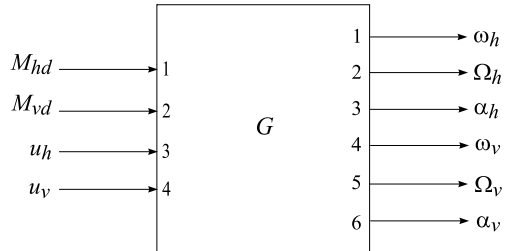


Fig. 18.8 Schematic diagram of the model input–output connection



```

k_H_v = ureal('k_H_v', 4.8746*10^3, 'Percentage', 10);
k_F_v = ureal('k_F_v', 3.07726*10^(-4), 'Percentage', 10);
k_f_v = ureal('k_f_v', 0.01271, 'Percentage', 5);
k_h_v = ureal('k_h_v', 0.004175, 'Percentage', 5);
R_v_nom = 0.050058*sin(alpha_v) - 0.093601*cos(alpha_v);
R_v = ureal('R_v', R_v_nom, 'Percentage', 5);

```

The uncertain TRAS model is obtained on the basis of the block-diagram, shown in Fig. 18.6, implementing the function `sysic`. This system model is created by the M-file `mod_tras.m`.

The uncertain TRAS system is described as a control plant by the equation

$$y = G \begin{bmatrix} M_d \\ u \end{bmatrix}$$

where

$$y = \begin{bmatrix} \alpha_h \\ \alpha_v \end{bmatrix}, \quad u = \begin{bmatrix} u_h \\ u_v \end{bmatrix}, \quad M_d = \begin{bmatrix} M_{hd} \\ M_{vd} \end{bmatrix}$$

The uncertain model of the Twin-Rotor Aerodynamic System is shown in Fig. 18.7.

The schematic diagram of the model input–output connection is shown in Fig. 18.8.

Let us introduce the representation

$$G = [G_d \quad G_u]$$

such that

$$y = G_d M_d + G_u u$$

In the last expression G_d is the plant transfer function matrix with respect to disturbances and G_u is the transfer function matrix with respect to the control signals.

The frequency response plot of the uncertain plant singular values, obtained from the transfer function G_u , is shown in Fig. 18.9.

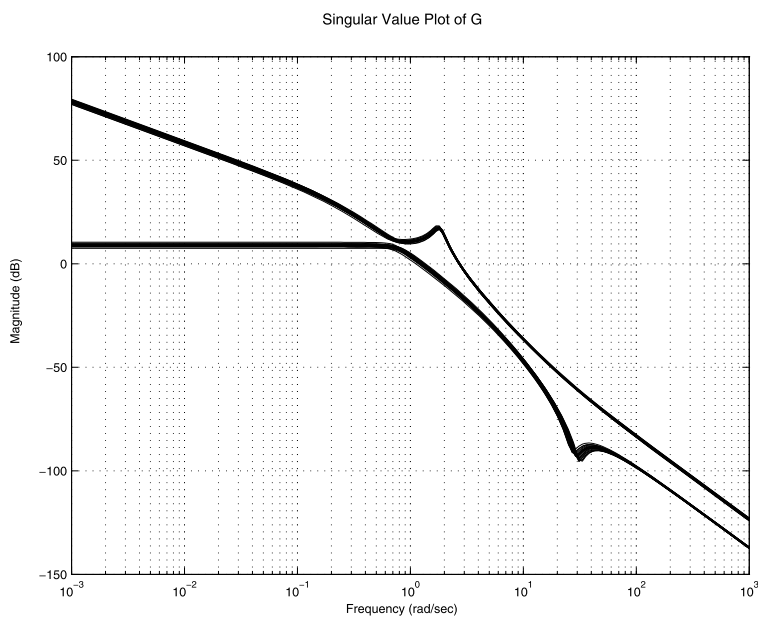


Fig. 18.9 Frequency responses of the uncertain plant

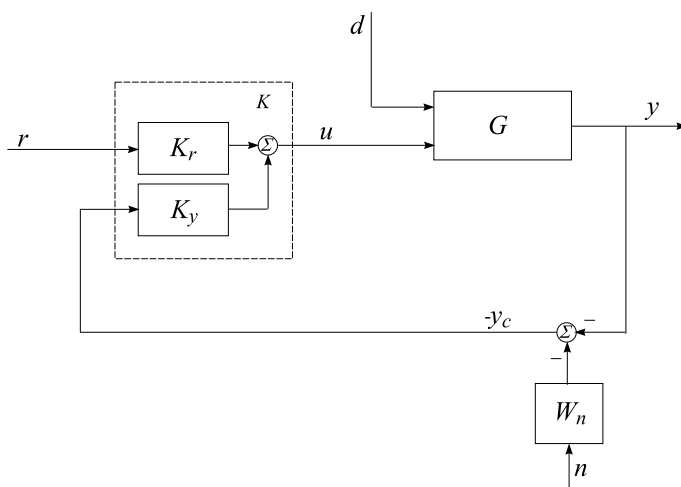


Fig. 18.10 Closed-loop system with 2-degree-of-freedom controller

18.5 Closed-Loop System Performance Requirements

The block-diagram of the closed-loop system with 2-degree-of-freedom controller is shown in Fig. 18.10.

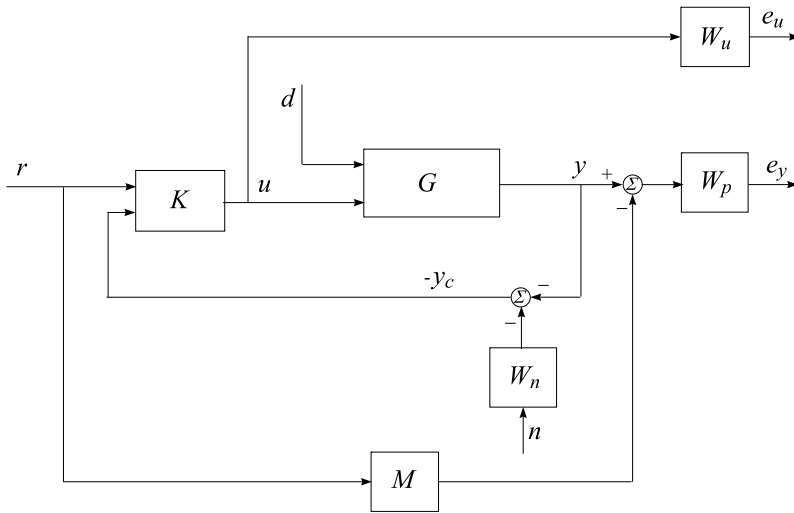


Fig. 18.11 Block-diagram of the closed-loop system with performance requirements

To obtain good performance of the system response we shall implement a 2-degree-of-freedom controller. The control actions are generated according to the expression

$$u = [K_r \quad K_y] \begin{bmatrix} r \\ -y_c \end{bmatrix} = K_r r - K_y y_c$$

where K_y is the output feedback transfer function matrix and K_r is the pre-filter transfer function matrix.

The block-diagram of the closed-loop system that includes the uncertain TRAS model, the feedback and the controller, as well as the elements reflecting the performance requirements, is shown in Fig. 18.11. The system has reference inputs (r), input disturbances (d) and noise (n) introduced in measurement of the angles α_h and α_v . (Here and further on the disturbance vector is denoted for brevity by d .) The TRAS uncertain model is the state space object G .

The system has two output signals (e_y and e_u). The block M is the ideal dynamics model that the designed closed-loop system should match to. The feedback of the system is realized by the vector $y_c = y + W_n n$, where the measurement noise n is a random vector with unit 2-norm and W_n is the transfer matrix of the noise shaping filters.

The weighted closed-loop system outputs e_y and e_u satisfy the equation

$$\begin{bmatrix} e_y \\ e_u \end{bmatrix} = \begin{bmatrix} W_p(S_o G_u K_r - M) & W_p S_o G_d & -W_p S_o G_u K_y W_n \\ W_u S_i K_r & -W_u S_i K_y G_d & -W_u S_i K_y W_n \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix} \quad (18.7)$$

where the matrix $S_i = (I + K_y G_u)^{-1}$ is the input sensitivity transfer function matrix and $S_o = (I + G_u K_y)^{-1}$ is the output sensitivity transfer function matrix.

Table 18.3 \mathcal{H}_∞ functions to be minimized

Function	Description
$W_p(S_o G_u K_r - M)$	Weighted difference between ideal and real closed-loop system
$W_p S_o G_d$	Weighted sensitivity to disturbance
$W_p S_o G_u K_y W_n$	Weighted sensitivity to noise
$W_u S_i K_r$	Weighted control action due to reference
$W_u S_i K_y G_d$	Weighted control action due to disturbance
$W_u S_i K_y W_n$	Weighted control action due to noise

The performance criterion requires the transfer function matrix from the exogenous input signals r , d , and n to the output signals e_y and e_u to be small in the sense of $\|\cdot\|_\infty$, for all possible uncertain plant models G . The transfer function matrices W_p and W_u are used to reflect the relative importance of the different frequency ranges for which the performance requirements should be fulfilled. The six transfer function matrices which constitute the transfer function matrix between the inputs and outputs of the extended system are described in Table 18.3.

The design problem for the Twin-Rotor Aerodynamic System is to find a linear controller $K(s)$ in the reference and measurable output

$$K = [K_r \quad K_y]$$

that has to ensure the following properties of the closed-loop system:

18.5.1 Robust Stability

The closed-loop system achieves robust stability if this system is internally stable for all possible plant models G .

18.5.2 Nominal Performance

The closed-loop system achieves nominal performance if the performance criterion is satisfied for the nominal plant model

$$\left\| \begin{bmatrix} W_p(S_{o,\text{nom}} G_{u,\text{nom}} K_r - M) & W_p S_{o,\text{nom}} G_{d,\text{nom}} & -W_p S_{o,\text{nom}} G_{u,\text{nom}} K_y W_n \\ W_u S_{i,\text{nom}} K_r & -W_u S_{i,\text{nom}} K_y G_{d,\text{nom}} & -W_u S_{i,\text{nom}} K_y W_n \end{bmatrix} \right\|_\infty < 1$$

18.5.3 Robust Performance

The closed-loop system should remain internally stable for all G and in addition the performance criterion

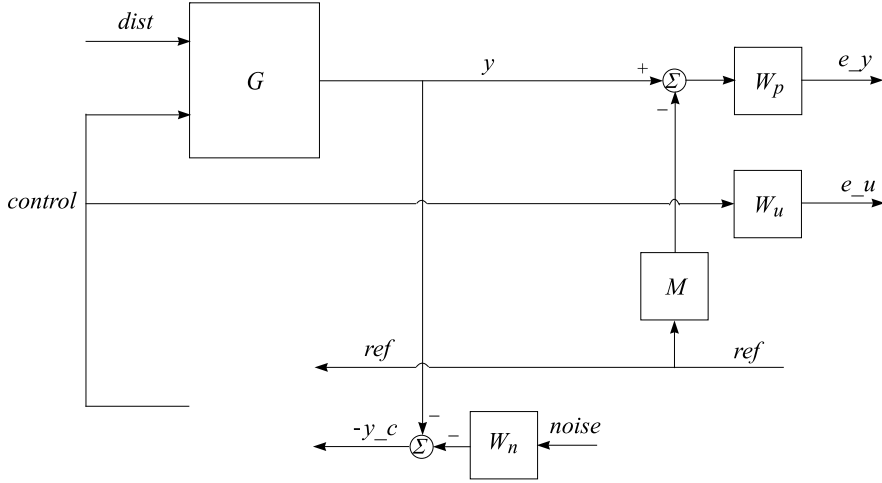


Fig. 18.12 Open-loop system structure

$$\left\| \begin{bmatrix} W_p(S_o G_u K_r - M) & W_p S_o G_d & -W_p S_o G_u K_y W_n \\ W_u S_i K_r & -W_u S_i K_y G_d & -W_u S_i K_y W_n \end{bmatrix} \right\|_{\infty} < 1 \quad (18.8)$$

should be satisfied for each G .

18.6 System Interconnections

The internal structure of the eight-input, eight-output open-loop system of 16th order, which is saved as the variable `sys_ic`, is shown in Fig. 18.12. The reference, disturbance, and noise are saved as the variables `ref`, `dist` and `noise`, respectively. The control action is saved as the variable `control`. All variables shown in the figure have two elements. The open-loop system interconnection is set by the M-file `dlp_tras` implementing the function `sysic`.

The schematic diagram showing the specific input/output arrangement of the variable `sys_ic`, is shown in Fig. 18.13.

The block-diagram used in the closed-loop system simulation is shown in Fig. 18.14. The corresponding closed-loop interconnection that is saved in the variable `sim_ic` is obtained by the M-file `sim_tras`.

Figure 18.15 shows the schematic diagram of the input/output arrangement of the variable `sim_ic`.

18.7 μ -Synthesis

It is possible to implement different methods in the TRAS controller design including several versions of \mathcal{H}_{∞} -optimization and μ -synthesis. According to the exper-

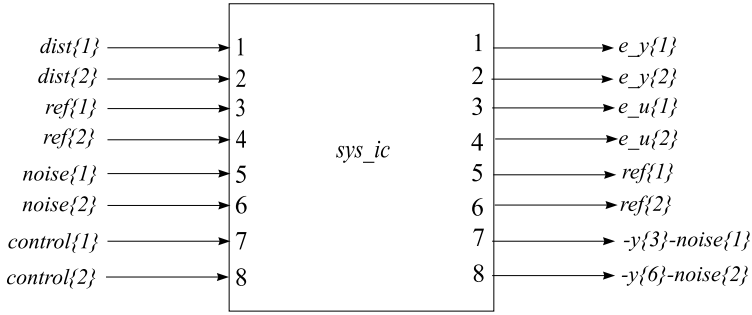


Fig. 18.13 Schematic input–output diagram of the open-loop system structure

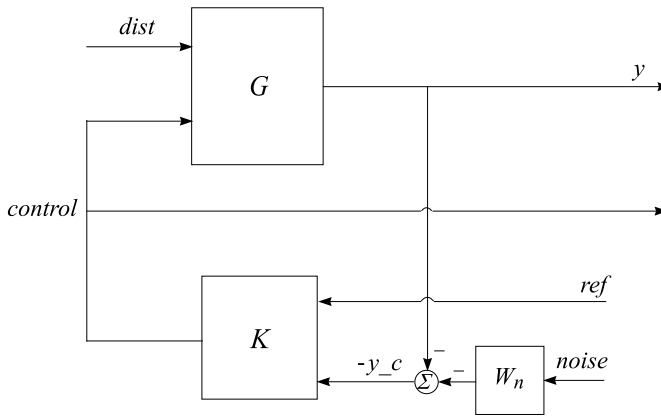


Fig. 18.14 Closed-loop system structure

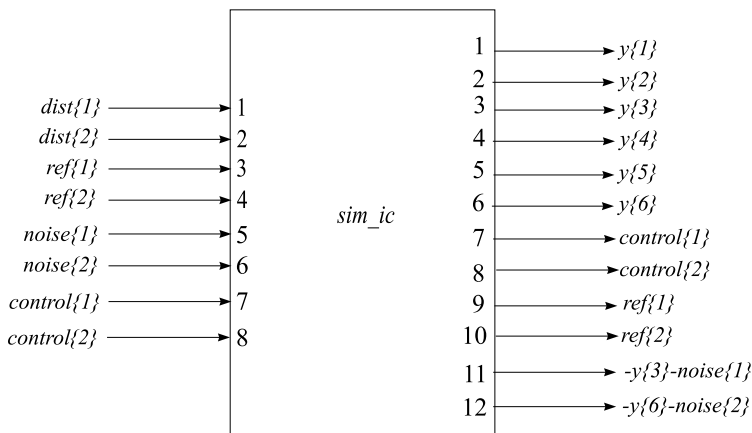
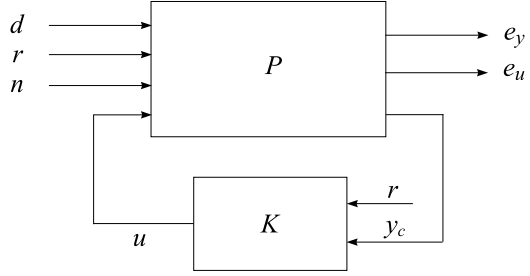


Fig. 18.15 Schematic input–output diagram of the closed-loop system structure

Fig. 18.16 Block-diagram of μ -synthesis

iments performed best results are obtained by using μ -synthesis of a 2-degree-of-freedom controller. For this reason we represent in this section only the design of such a type of controller.

The closed-loop system block-diagram corresponding to the μ -synthesis problem is shown in Fig. 18.16. The matrix P is the transfer function matrix of the extended open-loop system shown in Fig. 18.12.

The control actions to the plant are realized by a computer in real time with sampling frequency $f_s = 100$ Hz. For this reason the μ -synthesis is implemented to design a discrete-time controller at this sampling frequency.

Let us denote by $P_d(z) = F_U(N_d, \Delta)$ the transfer function matrix of the discretized eight-input eight-output open-loop system `sys_ic`, which consists of the uncertain plant model plus the weighting functions, and let the block-structure Δ_{P_d} is defined as

$$\Delta_{P_d} := \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_F \end{bmatrix} : \Delta \in \mathcal{R}^{10 \times 10}, \Delta_F \in \mathcal{C}^{6 \times 4} \right\}$$

The first block of the matrix Δ_{P_d} , the block Δ , corresponds to the parametric uncertainties, included in the model of the aerodynamic system. The second block Δ_F is a fictitious uncertainty block, used to include the performance requirements into the framework of the μ -approach. The inputs of this block are the weighted error signals e_y and e_u , and the outputs are the exogenous signals r , d , and n .

The aim of the μ -synthesis is to find a discrete stabilizing controller K_d , such that for each frequency $\omega \in [0, \pi/T_s]$, where $T_s = 2\pi/f_s$, the structured singular value μ satisfies the condition

$$\mu_{\Delta_{P_d}}[F_L(N_d, K_d)(j\omega)] < 1$$

where $F_L(N_d, K_d)$ is the closed-loop transfer function matrix. The fulfillment of this condition guarantees the robust performance of the closed-loop system, i.e.

$$\|F_U[F_L(N_d, K_d), \Delta_{P_d}]\|_{\infty} < 1$$

for all uncertainties Δ_{P_d} with $\|\Delta_{P_d}\|_{\infty} < 1$.

The transfer function matrix M of the ideal matching model is chosen as diagonal in order to suppress the interaction between the two channels and is taken as

$$M(s) = \begin{bmatrix} w_{m1} & 0 \\ 0 & w_{m2} \end{bmatrix}$$

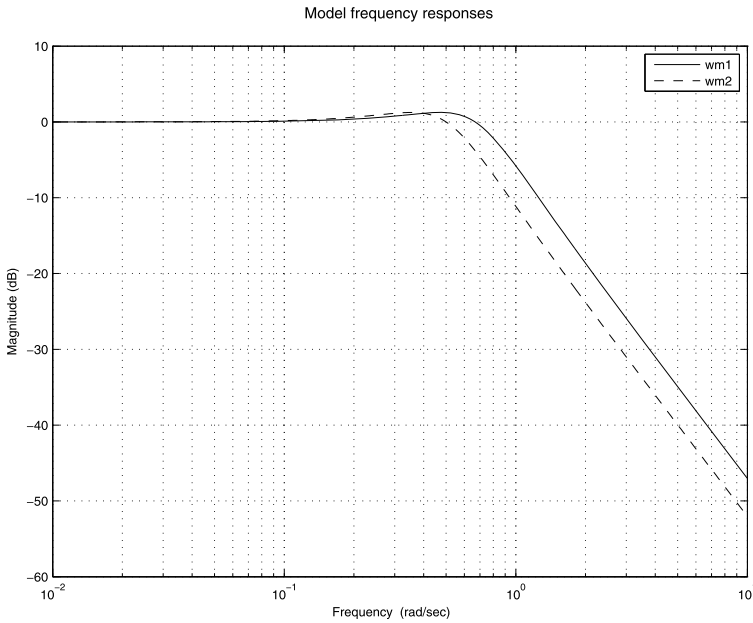


Fig. 18.17 Model frequency responses

where

$$w_{m1} = \frac{1}{1.5s^2 + 1.2s + 1}$$

$$w_{m2} = \frac{1}{2.0s^2 + 1.6s + 1}$$

In the choice of this model it was assumed that the azimuth dynamics is faster than the pitch dynamics.

The model magnitude responses are shown in Fig. 18.17.

The μ -synthesis is done for several performance weighting functions that ensure a good balance between system performance and robustness. On the basis of the experimental results, we choose the performance weighting function

$$W_p(s) = \begin{bmatrix} 8.7 \times 10^{-2} \frac{80s+1}{80s+10^{-3}} & -0.01 \\ 0.03 & 7.0 \times 10^{-1} \frac{500s+1}{500s+10^{-3}} \end{bmatrix}$$

and the control weighting function

$$W_u(s) = \begin{bmatrix} 4.0 \times 10^{-5} \frac{0.05s+1}{10^{-4}s+1} & 0 \\ 0 & 2.304 \times 10^{-4} \frac{0.1s+1}{10^{-4}s+1} \end{bmatrix}$$

The control weighting functions are chosen so that the azimuth control action to be in the range $[-0.8, 0.8]$ and the pitch control action in the range $[-0.5, 1]$. The weighting functions are set in the M-file `wts_tras_mu.m`.

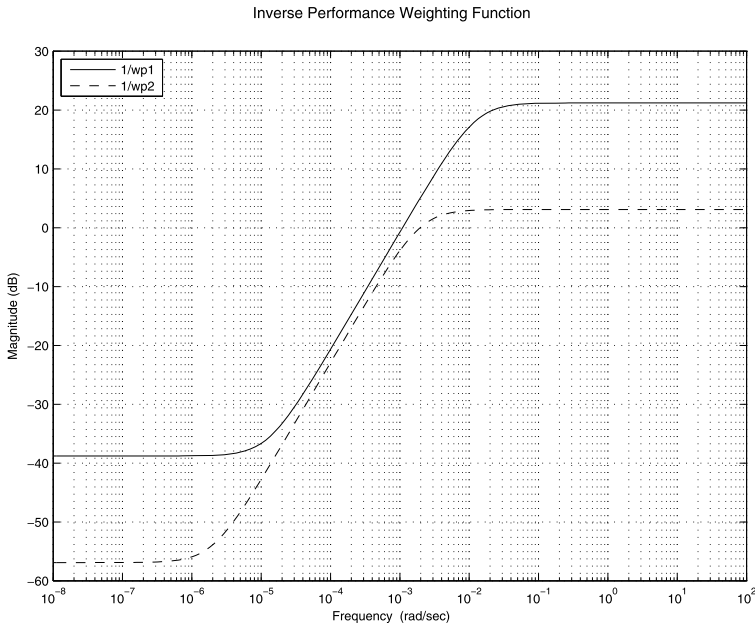


Fig. 18.18 Inverse performance weighting functions

The frequency responses of the inverse performance weighting function are given in Fig. 18.18.

The frequency responses of the inverse control weighting function are given in Fig. 18.19. The control weighting functions are chosen as high pass filters with appropriate bandwidth in order to impose constraints on the spectrum of the control actions.

The experiments with the control laws designed shows that the behavior of the real closed-loop system is very sensitive to the weighting functions used. That is why the precise tuning of the weighting functions requires a large volume of experiments.

The noise transfer function matrix is taken as

$$W_n(s) = \begin{bmatrix} w_n(s) & 0 \\ 0 & w_n(s) \end{bmatrix}$$

where the transfer function $w_n = 10^{-2} \frac{s}{s+1}$ is a high pass filter whose output is significant above 10 rad/s.

The magnitude frequency response obtained from the transfer function w_n is shown in Fig. 18.20. This weighting function corresponds to measurement error in the high frequency range about 1×10^{-2} rad.

The μ -synthesis is performed by using the M-file `dms_tras.m`, implementing the function `dksyn`.

The D - K iteration run is shown in Table 18.4.

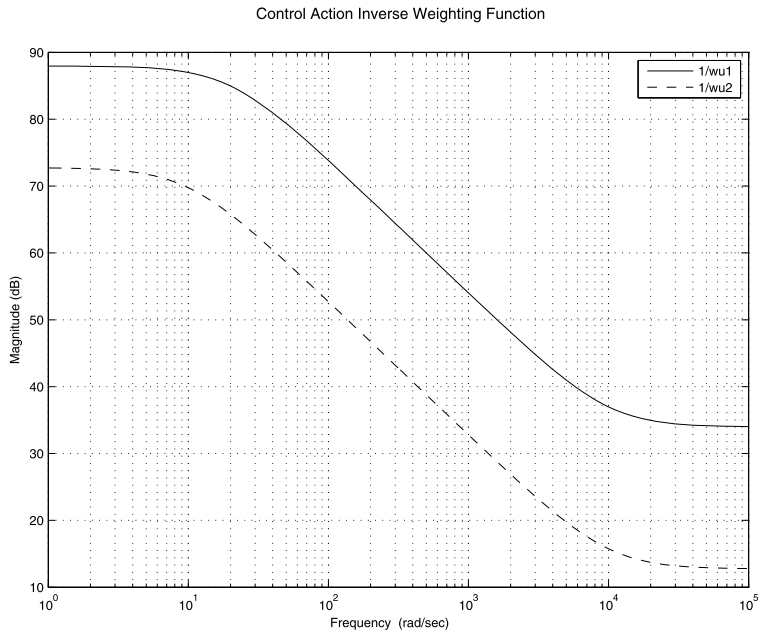


Fig. 18.19 Inverse control action weighting functions

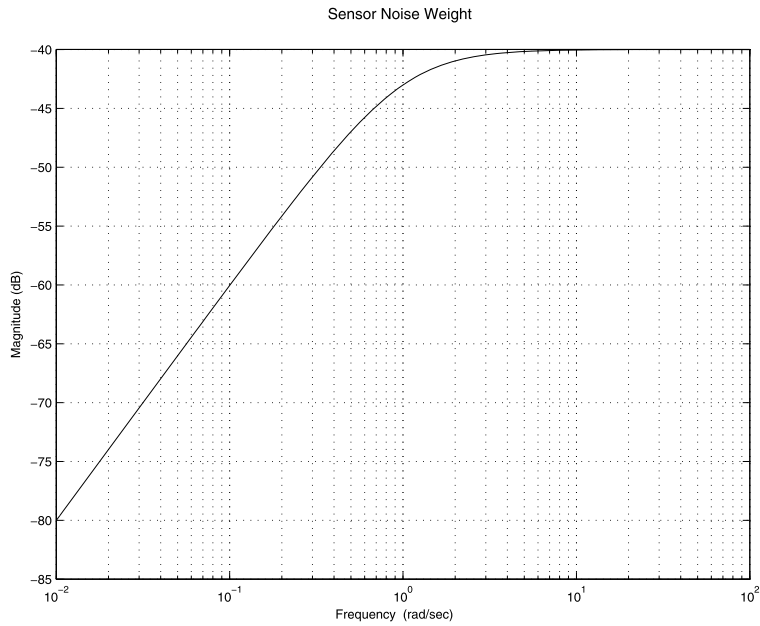


Fig. 18.20 Noise weighting function

Table 18.4 Results from μ -synthesis D – K iteration

Iteration	Controller order	μ
1	16	235.419
2	16	4.535
3	20	1.226
4	22	0.980
5	24	0.968

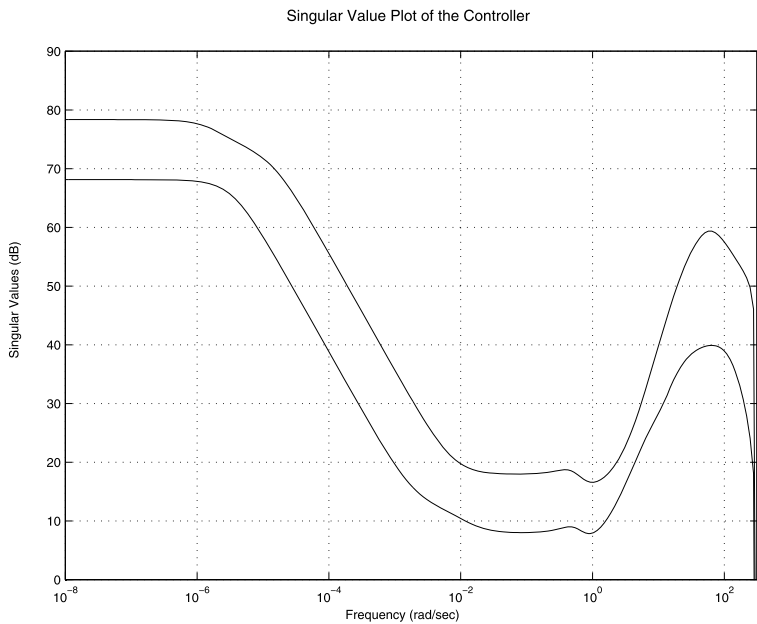


Fig. 18.21 Frequency responses of the μ -controller

Five iterations are performed that decrease the maximum value of μ to 0.971. The final controller obtained is of 24th order.

The singular value frequency response plot of the μ -controller singular values are shown in Fig. 18.21.

The robust stability and robust performance analysis of the closed-loop system with μ -controller, done by using the file `dmu_tras.m`, produces the following results.

report =

```
Uncertain System is robustly stable to modeled uncertainty.
-- It can tolerate up to 521% of the modeled uncertainty.
-- No modeled uncertainty exists to cause an instability
   at 0.001 rad/s.
```

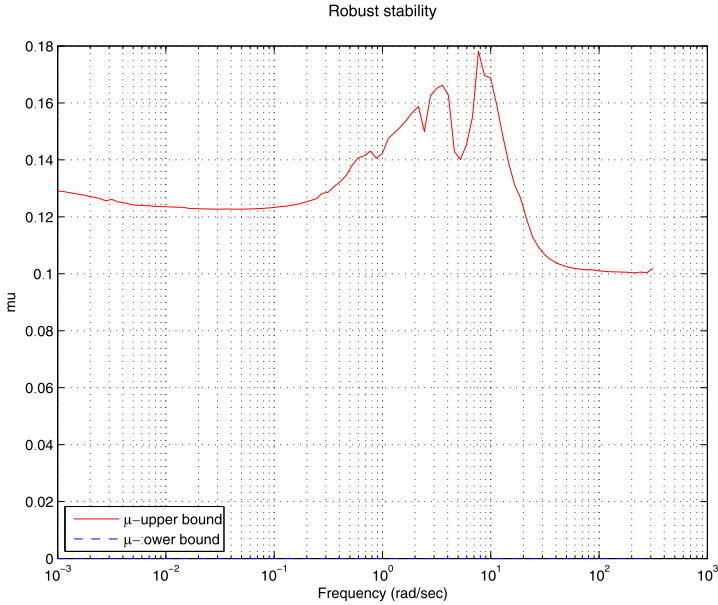



Fig. 18.22 Robust stability of the closed-loop system

report =

Uncertain System achieves a robust performance margin
of 1.041.

-- A model uncertainty exists of size 103% resulting in a
performance margin of 0.968 at 21.4 rad/s.

The frequency responses of the structured singular value that corresponds to robust stability analysis are shown in Fig. 18.22.

The frequency response of μ , corresponding to the robust performance analysis of the closed-loop sampled-data system, is shown in Fig. 18.23.

The frequency responses of the uncertain closed-loop system are obtained by the M-file `dfrs_tras.m`. The frequency responses of the singular values of the closed-loop transfer function matrix for random values of the plant uncertain parameters are shown in Fig. 18.24. It is seen that as a result of achieving robust performance, the closed-loop system frequency responses are close to these of the model (shown with dashed lines).

The frequency response plot of the singular values of disturbance transfer function matrix (output sensitivity function S_o) is shown in Fig. 18.25. (The inverse performance functions are shown with dashed lines.) The disturbance attenuation is more than 100 times (40 dB).

The frequency response plot of the singular values of the transfer function matrix from noises to outputs for the controller Kd_{mu} are shown in Fig. 18.26. The

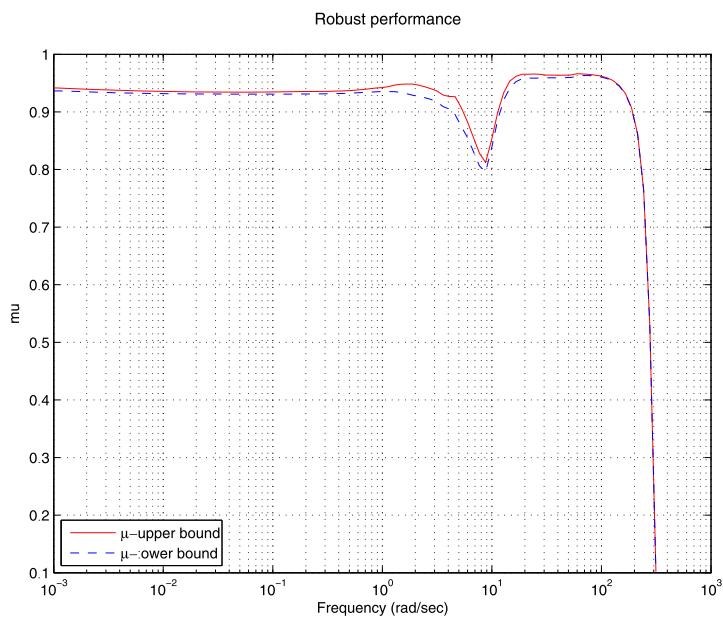


Fig. 18.23 Robust performance of the closed-loop system

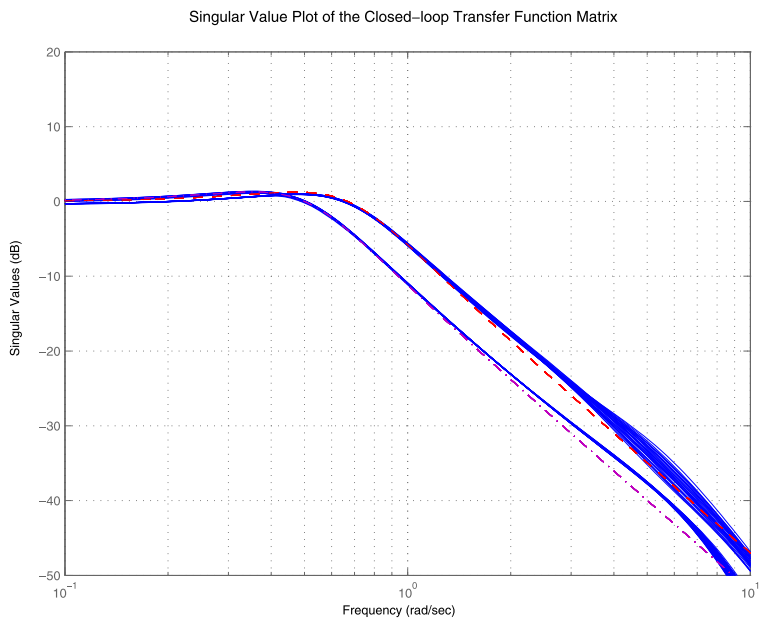


Fig. 18.24 Frequency responses of the closed-loop system (—) and model (---, -.)

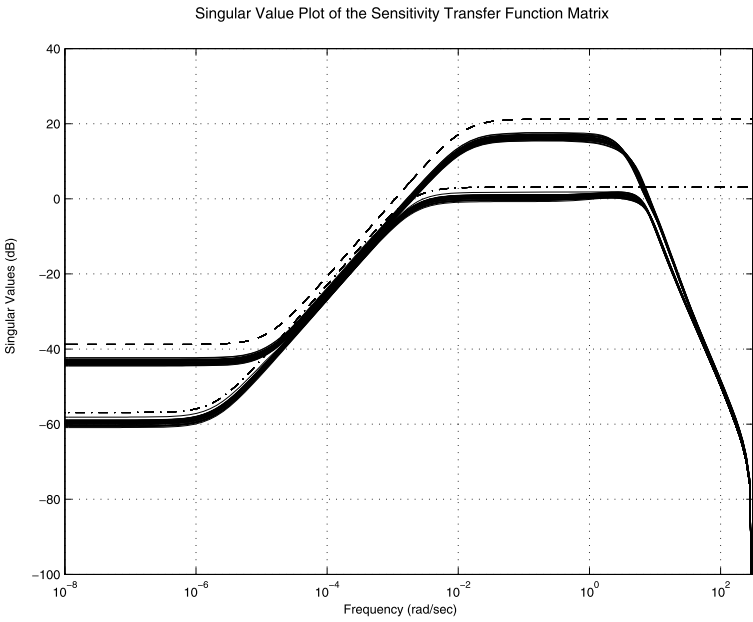


Fig. 18.25 Singular value plot of the sensitivity function (—) and the inverse performance weighting function (---, -.)

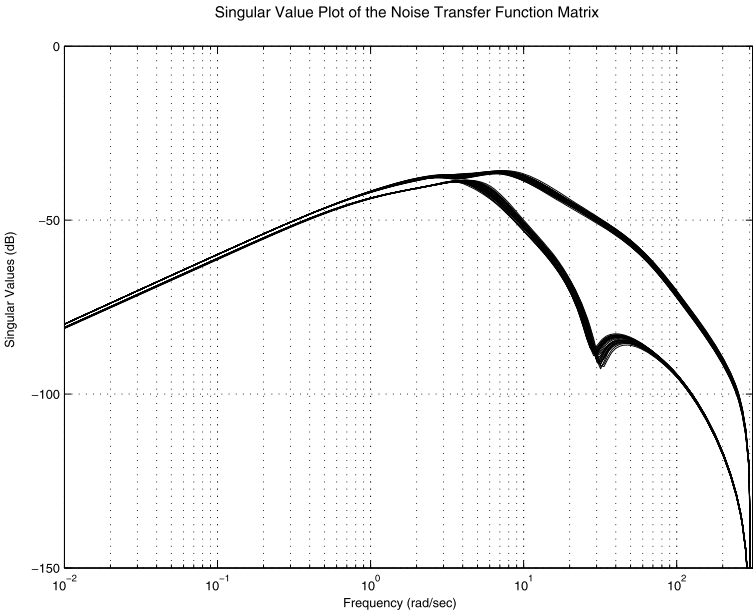


Fig. 18.26 Singular value plot of the noise transfer function

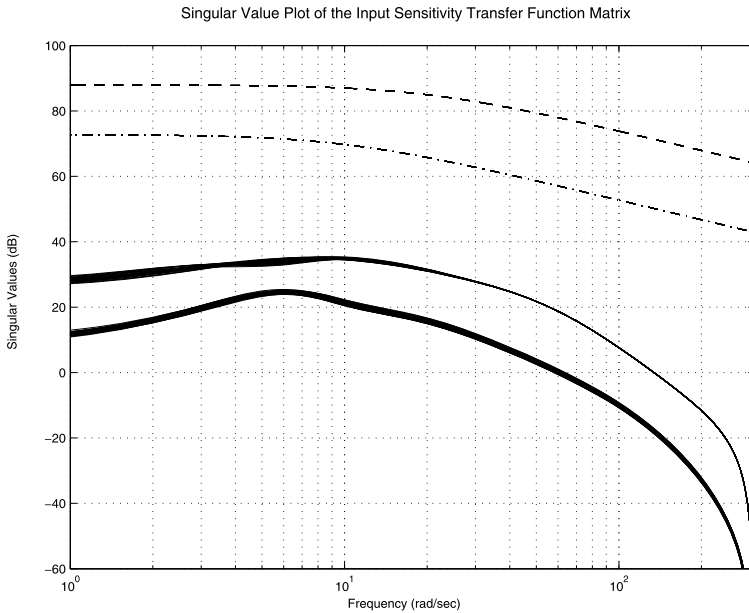


Fig. 18.27 Frequency responses of the input sensitivity function (—) and inverse control weighting function (---, -.-)

strongest influence have the noises with frequency between 5 rad/s and 10 rad/s, but their effect on the closed-loop system output is negligible.

In Fig. 18.27 we show the frequency responses of the input sensitivity. The maximum influence of the disturbances on the plant input is in the range around 10 rad/s.

The frequency response of the noise-to-control loop (Fig. 18.28) shows that one may expect high level of the noises at the actuator inputs. In Fig. 18.28 we show the frequency response plot of the singular values of the transfer function matrix from noise to control actions. The maximum of the larger singular value is equal to 20 dB, which shows that the magnitude of the noises at the plant input is 10 times larger than the magnitude of the noises at the sensor outputs. These large noises are due to the relatively high gains in the controller (up to 10^4).

The singular values of the open-loop system with μ -controller are shown in Fig. 18.29.

Consider now the transient responses of the closed-loop system. These responses are obtained for 30 different values of the uncertain parameters implementing the M-file `dsl_tras.m`.

The transient responses of the sampled-data closed-loop system, obtained for different random values of the uncertain parameters, are shown in Fig. 18.30. These processes have acceptable performance.

The control actions for different values of the uncertain parameters are shown in Fig. 18.31. The motor input signals are less than 1 for all possible values of the uncertain parameters.

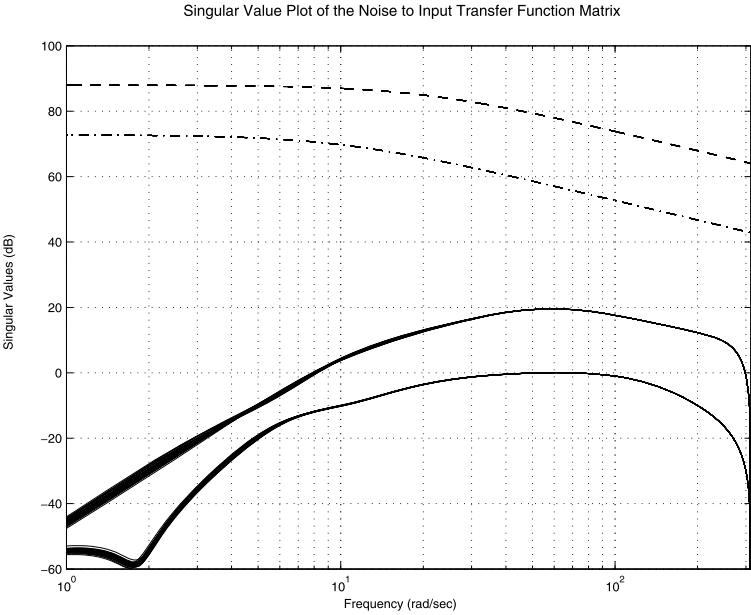


Fig. 18.28 Frequency response of the noise-to-control loop (–) and inverse control weighting function (–, –.)

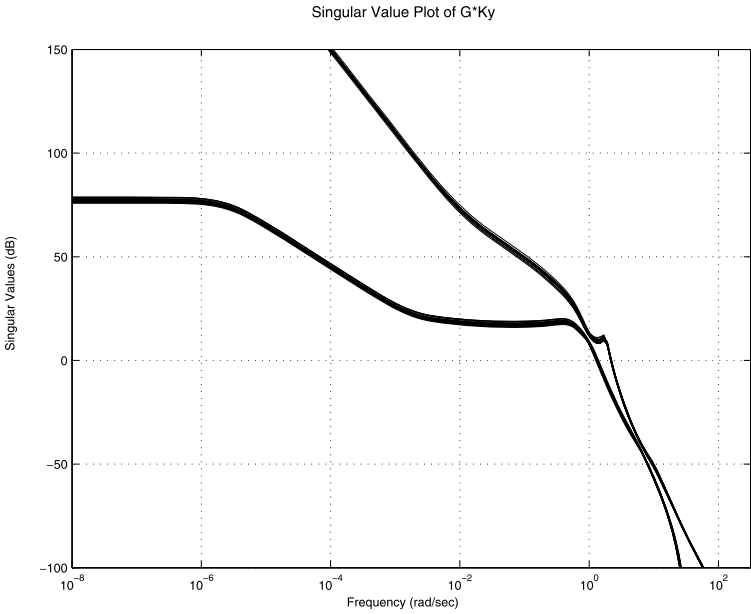


Fig. 18.29 Singular value plot of the open-loop system

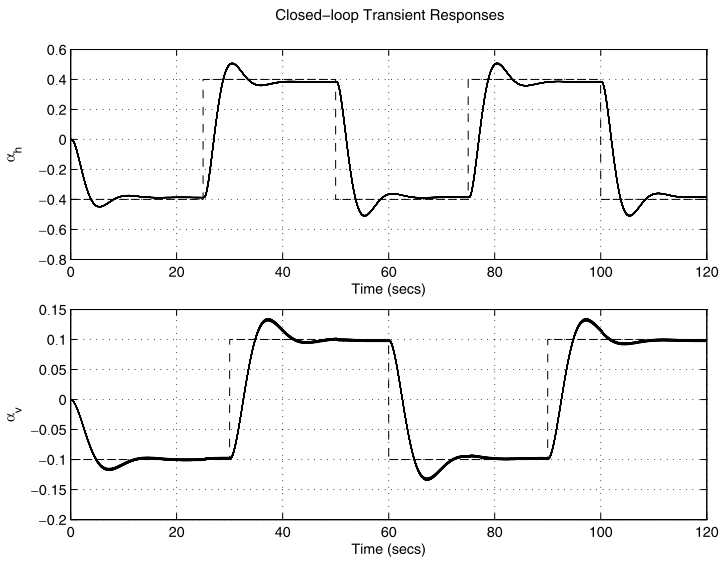


Fig. 18.30 Transient responses of the uncertain linearized closed-loop system

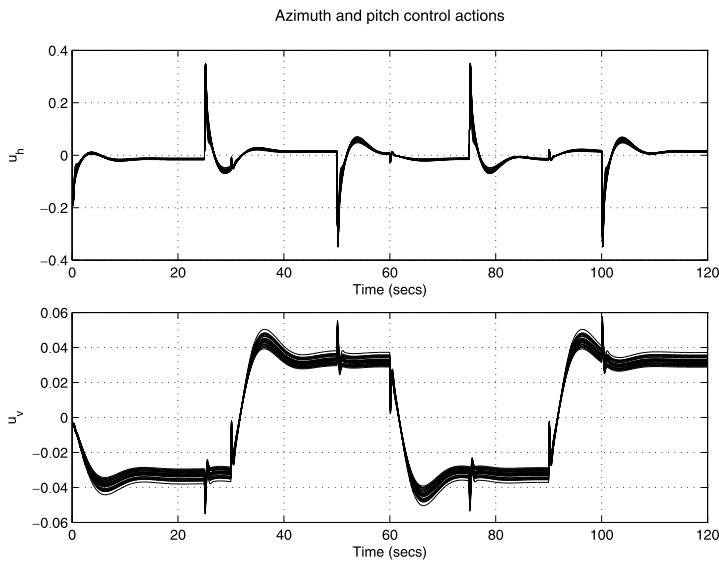


Fig. 18.31 Control actions of the uncertain linearized closed-loop system

18.8 Nonlinear System Simulation

The nonlinear sampled-data closed-loop control system is simulated using the Simulink® model `TRAS_2dof_model.mdl`. This model allows to simulate the

closed-loop system behavior for different references and disturbance setting noises equal to zero. The model utilizes the nonlinear differential equations (18.1)–(18.6) describing the Twin-Rotor Aerodynamic System.

The Simulink® model `TRAS_2dof_model.mdl` is shown in Fig. 18.32. The azimuth control actions are bounded in the range $[-0.8, 0.8]$ and the pitch control actions—in the range $[-0.5, 1]$.

The simulation of the nonlinear system for several references and disturbances produces results that are close to the results obtained for the nominal linearized model.

In Fig. 18.33 we show the transient responses to the reference of the nonlinear system with μ -controller.

In Fig. 18.34 we show the control actions of the nonlinear system with μ -controller.

18.9 Experimental Results

The experiments with the controller designed are done by using the laboratory set-up, shown in Fig. 18.1 along with a PC with MATLAB®, ver. R2007b. The generation of the C driving program is done by using the Simulink Coder®. For this aim a Simulink® model of the closed-loop system is used with built-in driver for interface with the plant.

The Simulink® model `TRAS_2dof_exper.mdl` of the twin-rotor aerodynamic system that is used to generate the control actions is shown in Fig. 18.35. As in the simulation of the nonlinear system the azimuth control actions are bounded in the interval $[-0.8, 0.8]$ and the pitch control actions in the interval $[-0.5, 1]$.

The experimentally obtained transient responses of the closed-loop system, controlled in real time with the μ -controller designed, are shown in Fig. 18.36. The comparison with the transient responses of the linearized closed-loop system given in Fig. 18.31 shows a good coincidence between the theoretical and experimental results.

The experimentally obtained control actions are shown in Fig. 18.37. Due to the high level of the noises at the actuator inputs, the control actions are severely contaminated by errors.

To extract the true actuator inputs, the control signals are filtrated by using appropriate first order Butterworth filter. The corresponding results are shown in Fig. 18.38 and are close to the theoretical results shown in Fig. 18.11.

18.10 Conclusions

The experience gained in the design and experimental implementation of the Twin-Rotor Aerodynamic System controller allows one to derive the following conclusions.

Simulink model of the Two Rotor Aerodynamical System with 2-DOF controller

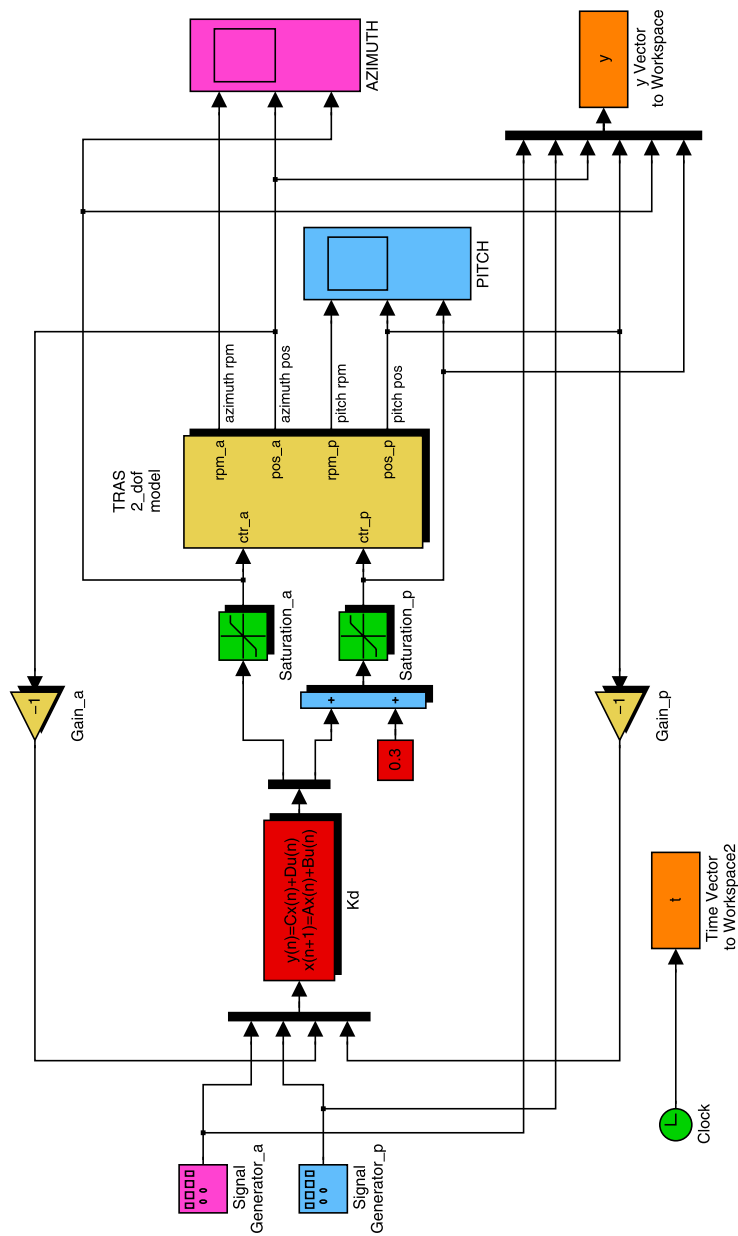


Fig. 18.32 Nonlinear system simulation model

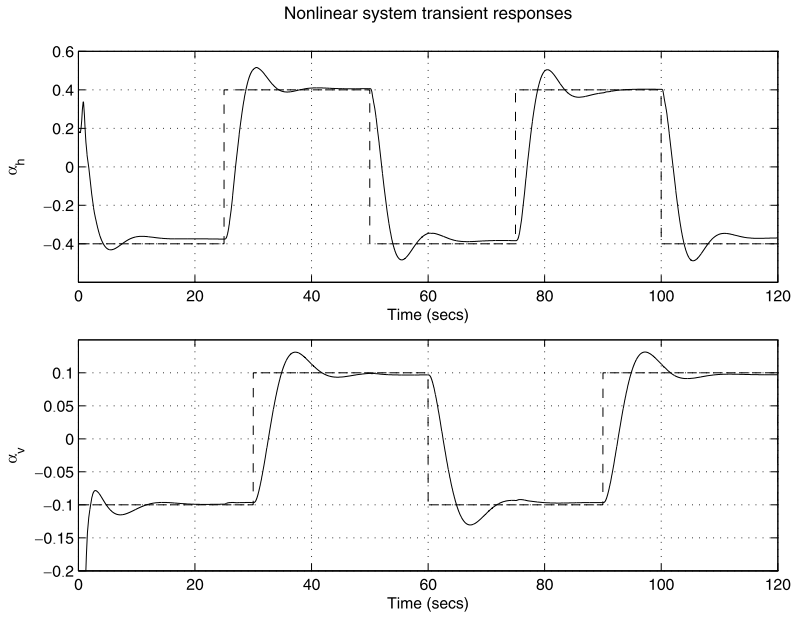


Fig. 18.33 Transient responses of the nonlinear system

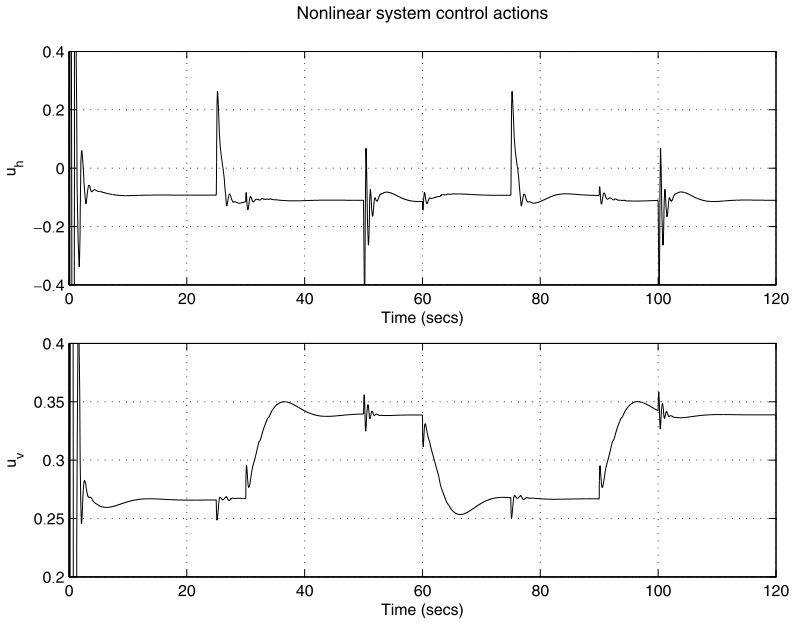


Fig. 18.34 Control actions in the nonlinear system

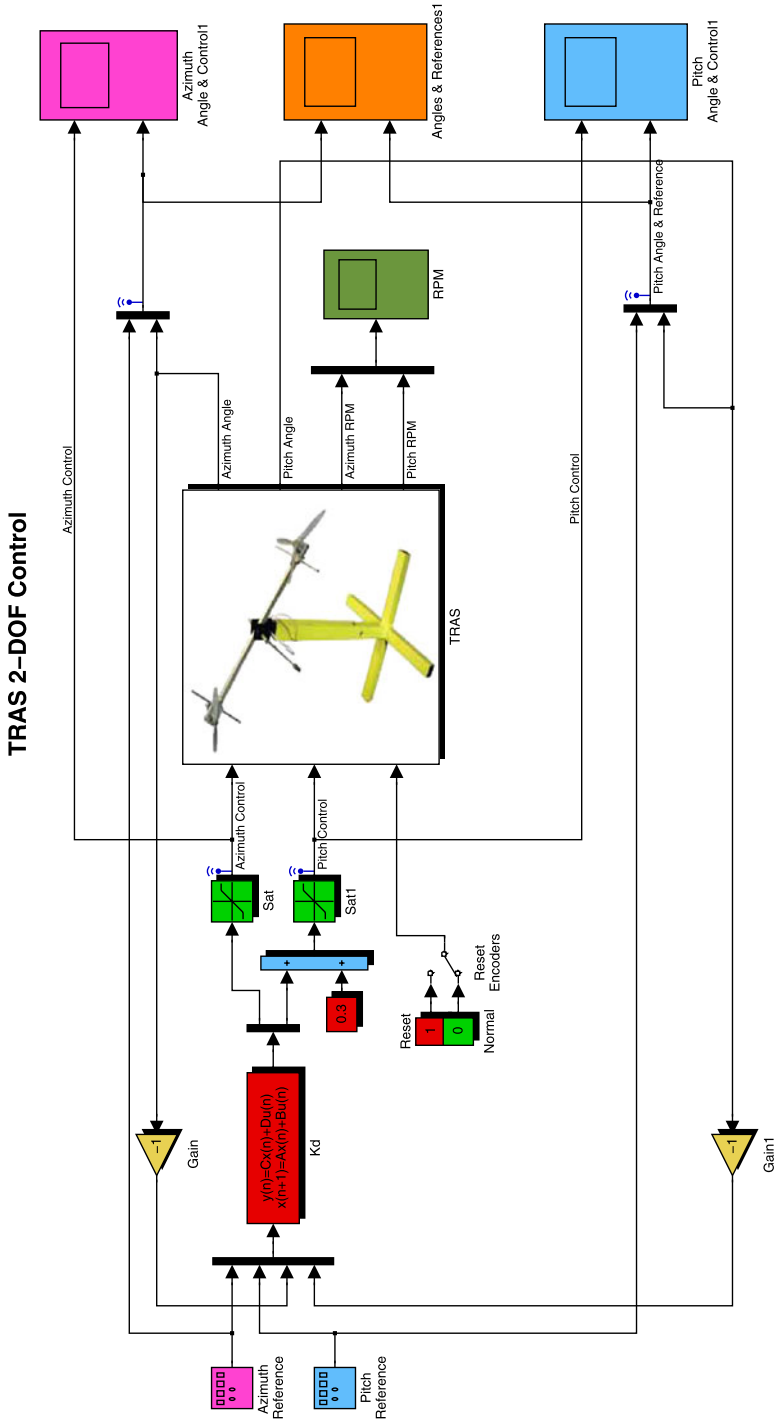


Fig. 18.35 Simulation model used for real time control

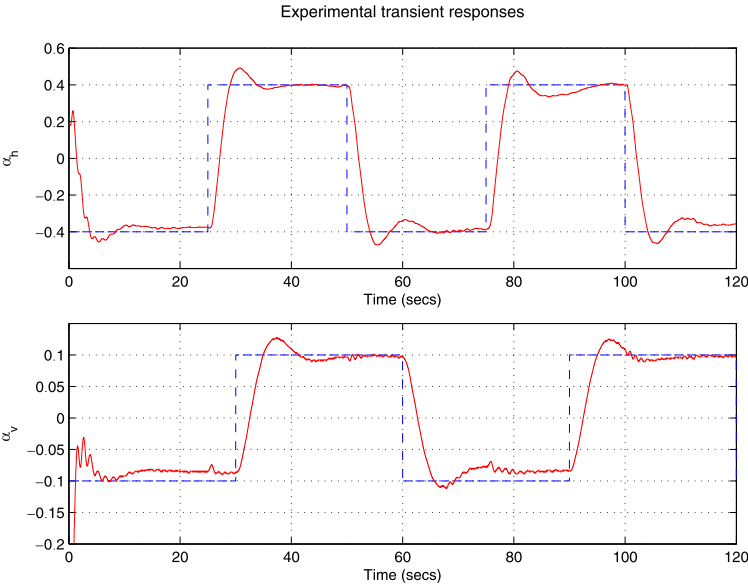


Fig. 18.36 Experimental transient responses

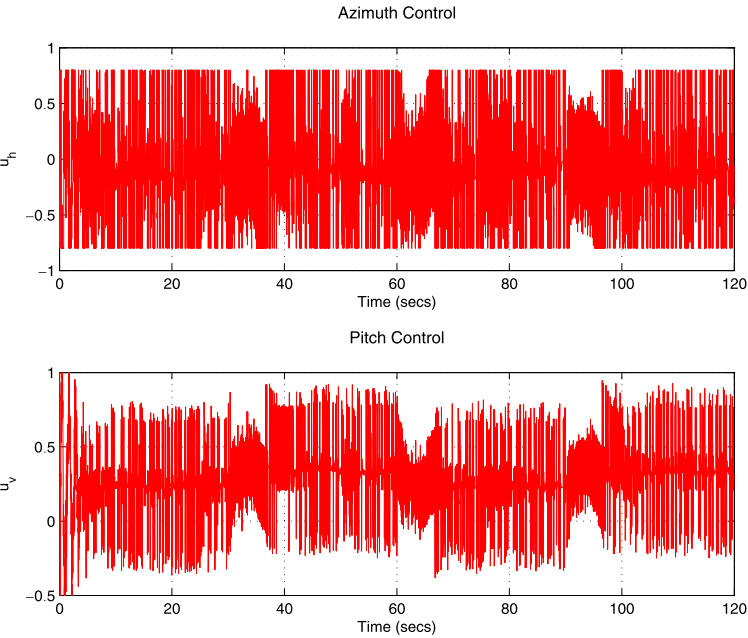


Fig. 18.37 Experimental control actions (noisy data)

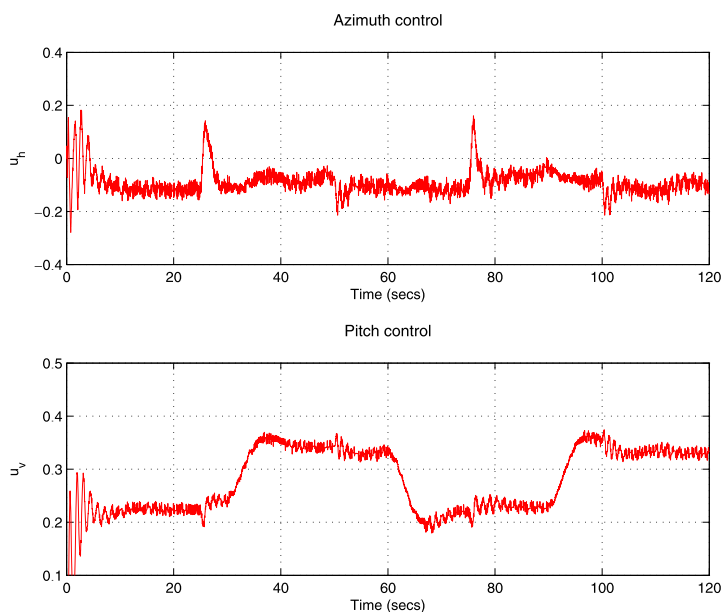


Fig. 18.38 Experimental control actions (after filtration)

- The modern technologies for real-time control allow implementation of high-order controllers (in the given case $n = 24$), which help to achieve better performance and robustness of the closed-loop system.
- Rapid tuning of the controller is possible as well as prototyping of a large number of controllers.
- The strong contamination of the input signals by noises leads to the necessity to suppress the high-frequency components of the control actions.
- Due to the high level of noises in the control inputs the actuators may saturate, which may lead to generation of auto-oscillations in the closed-loop system.

18.10.1 Notes and References

The derivation of the Twin-Rotor Aerodynamic System nonlinear model presented in this chapter follows [165].

The Twin-Rotor Aerodynamic System presents a simplified model of a helicopter in which the main rotor thrust and tail rotor thrust are controlled by changing the corresponding propeller speed. (In the real helicopter this is done by changing the collective and cyclic pitch angles of the propeller blades.) The behavior of this system illustrates well the difficulties arising in control of strongly coupled two-channel nonlinear systems. Good disturbance attenuation and channel decoupling is possible to achieve by using \mathcal{H}_∞ -design or μ -synthesis based on the linearized plant

model. A nonlinear \mathcal{H}_∞ disturbance rejection controller for the Twin-Rotor Aerodynamic System, showing a good performance for a range of operating conditions, is proposed in [104].

The control of unmanned rotorcraft vehicles is an area attracting intensive research in the recent years. The reader interested in this subject may consult the books [19, 105, 118, 166].

Chapter 19

Robust Control of Self-balancing Two-Wheeled Robot

This chapter presents the design and experimentation of a 2-degree-of-freedom robust controller for a self-balancing two-wheeled LEGO® Mindstorms NXT robot. A 12th order discrete-time controller is designed by using the techniques of μ -synthesis. The closed-loop control system achieves robust stability and robust performance in the presence of two uncertain friction coefficients. The experimental results show that the robot preserves stability in the vertical plane for deviations greater than 16° .

19.1 Introduction

In recent years, there has been a growing interest in research and education in implementation of miniature robots, build on the basis of LEGO® Mindstorms NXT developer kit (see for instance [10, 13, 144]). The control of such robots is done by 32-bit ATMEL ARM 7 (AT91SAM7S256) microcontroller with 48 MHz speed. This microcontroller works under the operational system nxtOSEK and has 64 KB RAM, which makes it suitable for implementation of sufficiently complex control laws.

The LEGO® Mindstorms NXT kit is used by Yoriyama Yamamoto to build the self-balancing two-wheeled robot NXTway-GS [188], which implements a linear quadratic regulator for robot digital control (stabilization of vertical body position and achieving a reference position in the horizontal plane). The software product Embedded Coder Robot NXT developed by Takashi Chikamasa [22] is used to implement additional tasks related to the robot control (system initialization, avoiding obstacles and battery voltage checking).

The general view of the NXTway-GS robot in self-balancing mode is shown in Fig. 19.1.

The robot balancing is achieved by rotating the wheels in the appropriate direction. The computation of control actions to both DC drive motors is realized in single precision on the basis of signals from the micro-electromechanical (MEM)

Fig. 19.1 NXTway-GS robot in self-balancing mode



gyroscopic sensor which measures the angular rate (and, after integration, the tilt angle) of the robot body in the vertical plane and signals from rotary encoders which measure the wheels rotation angles. The control of the DC motors is executed by Pulse Width Modulated (PWM) signals. To avoid obstacles the robot is equipped with an ultrasonic sensor.

In this chapter we present the design of a robust controller for NXTway-GS robot with the aim to implement in maximum degree the available software for real-time control presented in [188]. Since the robust controllers are of higher order the basic problem is to check the possibility to implement such controllers on the available microcontroller working with sampling frequency $f_s = 250$ Hz in the stabilization loop. The results obtained show that the microcontroller under consideration implements without difficulties a robust discrete controller of 12th order that allows to improve the closed-loop system performance. Results from the simulation of the closed-loop system as well as experimental results obtained during the real implementation of the controller designed are given.

19.2 Uncertain Model of the Two-Wheeled Robot

The nonlinear differential equations describing the robot motion are derived in detail in [188] by using the Lagrange method. These equations are linearized analytically around the balance point (equilibrium position) resulting in a state space description of fourth order for the vertical plane motion and a second order description for the

rotation around vertical axis. In the first case, the average angle θ of left and right wheel rotations and the body tilt angle ψ (measured from the vertical position), as well as their derivatives (the corresponding angular rates), are used as components of the state vector. In the second case, state variables are the body yaw angle ϕ and its derivative. Introducing the state and control action vectors as

$$x_1 = \begin{bmatrix} \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \quad x_2 = \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix}, \quad u = \begin{bmatrix} u_l \\ u_r \end{bmatrix} \quad (19.1)$$

where u_l, u_r are the control actions to the left and right wheel motors, respectively, the equations of the linearized system are obtained in the form

$$\dot{x}_1 = A_1 x_1 + B_1 u \quad (19.2)$$

$$\dot{x}_2 = A_2 x_2 + B_2 u \quad (19.3)$$

where the expressions for the elements of matrices A_1, B_1, A_2, B_2 , as well as the values of corresponding parameters, are given in [188]. We present only the expressions for matrices A_1 and B_1 , which are used later on in the design of a robust controller:

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (19.4)$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ b_3 & b_3 \\ b_4 & b_4 \end{bmatrix} \quad (19.5)$$

where

$$\begin{aligned} a_{32} &= -g M L e_{12} / \det_E \\ a_{42} &= g M L e_{11} / \det_E \\ a_{33} &= -2(\sigma e_{22} + \beta e_{12}) / \det_E \\ a_{43} &= 2(\sigma e_{12} + \beta e_{11}) / \det_E \\ a_{34} &= 2\beta(e_{22} + e_{12}) / \det_E \\ a_{44} &= -2\beta(e_{11} + e_{12}) / \det_E \\ b_3 &= \alpha(e_{22} + e_{12}) / \det_E \\ b_4 &= -\alpha(e_{11} + e_{12}) / \det_E \\ e_{11} &= (2m + M)R^2 + 2J_w + 2n^2 J_m \\ e_{12} &= M L R - 2n^2 J_m \\ e_{22} &= M L^2 + J_\psi + 2n^2 J_m \end{aligned} \quad (19.6)$$

Table 19.1 Nominal parameters of the robot model

Parameter	Description	Value	Units
g	gravity acceleration	9.81	m/s ²
m	wheel mass	0.03	kg
R	wheel radius	0.04	m
J_w	wheel inertia moment	$mR^2/2$	kg m ²
M	body mass	0.6	kg
W	body width	0.14	m
D	body depth	0.04	m
H	body height	0.144	m
L	distance of the center of the mass from the wheel axle	$H/2$	m
J_ψ	body pitch inertia moment	$ML^2/3$	kg m ²
J_ϕ	body yaw inertia moment	$M(W^2 + D^2)/12$	kg m ²
J_m	DC motor inertia moment	1.0×10^{-5}	kg m ²
R_m	DC motor resistance	6.69	Ω
K_b	DC motor back e.m.f. constant	0.468	m
K_t	DC motor torque constant	0.317	Nm/A
n	gear ratio	1	
f_m	friction coefficient between body and DC motor	0.0022	
f_w	friction coefficient between body and motion surface	0.468 0.0001	

$$\det E = e_{11}e_{22} - e_{12}^2$$

$$\alpha = nK_t/R_m$$

$$\beta = nK_tK_b/R_m + f_m$$

$$\sigma = \beta + f_w$$

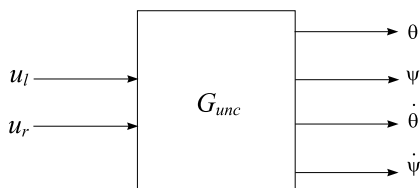
and the nominal values of the robot model parameters, in (19.6), are given in Table 19.1.

It should be noted that the robot model parameters are not determined with sufficient accuracy which may cause difficulties in controller designs, and thus robust controllers are required.

According to (19.6), the basic parameter of robot model is the body mass M , which determines almost all model parameters. Since this mass depends on the payload it is reasonable to assume it as an uncertain parameter of the robot model. In this case, however, the matrix Δ in the M - Δ model of the uncertain plant is a 52×52 matrix, in which the uncertainty of M occurs 52 times. This makes the synthesis of robust controller under such an assumption almost impossible in practice.

In the given case as uncertain parameters we consider the friction coefficient f_m between the robot body and DC motor and the friction coefficient f_w between the robot wheels and motion surface. The introduction of these sample uncertainties

Fig. 19.2 Uncertain model of the two-wheeled robot



allows to design easily a robust controller that ensures robust stability and robust performance of the closed-loop system with respect to the given uncertainty. In particular, we assume that the coefficient f_m is known with 20 % uncertainty and the coefficient f_w with 100 % uncertainty. The analysis performed later on shows that the coefficient f_m has more significant influence on the system dynamics.

The two real uncertain parameters f_m and f_w are defined in the program by using the following command lines:

```
fm = ureal('fm',0.0022,'Mode','Range','Percentage',20);
fw = ureal('fw',0.0001,'Mode','Range','Percentage',100);
```

As a result, for the subsystem (19.2) one obtains an uncertain plant G_{unc} with two inputs and four outputs (Fig. 19.2).

The uncertain system model is created by the M-file `mod_robot.m`.

The frequency response (in magnitude) of the nominal and uncertain plants (for random parameter values in the ranges assumed) are shown in Fig. 19.3. Obviously,

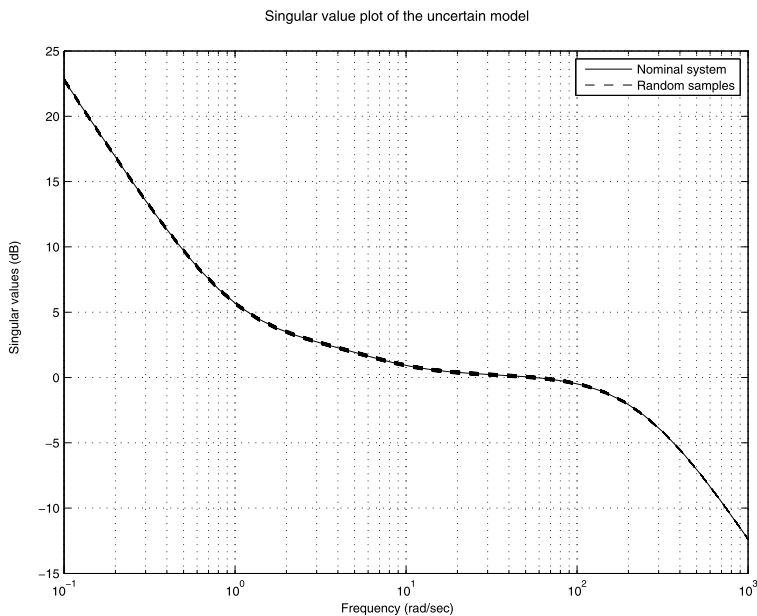


Fig. 19.3 Magnitude responses of the nominal and uncertain plant

the variation of the frequency response due to the variations in friction coefficients is relatively small.

19.3 Design of Robust Controller

The robot stabilization in upright position is a difficult task since the robot itself presents an inverted pendulum whose control requires a sufficiently accurate model. The design of a discrete-time controller for the subsystem (19.2) was done initially for the nominal plant model by using the \mathcal{H}_∞ -design. However, due to the inaccurate plant model the controller obtained was unable to stabilize the robot, although the simulation results were positive. That is why the final design is done by using the μ -synthesis, which allows to stabilize efficiently the robot in spite of the model inaccuracy. For better results it is appropriate to implement a 2-degree-of-freedom controller. The design is done for 250 Hz sampling frequency, which corresponds to the frequency of the signal at the output of gyro sensor. However, the following problem appears during the design. The experiments with different standard design configurations show that they cannot ensure good tracking of the reference angle θ , and hence the desired position in the motion plane. That is why apart from the constraints on the sensitivity function we add also a constraint on the integral of the error in wheels rotation angle. This allows to ensure sufficient accuracy in tracking the angle θ . The integral component is added also in the system feedback.

The subsystem (19.3) is controlled by a PID regulator as described in [188].

The closed-loop structure with the sensitivity function requirement on the output variables as well as the requirement on the control actions is shown in Fig. 19.4. The transfer function matrix W_p reflects the requirements on the system performance, the matrix W_u reflects the requirements on control actions and the transfer functions W_{n11} and W_{n22} reflects the influence of noises on the encoder and gyro measurements, respectively. The block *Intg* integrates the difference between the reference value and the output value of the angle θ and its output is used in the system feedback as well as an input to weighting function W_p .

The internal structure of the eight-input, 16-output open-loop system of 12th order, which is saved as the variable `sys_ic`, is shown in Fig. 19.5. The reference and noise are saved as the variables `ref` and `noise`, respectively. The control action is saved as the variable `control`. The input to the controller consists of the reference values of variables θ , ψ , $\dot{\theta}$, $\dot{\psi}$ as well as of four plant outputs and integrator output. The open-loop system interconnection is set by the M-file `olp_robot_2dof.m` implementing the function `sysic`.

The schematic diagram showing the specific input/output arrangement of the variable `sys_ic`, is shown in Fig. 19.6.

The closed-loop system block-diagram corresponding to the μ -synthesis problem is shown in Fig. 19.7. The matrix P is the transfer function matrix of the extended open-loop system shown in Fig. 19.5.

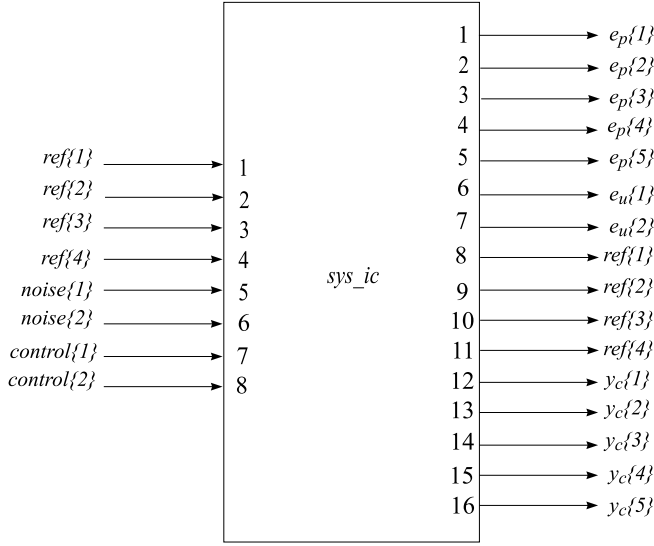
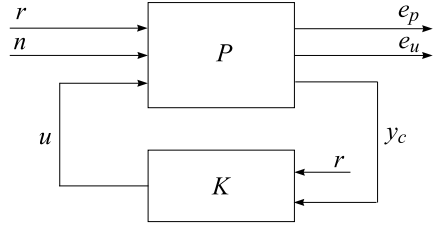


Fig. 19.6 Schematic diagram of the open-loop system interconnection

Fig. 19.7 Block diagram of the closed-loop system



In the given case the following weighting functions are found as appropriate in the controller design.

Performance weighting functions:

$$\begin{aligned}
 W_p &= \text{diag}(W_{p11}, W_{p22}, W_{p33}, W_{p44}, W_{p55}) \\
 W_{p11} &= 0.95 \frac{0.4s + 1}{5s + 0.06} \\
 W_{p22} &= 0.93 \\
 W_{p33} &= 0.15 \\
 W_{p44} &= 0.22 \frac{1.1s + 1}{1.0s + 1} \\
 W_{p55} &= 1.8 \frac{0.4s + 1}{5s + 0.06}
 \end{aligned} \tag{19.7}$$

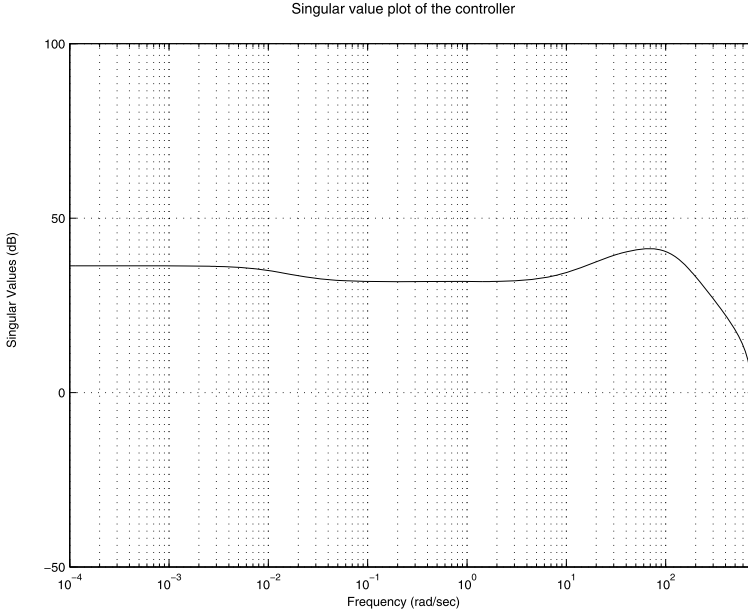


Fig. 19.8 Singular value plot of μ -controller

Control action weighting functions:

$$\begin{aligned}
 W_u &= \text{diag}(W_{u11}, W_{u22}) \\
 W_{u11} &= 0.02 \frac{1.2s + 1}{0.0024s + 1} \\
 W_{u22} &= 0.02 \frac{1.2s + 1}{0.0024s + 1}
 \end{aligned} \tag{19.8}$$

Noise shaping filters:

$$\begin{aligned}
 W_n &= \text{diag}(W_{n11}, W_{n22}) \\
 W_{n11} &= 0.1 \frac{1.0s + 2}{0.01s + 1} \\
 W_{n22} &= 1.0 \frac{1.0s + 2}{0.01s + 1}
 \end{aligned} \tag{19.9}$$

The design of 2-degree-of-freedom controller for the subsystem (19.2) is done by the M-file `dms_robot_2dof.m` that implements the function `dksyn` from Robust Control Toolbox^{®3}. As a result after three iterations one obtains a 12th order controller, which ensures a minimum value of the structured singular value μ equal to 0.930. This shows that the closed-loop system achieves robust performance in respect to the variation of both uncertain coefficients.

The singular value plot (the magnitude response) of the controller obtained is shown in Fig. 19.8. The maximum controller gain is 41 dB, which ensures accept-

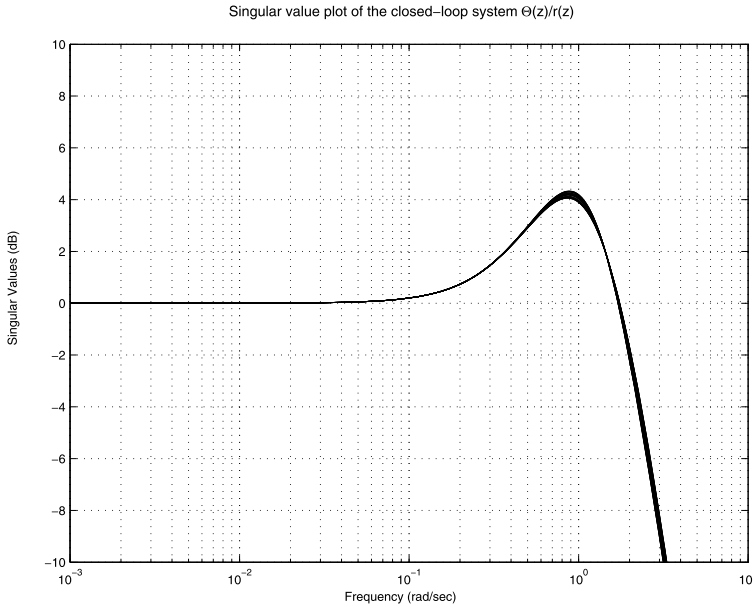


Fig. 19.9 Magnitude response of the uncertain closed-loop system

able control actions. The experiments show that larger controller gains lead to actuators saturation and instability of the real system.

19.4 Closed-Loop System Properties

After determining the controller, it is possible to compute several frequency responses and time responses of the closed-loop system that give profound information about its properties. The frequency responses are obtained by the M-file `dfrs_robot_2dof.m`.

In Fig. 19.9 we show a family of frequency response characteristics of the closed-loop with input and output the desired and actual angle θ , respectively, for different random values of the uncertain parameters in the prescribed range. It is seen that the responses have acceptable peaks, the frequency band width (at level -3 dB) being about 2.13 rad/s. For this band width it is possible to achieve good tracking of typical for the system under consideration reference signals.

The magnitude responses of the output sensitivity function of the closed-loop system along with the inverse performance function are shown in Fig. 19.10. It is seen that the disturbance attenuation in the low frequency range is better than this required by the performance weighting function.

In Fig. 19.11 we show the magnitude response of the loop from reference angle θ to the integral of the tracking error of this angle. Clearly, the prescribed frequency

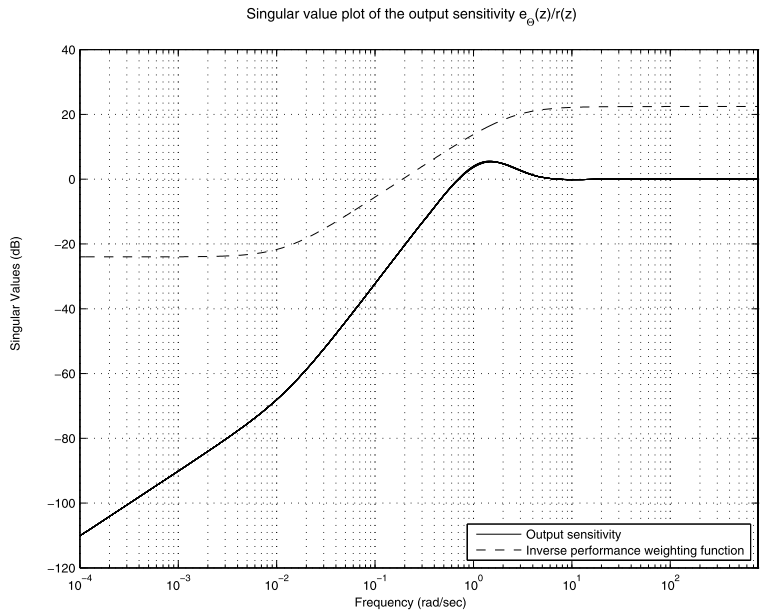


Fig. 19.10 Output sensitivity function of the closed-loop system

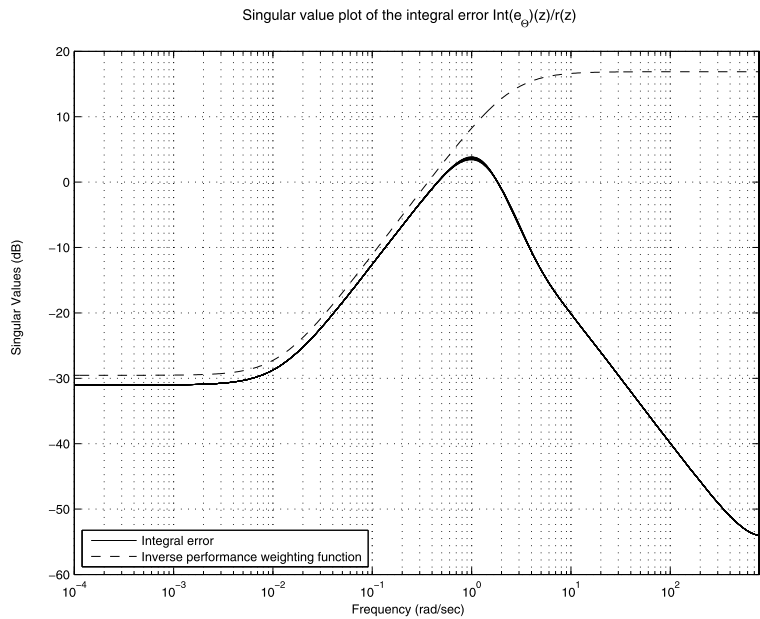


Fig. 19.11 Magnitude response of the integral component

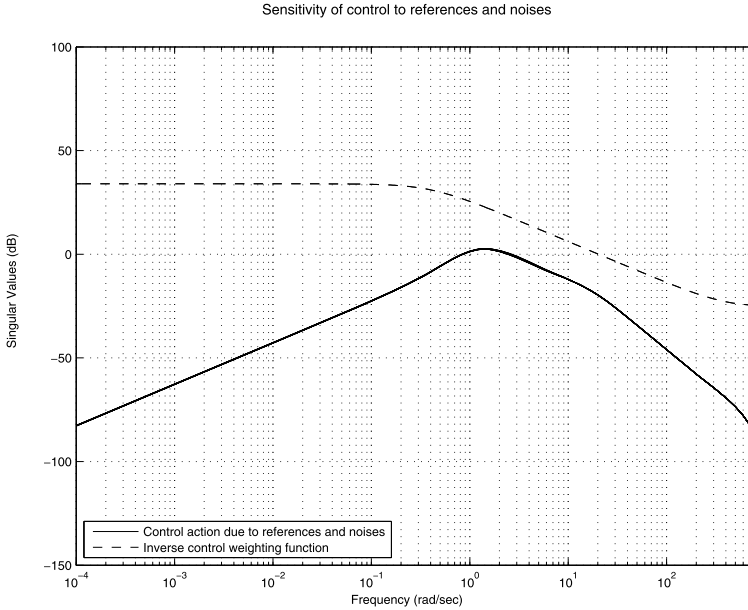


Fig. 19.12 Sensitivity of control action to references and noises

domain constraints on the integral error are fulfilled, the error attenuation in the low frequency range (where the reference spectrum lies) being 30 dB (i.e., more than 30 times). It is possible to achieve even better suppression of this error but in such a case the requirement for robust performance cannot be fulfilled (the value of μ increases).

In Fig. 19.12 we show the magnitude response of the loop from references and noises to the control actions (motor controls). It is seen that this response is lying below the frequency response of the inverse control weighting filter, i.e., the prescribed constraints on the controls are fulfilled. These constraints are chosen so that to avoid saturation of the actuators which generate the PWM signals to the motors.

The robust stability and robust performance analysis of the closed-loop system with discrete-time μ -controller is done by the file `dmu_robot_2dof.m`. The robust stability analysis of the uncertain closed-loop system shows that the upper bound on the structured singular value does not exceed the value of 0.063 (see Fig. 19.13). This means that the system may remain stable for much larger than the prescribed uncertainty in the corresponding parameters.

This is not the case in respect to the robust performance. It is seen from the frequency response characteristic of the structured singular value, shown in Fig. 19.14, that its maximum value is almost equal to 0.9304. This shows that there exists an uncertainty which is only 1.075 times larger than the existing one, for which the closed-loop system loses robust performance.

In Fig. 19.15 we show the worst case magnitude response of the closed-loop system obtained by the file `wcp_robot_2dof.m` implementing the function `wc-`

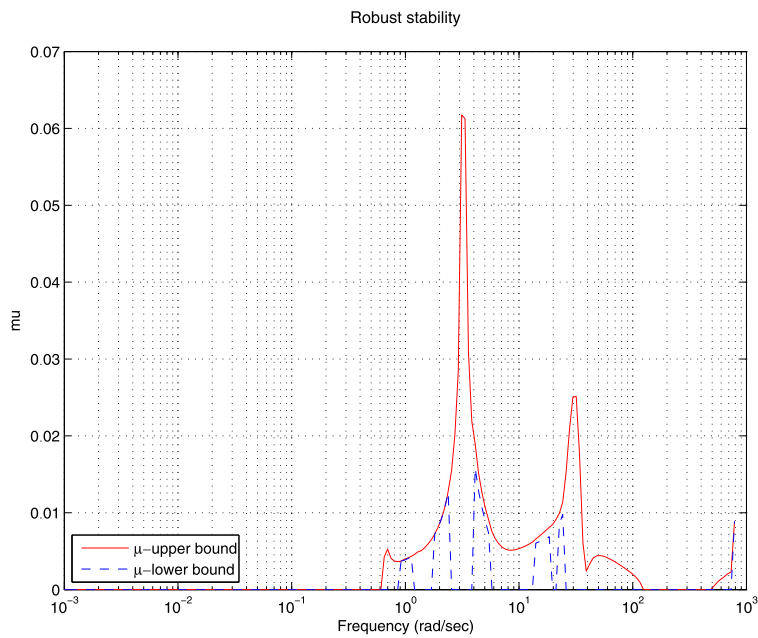


Fig. 19.13 Robust stability of the closed-loop system

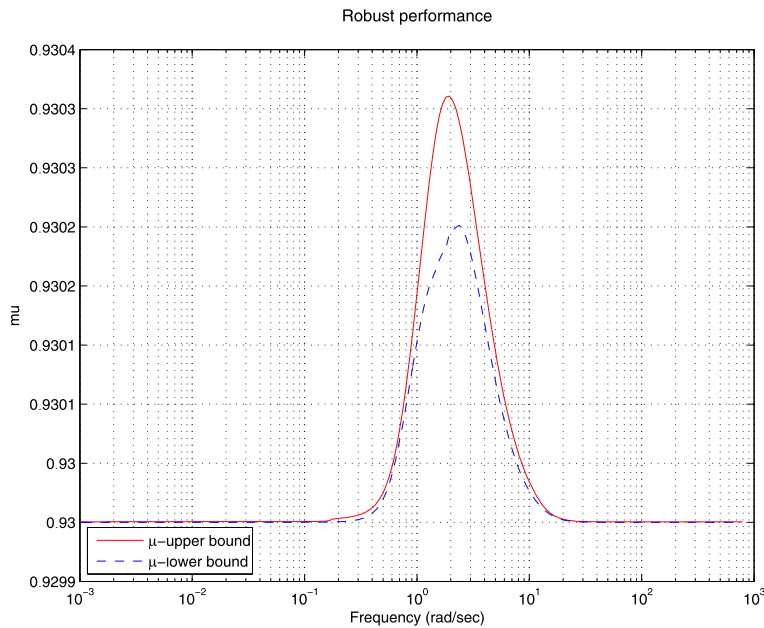


Fig. 19.14 Robust performance of the closed-loop system

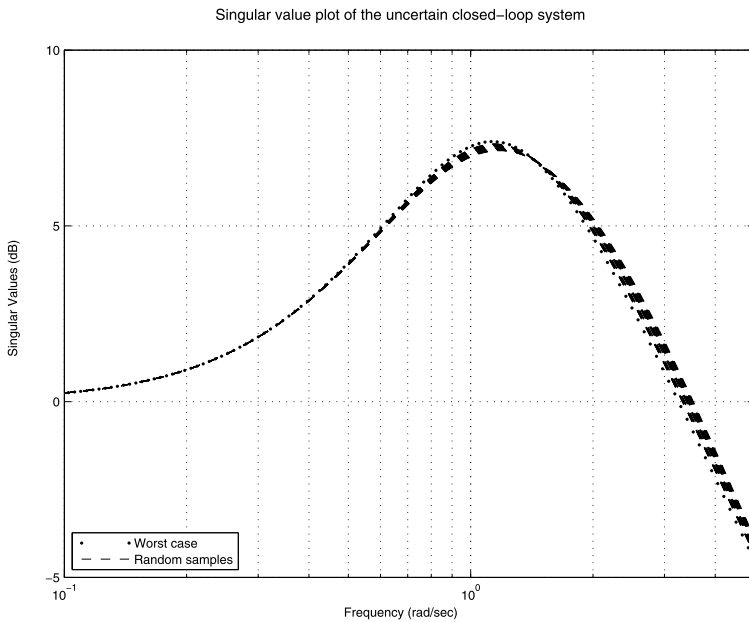


Fig. 19.15 Worst-case magnitude response of the closed-loop system

gain from Robust Control Toolbox[®] 3. Due to the robust performance achieved, the worst case response is slightly different from the nominal one.

The transient responses of the uncertain closed-loop system with discrete-time μ -controller is done by the file `mcs_robot_2dof.m`.

In Fig. 19.16 we show the reference angle θ , which is input to the controller as well as the transient response of the output angle θ . Multiplying these variables by the wheel radius $R = 0.04$ m it is possible to determine the accuracy of robot positioning in horizontal plane motion.

The body tilt angle, corresponding to the reference plane trajectory given in Fig. 19.16, is shown in Fig. 19.17. Obviously, this angle is in the range of ± 2 degrees.

The control action to the right motor, corresponding to the reference trajectory given in Fig. 19.16, is shown in Fig. 19.18.

The simulation of the nonlinear closed-loop system in the time domain is done by the program `nxtway_gs_vr.mdl` from [188], in which the controller, as implemented by the file `nxtway_gs_controller.mdl`, is modified such that to use the μ -regulator designed (see for details the next section). The simulation of the closed-loop system for zero reference and initial deviation of robot body from the vertical plane shows that the robot successfully and quickly returns to the balance point. Since the simulation program `nxtway_gs_vr.mdl` implements the linearized plant model, it is not possible to establish the maximum value of the vertical deviation for which the robot preserves stability. This is done experimentally as described in the next section.

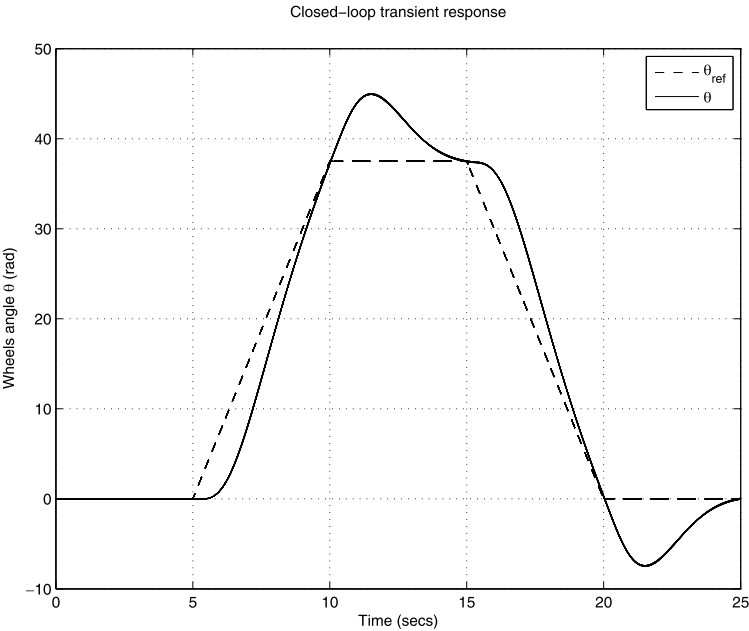


Fig. 19.16 Transient response of the closed-loop system

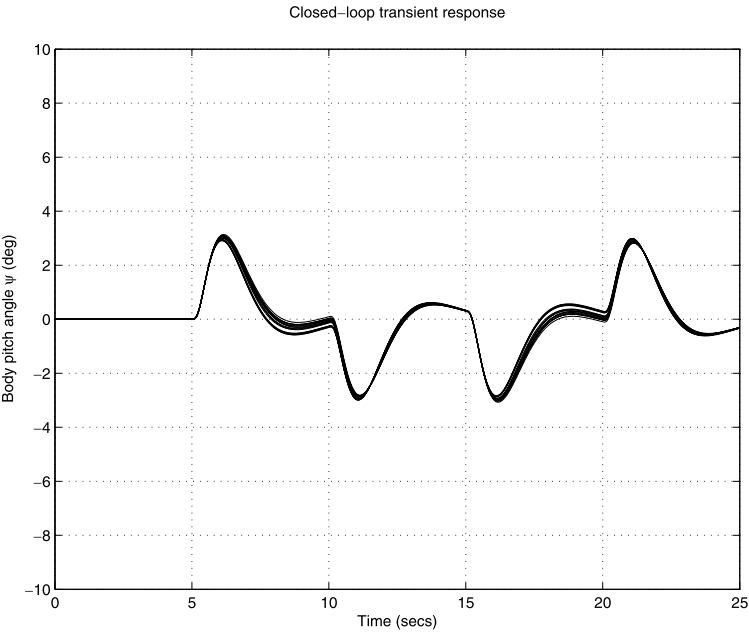


Fig. 19.17 Transient response of the closed-loop system: body tilt angle

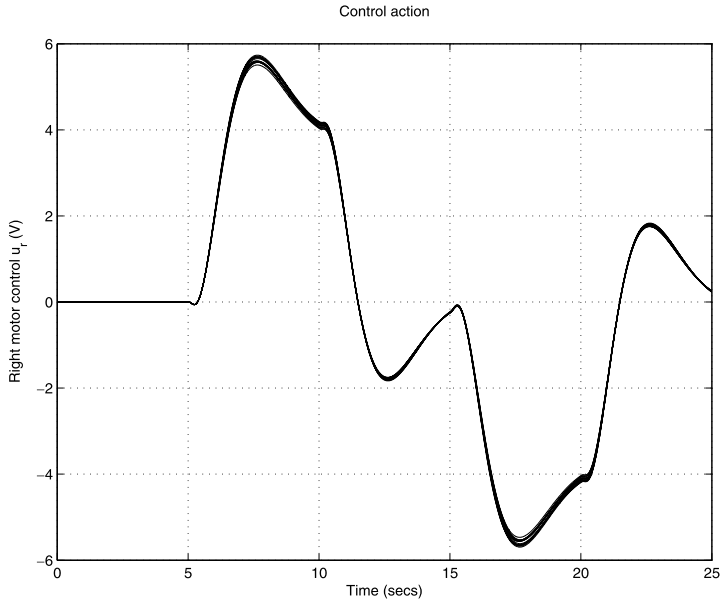


Fig. 19.18 Right motor control action

19.5 Experimental Results

The designed robust regulator of the two-wheeled robot motion is experimented in practice using the available software for automatic generation of C control code and its loading in the digital robot controller, as presented in [188].

For this aim we use the modified block-diagram of the controller model in Simulink®, shown in Fig. 19.19 and saved in the file `robust_controller`. The 12th order μ -regulator is represented by the state space block “Discrete Controller” and the integration of the tracking error is done by the block “Discrete Integrator”. This controller replaces the original linear quadratic state controller used in the Simulink® file `nxtway_gs_vr.mdl` adopted from [188]. The generation of the control code is done by using the Simulink Coder®. During the experiments one measures the control signals to both motors, the pitch angle ψ , the battery voltage and so on. The data transfer from robot to the personal computer is done by using the bluetooth-protocol. The file containing the results is loaded to MATLAB® which allows to visualize the experimental data. Since the angular rate $\dot{\psi}$ measured by the gyroscope and its integral are contaminated by noises there is some bias in the computed value of ψ , which is removed by the MATLAB® function `dtrend`.

In Fig. 19.20 we show the results from one of the experiments performed. The robot is subject to two different disturbance forces causing deviations from the vertical position at approximately $t = 54$ s and $t = 90$ s. For the bigger deviation of 16.5° at $t = 54$ s the regulator succeeds to stabilize the system and to return the

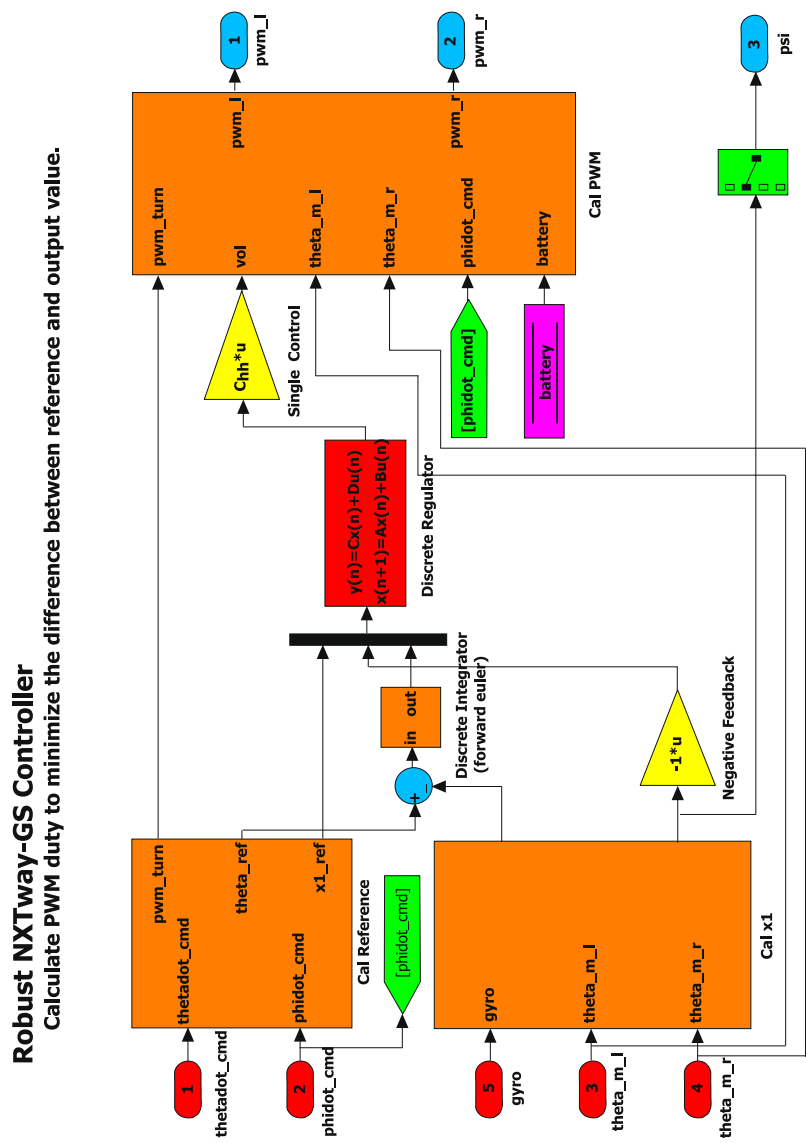


Fig. 19.19 Controller implementation

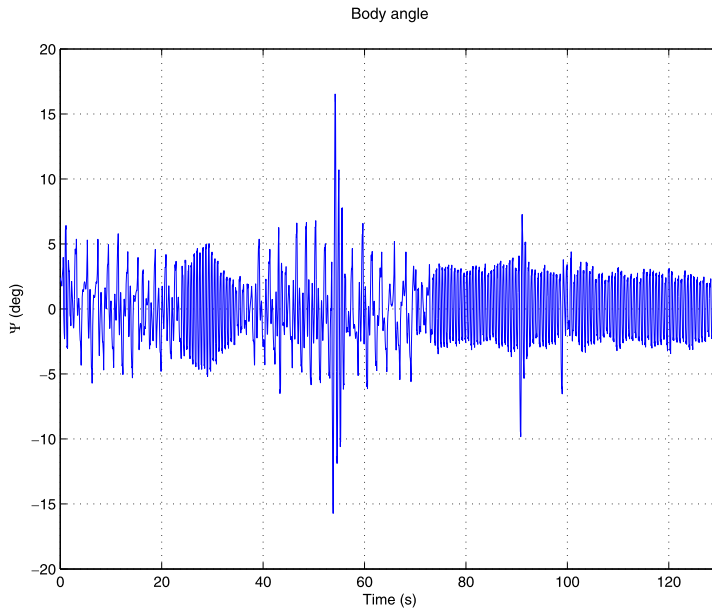


Fig. 19.20 Transient responses to disturbance forces

robot to the balance point. There are some oscillations in the steady-state mode of order $\pm 2^\circ$ that are due to the dead-zone in motor gears.

The control signal to the right wheel for the same experiment is shown in Fig. 19.21. In stabilization mode the control magnitude is between 40 % and 60 % from the maximum one which guarantees that the actuator saturation will be avoided. Since the control at $t = 54$ s reaches 100 % it is justified to assume that the deviation of 16.5° is the maximum allowable one for which the robot is kept stable.

19.6 Conclusions

The experience gained in the design and experimental evaluation of the Two-Wheeled Robot System controller allows to derive the following conclusions.

- The inclusion of all possible uncertainties in the plant model may lead to a very complicated model that makes robust controller design impossible in practice. In many cases it is sufficient to take a small number of uncertain parameters in order to find a robust controller that ensures acceptable performance in practice.
- The LEGO[®] Mindstorms NXT microcontroller is capable to realize complicated high-order control laws that may ensure stability of the closed-loop system for sufficiently large deviations from the equilibrium state.

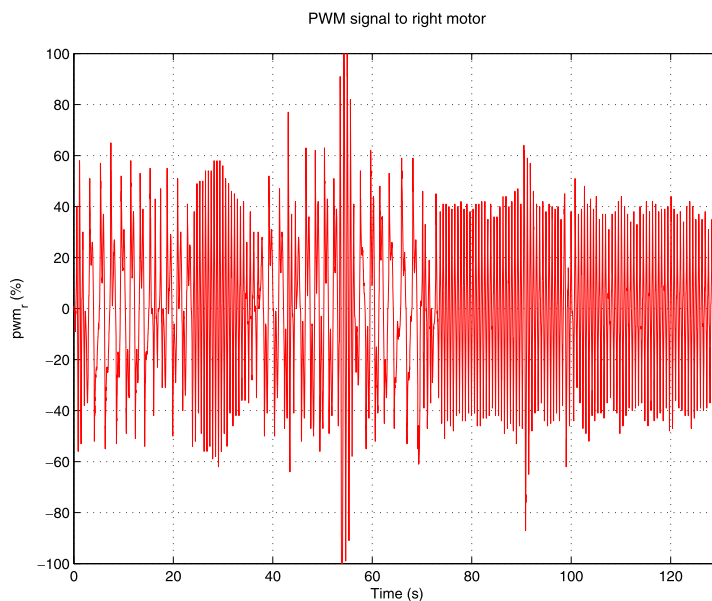


Fig. 19.21 Control action to the right wheel motor

- The presence of nonlinearities in the system, like dead-zone or backlash, may deteriorate significantly the closed-loop performance since the design of the robust controller is based on the assumption of a linearized plant model.
- The presence of bias and noises in the gyroscope signal may require inclusion of appropriate filters to determine sufficiently accurate estimates of the tilt angle and its derivative.

19.7 Notes and References

Different types of wheeled robot and their properties are described in [18]. The two-wheeled NXTway-GS robot, presented in this chapter, is an inverted-pendulum-type robot which should be stabilized around the vertical position by a control system. Usually, such robots are equipped with two servo drives for actuation, a gyroscope for measuring angle and angular velocity of pendulum body, encoders for measuring the position of the wheels and a microcontroller implementing a discrete real-time stabilization algorithm. Since the gyroscope measurements are subject to bias, random walk and noise, the body tilt angle is frequently estimated by using a Kalman filter. In some implementations it is appropriate to add an accelerometer or to use Inertial Measurement Unit with MEM sensors in order to eliminate the drift.

The most popular commercial product, built on the idea of self balancing two-wheeled robot, is the Segway® Personal Transporter (PT), produced by Segway Inc.,

USA [152]. Some of the Segway[®] PTs have maximum speed of 20 km/h and can travel as far as 38 km on a single battery charge.

Different dynamic models and methods for real-time control of two-wheeled robots may be found in [30, 85, 128], including self-tuning PID controllers [142], robust [170] and fuzzy controllers [186]. When the size of the robots gets bigger and mass gets smaller it is necessary to take into account the flexibility of robot body [143]. Many issues connected to the development of real-time systems for mobile robot control are considered in depth in the excellent book of Bräunl [16].

References

1. Abramovitch, D.Y.: Magnetic and optical disk control: parallels and contrasts. In: Proceedings of the 2001 American Control Conference, Arlington, VA, June 2001, pp. 421–428 (2001)
2. Abramovitch, D., Franklin, G.: A brief history of disk drive control. *IEEE Control Syst. Mag.* **22**, 28–42 (2002)
3. Abramovitch, D., Franklin, G.: Disk drive control: the early years. In: Proceedings of the 15th IFAC Congress. Session T-Th-M12, Barcelona, Spain, July 2002, pp. 1–12 (2002). CD-ROM
4. Abramovitch, D., Wang, F., Hurst, T., Franklin, G.: Disk drive pivot nonlinearity modelling Part I: Frequency domain. In: Proceedings of the 1994 American Control Conference, Baltimore, MD, June 1994, pp. 2600–2603 (1994)
5. Abramovitch, D., Hurst, T., Henze, D.: An overview of the PES Pareto method for decomposing baseline noise sources in hard disk position error signals. *IEEE Trans. Magn.* **34**, 17–23 (1998)
6. Adamjan, V.M., Arov, D.Z., Krein, M.G.: Analytic properties of Schmidt pairs for a Hankel operator and the generalized Schur–Takagi problem. *Math. USSR Sb.* **15**, 31–73 (1971)
7. Al-Saggaf, U.M., Franklin, G.F.: An error bound for a discrete reduced order model of a linear multivariable system. *IEEE Trans. Autom. Control* **AC-32**, 815–819 (1987)
8. Anderson, B.D.O., Moore, J.B.: *Linear Optimal Control*. Prentice Hall, Englewood Cliffs (1972)
9. Apkarian, P., Gahinet, P.: A convex characterization of gain-scheduled \mathcal{H}_∞ controllers. *IEEE Trans. Autom. Control* **40**, 853–864 (1995)
10. Azlan, N., Zainudin, F., Yusuf, H., Toha, S., Yusoff, S., Osman, N.: Fuzzy logic controlled miniature LEGO robot for undergraduate training system. In: Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications, ICIEA 2007, Harbin, PRC, May 2007, pp. 2184–2188 (2007)
11. Balas, G.J., Doyle, J.C., Glover, K., Packard, A., Smith, R.: μ -Analysis and Synthesis Toolbox: User's Guide. MUSYN/The Mathworks, Natick (1995)
12. Balas, G., Chiang, R., Packard, A., Safonov, M.: Robust Control Toolbox® User's Guide. The Mathworks, Natick (2012). http://www.mathworks.com/help/pdf_doc/robust/robust Ug.pdf
13. Behrens, A., Atorf, L., Schwann, R., Neumann, B., Schnitzler, R., Ballé, J., Herold, T., Telle, A., Noll, T.G., Hameyer, K., Aach, T.: MATLAB meets LEGO Mindstorms—a freshman introduction course into practical engineering. *IEEE Trans. Ed.* **53**, 306–317 (2010)
14. Bernstein, D.S., Haddad, W.M.: LQG control with an \mathcal{H}_∞ performance bound: a Riccati equation approach. *IEEE Trans. Autom. Control* **AC-34**, 293–305 (1989)
15. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: *Linear Matrix Inequality in Systems and Control Theory*. Society for Industrial and Applied Mathematics, Philadelphia (1994)

16. Bräunl, T.: *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*, 3rd edn. Springer, Berlin (2008). ISBN 978-3-540-70533-8
17. Callier, F.M., Desoer, C.A.: *Linear System Theory*. Springer, New York (1991)
18. Campion, G., Chung, W.: Wheeled robots. In: Siciliano, B., Khatib, O. (eds.) *Springer Handbook of Robotics*, pp. 391–410. Springer, Berlin (2008). Chap. 17
19. Castillo, P., Lozano, R., Dzul, A.E.: *Modelling and Control of Mini-Flying Machines*. Springer, London (2005). ISBN 1852339578
20. Chen, B.M., Lee, T.H., Peng, K., Venkataramanan, V.: *Hard Disk Drive Servo Systems*, 2nd edn. Springer, Berlin (2006)
21. Chew, K.K., Tomizuka, M.: Digital control of repetitive errors in disk drive systems. *IEEE Control Syst. Mag.* **10**, 16–20 (1990)
22. Chikamasa, T.: Embedded Coder Robot NXT demo. <http://www.mathworks.com/matlabcentral/fileexchange/13399>
23. Choi, B.W., Gu, D.W., Postlethwaite, I.: Low-order \mathcal{H}_∞ sub-optimal controllers. *IEE Proc. Part D, Control Theory Appl.* **141**, 243–248 (1994)
24. de Gaston, R.R.E., Safonov, M.G.: Exact calculation of the multiloop stability margin. *IEEE Trans. Autom. Control* **AC-33**, 156–171 (1988)
25. De Luca, A., Siciliano, B.: Trajectory control of a non-linear one-link flexible arm. *Int. J. Control* **50**, 1699–1715 (1989)
26. Desai, U.B., Pal, D.: A transformation approach to stochastic model reduction. *IEEE Trans. Autom. Control* **AC-29**, 1097–1100 (1984)
27. Desoer, C.A., Chan, W.S.: The feedback interconnection of lumped linear time-invariant systems. *J. Franklin Inst.* **300**, 335–351 (1975)
28. Desoer, C.A., Vidyasagar, M.: *Feedback Systems: Input-Output Properties*. Academic Press, New York (1975)
29. Ding, J., Tomizuka, M., Numasato, H.: Design and robustness analysis of dual stage servo system. In: *Proceedings of the 2000 American Control Conference*, Chicago, Illinois, June 2000, pp. 2605–2609 (2000)
30. Do, K.D., Seet, G.: Motion control of a two-wheeled mobile vehicle with an inverted pendulum. *J. Intell. Robot. Syst.* **60**, 577–605 (2010)
31. Doyle, J.C.: Analysis of feedback systems with structured uncertainties. *IEE Proc. Part D, Control Theory Appl.* **129**, 242–250 (1982)
32. Doyle, J.C.: Structured uncertainty in control system design. In: *Proceedings of the 24 IEEE Conference on Decision and Control*, December 1985, pp. 260–265 (1985)
33. Doyle, J.C.: A review of μ : for case studies in robust control. In: *Proceedings of 10th IFAC World Congress*, Munich, Germany, July 1987, pp. 395–402 (1987)
34. Doyle, J.C., Francis, B.A., Tannenbaum, A.R.: *Feedback Control Theory*. Macmillan, New York (1992)
35. Du, C., Zhang, J., Guo, G.: Vibration analysis and control design comparison of HDDs using fluid bearing and ball bearing spindles. In: *Proceedings of the 2002 American Control Conference*, Anchorage, AK, May 2002, pp. 1378–1383 (2002)
36. El Ghaoui, L., Niculescu, S.L. (eds.): *Advances in Linear Matrix Inequality Methods in Control: Advances in Design and Control*. Society for Industrial and Applied Mathematics, Philadelphia (2000)
37. Eltohamy, K.G.: Nonlinear optimal control of a triple inverted pendulum with single control input. *Int. J. Control* **69**, 239–256 (1998)
38. Enns, D.: *Model reduction for control systems design*. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA (1984)
39. Enns, D.: Model reduction with balanced realizations: an error bound and a frequency weighted generalization. In: *Proceedings of the 23rd IEEE Conference on Decision and Control*, Las Vegas, NV, pp. 127–132 (1984)
40. Evans, R.B., Griesbach, J.S., Messner, W.C.: Piezoelectric microactuator for dual stage control. *IEEE Trans. Magn.* **35**, 977–982 (1999)

41. Fan, M.K.H., Tits, A.L.: Characterization and efficient computation of the structured singular value. *IEEE Trans. Autom. Control* **AC-31**, 734–743 (1986)
42. Fan, M.K.H., Tits, A.L., Doyle, J.C.: Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Trans. Autom. Control* **AC-36**, 25–38 (1991)
43. Fernando, K.V., Nicholson, H.: Singular perturbational model reduction of balanced systems. *IEEE Trans. Autom. Control* **AC-27**, 466–468 (1982)
44. Fernando, K.V., Nicholson, H.: Singular perturbational approximation for discrete-time systems. *IEEE Trans. Autom. Control* **AC-28**, 240–242 (1983)
45. Francis, B.A.: *A Course in \mathcal{H}^∞ Control Theory*. Lecture Notes in Control and Information Sciences, vol. 88. Springer, Berlin (1987)
46. Franklin, G.F., Powell, J.D., Workman, M.L.: *Digital Control of Dynamic Systems*, 3rd edn. Addison-Wesley, Menlo Park (1998)
47. Furuta, K., Kajiwara, K., Kosuge, K.: Digital control of a double inverted pendulum on an inclined rail. *Int. J. Control* **32**, 907–924 (1980)
48. Furuta, K., Ochia, T., Ono, N.: Attitude control of a triple inverted pendulum. *Int. J. Control* **39**, 1351–1365 (1984)
49. Gahinet, P., Apkarian, P.: A linear matrix inequality approach to \mathcal{H}_∞ control. *Int. J. Robust Nonlinear Control* **4**, 421–448 (1994)
50. Gahinet, P., Nemirovski, A., Laub, A.J., Chilali, M.: *LMI Control Toolbox*. The MathWorks, Natick (1995)
51. Garcia-Benitez, E., Walkins, J., Yurkovich, S.: Nonlinear control with acceleration feedback for a two-link flexible robot. *Control Eng. Pract.* **1**, 989–997 (1993)
52. Ge, S., Lee, T., Zhu, G.: Improving regulation of a single-link flexible manipulator with strain feedback. *IEEE Trans. Robot. Autom.* **14**, 179–185 (1998)
53. Geniele, H., Patel, R., Khorasani, K.: End-point control of a flexible-link manipulator: theory and experiment. *IEEE Trans. Control Syst. Technol.* **5**, 559–570 (1997)
54. Ghanekar, M., Wang, D., Heppler, G.: Scaling laws for linear controllers of flexible link manipulators characterized by nondimensional groups. *IEEE Trans. Robot. Autom.* **13**, 117–127 (1997)
55. Glover, K.: All optimal Hankel-norm approximations of linear multivariable systems and their l^∞ -error bounds. *Int. J. Control* **39**, 1115–1193 (1984)
56. Glover, K.: Multiplicative approximation of linear multivariable systems with \mathcal{L}_∞ error bounds. In: *Proceedings of the 1986 American Control Conference*, Minneapolis, MN, pp. 1705–1709 (1986)
57. Glover, K., Doyle, J.C.: State-space formulae for all stabilizing controllers that satisfy an \mathcal{H}^∞ norm bound and relations to risk sensitivity. *Syst. Control Lett.* **11**, 167–172 (1988)
58. Glover, K., Jonckheere, E.A.: A comparison of two Hankel norm methods for approximating spectra. In: Byrnes, C.I., Lindquist, A. (eds.) *Modelling, Identification and Robust Control*. North-Holland, Amsterdam (1986)
59. Goh, T.B., Li, Z., Chen, B.M., Lee, T.H., Huang, T.: Design and implementation of a hard disk drive servo system using robust and perfect tracking approach. In: *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona, December 1999, pp. 5247–5252 (1999)
60. Green, M.: Balanced stochastic realizations. *Linear Algebra Appl.* **98**, 211–247 (1988)
61. Green, M.: A relative error bound for balanced stochastic truncation. *IEEE Trans. Autom. Control* **AC-33**, 961–965 (1988)
62. Green, M., Limebeer, D.J.N.: *Linear Robust Control*. Prentice Hall, Englewood Cliffs (1995)
63. Gu, D.W., Tsai, M.C., Postlethwaite, I.: State-space formulae for discrete-time optimization. *Int. J. Control* **49**, 1683–1723 (1989)
64. Gu, D.W., Postlethwaite, I., Tsai, M.C.: \mathcal{H}_∞ super-optimal solutions. In: Leondes, C.T. (ed.) *Advances in Control and Dynamic Systems*, vol. 51, pp. 183–246. Academic Press, San Diego (1992)
65. Gu, D.W., Choi, B.W., Postlethwaite, I.: Low-order stabilizing controllers. *IEEE Trans. Autom. Control* **AC-38**, 1713–1717 (1993)

66. Gu, D.W., Goh, S.J., Fitzpatrick, T., Postlethwaite, I., Measor, N.: Application of an expert system for robust controller design. In: Control'96, vol. 2, pp. 1004–1009 (1996)
67. Gu, D.W., Petkov, P.H., Konstantinov, M.M.: Direct Formulae for the \mathcal{H}_∞ Sub-Optimal Central Controller. Technical Report 1998-7, Niconet Report (1998). <http://www.win.tue.nl/niconet>
68. Gu, D.W., Petkov, P.H., Konstantinov, M.M.: Formulae for discrete \mathcal{H}_∞ loop shaping design procedure controllers. In: Proceedings of the 15th IFAC World Congress, Paper 276, Session T-Mo-M15, Barcelona, Spain, July 2002. CD-ROM
69. Guo, L., Lee, H.S., Hudson, A., Chen, S.-H.: A comprehensive time domain simulation tool for hard disk drive TPI prediction and mechanical/servo enhancement. IEEE Trans. Magn. **35**, 879–884 (1999)
70. Hammarling, S.: Numerical solution of the stable non-negative definite Lyapunov equation. IMA J. Numer. Anal. **2**, 303–323 (1982)
71. Hara, S., Hara, T., Yi, L., Tomizuka, M.: Two degree-of-freedom controllers for hard disk drives with novel reference signal generation. In: Proceedings of the 1999 American Control Conference, San Diego, CA, June 1999, pp. 4116–4121 (1999)
72. Helton, J.W.: Worst case analysis in the frequency domain: the \mathcal{H}^∞ approach to control. IEEE Trans. Autom. Control **AC-30**, 1154–1170 (1985)
73. Helton, J.W., Merino, O.: Classical Control Using \mathcal{H}^∞ Methods. Society for Industrial and Applied Mathematics, Philadelphia (1998)
74. Hernandez, D., Park, S.-S., Horowitz, R., Packard, A.K.: Dual stage track-following servo design for hard disk drives. In: Proceedings of the 1999 American Control Conference, San Diego, CA, June 1999, pp. 4116–4121 (1999)
75. Hoyle, D., Hyde, R., Limebeer, D.J.N.: An \mathcal{H}^∞ approach to two-degree-of-freedom design. In: Proceedings of the 30th IEEE Conference on Decision and Control, Brighton, UK, December 1991, pp. 1581–1585 (1991)
76. Hsu, C.S., Yu, X., Yeh, H.H., Banda, S.S.: \mathcal{H}_∞ compensator design with minimal order observer. In: Proceedings of the 1993 American Control Conference, San Francisco, CA, June 1993
77. Huang, Y., Mathur, P., Messner, W.C.: Robustness analysis on a high bandwidth disk drive servo system with an instrumented suspension. In: Proceedings of the 1999 American Control Conference, San Diego, CA, June 1999, pp. 3620–3624 (1999)
78. International Society Niconet. SLICOT: Numerical Subroutine Library for Control and Systems Theory (2001). <http://www.win.tue.nl/niconet>
79. Ishikawa, J., Yanagita, Y., Hattori, T., Hashimoto, M.: Head positioning control for low sampling rate systems based on two degree-of-freedom control. IEEE Trans. Magn. **32**, 1787–1792 (1996)
80. Iwasaki, T., Skelton, R.E.: All low order \mathcal{H}_∞ controllers with covariance upper bound. In: Proceedings of the 1993 American Control Conference, San Francisco, CA, June 1993
81. Kajiwar, H., Apkarian, P., Gahinet, P.: LPV techniques for control of an inverted pendulum. IEEE Control Syst. Mag. **19**, 44–54 (1999)
82. Karkoub, M., Balas, G., Tamma, K., Donath, M.: Robust control of flexible manipulators via μ -synthesis. Control Eng. Pract. **8**, 725–734 (2000)
83. Kempf, C., Messner, W., Tomizuka, M., Horovitz, R.: Comparison of four discrete-time repetitive control algorithms. IEEE Control Syst. Mag. **13**, 48–54 (1993)
84. Kim, S.W., Anderson, B.D.O., Madieviski, A.G.: Error bound for transfer function order reduction using frequency weighted balanced truncation. Syst. Control Lett. **24**, 183–192 (1995)
85. Kim, Y., Kim, S.H., Kwak, Y.K.: Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot. J. Intell. Robot. Syst. **44**, 25–46 (2006)
86. Kobayashi, M., Yamaguchi, T., Horowitz, R.: Track-seeking controller design for dual-stage actuator in magnetic disk drives. In: Proceedings of the 2000 American Control Conference, Chicago, Illinois, June 2000, pp. 2610–2614 (2000)

87. Kozierek, C.M.: The PC Guide: The Reference Section on Hard Disk Drives. June 2012. <http://www.pcguide.com>
88. Kundur, P.: Power System Stability and Control. McGraw-Hill, New York (1994). ISBN-13: 978-0070359581
89. Kung, S.: A new low-order approximation algorithm via singular value decomposition. In: Proceedings of the 18th IEEE Conference on Decision and Control, Ft. Lauderdale, Florida, December 1979
90. Kung, S., Lin, D.W.: Optimal Hankel norm model reduction: multivariable systems. IEEE Trans. Autom. Control **AC-26**, 832–852 (1981)
91. Larcombe, P.J.: On the generation and solution of the symbolic, open-loop characteristic equation for a double inverted pendulum. Int. J. Syst. Sci. **24**, 2379–2390 (1993)
92. Laub, A.J.: On computing balancing transformations. In: Proceedings of the Joint 1980 American Control Conference, San Francisco, CA, August 1980, p. 8 (1980)
93. Laub, A.J., Heath, M.T., Paige, C.C., Ward, R.C.: Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. IEEE Trans. Autom. Control **AC-32**, 115–121 (1987)
94. Lee, H.S.: Controller optimization for minimum position error signals of hard disk drives. In: Proceedings of the 2000 American Control Conference, Chicago, Illinois, June 2000, pp. 3081–3085 (2000)
95. Lee, S.-H., Kang, H.J., Chung, C.C.: Robust fast seek control of a servo track writer using a state space disturbance observer. IEEE Trans. Control Syst. Technol. **20**, 346–355 (2012)
96. Li, Y., Thang, B., Shi, Z., Lu, Y.: Experimental study for trajectory tracking of a two-link flexible manipulators. Int. J. Syst. Sci. **31**, 3–9 (2000)
97. Limebeer, D.J.N.: The specification and purpose of a controller design study. In: Proceedings of the 30th IEEE Conference on Decision and Control, Brighton, UK, December 1991, pp. 1579–1580 (1991)
98. Limebeer, D.J.N., Kasenally, E.M., Perkins, J.D.: On the design of robust two degree of freedom controllers. Automatica **29**, 157–168 (1993)
99. Lin, J.L.: Control system design for robust stability and robust performance. PhD thesis, Department of Engineering, University of Leicester, Leicester, UK, May 1992
100. Lin, C.-A., Chiu, T.-Y.: Model reduction via frequency weighted balanced realization. Control Theory Adv. Technol. **8**, 341–351 (1992)
101. Lin, J.L., Postlethwaite, I., Gu, D.W.: μ -K iteration: a new algorithm for μ -synthesis. Automatica **29**, 219–244 (1993)
102. Lindquist, A., Picci, G.: On the stochastic realization problem. SIAM J. Control Optim. **17**, 365–389 (1979)
103. Liu, Y., Anderson, B.D.O.: Singular perturbation of balanced systems. Int. J. Control **50**(4), 1379–1405 (1989)
104. López-Martínez, M., Vivas, C., Ortega, M.G.: A multivariable nonlinear \mathcal{H}_∞ controller for a laboratory helicopter. In: Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain, December 2005, pp. 4065–4070 (2005)
105. Lozano, R. (ed.): Unmanned Aerial Vehicles Embedded Control. Wiley-ISTE, Chichester (2010). ISBN 978-1-84821-127-8
106. Lundström, P., Skogestad, S., Doyle, J.C.: Two-degree-of-freedom controller design for an ill-conditioned distillation process using μ -synthesis. IEEE Trans. Control Syst. Technol. **7**, 12–21 (1999)
107. Lunz, J.: Robust Multivariable Feedback Control. Prentice Hall, London (1989)
108. MacDuffee, C.C.: The Theory of Matrices. Chelsea, New York (1950)
109. Maciejowski, J.M.: Multivariable Feedback Design. Addison-Wesley, Wokingham (1989). ISBN 0-201-18243-2
110. Magee, D.P.: Optimal filtering to improve performance in hard disk drives: simulation results. In: Proceedings of the 1999 American Control Conference, San Diego, CA, June 1999, pp. 71–75 (1999)

111. McFarlane, D.C., Glover, K.: Robust Controller Design Using Normalized Coprime Factor Plant Descriptions. Lecture Notes in Control and Information Sciences, vol. 138. Springer, Berlin (1990)
112. McFarlane, D., Glover, K.: A loop shaping design procedure using \mathcal{H}_∞ synthesis. IEEE Trans. Autom. Control **AC-37**, 749–769 (1992)
113. Medrano-Cerda, G.A.: Robust stabilization of a triple inverted pendulum-cart. Int. J. Control **68**, 849–865 (1997)
114. Medrano-Cerda, G.A.: Robust computer control of an inverted pendulum. IEEE Control Syst. Mag. **19**, 58–67 (1999)
115. Meier, Z., Farwig, H., Unbehauen, H.: Discrete computer control of a triple-inverted pendulum. Optim. Control Appl. Methods **11**, 157–171 (1990)
116. Meirovitch, L.: Elements of Vibration Analysis. McGraw-Hill, New York (1986)
117. Messner, W., Ehrlich, R.: A tutorial on controls for disk drives. In: Proceedings of the 2001 American Control Conference, Arlington, VA, June 2001, pp. 408–420 (2001)
118. Mettler, B.: Identification Modeling and Characteristics of Miniature Rotorcraft. Kluwer Academic, Boston (2003). ISBN 1-4020-7228-7-096-1
119. Meyer, D.G.: Fractional balanced reduction: Model reduction via fractional representation. IEEE Trans. Autom. Control **AC-35**(3), 1341–1345 (1990)
120. Moore, B.C.: Principle component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans. Autom. Control **AC-26**, 17–31 (1981)
121. Morari, M., Zafiriou, E.: Robust Process Control. Prentice Hall, Englewood Cliffs (1989)
122. Mori, S., Nishihara, H., Furuta, K.: Control of an unstable mechanical system. Control of pendulum. Int. J. Control **23**, 673–692 (1976)
123. Murad, G.: Robust multivariable control of industrial production processes: a discrete-time multi-objective approach. PhD thesis, Department of Engineering, University of Leicester, Leicester, UK (1995)
124. Murad, G., Postlethwaite, I., Gu, D.-W., Whidborne, J.F.: Robust control of a glass tube shaping process. In: Proceedings of the Second European Control Conference, Groningen, The Netherlands, June–July 1993, vol. 4, pp. 2350–2355 (1993)
125. Murad, G., Postlethwaite, I., Gu, D.-W., Samar, R.: On the structure of an \mathcal{H}^∞ two-degree-of-freedom internal model-based controller and its application to a glass tube production process. In: Proceedings of the Third European Control Conference, Rome, September 1995, pp. 595–600 (1995)
126. Murad, G., Gu, D.-W., Postlethwaite, I.: Robust internal model control of a binary distillation column. In: IEEE International Conference on Industrial Technology, Shanghai, China, December 1996, pp. 194–198 (1996)
127. Murad, G., Gu, D.-W., Postlethwaite, I.: A Direct Model Reduction Approach for Discrete-Time Non-Minimal State-Space Systems. Internal Report 94-23, University of Leicester, Leicester, UK, September 1994
128. Nawawi, S.W., Ahmad, M.N., Osman, J.H.S.: Real-time control system for a two-wheeled inverted pendulum mobile robot. In: Fürstner, I. (ed.) Advanced Knowledge Application in Practice, pp. 299–312. Sciyo, Rijeka (2010). Chap. 16
129. Nehari, Z.: On bounded bilinear forms. Ann. Math. **65**, 153–162 (1957)
130. Nesterov, Y., Nemirovski, A.: Interior-Point Polynomial Algorithms in Convex Programming. Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia (1993)
131. Ng, W.Y.: Interactive Multi-Objective Programming as a Framework for Computer-Aided Control System Design. Lecture Notes in Control and Information Sciences. Springer, Berlin (1989)
132. Packard, A., Doyle, J.C.: The complex structured singular value. Automatica **29**, 71–109 (1993)
133. Packard, A., Pandey, P.: Continuity properties of the real/complex structured singular value. IEEE Trans. Autom. Control **AC-38**, 415–428 (1993)

134. Packard, A., Fan, M.K.H., Doyle, J.C.: A power method for the structured singular value. In: Proceedings of the 27th IEEE Conference on Decision and Control, Austin, Texas, December 1988, pp. 2132–2137 (1988)
135. Pao, L.Y., Franklin, G.F.: Time-optimal control of flexible structures. In: Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, HI, December 1990, pp. 2580–2581 (1990)
136. Pappas, T., Laub, A.J., Sandell, N.R.: On the numerical solution of the discrete-time algebraic Riccati equation. *IEEE Trans. Autom. Control* **AC-25**, 631–641 (1980)
137. Park, J., Asada, H.: Dynamic analysis of noncollocated flexible arms and design of torque transmission mechanisms. *J. Dyn. Syst. Meas. Control* **116**, 201–207 (1994)
138. Pernebo, L., Silverman, L.M.: Model reduction via balanced state space representations. *IEEE Trans. Autom. Control* **AC-27**, 382–387 (1982)
139. Postlethwaite, I., Lin, J.L., Gu, D.W.: Robust control of a high purity distillation column using μ -K iteration. In: Proceedings of the 30th IEEE Conference on Decision and Control, Brighton, UK, December 1991
140. Redheffer, R.M.: Remarks on the basis of network theory. *J. Math. Phys.* **28**, 237–258 (1950)
141. Redheffer, R.M.: On a certain linear fractional transformation. *J. Math. Phys.* **39**, 269–286 (1960)
142. Ren, T.-J., Chen, T.-C., Chen, C.-J.: Motion control for a two-wheeled vehicle using a self-tuning PID controller. *Control Eng. Pract.* **16**, 365–375 (2008)
143. Ruan, X.-g., Ren, H.-g., Li, X.-y., Wang, Q.-y.: Dynamic model and analysis of the flexible two-wheeled mobile robot. In: Xiong, C., Liu, H., Huang, Y., Xiong, Y. (eds.) *Intelligent Robotics and Applications. Lecture Notes in Artificial Intelligence*, vol. 5314, pp. 933–942. Springer, Berlin (2008)
144. RWTH-Mindstorms NXT Toolbox for MATLAB. Aachen, Germany (2008). <http://www.mindstorms.rwth-aachen.de>
145. Safonov, M.G.: Stability margins of diagonally perturbed multivariable feedback systems. *IEEE Proc. Part D, Control Theory Appl.* **129**, 251–256 (1982)
146. Safonov, M.G., Chiang, R.Y.: A Schur method for balanced-truncation model reduction. *IEEE Trans. Autom. Control* **AC-34**, 729–733 (1989)
147. Safonov, M.G., Chiang, R.Y., Limebeer, D.J.N.: Hankel model reduction without balancing—a descriptor approach. In: Proceedings of the 26th IEEE Conference on Decision and Control, Los Angeles, CA, December 1987
148. Safonov, M.G., Limebeer, D.J.N., Chiang, R.Y.: Simplifying the \mathcal{H}^∞ theory via loop-shifting, matrix-pencil and descriptor concepts. *Int. J. Control* **50**, 2467–2488 (1989)
149. Samar, R., Postlethwaite, I., Gu, D.-W.: Model reduction with balanced realizations. *Int. J. Control* **62**, 33–64 (1995)
150. Sánchez-Peña, R.S., Sznaiier, M.: *Robust Systems: Theory and Applications*. Wiley, New York (1998)
151. Sefton, J., Glover, K.: Pole-zero cancellations in the general \mathcal{H}_∞ problem with reference to a two block design. *Syst. Control Lett.* **14**, 295–306 (1990)
152. Segway Personal Transporters. Bedford, NH (2012). <http://www.segway.com>
153. Sezginer, R.S., Overton, M.L.: The largest singular value of $e^X A_0 e^{-X}$ is convex on convex sets of commuting matrices. *IEEE Trans. Autom. Control* **AC-35**, 229–230 (1990)
154. Skogestad, S.: Dynamics and control of distillation columns: a tutorial introduction. *Chem. Eng. Res. Des.* **75**, 539–562 (1997)
155. Skogestad, S., Postlethwaite, I.: *Multivariable Feedback Control*, 2nd edn. Wiley, Chichester (2005)
156. Skogestad, S., Morari, M., Doyle, J.C.: Robust control of ill-conditioned plants: high purity distillation. *IEEE Trans. Autom. Control* **33**, 1092–1105 (1988)
157. Sreeram, V., Anderson, B.D.O., Madieviski, A.G.: New results on frequency weighted balanced reduction technique. In: Proceedings of the 1995 American Control Conference, Seattle, WA, June 1995, pp. 4004–4009 (1995)

158. Stoorvogel, A.A.: The \mathcal{H}_∞ Control Problem: A State Space Approach. Prentice Hall, Englewood Cliffs (1992)
159. Sutton, R., Halikias, G., Plummer, A., Wilson, D.: Modelling and \mathcal{H}_∞ control of a single-link flexible manipulator. *Proc. Inst. Mech. Eng., Part I, J. Syst. Control Eng.* **213**, 85–104 (1999)
160. Tang, K.S., Man, K.F., Gu, D.-W.: Structured genetic algorithm for robust \mathcal{H}_∞ control systems design. *IEEE Trans. Ind. Electron.* **43**(5), 575–582 (1996)
161. Tits, A.L., Fan, M.K.H.: On the small- μ theorem. *Automatica* **31**, 1199–1201 (1995)
162. Tombs, M.S., Postlethwaite, I.: Truncated balanced realization of a stable non-minimal state space system. *Int. J. Control* **46**(4), 1319–1330 (1987)
163. Tomizuka, M.: Dealing with periodic disturbances in controls of mechanical systems. *Annu. Rev. Control* **32**, 193–199 (2008)
164. Tsacouridis, V.A., Medrano-Cerda, G.A.: Discrete-time H_∞ control of a triple inverted pendulum with single control input. *IEE Proc., Control Theory Appl.* **146**, 567–577 (1999)
165. Two Rotor Aerodynamical System. User's Manual. Krakow, Poland (2006). <http://www.inteco.com.pl>
166. Valavanis, K.P. (ed.): *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*. Springer, Berlin (2007). ISBN 978-1-4020-6113-4
167. van der Linden, G.-W., Lambrechts, P.F.: H_∞ control of an experimental inverted pendulum with dry friction. *IEEE Control Syst. Mag.* **19**, 44–50 (1993)
168. Varga, A.: Balancing free square-root algorithm for computing singular perturbation approximations. In: *Proceedings of the 30th IEEE Conference on Decision and Control*, Brighton, UK, December 1991, pp. 1062–1065 (1991)
169. Varga, A.: On stochastic balancing related model reduction. In: *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, December 2000, pp. 2385–2390 (2000)
170. Vermeiren, L., Dequidt, A., Guerra, T.M., Rago-Tirmant, H., Parent, M.: Modeling, control and experimental verification on a two-wheeled vehicle with free inclination: an urban transportation system. *Control Eng. Pract.* **19**, 744–756 (2011)
171. Vidyasagar, M., Kimura, H.: Robust controllers for uncertain linear multivariable systems. *Automatica* **22**, 85–94 (1986)
172. Vidyasagar, M., Schneider, H., Francis, B.: Algebraic and topological aspects of feedback stabilization. *IEEE Trans. Autom. Control* **AC-27**, 880–894 (1982)
173. Walker, D.J.: Relationship between three discrete-time \mathcal{H}_∞ algebraic Riccati equation solutions. *Int. J. Control* **52**, 801–809 (1990)
174. Wang, F., Hurst, T., Abramovitch, D., Franklin, G.: Disk drive pivot nonlinearity modelling Part II: Time domain. In: *Proceedings of the 1994 American Control Conference*, Baltimore, MD, June 1994, pp. 2604–2607 (1994)
175. Wang, G., Sreeram, V., Liu, W.Q.: A new frequency-weighted balanced truncation method and an error bound. *IEEE Trans. Autom. Control* **44**, 1734–1737 (1999)
176. Whidborne, J.F., Liu, G.P.: *Critical Control Systems*. Research Studies, Taunton (1993)
177. Whidborne, J.F., Murad, G., Gu, D.-W., Postlethwaite, I.: Robust control of an unknown plant—the IFAC 93 benchmark. *Int. J. Control* **61**, 589–640 (1994)
178. Whidborne, J.F., Postlethwaite, I., Gu, D.-W.: Robust controller design using \mathcal{H}^∞ loop-shaping and the method of inequalities. *IEEE Trans. Control Syst. Technol.* **2**, 455–461 (1994)
179. Whidborne, J.F., Gu, D.-W., Postlethwaite, I.: Algorithms for solving the method of inequalities—a comparative study. In: *Proceedings of the 1995 American Control Conference*, June 1995
180. Whidborne, J.F., Postlethwaite, I., Gu, D.-W.: Multiobjective control system design—a mixed optimization approach. In: Agarwal, R.P. (ed.) *Recent Trends in Optimization Theory and Applications*. World Scientific Series in Applicable Analysis, vol. 5, pp. 467–482. World Scientific, Singapore (1995)

181. Whidborne, J.F., Postlethwaite, I., Gu, D.-W.: Robust control of a paper machine. *Control Eng. Pract.* **3**, 1475–1478 (1995)
182. White, W.N., Fales, R.C.: Control of a double inverted pendulum with hydraulic actuation: a case study. In: *Proceedings of the 1999 American Control Conference*, San Diego, CA, June 1999, pp. 495–499 (1999)
183. White, M.T., Tomizuka, M., Smith, C.: Rejection of disk drive vibration and shock disturbances with a disturbance observer. In: *Proceedings of the 1999 American Control Conference*, San Diego, CA, June 1999, pp. 4127–4131 (1999)
184. Willems, J.C.: Least stationary optimal control and the algebraic Riccati equation. *IEEE Trans. Autom. Control* **AC-16**(6), 621–634 (1971)
185. Willems, J.C.: Dissipative dynamical systems I: General theory. II: Linear systems with quadratic supply rates. *Arch. Ration. Mech. Anal.* **45**, 321–334 (1972)
186. Wong, C.-C., Wang, H.-Y., Lim, S.-A., Cheng, C.T.: Fuzzy controller designed by GA for two-wheeled mobile robots. *Int. J. Fuzzy Syst.* **9**, 22–30 (2007)
187. Yamakita, M., Hoshino, T., Furuta, K.: Control practice using pendulum. In: *Proceedings of the 1999 American Control Conference*, San Diego, CA, June 1999, pp. 490–494 (1999)
188. Yamamoto, Y.: NXTway-GS (Self-Balancing Two-Wheeled Robot) controller design). <http://www.mathworks.com/matlabcentral/fileexchange/19147-nxtway-gs-self-balancing-two-wheeled-robot-controller-design>
189. Youla, D.C., Jabr, H.A., Lu, C.N.: Single-loop feedback stabilization of linear multivariable dynamical plants. *Automatica* **10**, 159–173 (1974)
190. Youla, D.C., Jabr, H.A., Bongiorno, J.J.: Modern Weiner–Hopf design of optimal controllers, Part II: The multivariable case. *IEEE Trans. Autom. Control* **AC-21**, 319–338 (1976)
191. Zakian, V., Al-Naib, U.: Design of dynamical and control systems by the method of inequalities. *Proc. IEE Control Sci.* **120**, 1421–1427 (1973)
192. Zames, G.: Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms and approximate inverses. *IEEE Trans. Autom. Control* **AC-26**, 301–320 (1981)
193. Zames, G., Francis, B.A.: Feedback, minimax sensitivity, and optimal robustness. *IEEE Trans. Autom. Control* **AC-28**, 585–600 (1983)
194. Zeiger, H.P., McEwen, A.J.: Approximate linear realization of given dimension via Ho’s algorithm. *IEEE Trans. Autom. Control* **AC-19**, 153 (1974)
195. Zhou, K., Doyle, J.C.: *Essentials of Robust Control*. Prentice Hall, Upper Saddle River (1998)
196. Zhou, K., Doyle, J.C., Glover, K.: *Robust and Optimal Control*. Prentice Hall, Upper Saddle River (1995)

This page intentionally left blank

Index

Symbols

M – Δ configuration, 27
 \mathcal{H}_∞ gain-scheduled controllers, 236
 \mathcal{H}_∞ loop-shaping design procedure, 53
 \mathcal{H}_∞ mixed sensitivity design, 179
 \mathcal{H}_∞ suboptimal problem, 34
 ∞ -norm of a system, 11
 μ – K iteration method, 70
 μ -synthesis, 204
 μ -synthesis method, 69
 2-degree-of-freedom control scheme, 33

A

Absolute-error approximation methods, 74
 Additive perturbation, 13
 Admissible point, 62
 Admissible set, 62
 Affine parameter-dependent systems, 223
 Algebraic Riccati equation, 35
 Assumed-modes method, 368
 Asymptotic stability, 6
 Asymptotically stable system, 94

B

Balanced Realization Algorithm, 75
 Balanced residualization, 76
 Balanced Stochastic Truncation method, 84
 Balanced system, 75
 Balanced truncation method, 75
 Bezout identity, 8
 bode, 127
 bodemag, 127
 Bottom product, 329
 Bounded-Input–Bounded-Output stability, 6
 Bounded-real lemma, 97

C

Central controller, 38
 Central \mathcal{H}_∞ suboptimal controller, 44
 Cholesky factors, 37
 Clamping hub, 368
 Closed-loop system, 4
 Complementary sensitivity function, 26
 Condensor, 329
 Continuous-time system, 5
 Control-system, 4
 Controllability Gramian, 75
 Coprime factorization, 7
 Coprime transfer functions, 8

D

D – K iteration, 69
 Descriptor systems, 221
 Diagonal scaling problem, 30
 Discrete Lyapunov equations, 80
 Discrete-time \mathcal{H}_∞ case, 45
 Discrete-time Riccati equation, 46
 Discrete-time system, 5
 Disk sectors, 250
 Disk tracks, 250
 Distillate product, 328
 Distillation column, 328
 Disturbance, 26
 Disturbance attenuation, 26
 dksyn, 205
 dtrend, 450
 Dualization Lemma, 103

E

Elastic deflection, 368
 Embedded Coder Robot NXT, 435
 Embedded servo, 251
 Error signal, 26

F

Fast part, 16
 feedback, 125
 Feedback system, 4
 Feedforward compensator, 54
 Feedforward path, 33, 53
 Fictitious output vector, 17
 Finsler's Lemma, 102
 Flexible-link manipulator, 368
 Fractional Balanced Truncation method, 82
 Fractional Singular Perturbation
 Approximation method, 82
 frd, 107
 Frequency-Weighted approximation methods,
 86
 Frequency-Weighted Balanced Truncation, 88
 Frequency-Weighted Moduli Truncation
 Method, 89
 Frequency-Weighted Singular Perturbation
 Approximation, 89
 Full uncertainty blocks, 21
 Furuta pendulum, 325

G

Gain scheduling, 234
 Generalized plant, 32
 gridureal, 129

H

Hankel approximation problem, 52
 Hankel norm, 51, 77
 Hankel singular value, 75
 Hankel-norm approximation, 77
 Hard Disk Drive, 249
 Hard Disk Drive servo control system, 251
 hinfgs, 236
 hinfsyn, 190

I

iconnect, 132
 Ill-conditioned problem, 35
 Ill-posed feedback system, 48
 impulse, 127
 Induced norm, 10
 Input, 4
 Input complementary sensitivity matrix, 116
 Input loop transfer matrix, 116
 Input multiplicative perturbation, 14
 Input sensitivity matrix, 116
 Interconnected system, 32
 Interior-point methods in convex optimization,
 94
 Internal stability, 6
 Inverse additive perturbation, 14

Inverse input multiplicative perturbation, 14
 Inverse output multiplicative perturbation, 15

L

Left coprime factor perturbations, 15
 LEGO® Mindstorms NXT, 435
 lftdata, 131, 148
 Linear Fractional Transformation
 lower, 20
 upper, 19
 Linear Matrix Inequality (LMI), 93
 LMI feasible problem, 93
 Loop-Shaping Design, 175
 Loop shaping design procedure, 53
 loopsens, 116
 loopsyn, 175
 Low-order controller, 73
 Lower bound on μ , 30
 LPV systems, 221
 LQG methods, 3
 LTI models, 107
 ltiss, 223
 ltisys, 222
 LV-configuration of the distillation column,
 331
 Lyapunov equations, 75

M

Mass-damper-spring system, 121, 222
 Measurement noise, 26
 Method of inequalities, 61
 MIMO system, 4
 Minimal system, 5
 Mixed optimization loop-shaping design
 method, 60
 Mixed sensitivity problem, 31
 mixsyn, 179
 Model with additive uncertainty, 135
 Model with multiplicative uncertainty, 138
 Multivariable system, 4
 mussv, 148

N

Natural frequencies, 370
 Noise rejection, 26
 Nominal model, 13
 Nominal parameter value, 17
 Nominal performance, 67
 Nominal stability, 67
 Noncollocated controller structure, 378
 Nonrepeatable runout, 251
 Norm of a signal, 9
 Normalized system, 38

NXTway-GS, 435

nyquist, 127

O

Observability Gramian, 75

Open-loop system, 4

Order reduction, 74

Orthonormal matrix, 78

Output, 4

Output complementary sensitivity matrix, 116

Output loop transfer matrix, 116

Output multiplicative perturbation, 14

Output sensitivity matrix, 116

P

Parameter box, 224

Parameter-dependent systems, 221

Parametric uncertainty, 18

pdlstab, 229

pdsimul, 232

Performance margin, 156

Performance specifications, 26

Platters, 249

Polytopic system, 227

Position bursts, 251

Position Error Signal, 251

Postcompensator, 53

Power signal, 10

Precompensator, 53

Projection Lemma, 101

psys, 225

pvec, 224

Q

Quadratically stable system, 229

quadstab, 229

R

Read/write heads, 249

Reboiler, 329

Reference input, 26

Relative-error approximation methods, 84

Repeatable runout, 251

Resonant modes, 252

Right coprime factor perturbations, 15

Robust control system, 23

Robust performance, 67

Robust stability, 67

Robust stabilization conditions, 25

Robustly stable system, 23

robustperf, 157

robuststab, 146

Runout, 251

S

S-procedure, 102

Scalar uncertainty blocks, 21

Scaled model, 331

Scaling matrix

input, 331

output, 331

Schur Complement Formula, 98

sconnect, 241

Seeking mode, 251

Segway® Personal Transporter, 453

Sensitivity function, 26

sigma, 113

sinfo, 223

Singular perturbation approximation, 76

Singular value decomposition, 40

SISO system, 4

sleft, 237

Sliders, 250

Slow part, 16

Small-Gain Theorem, 23

Space eigenfunction, 369

Spectral radius of a matrix, 28

splot, 223

ss, 107

Stability, 5

Stability margin, 28

Stabilizing controller, 23

step, 127

Structured singular value, 27

Structured uncertainty, 18

Suboptimal discrete-time loop-shaping

controller, 56

sysic, 133

SYSTEM matrix, 222

System norms, 10

T

tf, 107, 125

Time-invariant system, 5

Track following mode, 251

Trade-off between nominal performance and robustness of the closed-loop system, 325

Transfer function matrix, 5

Triple inverted pendulum, 292

Twin-Rotor Aerodynamic System (TRAS), 401

Two-Wheeled Self-Balancing Robot, 435

U

ucomplex, 130

ucomplexm, 130

ufrd, 130

ultidyn, 134
umat, 130
Unmodeled dynamics, 13, 17, 140
Unstructured uncertainty, 13
Upper bound on μ , 30
ureal, 123
usample, 127
uss, 124
usubs, 127

V

Vertex controllers, 235
Vertex system, 227
Voice Coil Motor, 250

W

wcgain, 165
Weighting functions, 26
Worst-case gain, 165

X

X-Riccati equation, 47

Y

Youla Parameterization Theorem, 8

Z

Z-Riccati equation, 47
zpk, 107