

Advanced Textbooks in Control and Signal Processing

Eugene Lavretsky
Kevin A. Wise

Robust and Adaptive Control

With Aerospace Applications

Second Edition



Advanced Textbooks in Control and Signal Processing

Series Editors

Michael J. Grimble, Electronic and Electrical Engineering Department, University of Strathclyde, Glasgow, UK

Lorenzo Marconi , Department of Electrical, Electronic, and Information Engineering, Università di Bologna, Bologna, Italy

Advanced Textbooks in Control and Signal Processing is a series designed to present ideas in these disciplines in a manner suitable for students and tutors involved in advanced-undergraduate and graduate courses. The subject matter may be theoretical or applied but must be suitable for learning at this level – these books do not often contain the latest cutting-edge research but take more of a long view of a subject as a whole and deal with general areas rather than specific niches.

Volumes in this series are designed to be adopted as classroom textbooks and are often augmented with accessory material available from the Web. This can be anything from MATLAB® files to pdf slides to be projected in class as teaching aids. Solutions manual pdfs available for download by class instructors are common additions and books are peer reviewed with view to the requirements of classroom tuition.

Titles in this series should be between 200 and 500 pages in length and should contain pedagogical material to assist the student: problems (preferably with solutions in a separate pdf); examples; learning plans; case studies; thorough background reference lists; etc.

Authors wishing to submit a proposal for this series should contact the

Series Editors

Professor **Michael J. Grimble** Department of Electronic and Electrical Engineering, Royal College Building, 204 George Street, Glasgow G1 1XW, United Kingdom

e-mail: m.j.grimble@strath.ac.uk

Professor **Lorenzo Marconi** of the Università di Bologna, Department of Electrical, Electronic, and Information Engineering “Guglielmo Marconi”, Via Zamboni 33, 40126 Bologna, Italy

e-mail: lorenzo.marconi@unibo.it

or the

In-house Editor

Mr. **Oliver Jackson** Springer London, 4 Crinan Street, London, N1 9XW, United Kingdom

e-mail: oliver.jackson@springer.com

Indexed by SCOPUS and Engineering Index.

Publishing Ethics:

Researchers should conduct their research from research proposal to publication in line with best practices and codes of conduct of relevant professional bodies and/or national and international regulatory bodies. For more details on individual ethics matters please see:

<https://www.springer.com/gp/authors-editors/journal-author/journal-author-help/pdesk/publishing-ethics/14214>

Eugene Lavretsky · Kevin A. Wise

Robust and Adaptive Control

With Aerospace Applications

Second Edition



Eugene Lavretsky
Los Angeles, CA, USA

Kevin A. Wise
Saint Charles, MO, USA

ISSN 1439-2232 ISSN 2510-3814 (electronic)
Advanced Textbooks in Control and Signal Processing
ISBN 978-3-031-38313-7 ISBN 978-3-031-38314-4 (eBook)
<https://doi.org/10.1007/978-3-031-38314-4>

Mathematics Subject Classification: 37N35

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See <https://www.mathworks.com/trademarks> for a list of additional trademarks.

1st edition: © Springer-Verlag London 2013.

2nd edition: © The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are solely and exclusively licensed 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.

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.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Series Editor's Foreword to the Second Edition

It is a pleasure to welcome the second edition of this text covering the important subject of *Robust and Adaptive Control* to the Springer *Advanced Textbooks in Control and Signal Processing* series. The text is updated but many of the earlier supportive comments by the former joint Series Editor, Prof. Michael Johnson, are still applicable.

The topics of control engineering and signal processing continue to flourish and evolve. In common with general scientific investigations, new ideas, concepts, and interpretations emerge spontaneously, and these are then discussed, used, discarded, or subsumed into the prevailing subject paradigm. Sometimes, these innovative concepts coalesce into a new subdiscipline within the broad subject tapestry of control and signal processing. This preliminary battle between old and new usually takes place at conferences, through the Web and in the relevant journals. After a little more maturity has been acquired by the new concepts 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.

This Springer series is 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 series. An aim of the series is to create a library that covers the main subjects in the control and signal processing fields. It is a series of high-quality books that covers some fundamental areas and many more advanced topics in control and signal processing.

We are therefore pleased to have this textbook from Eugene Lavretsky and Kevin Wise, *Robust and Adaptive Control: With Aerospace Applications*, in the series. This textbook deals with the fundamentals of robust and adaptive control and has strong material derived from aerospace applications. The reader will appreciate the industrial context where the aircraft operates across a wide range of flight conditions, so that there are many operating and design points. The aerospace industry usually solves this problem by using many strategically selected control design points and employs gain scheduling in the flight controllers. This is very

convenient for efficient implementation and certification of the system but can limit the performance achievable in some operating regions.

Chapter 1 introduces the basic dynamic equations needed in aircraft modeling and in so doing reveals the complexity of the multivariable system problem. Chapter 2 introduces some of the tools needed in flight control systems analysis and design. Frequency response methods described in Chap. 3 are particularly useful in aircraft systems where sensitivities are important, resonances occur, and disturbances can have known frequency spectra. This chapter also covers perturbations, robustness, and stability issues. Chapter 4 on linear quadratic regulators (LQRs) is important since linear quadratic Gaussian design is one of the accepted modern control methods for this type of aerospace application. It covers the command tracking design problem in some detail, and there are numerous examples.

Chapter 5 is on the other popular linear multivariable control method, namely the state feedback H_∞ optimal control approach. The standard system model description that is normally available in software tools is used. Chapter 6 moves on to linear quadratic output feedback control problems where state observers are needed. The chapter begins by introducing projective control theory, which relates to an LQR solution but is not well known. It also describes the state estimation problem and role of observers in the separation theorem.

The second part of the book moves on from the introductory material and involves robust adaptive control. Chapter 7 is on direct model reference adaptive control (MRAC) where the aircraft is controlled to follow a specified reference model. Model following with adaptive mechanisms can be needed even when robust control methods are employed. Robust control designs can be too conservative, and adaptation can therefore be used to improve performance.

The techniques of Lyapunov stability theory are important tools, and Chap. 8 is devoted to this topic. These tools are needed to provide closed-loop performance guarantees for the adaptive control algorithms. Chapter 9 on state feedback direct MRAC builds on the ideas developed and considers topics like dynamic inversion. The aircraft control examples make the adaptive control ideas more accessible. The introduction of integral control in command tracking controllers is often needed in practical applications, and this is described in Chap. 10.

The operation of robust adaptive control in the presence of parametric uncertainties and bounded noise is considered in Chap. 11. Three methods are introduced to try to improve robustness, and the examples are very informative. Chapter 12 involves another less familiar topic, namely approximation-based adaptive control, where neural networks are introduced. It allows for unstructured uncertainties, such as nonlinear state-dependent functions and bounded time-varying process noise. Chapter 13 discusses the modifications needed for the very difficult problem of nonlinear dynamical systems that also contain uncertainties.

Chapter 14 involves output feedback with observer-based loop transfer recovery (LTR) and adaptive augmentation. The LTR method of improving robustness depends upon asymptotic properties of optimal controls and may not be suitable in some problems but it is easy to understand and try. The examples are again

useful. The section on respecting unstable dynamics is particularly relevant to some aircraft control design problems. The last chapter, 15, on robust and adaptive output feedback control, is for systems with difficult dynamics including both unstable and non-minimum phase systems. The output feedback observer-based control design approach can achieve closed-loop stability and desired tracking performance and stability margins.

The book is very suitable for students, researchers, and engineers. The chapters are supported, for example, by reference lists and sets of exercises for the reader. The authors are very well-known Boeing Technical Fellows and are distinguished award winners. Unusually, they have strong teaching and research records in addition to their extensive industrial experience. This new edition of the textbook is an excellent addition to the *Advanced Textbooks in Control and Signal Processing* series.

April 2023

Michael J. Grimble
University of Strathclyde
Glasgow, Scotland, UK

Preface to the Second Edition

This book was written with the main goal to share our lessons-learned and design insights into the development and analysis of robust and adaptive control systems. Our focus has been on control system design methods that are practical yet have a formal basis for performing their design, analysis, and performance evaluations. Since we published the first edition of our book, many schools have adopted it as a teaching material and companies have started using the book as a reference for their design practices. Not surprisingly, we began to receive comments and suggestions from students, scientists, engineering practitioners, and others on how to improve the book contents. Also, we have had further opportunities to apply these methods in the design and analysis of both piloted and unpiloted aircraft systems, gaining additional insights we would like to share with the engineering community. We are excited about the contents of this second edition and the benefits it will bring to readers.

In Chap. 1, we added a detailed self-contained discussion about the mathematical foundations and the development of a six-degrees-of-freedom (6-DoF) flight simulation environment for a fixed-wing aircraft. This new material is supplemented with Appendix A, where a description of the Aircraft Flight Simulation (aFltSim) software is given. This material will aid aerospace inclined readers, with specific interests in aircraft dynamics and control.

Chapter 2 is new. It presents a self-contained introduction to linear systems and control, with a summary of key concepts. We summarize important time domain and frequency domain analysis methods for linear systems. This chapter focuses on presenting and understanding the gain and phase characteristics of certain transfer functions used in application. Integrators and differentiators, lead and lag filters, notch filters, etc., are all used today to provide specific control actions. These transfer functions are designed to condition output and input signals and are often used to remove senior noise and high frequency signal contents. They represent key elements that are combined with our optimal control systems presented in the book chapters.

Chapter 3 is a modified version of the material previously contained in Chap. 5. We moved this chapter earlier in the text as the frequency domain analysis methods are key and central to all of the control design methods presented in later chapters.

Chapter 4 contains optimal control and servomechanism theory. We combined Chaps. 2 and 3 from the first edition into a signal chapter, adding additional examples to improve understanding of this material.

Chapter 5 on H_∞ state feedback control remains largely unchanged.

Material in Chap. 6 has been significantly modified, and the chapter title is changed to “State Observers and Output Feedback Control”. In this chapter we focus on output feedback control architectures that implement linear quadratic regulator and servomechanism control laws. This chapter now has a self-contained treatment of state observers and observer-based control design using output feedback dynamic controllers. The output feedback full state observer design and analysis method is named observer-based loop transfer recovery (OBLTR). We added the much-needed theoretical background on asymptotic expansions and the squaring-up algorithm used in the observer design. These technologies provide mathematical foundations for the design and analysis methods of the OBLTR controllers. New examples are added to illustrate the OBLTR methodology. This section provides a direct link to combine robust and adaptive controllers in Part II.

Chapters 7–12 remain unaffected.

A design example is added to Chap. 13 to illustrate step-by-step design steps and applicability of the observer-like state feedback model reference control.

In Chap. 14, we added a new stronger result on the closed-loop stability of OBLTR controllers with adaptive augmentation: We prove an asymptotic closed-loop stability for a sufficiently small parameter v , as opposed to a uniform ultimate boundedness. Two new design studies are also added to this chapter. Both examples are intended to provide further insights into the Adaptive-OBLTR control design, analysis, and simulation testing.

Chapter 15 presents an extension of the Adaptive-OBLTR control design to a class of multi-input–multi-output square non-minimum phase uncertain systems with arbitrary relative degree. In this chapter, we added several trade studies to illustrate numerical features of this technology.

As in the original publication, the book second edition continues to emphasize formal practical methods for control systems design, analysis, validation, and verification. We wrote this edition based on our lessons-learned from years of practicing controls engineering in the aerospace industry, while teaching undergraduate and graduate courses in dynamics and controls. Many of these technologies were developed, implemented, and flight verified on several aerial vehicles. Some have been transitioned into production programs and will remain operational during the lifespan of those systems.

We added numerical examples using models and other data that are representative of real systems. Model parameters were modified to avoid potential data sensitivity issues that might be associated with these models. Nevertheless, we hope that the readers will find these systems useful to practice their controls skills, while reading the revised book material.

Prerequisites for understanding the book contents remain the same and include a basic knowledge of control systems. Every section is written to be as self-contained

as possible. However, certain classical subjects in dynamics and automatic control are omitted. For example, it is assumed that readers are familiar with the derivations of Newton's second law for rigid body dynamics. Also, ordinary differential equations and Laplace transforms are prominently used throughout the book second edition.

We wrote this book to mirror our personal learning, professional passion, and a decade-long experience in aerospace industry. From that point of view, we sincerely hope that students, engineers, and scientists will find the book contents useful. This material can be utilized to teach undergraduate and graduate courses in dynamics and control systems.

Solving real control problems is always a team effort. In industry, you never work alone. It takes a multi-disciplinary team to succeed and to ultimately complete a development program. We were simply lucky enough to be at the right place and at the right time and have the required skills to contribute. In that regard, we always felt humbled to stand on "shoulders" of those who came before us and from whom we were able to learn the trade of controls engineering. Toward that end, we would like to thank all of our teachers, colleagues, and friends who have helped and guided us to become better engineers. Our gratitude extends to our students who always kept us on our "toes" when teaching mathematically rigorous courses in dynamics and control. But above all, we are indebted to our families for their never-ending unconditional love and support.

Los Angeles, CA, USA
Saint Louis, MO, USA

Eugene Lavretsky
Kevin A. Wise

Preface to the First Edition

After working in the aerospace industry for close to a quarter of a century, both of us felt strongly about writing this book, with the main goal to share our lessons-learned and design insights into the development and analysis of robust and adaptive control systems. Our focus is on the systems that are practical yet have a formal basis for performing their design, analysis, and performance evaluations. During our professional careers at the Boeing Company, we have had a multitude of opportunities to design and flight test Guidance, Navigation, and Control (GN&C) algorithms for a variety of platforms, ranging from commercial aircraft to fully autonomous experimental aerial vehicles. Over time and after numerous trade studies, we have collected a number of GN&C methods that have performed well on a variety of aircraft systems. So, we decided to write this book and share with the reader our experiences and lessons-learned in the design, analysis, and evaluation of control technologies, with an emphasis on flight systems. The latter is not a prerequisite for understanding the book material, as these methods and design insights apply to all control systems. Aerospace applications and examples presented in this book are rather a motivation to challenges in constructing reliable and numerically efficient control algorithms.

Many parts of this book are based on undergraduate and graduate control courses that we have taught over the years at The Washington University in Saint Louis, The University of Missouri—Rolla, The Southern Illinois University in Edwardsville MO, and at the California Institute of Technology (Caltech). As such, the book material is quite suitable for senior undergraduate and graduate students, as well as for practicing engineers and research scientists, who have had an exposure to basic principles in controls and dynamics, such as an undergraduate-level control course, covering classical methods (Root locus, Bode diagrams, and Nyquist plots). In addition, we assume that the reader has a basic understanding of linear algebra and ordinary differential equations and is familiar with using state-space methods for analysis and numerical modeling of dynamical systems. These are the prerequisites.

Motivated and driven by aerospace applications, the book focuses on systems whose dynamics are continuous. Extensions of these methodologies to discrete and hybrid systems are possible and can be found elsewhere in the vast literature devoted to control of dynamical systems.

Overall, the book is self-contained, while covering theoretical development and practical applications of formal methods in robust and optimal linear control, robust stability analysis, Lyapunov stability theory, and model reference adaptive control (MRAC). Throughout the book, we present detailed simulation examples and case studies to illustrate key design steps and the benefits of applying robust and adaptive control methodologies to transport aircraft and experimental aerial platforms.

There are two major parts in the book. Part I presents robust control design and analysis methods for linear time-invariant systems. Part II focuses on MRAC methods for systems with nonlinear and uncertain dynamics.

Readers will benefit from the two-part distinct structure of this book. Such an arrangement enables a seamless transition from the classical linear control concepts to the state-of-the-art in adaptive systems, while illustrating each design with realistic aerospace applications. Also, the two-part book organization allows us to present self-contained material, covering linear and robust adaptive control techniques for dynamical systems that operate in the presence of uncertainties. Toward that end, we consistently give structured descriptions of both classical and advanced control techniques, key design procedures and guidelines, worked examples, and MATLAB® simulations. Each part ends with a set of educational and challenging exercises that are directly related to the material presented. All these features constitute the books educational value.

Part I begins with an introduction to challenges in control design, analysis, and simulation of aerial vehicles. General aviation background and current trends that lead to the need for more advanced control are discussed. Also presented is a brief survey of control-theoretic methods for existing and future aerial vehicles. The theoretical portion of Part I starts with the introduction of robust and optimal linear control methods for linear systems. Command tracking using linear quadratic regulators (LQR) with integral action is presented. This part also covers two output feedback design methods, such as projective control and linear quadratic Gaussian control with loop transfer recovery (LQG/LTR). These algorithms are employed to develop baseline control architectures for linear systems with known dynamics.

Part II begins with self-contained material on the design and analysis of adaptive state feedback controllers for linear and nonlinear uncertain dynamical systems in continuous-time domain. An overview of Lyapunov stability theory is given, followed by theoretical fundamentals for MRAC systems. Next, approximation properties of artificial neural networks and their applications to the design of direct adaptive systems are introduced, and several approximation-based MRAC methods are discussed. The part proceeds with the development of state feedback adaptive augmentation architectures for robust baseline linear controllers, followed by extensions and modifications to achieve transient performance in adaptive systems, as well as to accommodate output feedback constraints. In this part, we also present adaptive augmentation design methods to combine robust baseline controllers with adaptive feedback.

Throughout the book, we discuss motivations to the design, analysis, and implementation of robust and adaptive controllers, with the aim to address realistic challenges that often arise in the flight control of aerial vehicles and other systems.

Acknowledgments

First and foremost, we would like to thank our families for their love, understanding, and patience. Often, the book writing took us away from our family lives. Thank you for your infinite support in allowing us to complete such an undertaking.

We are indebted to professional colleagues and friends, whom we have had the privilege to closely interact during our professional and social activities, before and while writing this book. We would like to express our gratitude to Anuradha Annaswamy, Anthony Calise, Petros Ioannou, Irene Gregory, Jonathan How, Naira Hovakimyan, Wassim Haddad, Richard Murray, Dennis Bernstein, Gary Balas, Siva Banda, David Doman, Zac Dydek, Travis Gibson, Ross Gadian, Michael Steinhour, Joseph Brinker, Ken Buckholtz, and many others who have helped us better understand and appreciate the ever changing and challenging field of controls and dynamics.

We are grateful to Oliver Jackson and Michael Johnson for their editorial comments and suggestions toward improving readability, style, and clarity of the book material.

Finally, we would like to acknowledge The Boeing Company for giving us opportunities, associated challenges, and responsibilities in leading the development of flight systems for aircraft, helicopters, spacecraft, and other aerial platforms.

Los Angeles, CA, USA
Saint Louis, MO, USA

Eugene Lavretsky
Kevin A. Wise

Contents

Part I Robust Control

1	Introduction	3
1.1	Why Robust and Adaptive Control?	4
1.2	About This Book	5
1.3	Aircraft Flight Dynamics Equations of Motion	6
1.4	High-Fidelity Flight Simulation Environment	15
1.5	Simplified Flight Dynamics for Control Design	29
1.6	Summary	41
1.7	Exercises	41
	References	43
2	Linear Time-Invariant Systems and Control	45
2.1	Model-Based Control Engineering	45
2.2	Control System Design Goals and Objectives	49
2.3	Feedback and Feedforward Control	49
2.4	State-Space Systems	52
2.4.1	Time and Frequency Domain Modeling of State-Space Systems	54
2.4.2	Control-Oriented Models for Linear Time-Invariant Systems	77
2.4.3	State-Space Similarity Transformations	81
2.4.4	Eigenvalues and Eigenvectors	82
2.4.5	Computing the State Transition Matrix	84
2.5	Stability, Controllability, and Observability	87
2.5.1	Stability of LTI Systems	88
2.5.2	Controllability of LTI Systems	89
2.5.3	Observability of LTI Systems	92
2.6	Norms of Vectors and Matrices in Euclidean Spaces	95
2.7	Summary	97
2.8	Exercises	97
	References	101

3 Frequency Domain Analysis	103
3.1 Introduction	103
3.2 Transfer Functions and Transfer Function Matrices	104
3.3 Multivariable Stability Margins	110
3.3.1 Singular Values	111
3.3.2 Multivariable Nyquist Theory	114
3.3.3 Singular Value-Based Stability Margins for MIMO Systems	119
3.4 Control System Robustness Analysis	138
3.4.1 Analysis Models for Uncertain Systems	140
3.4.2 Singular Value Robustness Tests	149
3.4.3 Real Stability Margin	159
3.5 Conclusions	173
3.6 Exercises	174
References	176
4 Optimal Control and Linear Quadratic Regulators	177
4.1 Introduction	177
4.2 Optimal Control and the Hamilton–Jacobi–Bellman Equation	179
4.2.1 The HJB Equation for Nonlinear Systems Affine in Control	185
4.3 Linear Quadratic Regulator (LQR)	195
4.3.1 Infinite-Time LQR Problem	197
4.3.2 Guaranteed Stability Robustness for State Feedback LQR	203
4.3.3 LQR Design and Asymptotic Properties	211
4.4 Command Tracking and Robust Servomechanism Control	214
4.4.1 Servomechanism Control Design Model	216
4.4.2 Servomechanism Model Controllability	220
4.4.3 Servomechanism Control Design	222
4.5 Conclusions	240
4.6 Exercises	241
References	244
5 State Feedback H_∞ Optimal Control	247
5.1 Introduction	247
5.2 Norms for Signals and Systems	249
5.3 Stability and Performance Specifications in the Frequency Domain	254
5.4 Loop Shaping Using Frequency-Dependent Weights	256
5.5 State Feedback H_∞ Optimal Control	259
5.6 Controller Design Using γ -Iteration	262
5.7 Conclusions	269
5.8 Exercises	269
References	270

6	Output Feedback Control and State Observers	271
6.1	Output Feedback Using Projective Controls	272
6.2	Full-Order State Observers for Linear Time-Invariant Systems	288
6.2.1	The Separation Principle	291
6.2.2	Observer-Based Optimal Servomechanism Design	293
6.2.3	Asymptotic Properties of the Algebraic Riccati Equation	297
6.2.4	The Squaring-Up Method	312
6.3	Observer-Based Control with Loop Transfer Recovery	324
6.3.1	OBLTR Design Process and Examples	330
6.4	Summary	351
6.5	Exercises	351
	References	354

Part II Robust Adaptive Control

7	Direct Model Reference Adaptive Control: Motivation and Introduction	359
7.1	Model Reference Control: Motivational Example	359
7.2	Introduction to Direct Model Reference Adaptive Control	363
7.3	Direct Model Reference Adaptive Control of Scalar Linear Systems with Parametric Uncertainties	368
7.4	Historical Roots and Foundations of Model Reference Adaptive Control	370
7.5	Exercises	370
	References	371
8	Lyapunov Stability of Motion	373
8.1	Dynamical Systems	373
8.2	Existence and Uniqueness of Solutions	375
8.3	System Equilibrium	382
8.4	Lyapunov Stability Definitions	384
8.5	Lyapunov Stability Theorems	389
8.6	Uniform Ultimate Boundedness	397
8.7	Barbalat's Lemma	404
8.8	Summary and Historical Remarks	409
8.9	Exercises	409
	References	411
9	State Feedback Direct Model Reference Adaptive Control	413
9.1	Introduction	413
9.2	Command Tracking	415
9.3	Direct MRAC Design for Scalar Systems	416
9.4	Dynamic Inversion MRAC Design for Scalar Systems	425

9.5	MRAC Design for Multi-Input–Multi-Output Systems	432
9.6	Summary	442
9.7	Exercises	443
	References	443
10	Model Reference Adaptive Control with Integral Feedback Connections	445
10.1	Introduction	445
10.2	Control Design	447
10.3	MRAC Augmentation of an Optimal Baseline Controller	456
10.4	Summary	467
10.5	Exercises	467
	References	468
11	Robust Adaptive Control	469
11.1	MRAC Design in the Presence of Bounded Disturbances	469
11.2	MRAC Design Modifications for Robustness	472
11.2.1	The Dead-Zone Modification	472
11.2.2	The σ -Modification	477
11.2.3	The e -Modification	480
11.3	The Projection Operator	482
11.4	Projection-Based MRAC Design	491
11.5	Summary and Discussion	503
11.6	Exercises	504
	References	506
12	Approximation-Based Adaptive Control	507
12.1	Motivation	507
12.2	Basic Definitions	509
12.3	Approximation Properties of Feedforward Neural Networks	512
12.4	Adaptive Control with State Limiting Constraints	514
12.5	Summary	535
12.6	Exercises	536
	References	537
13	Adaptive Control with Improved Transient Dynamics	539
13.1	Motivation	540
13.2	Asymptotic Orders and Singular Perturbations	547
13.3	Asymptotic Properties of the Algebraic Riccati Equation	552
13.4	System Dynamics and Control Problem Formulation	559
13.5	Observer-Like Model Reference Adaptive Control	561
13.6	Transient Dynamics Analysis	566
13.7	Summary	578
13.8	Exercises	578
	References	579

14 Output Feedback Servomechanism with Observer-Based Loop	
Transfer Recovery and Adaptive Augmentation	581
14.1 Introduction	582
14.2 Optimal Control with “Cheap” Input and “Expensive” Output	583
14.3 Optimal Cost Asymptotic Analysis	584
14.4 ARE Asymptotic Analysis	585
14.5 Adaptive Output Feedback Design and Analysis	587
14.6 Adaptive Flight Control of a Flexible Transport Aircraft	597
14.7 Design Case Study: Control of the “Respect the Unstable” Dynamics	609
14.8 Design Case Study: (OBLTR + Adaptive) Flight Control of Aircraft MIMO Roll–Yaw Dynamics	622
14.9 Conclusions	634
14.10 Exercises	635
References	636
15 Robust and Adaptive Output Feedback Control for Square Non-Minimum Phase Systems	639
15.1 Introduction	640
15.2 Problem Motivation	641
15.3 The Squaring-Up Design for Non-Minimum Phase Systems with Arbitrary Relative Degree	646
15.4 Observer-Based Loop Transfer Recovery (OBLTR) Servo-Controller for Square Systems	655
15.5 Loop Transfer Recovery and OBLTR Stability Margins for Square Systems	661
15.6 OBLTR Adaptive Augmentation for Square Non-Minimum Phase Systems	670
15.7 Summary	678
15.8 Exercises	679
References	680
Appendix A: Aircraft Flight Simulation (aFltSim) Software	683
Index	705

Part I

Robust Control



Introduction

1

In this book, we introduce formal methods and practical tools from dynamics and control. Our intent is to design, analyze, and evaluate robust and adaptive control algorithms for continuous dynamical systems. We begin our material with a brief introduction and an overview of flight dynamics and control. This material is supplementary. It is not a required reading. However, readers who are interested in reading a concise summary of the aerospace-related problems will find this chapter helpful. After that we continue with the presentation of various control algorithms to regulate linear time-invariant systems, while enforcing adequate stability margins and thus providing necessary robustness properties to the overall design. State feedback optimal servo-controllers for command tracking will be derived and discussed. We also introduce output feedback architectures to design optimal controllers for linear time-invariant systems whose states are not available as measurements. In addition to design, we present numerically efficient analysis methods to evaluate the robustness of the synthesized linear controllers. All these topics are covered in Part I of the book, whereby many examples are given to help the reader isolate and focus on the key points and features of the presented methodologies. Part II covers direct adaptive methods to control both linear and nonlinear systems with uncertainties in their dynamics. The material is self-contained. It includes a brief introduction into Lyapunov's stability, followed by an exposition of model reference adaptive controllers, with their design and analysis methods illustrated on a series of practical examples from aerospace applications. After we

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-38314-4_1.

introduce what is now known as the “classical adaptive control”, we focus our attention on extensions and design modifications for robustness to improve transient performance and utilize output measurements (as oppose to state feedback connections). Also, in this chapter we offer design and analysis methods to combine robust and adaptive controllers into a unified resilient to uncertainties control architecture. Throughout the book, we always strive to motivate and rationalize problem formulations, their solutions, and practicality of the methods.

1.1 Why Robust and Adaptive Control?

Robust control can be thought of as an online policy capable of regulating systems (plants) whose dynamics may contain bounded (in some sense) uncertainties. Such an algorithm would often utilize feedback–feedforward command-state-output connections to generate appropriate control inputs, so that the plant output moves along the prescribed “trajectories”. The main idea here is to design a control system that would work satisfactory for a set of plants, whether linear or nonlinear, while assuming the worst-case conditions on the “unknown unknowns” in the system dynamics.

Discarding ad hoc designs, it would be safe to say that all formal and reliable control methods are model based. We often start with a mathematical model that resembles the process of interest in a selected domain of operation. The model may or may not be accurate in capturing significant and other effects in the process dynamics. In order to overcome potential modeling deficiencies, we seek a robust solution, designed based on the model, yet capable of controlling the real process, and not just the model. We would also want a controller whose performance “gracefully degrades” in the presence of uncertainties. The graceful degradation property is highly desirable, since it becomes the only assurance that the controller would not abruptly break down, if and when the system encounters slightly unprecedented events during its intended operation.

Embedding robustness properties into a control solution should be treated as one of the main criteria in any design. For example, achieving closed-loop stability and tracking performance, while providing adequate stability margins, are the three main goals, especially when dealing with linear system approximations of real processes. In Part I, we will present control design methods and techniques to achieve and enforce the three must-have properties: (1) closed-loop stability; (2) robustness; and (3) tracking performance.

Once a robust control solution is found, one may wonder if its robustness properties and applicability domains can be further extended to cover a wider class of uncertainties in the process dynamics. In Part II of the book, we will attempt to address this problem using formal methods from adaptive systems. We shall employ nonlinear design tools and show that indeed it is possible to construct adaptive controllers that would cope with a certain class of bounded and unbounded state-dependent nonlinear uncertainties that may arise in the plant dynamics during operation.

What is the difference between robust and adaptive controllers? A robust controller is designed to operate under the worst-case condition assumption. Such a controller may use excessive actions to regulate the process. In contrast, an adaptive controller would try to perform an online estimation of the process uncertainty and then produce a control input to anticipate, overcome, or minimize the undesirable deviations from the prescribed closed-loop plant behavior. In addition to their adaptive properties, these controllers can be constructed to “learn” and remember. Learning refers to remembering/recognizing certain patterns, and acting based on prior knowledge or “memory”. A tracking error integrator in the feedback loop is a simple example of a learning controller. It accumulates and integrates regulation errors based on previous and current data. We present adaptive controllers that can be viewed as nonlinear extensions of linear feedback integrators. In other words, adaptive loops form their output by integrating nonlinear functions of the system tracking errors.

We would like to emphasize that adaptive control is not the ultimate solution for all problems. This concept represents merely another formal method to design controllers for a wide class of process uncertainties and with performance guarantees.

Our professional experience comes from the design of robust and adaptive flight controllers for a variety of airborne platforms. Most of them are in operation today. Over the years, we have found that it is not robust versus adaptive, but rather a combination of both controllers that works best, in the sense of maintaining closed-loop stability, enforcing robustness to uncertainties, and delivering the desired closed-loop command tracking performance in the presence of unanticipated events. In essence, the (Robust + Adaptive) architecture combination is our “secret” control design recipe that we would like to share with the reader.

1.2 About This Book

This book is written to provide a self-contained introduction to linear robust control methods, followed by an exploration of adaptive systems. Part I is solely devoted to robust control methods for linear time-invariant continuous systems. This part can be taught in a semester-long course to students who have had a basic introduction to control systems. Part II covers a series of topics in adaptive control in a progressive complexity, starting with the detailed introduction to model reference adaptive controllers for linear systems and ending with the adaptive output feedback control methods for a class of nonlinear uncertain dynamics. The mathematical prerequisites for this part consist of basic concepts in linear algebra and ordinary differential equations. Prior to adaptive control, we give an introduction and an overview of the Lyapunov stability theory, which becomes the essential tool for the development of all design and analysis methods in this part of the book. The contents here can be covered within a semester. It is also possible to condense the material for use in a 9- to 10-week graduate-level quarter-long course.

The two parts of the book can be combined in a single course arrangement, whereby selected linear control methods from Part I are taught, followed by a subset of adaptive control techniques. For example, one may elect to start with linear optimal control and then discuss methods to combine robust and adaptive controllers into a single system capable of mitigating a wide range of uncertainties.

Finally, the book can serve as an ample reference for research scientists and control practitioners, who are interested in the development and application of robust linear and/or adaptive control methods to control a wide variety of systems, including but not limited to those from aerospace.

1.3 Aircraft Flight Dynamics Equations of Motion

This chapter presents fundamentals in modeling an aircraft as a rigid body moving in three-dimensional Cartesian coordinates. The aircraft dynamics are defined with respect to the vehicle body-fixed coordinate frame of reference (the body axes) which in turn undergoes translational and rotational displacements relative to the Earth-fixed inertial reference frame (the Earth axes). Detailed derivations can be found in [1–4].

We define both coordinate systems as right-hand orthogonal. In aerospace applications, the Earth-fixed axis system is often referred to as the North–East–Down (NED) frame of reference. We further assume that the NED frame is inertial, and therefore Newton’s laws of motion are valid.

In the NED axis system, the ($X_I - Y_I$) plane is parallel to the Earth’s surface while Z_I points down toward the Earth’s center and parallel to the gravity vector \bar{g} . The NED origin 0 is fixed at an arbitrary location on the Earth’s surface.

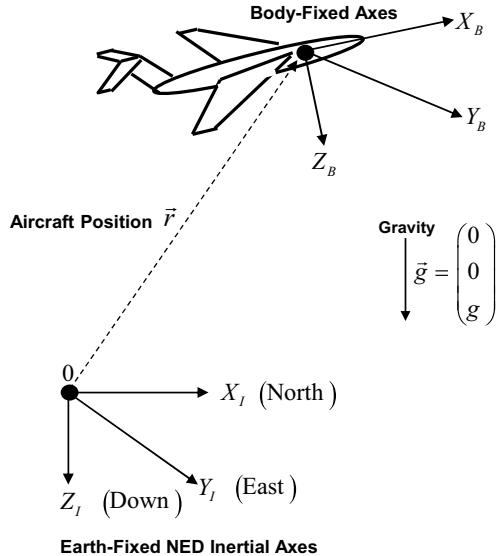
The origin of the body axis system is fixed at the aircraft center of gravity (CG), X_B points forward along the vehicle fuselage line, Y_B is positive out the right wing, and Z_B direction is positive downward out the bottom of the vehicle.

Throughout the book, we ignore Earth curvature effects on atmospheric flight dynamics and assume that gravity is uniformly constant. This constitutes the Flat-Earth assumption which is a reasonable supposition for modeling atmospheric flight dynamics of an aircraft on a short-to-medium time interval [1, 2]. It also simplifies aircraft equations of motion, allows to gain insights into the vehicle primary modes, and facilitates flight control development.

Rigid Aircraft Equations of Motion

Aircraft flight dynamics can be accurately represented and derived based on the classical Newton–Euler formulations for the rigid body equations of motion [1], written with respect to the Earth-fixed inertial coordinate frame of reference (Fig. 1.1) and expressed in the vehicle body-fixed coordinate system. In this context, the aircraft is viewed as a single rigid body moving through the three-dimensional Cartesian aerospace. The Newton–Euler equations describe the vehicle translational and rotational components of the overall motion in the

Fig. 1.1 Earth-fixed and body-fixed reference frames



Earth-fixed inertial coordinates.

$$\frac{d}{dt} (m \vec{V})_I = (\vec{F} + m \vec{g})_I, \quad \frac{d}{dt} (J \vec{\omega})_I = (\vec{M})_I \quad (1.1)$$

These relations define time-dependent changes in the aircraft momentum $(m \vec{V})_I$ and its moment of momentum $(J \vec{\omega})_I$, subject to the total external force $(\vec{F} + m \vec{g})_I$ and the total moment $(\vec{M})_I$ acting on the vehicle. The index “I” implies that the time derivative operator $\frac{d}{dt}$ is applied to the total aircraft velocity vector \vec{V} and to the vehicle angular velocity vector $\vec{\omega}$ in the inertial frame of reference. The gravity vector is \vec{g} , the aircraft mass is m , and the inertia tensor (3×3) matrix is J .

For the sake of clarity, we assume that \vec{g} , m , and J are constant. Rewriting (1.1) in the aircraft body-fixed coordinates with the frame origin attached at the vehicle CG results in the vehicle six-degrees-of-freedom (6-DoF) system of ordinary differential equations.

$$\begin{aligned}
\text{Translational DOF : } m \underbrace{\begin{pmatrix} \dot{u}_b \\ \dot{v}_b \\ \dot{w}_b \end{pmatrix}}_{\vec{V}} &= -m \left[\underbrace{\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix}}_{\vec{\omega}} \times \underbrace{\begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix}}_{\vec{V}} \right] \\
&\quad + \underbrace{\begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix}}_{\vec{F} = \vec{F}_a + \vec{F}_T} + m \underbrace{\|\vec{g}\|}_{g} \underbrace{\begin{pmatrix} -\sin \theta \\ \cos \theta \sin \varphi \\ \cos \theta \cos \varphi \end{pmatrix}}_{\vec{g}} \\
\text{Rotational DOF : } J \underbrace{\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix}}_{\vec{\omega}} &= - \left[\underbrace{\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix}}_{\vec{\omega}} \times J \underbrace{\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix}}_{\vec{\omega}} \right] + \underbrace{\begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix}}_{\text{Total Moment: } \vec{M}}
\end{aligned} \tag{1.2}$$

The aircraft translational dynamics prescribe time-dependent changes of the velocity vector $\vec{V} = (u_b \ v_b \ w_b)^T$ with three body velocity components that represent the vehicle forward, lateral, and vertical speeds along the body-fixed axes, respectively. According to Newton's second law of motion, the vehicle translational dynamics are driven by the sum of all forces that are acting on the body externally. Those include the sum of aerodynamic and propulsive forces $\vec{F} = \vec{F}_a + \vec{F}_T$, as well as the gravitational force $m \vec{g}$, whereby the aircraft mass m and the gravity vector \vec{g} are assumed to be constant.

The aircraft rotational dynamics define evolution of the vehicle angular velocity vector $\vec{\omega} = (p_b \ q_b \ r_b)^T$ whose components are comprised of the body axis roll, pitch, and yaw angular rates, in that order. The system is driven by the total moment vector \vec{M} . It is also assumed that the aircraft inertia tensor matrix $J \in R^{3 \times 3}$ is constant.

Figure 1.2 shows a sketch of a generic aircraft with its body-fixed coordinate frame of reference and conventional control surfaces such as ailerons, elevators, and rudder to control roll, pitch, and yaw angular displacements.

The aircraft 6-DoF equations of motion (1.2) define the vehicle dynamics with respect to the body-fixed reference frame as shown in Fig. 1.2. This system depends on the gravity vector.

$$\vec{g} = g \begin{pmatrix} -\sin \theta \\ \cos \theta \sin \varphi \\ \cos \theta \cos \varphi \end{pmatrix} \tag{1.3}$$

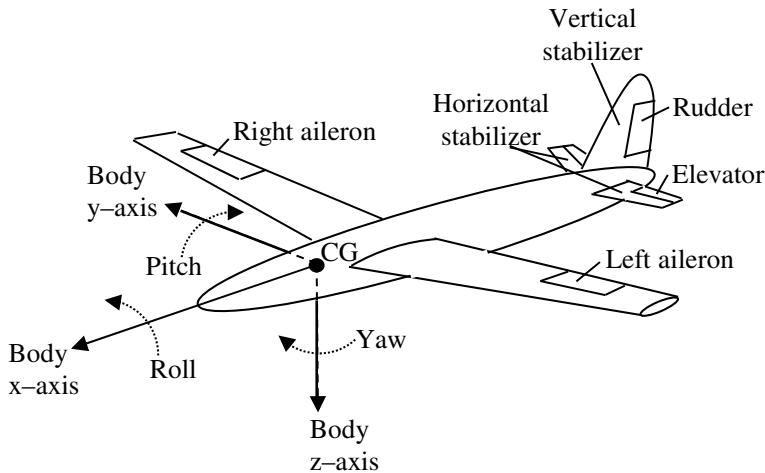


Fig. 1.2 Aircraft body-fixed coordinate frame

The three components of \vec{g} are written using the Euler bank, pitch, and yaw angles φ , θ , ψ . The triplet defines angular orientation of the aircraft body-fixed frame relative to the Earth-fixed inertial NED coordinate system. Euler angles transform NED frame into the body axis system via the three consecutive rotations about the Z_I , Y_I , and X_I axes, in that order. The rotation sequence is yaw–pitch–roll. It defines the three Euler angles ($\psi - \theta - \varphi$) and must be maintained for consistency of the definition (Fig. 1.3).

Next, we can use Euler angles to transform the aircraft velocity vector between the body-fixed axes and the NED inertial system. Based on the definition of the Euler rotations (Fig. 1.3), velocity transformation from NED to the vehicle body axes is written first and according to the predetermined rotation sequence: $\psi \Rightarrow \theta \Rightarrow \varphi$.

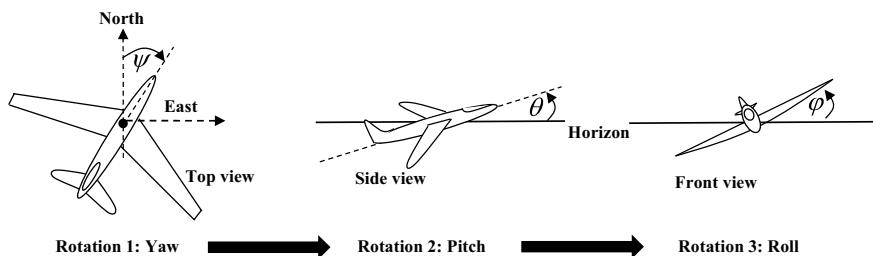


Fig. 1.3 Rotation sequence of Euler angles: yaw \rightarrow pitch \rightarrow roll

$$\begin{aligned}
\begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \\
&= \underbrace{\begin{pmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi & \sin \varphi \cos \theta \\ \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi & \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi & \cos \varphi \cos \theta \end{pmatrix}}_{T_{I \Rightarrow B}} \\
&\times \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = T_{I \Rightarrow B} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \tag{1.4}
\end{aligned}$$

In (1.4), $(\dot{x}, \dot{y}, \dot{z})$ are the north, east, and down velocity components, respectively, and $T_{I \Rightarrow B}$ is the total transformation matrix from body-fixed to inertial axes.

Inverting (1.4) yields transformation of the aircraft velocity vector from the aircraft body-fixed frame into the inertial NED system.

$$\begin{aligned}
\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} &= \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} \\
&= \underbrace{\begin{pmatrix} \cos \theta \cos \psi & \sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi \\ \cos \theta \sin \psi & \sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi & \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi \\ -\sin \theta & \sin \varphi \cos \theta & \cos \varphi \cos \theta \end{pmatrix}}_{T_{B \Rightarrow I}} \\
&\times \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} = T_{B \Rightarrow I} \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} \tag{1.5}
\end{aligned}$$

These are the vehicle navigation equations, and $T_{B \Rightarrow I} = T_{I \Rightarrow B}^{-1}$ is the transformation matrix from NED reference frame to the aircraft body axis system. We can define geometric altitude as the aircraft height above the ground.

$$h = -z \tag{1.6}$$

Then the navigation equations (1.5) can be rewritten.

$$\begin{aligned}
\dot{x} &= u_b \cos \theta \cos \psi + v_b (\sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi) \\
&\quad + w_b (\cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi) \\
\dot{y} &= u_b \cos \theta \sin \psi + v_b (\sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi) \\
&\quad + w_b (\cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi) \\
\dot{h} &= u_b \sin \theta - v_b \sin \varphi \cos \theta - w_b \cos \varphi \cos \theta \tag{1.7}
\end{aligned}$$

The Earth-to-body transformation matrix $T_{I \Rightarrow B}$ applies to any three-dimensional vector in inertial space to convert it into the body-fixed coordinates. It can be shown [4] that this matrix satisfies the navigational strap-down kinematics relation.

$$\dot{T}_{I \Rightarrow B} = - \underbrace{\begin{pmatrix} 0 & -r_b & q_b \\ r_b & 0 & -p_b \\ -q_b & p_b & 0 \end{pmatrix}}_{\Omega} T_{I \Rightarrow B} = -\Omega T_{I \Rightarrow B} \quad (1.8)$$

This is a three-dimensional dynamic system that describes relative rotation between two right-handed frames of references, such as inertial and body axis frames. System (1.8) connects Euler angles (φ, θ, ψ) with body angular rates (p_b, q_b, r_b), where Ω is the cross-product matrix of body angular rates. We immediately note that Ω originates directly from the dynamics of a rotating constant vector $\vec{e} = (\vec{e}_x \ \vec{e}_y \ \vec{e}_z)^T$ in the Cartesian space.

$$\begin{aligned} \dot{\vec{e}} &= - \left[\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \times \begin{pmatrix} \vec{e}_x \\ \vec{e}_y \\ \vec{e}_z \end{pmatrix} \right] = \begin{vmatrix} i & j & k \\ p_b & q_b & r_b \\ \vec{e}_x & \vec{e}_y & \vec{e}_z \end{vmatrix} = \begin{pmatrix} q_b \vec{e}_z - r_b \vec{e}_y \\ r_b \vec{e}_x - p_b \vec{e}_z \\ p_b \vec{e}_y - q_b \vec{e}_x \end{pmatrix} \\ &= \begin{pmatrix} 0 & -r_b & q_b \\ r_b & 0 & -p_b \\ -q_b & p_b & 0 \end{pmatrix} \begin{pmatrix} \vec{e}_x \\ \vec{e}_y \\ \vec{e}_z \end{pmatrix} = -\Omega \vec{e} \end{aligned} \quad (1.9)$$

Applying (1.9) to the inertial triplet of unit vectors $(\vec{i}, \vec{j}, \vec{k})$ gives (1.8). Furthermore, substituting the explicit definition of $T_{I \Rightarrow B}$ from (1.4) into the strap-down kinematics (1.8) and evaluating elements in the second and the third columns gives Euler angles derivatives as functions of the body angular rates.

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \quad (1.10)$$

This is the aircraft rotational kinematics. Together (1.2) and (1.10) completely define the 6-DoF aircraft dynamics in body axes and the 3-DoF vehicle orientation with respect to the NED inertial frame.

Euler rotational kinematics equations (1.10) can be written explicitly.

$$\begin{aligned} \dot{\varphi} &= p_b + \tan \theta (q_b \sin \varphi + r_b \cos \varphi) \\ \dot{\theta} &= q_b \cos \varphi - r_b \sin \varphi \\ \dot{\psi} &= \frac{1}{\cos \theta} (q_b \sin \varphi + r_b \cos \varphi) \end{aligned} \quad (1.11)$$

It is clear that the system is not defined at $\theta = \pm 90^\circ$. This puts a restriction on modeling aircraft pitch angle dynamics to within 90-degree bounds. In that respect, the three-variable Euler orientation expression (1.11) is not applicable for simulating flights around the Earth and modeling of spinning bodies. Other parameterizations exist to overcome angular singularities in (1.11) such as the quaternion four-variable representation [4]. However, Euler kinematics are well suited for modeling aircraft atmospheric flight dynamics with the focus on flight control development [1–4].

Aerodynamic Forces and Moments

The aircraft 6-DoF equations of motion (1.2) are driven by external forces $\vec{F} \in R^3$ and moments $\vec{M} \in R^3$. These quantities may also contain uncertainties due to the vehicle aerodynamics, propulsion, and flight environment, as oppose to the well-defined and known gravity force $m \vec{g} \in R^3$ whose magnitude defines the aircraft total gross weight $W = \|m \vec{g}\| = m g$.

On a typical aircraft development program, aerodynamic forces and moments data originate from a series of wind tunnel testing. In addition, theoretical predictions are often used to augment the vehicle wind tunnel aero database. Later these estimates would be revised if and when flight test data become available.

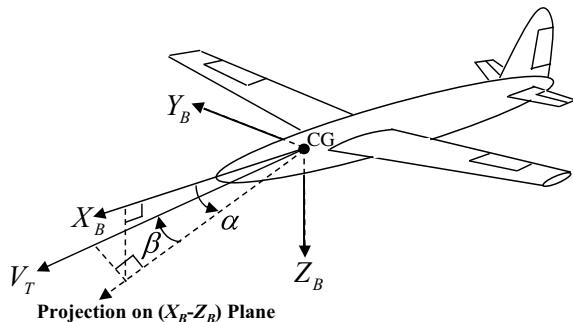
In this section we will define basic knowledge and principles in quantifying an aircraft aero database in order to build a high-fidelity flight simulation testing environment and subsequently to design a flight control system.

We begin by introducing the three aerodynamic parameters that define aircraft motion in the body-fixed axes with respect to atmosphere. The airmass-related parameters are: true airspeed V_T , angle of attack (AOA) α , and angle of sideslip (AOS) β .

$$V_T = \sqrt{u_b^2 + v_b^2 + w_b^2}, \quad \alpha = \arctan\left(\frac{w_b}{u_b}\right), \quad \beta = \arcsin\left(\frac{v_b}{V_T}\right) \quad (1.12)$$

Figure 1.4 shows these important parameters.

Fig. 1.4 True airspeed, angle of attack, and angle of sideslip



Inverting (1.12) we can rewrite the aircraft body axis velocity components in terms of the airmass parameters.

$$u_b = V_T \cos \alpha \cos \beta, \quad v_b = V_T \sin \beta, \quad w_b = V_T \sin \alpha \cos \beta \quad (1.13)$$

Equations (1.13) are often used to compute the aircraft velocity vector based on the three airmass signals that in turn can be measured and/or estimated during flight.

Another two very important quantities that govern the aircraft motion are the vehicle dynamic pressure \overline{Q} and Mach number M .

$$\overline{Q} = \frac{\rho(h)V_T^2}{2}, \quad M = \frac{a(h)}{V_T} \quad (1.14)$$

Dynamic pressure and Mach number calculations are carried out based on the standard day air density $\rho(h)$ and the speed of sound $a(h)$. These two functions depend on the vehicle altitude h . They represent variations in the static atmospheric model [5] over a range of geopotential heights referenced to Earth's mean sea level. Note that under the assumed Flat-Earth conditions, values for geopotential and geometric altitude h are equal to each other.

Excluding gravitational force ($m \vec{g}$), the external forces \vec{F} and moments \vec{M} in the 6-DoF model (1.2) represent the sum of the aircraft aerodynamic and propulsive effects. As such, they depend on the airmass parameters (1.12), dynamic pressure and Mach number (1.14), altitude h , angular rates (p_b, q_b, r_b), and on the vehicle control input vector $\bar{\delta}$ often composed of ailerons, elevators, rudders, and thrust.

$$\vec{F}, \vec{M} = \vec{F}, \vec{M}(h, M, \overline{Q}, V_T, \alpha, \beta, p_b, q_b, r_b, \bar{\delta}) \quad (1.15)$$

Dependencies in (1.15) can be modified to account for unsteady aerodynamic effects due to translational and rotational accelerations [1].

In aerospace applications, it is a standard practice to represent the aerodynamic forces and moments in terms of non-dimensional coefficients that are defined in the vehicle body axes [1–4]

$$\vec{F}_a = \overline{Q} S_w \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix}, \quad \vec{M}_a = \overline{Q} S_w \begin{pmatrix} \bar{b} & 0 & 0 \\ 0 & \bar{c} & 0 \\ 0 & 0 & \bar{b} \end{pmatrix} \begin{pmatrix} C_l \\ C_m \\ C_n \end{pmatrix} \quad (1.16)$$

The factorization (1.16) depends on the wing reference data represented by wing area S_w , wing chord \bar{c} , and wing span \bar{b} . Definitions of the six non-dimensional aerodynamic coefficients are listed in Table 1.1.

Coefficient primary dependencies are well known from practice [1].

$$C_i = C_i(\alpha, \beta, M, \bar{\delta}), \quad i = x, y, z, l, m, n \quad (1.17)$$

Table 1.1 Six aerodynamic coefficients

C_x	Axial force coefficient
C_y	Lateral force coefficient
C_z	Vertical force coefficient
C_l	Rolling moment coefficient
C_m	Pitching moment coefficient

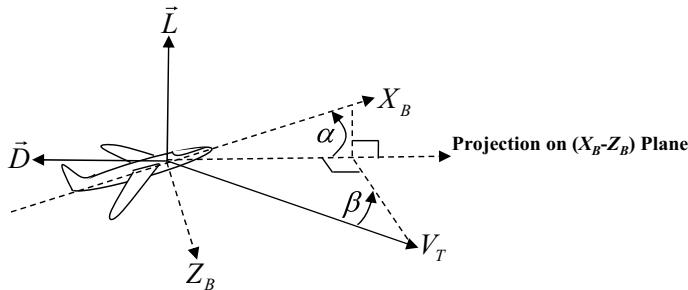


Fig. 1.5 Aerodynamic lift and drag forces

The aerodynamic force \vec{F}_a is often decomposed into the aircraft lift \vec{L} and drag \vec{D} forces (Fig. 1.5).

The lift-drag force decomposition is defined in the body axis X_B-Z_B . In other words, \vec{F}_a is projected onto X_B-Z_B vertical plane and then decomposed. By definition, the lift force \vec{L} is perpendicular to true airspeed V_T and resides in the X_B-Z_B vertical plane. The drag force component \vec{D} is in the same plane but parallel and opposite to the projection of the airspeed vector onto the vertical plane X_B-Z_B . Similar to (1.16), magnitudes for the aerodynamic lift and drag forces are written in terms of non-dimensional coefficients.

$$L = \overline{Q} S_w C_L, \quad D = \overline{Q} S_w C_D \quad (1.18)$$

Based on their definition, we can calculate lift and drag coefficients from axial and vertical terms rotated with α .

$$\begin{aligned} C_L &= -C_z \cos \alpha + C_x \sin \alpha \\ C_D &= -C_z \sin \alpha - C_x \cos \alpha \end{aligned} \quad (1.19)$$

Non-dimensional aerodynamic coefficient data (C_L , C_D , C_M , C_l , C_m , C_n) are vehicle-specific although generic trends in their components can often be observed [1]. During aircraft development, the coefficient values would be gathered in a wind tunnel, estimated using theoretical predictions, and later on augmented using system identification parameters from flight test data [6]. Data collection process

also includes definitions and modeling of propulsion induced forces and moments. The resulting (aerodynamic + propulsion) database provides foundations for aircraft high-fidelity flight simulation development.

1.4 High-Fidelity Flight Simulation Environment

Setting up a sufficiently precise flight simulation software architecture is one of the key steps in a model-based aircraft development framework. In this section, we are going to discuss this topic in details.

Modeling system dynamics with a set of ordinary differential equations (ODEs) starts with the definition of the model states (degrees-of-freedom), external inputs (controls, disturbances), and outputs (measurements and other calculated online signals).

The system dynamics and output are

$$\dot{X} = F(X, Y, U), \quad Y = H(X, U) \quad (1.20)$$

External inputs U may include actual controls that can be manipulated as well as disturbances that may be known or unknown. States X are internal signals. They define degrees-of-freedom for the system dynamics. The number of states is the same as the number of ODEs selected to model the system evolution in time. The outputs Y may include states and other signals that are computable from states and inputs. Some of the output components are fed back into the system while others are saved for further analysis. The triplet (X, Y, U) contains all signals that are required to compute and propagate the system state dynamics one step ahead in time. The functions $F(X, Y, U)$ and $H(X, U)$ can be defined analytically or they might be represented by interpolation of pre-existing database via lookup tables. Their definitions must ensure existence and uniqueness of the system trajectory solutions that arise from the set of ODEs selected to model the process dynamics. In other words, starting from an initial condition X_0 , the system (1.20) must have the unique solution forward in time.

An algorithm to compute the system time-dependent evolution is implemented numerically, and therefore the analytical integrator $\frac{1}{s}$ must be replaced by its discrete equivalent. For example, the forward Euler integration method can be used to implement iterations for the simulation process [4, 7].

One-Step-Ahead Propagation : $X(k+1) = X(k) + \Delta t F(X(k), Y(k), U(k)),$

$$k = 0, 1, \dots, N$$

System Output : $Y(k) = H(X(k), U(k))$

State Initialization : $X(0) = X_0$

(1.21)

As defined in (1.21), the system state $X(k)$ at time t_k is propagated one step ahead to compute the next state value $X(k+1)$ at time $t_{k+1} = t_k + \Delta t$, with all calculations carried out based on the current state, output, and input values ($X(k)$, $Y(k)$, $U(k)$). Starting from an initial state X_0 , iterations (1.21) continue over a predetermined finite time interval $[t_0, t_N]$.

In what follows, we will use the generic framework of (1.21) to develop an accurate flight simulation environment. Toward that end, we combine the aircraft 6-DoF dynamics (1.2), the Euler kinematics (1.10), and the navigation equations (1.5) into a single twelve-degrees-of-freedom dynamical system.

$$\begin{aligned} \begin{pmatrix} \dot{u}_b \\ \dot{v}_b \\ \dot{w}_b \end{pmatrix} &= - \left[\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \times \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} \right] + \frac{1}{m} \left(\bar{Q} S_w \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \right) \\ &\quad + g \begin{pmatrix} -\sin \theta \\ \cos \theta \sin \varphi \\ \cos \theta \cos \varphi \end{pmatrix} \\ \begin{pmatrix} \dot{p}_b \\ \dot{q}_b \\ \dot{r}_b \end{pmatrix} &= -J^{-1} \left[\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \times J \begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \right] + J^{-1} \bar{Q} S_w \begin{pmatrix} \bar{b} C_l \\ \bar{c} C_m \\ \bar{b} C_n \end{pmatrix} + J^{-1} \begin{pmatrix} T_l \\ T_m \\ T_n \end{pmatrix} \\ \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} &= \begin{pmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \\ \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} &= \begin{pmatrix} \cos \theta \cos \psi & \sin \varphi \sin \theta \cos \psi & -\cos \varphi \sin \psi & \cos \varphi \sin \theta \cos \psi & +\sin \varphi \sin \psi \\ \cos \theta \sin \psi & \sin \varphi \sin \theta \sin \psi & +\cos \varphi \cos \psi & \cos \varphi \sin \theta \sin \psi & -\sin \varphi \cos \psi \\ -\sin \theta & \sin \varphi \cos \theta & & & \cos \varphi \cos \theta \end{pmatrix} \\ &\quad \times \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} \end{aligned} \tag{1.22}$$

The system state has twelve components.

$$X = (u_b \ v_b \ w_b \ p_b \ q_b \ r_b \ \varphi \ \theta \ \psi \ x \ y \ z)^T \tag{1.23}$$

The aircraft control inputs are denoted by $\vec{\delta}$. For a conventionally configured aircraft the vehicle control inputs are aileron δ_a , elevator δ_e , rudder δ_r , and thrust δ_T .

$$\vec{\delta} = (\delta_a \ \delta_e \ \delta_r \ \delta_T)^T \tag{1.24}$$

Aircraft may have redundant control surfaces. Tailless vehicles have no rudders. In this case a control allocation algorithm is needed to map actual to virtual inputs (1.24) and vice versa. We will discuss control allocation methods later in the book.

An aircraft may have multiple surfaces that are designated to control the vehicle rolling motion. For example, a conventional aircraft would have at a minimum two ailerons, one on each side, as shown in Fig. 1.1. In that case, a total aileron command can be defined as a half-difference of the right-minus-left or left-minus-right of the aileron deflections. Both definitions are valid. However, they will result in sign-opposite control responses. The same concept is applied to any surfaces that move asymmetrically in order to produce the desired rotational dynamics.

The thrust input δ_T may represent a position or an angle of the vehicle thrust (power) lever mechanical device. It controls the thrust output of the aircraft engines. For pure simulation purposes δ_T can also be defined as a percent of the thrust commanded.

Since aircraft forces and moments depend on altitude, Mach number, dynamic pressure, and three aero parameters (1.15), we need to add all of those as the model outputs. Calculations are conducted based on previously derived relations (1.12), (1.14) that are summarized below.

$$\begin{aligned} V_T &= \sqrt{u_b^2 + v_b^2 + w_b^2}, \quad \alpha = \arctan\left(\frac{w_b}{u_b}\right), \quad \beta = \arcsin\left(\frac{v_b}{V_T}\right) \\ h &= -z_I, \quad \bar{Q} = \frac{\rho(h)V_T^2}{2}, \quad M = \frac{a(h)}{V_T} \end{aligned} \quad (1.25)$$

In (1.25) without a loss of generality, aircraft geometric altitude h is defined as the opposite to the system vertical displacement z_I in the NED reference frame. Then altitude-dependent values for the air density $\rho(h)$ and the speed of sound $a(h)$ can be computed online using standard atmospheric tables [5].

In addition to (1.25), we add three translational acceleration signals (A_x , A_y , A_z) to the simulation output vector.

$$A_x = \frac{F_x}{m g}, \quad A_y = \frac{F_y}{m g}, \quad A_z = \frac{F_z}{m g} \quad (1.26)$$

These signals are collected during flight by accelerometer devices mounted on the vehicle. Accelerometer outputs measure the difference (in $g - s$) between the aircraft total inertial acceleration and the gravitational acceleration, with respect to the aircraft body axes, at the point on the aircraft structure where they are mounted. For the sake of clarity, we assume that the three accelerometers are located at the vehicle CG.

The simulation output vector is defined in (1.27).

$$Y = (X^T \ h \ M \ \bar{Q} \ V_T \ \alpha \ \beta \ A_x \ A_y \ A_z)^T \quad (1.27)$$

We may also add other output signals to (1.27) that might be needed for simulation initialization, linearization, control design, or during a post-simulation analysis. For example, time derivatives of the aero parameters (\dot{V}_T , $\dot{\alpha}$, $\dot{\beta}$) are often saved as outputs. Their definition and calculations follow directly from (1.12).

In flight, an aircraft will frequently encounter environmental disturbances such as slowly varying winds and rapidly changing gusts. Both translational \vec{v}_{wg} and rotational $\vec{\omega}_{wg}$ wind-gust velocities are important to consider.

$$\vec{v}_{wg} = (u_{wg} \ v_{wg} \ w_{wg})^T, \quad \vec{\omega}_{wg} = (p_{wg} \ q_{wg} \ r_{wg})^T \quad (1.28)$$

In (1.28) wind-gust components are expressed in the aircraft body axis system. They directly affect the aerodynamic forces and moments acting on the vehicle since those are created by the aircraft motion with respect to the incoming airmass. In order to incorporate wind-gust effects into a flight simulation, we need to define the aircraft translational and rotational velocity vectors relative to the surrounding air [1].

$$\vec{v}_a = (u_b - u_{wg} \ v_b - v_{wg} \ w_b - w_{wg})^T, \quad \vec{\omega}_a = (p_b - p_{wg} \ q_b - q_{wg} \ r_b - r_{wg})^T \quad (1.29)$$

In this case, using (1.29) the aerodynamic parameters (1.12) must be redefined.

$$V_T = \sqrt{u_a^2 + v_a^2 + w_a^2}, \quad \alpha = \arctan\left(\frac{w_a}{u_a}\right), \quad \beta = \arcsin\left(\frac{v_a}{V_T}\right) \quad (1.30)$$

As a result, the aircraft forces and moments (1.15), the six aerodynamic coefficients (1.17) as well as the vehicle accelerations (1.26) all become dependent on the airmass-related velocities.

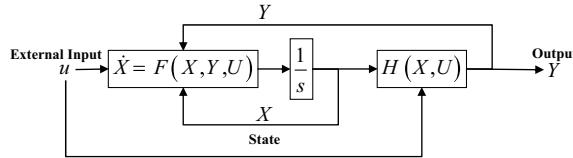
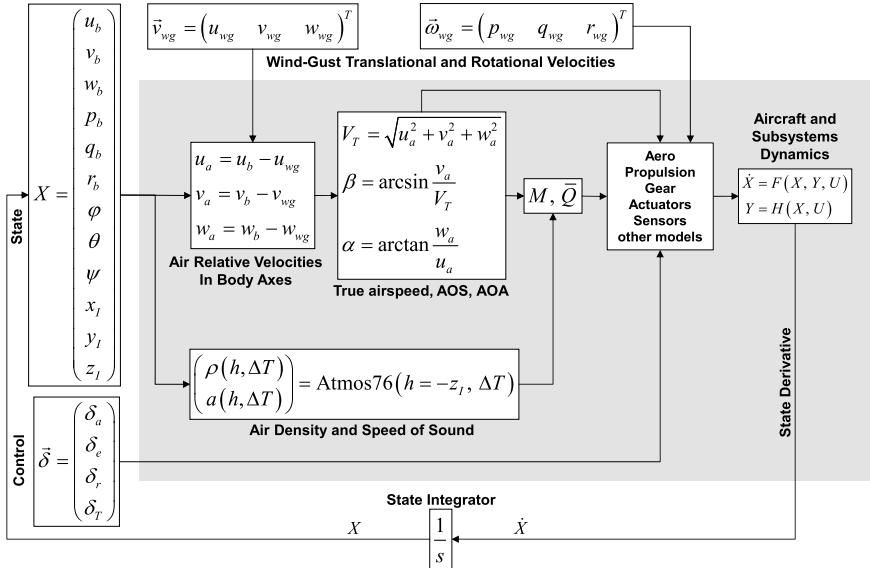
The wind-gust velocity components (1.28) are unknown external inputs. It is important to understand that from a simulation point of view, we need to model wind-gust signals but not rely on them as the online-available data for control usage or for any other real-time operations. On the other hand, the aircraft accelerations (1.26) are known and collected in flight as measurement outputs from the vehicle accelerometers. These signals are the link between airmass-induced aerodynamics and the aircraft motion with respect to the NED inertial frame of reference.

Adding wind-gust velocities (1.28) to the aircraft control inputs $\vec{\delta}$ gives the total vector of external inputs for simulation.

$$U = (\vec{\delta}^T \ \vec{v}_{wg}^T \ \vec{\omega}_{wg}^T)^T \quad (1.31)$$

Internal states X (1.23), system outputs Y (1.27), and external inputs U (1.31) are now well defined and ready to support the simulation block diagram shown in Fig. 1.7.

The diagram flow reflects that of the generic simulation in Fig. 1.6 but it makes the latter become aircraft specific. In addition to previously discussed signals and models, the atmosphere model is shown to depend on a temperature deviation ΔT from standard day to account for non-standard atmospheric changes in the

**Fig. 1.6** Generic simulation block diagram**Fig. 1.7** Aircraft flight simulation block diagram

model outputs. Other models can be added to simulate subsystems such as control actuators and sensors. Formally speaking, every model of a subsystem within the simulation framework has its own set of internal states, and those signals need to be appropriately initialized and managed.

The aircraft dynamics (1.22) can be analyzed in terms of the vehicle longitudinal (pitch) and lateral-directional (roll-yaw) degrees-of-freedom. The longitudinal motion is defined in the vertical pitch plane while the lateral-directional motion prescribes the aircraft horizontal roll-yaw dynamics. The aircraft pitch and roll-yaw states, inputs and outputs are shown below.

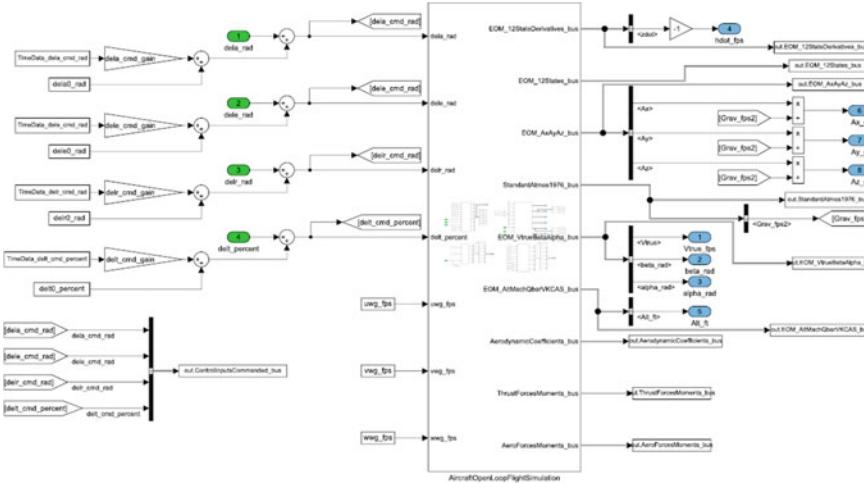


Fig. 1.8 aFltSim top-level block diagram

$$\begin{aligned}
 \text{Pitch States} : x &= (u_b \ w_b \ q_b \ \theta \ z)^T \\
 \text{Pitch Inputs} : u &= (\delta_e \ \delta_T)^T \\
 \text{Pitch Outputs} : y &= (V_T \ \alpha \ \dot{h} \ h \ A_x \ A_z)^T \\
 \\
 \text{Roll-Yaw States} : x &= (v_b \ p_b \ r_b \ \varphi)^T \\
 \text{Roll-Yaw Inputs} : u &= (\delta_a \ \delta_r)^T \\
 \text{Roll-Yaw Outputs} : y &= (\beta \ A_y)^T
 \end{aligned} \tag{1.32}$$

The vehicle horizontal ($X-Y$) inertial positions and other output measurements can be easily added to the list.

In most applications, the 6-DoF aircraft dynamics can be separated into two decoupled subsystems. The decoupling is based on the definition (1.32). We will always explore this useful feature for control design purposes.

Appendix A defines aircraft flight simulation (aFltSim) software implemented within MATLAB®/Simulink® environment. Throughout the book, we will use aFltSim to illustrate main concepts, ideas, and numerical examples. For the sake of completeness, we show Simulink® representation of the top-level simulation block diagram (Fig. 1.8).

This is an open-loop flight simulation of a medium-size fixed-wing aircraft. It is representative of the McDonnell F-4 Phantom fighter jet. The aerodynamics and propulsion data were taken from [8].

The simulation has four primary control inputs (green ovals) and eight main outputs (blue ovals). These signals are used to define trim and linearization data.

Of note, the total aileron deflection δ_a is defined as the half-difference of the right-minus-left aileron surfaces.

$$\delta_a = \frac{(\text{Right Aileron} - \text{Left Aileron})}{2} \quad (1.33)$$

Consequently, a positive aileron command would result in the right aileron moving trailing edge and the left aileron moving trailing edge up. As a result, the aircraft will roll left wing down, which in turn defines negative roll dynamics. This is consistent with the sign convention for the elevator and the rudder inputs, whereby a positive surface deflection produces a negative moment on the vehicle.

Other simulation outputs are added for analysis. Also, three wind-gust velocities (uwg_fps , vwg_fps , wwg_fps) are available as external inputs for simulating the aircraft dynamics in the presence of environmental disturbances. Within aFltSim, signal names are defined to signify the meaning (abbreviated) and the units (e.g., feet, radians) for each parameter. The open-loop aircraft simulation can be driven using the four control inputs (aileron, elevator, rudder, and thrust) as functions of time.

Figure 1.9 shows the aircraft flight simulation core diagram, called from the main top-level architecture.

Note that the core block diagram in Fig. 1.9 is based on and consistent with the generic simulation architecture in Fig. 1.7. Also, all of the simulation input/output signals are defined according to Eqs. (1.23)–(1.31). Further details can be found in Appendix A.

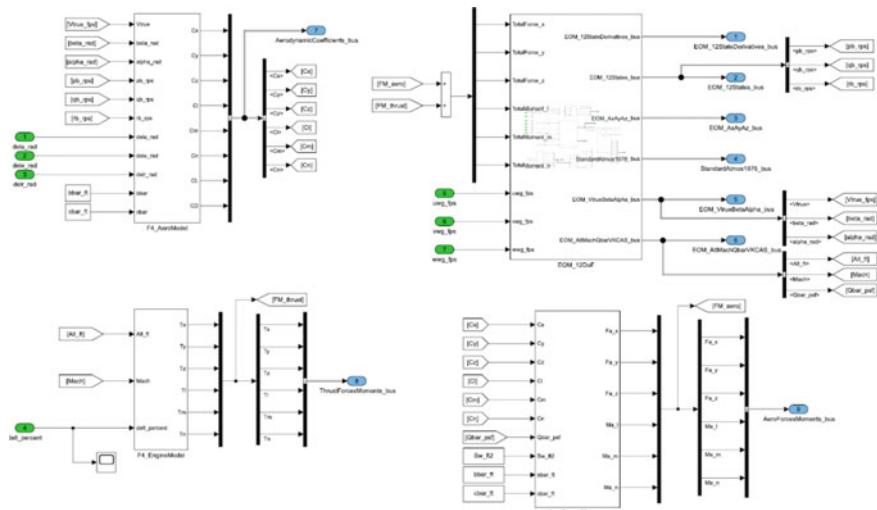


Fig. 1.9 aFltSim core block diagram

Aircraft Trim Conditions

Aircraft flight simulation process begins with initialization of the system internal states and its external inputs. For an aircraft that means finding states and inputs such that the system trajectory starts at an equilibrium which is also called “trim”. In terms of a generic system (1.20), an equilibrium is a set of all state-control pairs $(X_{\text{eq}}, U_{\text{eq}})$ that satisfy the algebraic equations.

$$\begin{aligned}\dot{X}_{\text{eq}} &= F(X_{\text{eq}}, Y_{\text{eq}}, U_{\text{eq}}), \quad \dot{X}_{\text{eq}}, X_{\text{eq}} \in R^n, \quad U_{\text{eq}} \in R^m \\ Y_{\text{eq}} &= H(X_{\text{eq}}, U_{\text{eq}}) \in R^P\end{aligned}\tag{1.34}$$

These are not ODEs. The equilibrium time derivative \dot{X}_{eq} and the output Y_{eq} are given. They represent the desired “targets” to achieve. There are $(n + p)$ nonlinear algebraic equations in (1.34), with $(n + m)$ unknowns $(X_{\text{eq}}, U_{\text{eq}})$. In general, equilibrium calculations are numerical in nature and are performed using an optimization method such as gradient descent with constraints [4, 7]. Within the selected optimization framework, the unknown equilibrium pairs $(X_{\text{eq}}, U_{\text{eq}})$ are calculated iteratively such that the resulting left-hand side values of (1.34) approach the desired targets $(\dot{X}_{\text{eq}}, Y_{\text{eq}})$.

It is important to realize that for the aircraft dynamics (1.22), a trimmed state-control pair $(X_{\text{eq}}, U_{\text{eq}})$ is nonzero. It is computed given a nonzero target state derivative \dot{X}_{eq} and a nonzero target output Y_{eq} . Necessity of a nonzero equilibrium solution directly follows from the aircraft state and output definitions (1.23), (1.27) that include speed-related quantities such as true airspeed, dynamic pressure, Mach number, and inertial velocity components. In order to sustain a trimmed flight trajectory, an aircraft would also require a nonzero angle of attack and a nonzero pitch angle to produce aerodynamic lift force that would counter the vehicle gross weight.

In aircraft dynamics, the most common equilibrium condition is the so-called Wings-Level 1g (WL1g) trim. It defines initial flight conditions for the vehicle to maintain a wings-level flight trajectory on a straight line with a constant airspeed and at a given fixed altitude. In order to do that, the aircraft must be able to generate a vertical force F which is equal and opposite to the gravitational force whose magnitude equals to the vehicle gross weight $m g$. Scaling the required vertical force by the aircraft mass implies that in order to maintain WL1g trim the aircraft must generate 1g acceleration in the vertical direction.

By definition, the WL1g flight trajectory requires finding a set of a “trim-with” set of states and controls such that six accelerations, vertical speed, angle of sideslip, and bank angle are equal to zero, while airspeed and altitude are given and held constant. These are the trim targets. We will also call them “trim-to” variables.

$$\begin{aligned}
 \text{Trim-With : } & (u_b \ v_b \ w_b \ p_b \ q_b \ r_b \ \theta \ \delta_a \ \delta_e \ \delta_r \ \delta_T) \\
 \Downarrow \\
 \text{Trim-To : } & (\dot{u}_b = 0 \ \dot{v}_b = 0 \ \dot{w}_b = 0 \ \dot{p}_b = 0 \ \dot{q}_b = 0 \ \dot{r}_b = 0 \ \dot{\theta} = 0 \ M_0 \ \beta_0 \ \varphi_0 \ h_0)
 \end{aligned} \tag{1.35}$$

Note that in (1.35), four of the trim-to components, Mach number M_0 , angle of sideslip β_0 , bank angle φ_0 , and altitude h_0 , are given and fixed. They are denoted with the 0-subscript. The other components represent the 6-DoF trim-to acceleration target values. For WL1g trim, $\beta_0 = \varphi_0 = 0$. The heading angle ψ_0 does not influence aircraft 6-DoF equations, and thus it can be set to any constant value.

It is not so difficult to argue that for any fixed heading angle ψ_0 , the WL1g 11-dimensional trim-to set and the 11-dimensional trim-with set uniquely define the 16-dimensional equilibrium pair $(X_{\text{eq}}, U_{\text{eq}})$ (1.23), (1.24) and the corresponding 12-dimensional state derivative vector in (1.34). Therefore, we have a well-posed WL1g (11×11) optimization problem. The unique WL1g trim solution would be solved and found numerically via an iterative optimization method of the user choice. The goal of an optimizer is to iteratively find a set of the selected trim-with inputs such that the target trim-to outputs are achieved within the user-specified numerical tolerances. An optimization block diagram can be drawn starting with the aircraft 6-DoF flight simulation illustration (Fig. 1.7) and replacing the simulation integrator function with an optimizer logic.

Example 1.1 aFltSim WL1g Trim Data We set the aircraft gross weight to 38,924 lbs and perform a WL1g trim at 25,000 ft altitude and 0.6 Mach number. The trim

Table 1.2 aFltSim WL1g trimmed state data for Example 1.1

Trimmed state name	Trimmed state value	Trimmed state derivative
u_b (ft/s)	605.75	0
v_b (ft/s)	0	0
w_b (ft/s)	68.112	0
p_b (rad/s)	0	0
q_b (rad/s)	0	0
r_b (rad/s)	0	0
φ (rad)	0	0
θ (rad)	0.11197	0
ψ (rad)	0	0
x (ft)	0	609.57
y (ft)	0	0
z (ft)	- 25,000	0

conditions are computed numerically and are shown in Table 1.2. The units of measurement are expressed in the British Imperial System, that is, linear velocities are in ft/s, angular velocities are in rad/s, angles are in rad, and positions are in ft.

The required trimmed control inputs (rad) are listed in Table 1.3. The three control surfaces are defined in radians, while the thrust input is expressed in percent.

Finally, the eight essential trimmed simulation outputs are given in Table 1.4.

Clearly, the data show that a WL1g trim is achieved at the noted flight conditions. The reader should be able to correlate the trimmed values to the 6-DoF equations that define these signals. ■

Another commonly used aircraft trim configuration is called the steady-heading sideslip (SHSS) flight. This is a trimmed flight trajectory similar to WL1g except the vehicle flies at a constant nonzero sideslip angle $\beta_0 \neq 0$, which needs to be defined within the trim-to set (1.35).

We often consider an aircraft flying in a steady turn while maintaining the desired altitude and constant airspeed. In flight dynamics this trim maneuver is referred to as the Wind-Up Turn (WUT). It is performed by finding trim-to inputs such that trim-with outputs are achieved with zero sideslip angle $\beta_0 = 0$ and a desired nonzero bank angle $\varphi_0 \neq 0$. It can be shown that increasing bank angle during a WUT maneuver leads to an increase of the aircraft angle of attack α such that the corresponding vertical force counters the aircraft weight and as a result the altitude is kept constant during the test. Formal definitions and detailed discussions of other common trimmed flight conditions can be found in Refs. [1–4].

Table 1.3 aFltSim WL1g trimmed control input data for Example 1.1

Trimmed control input name	Trimmed control input value
δ_a (rad)	0
δ_e (rad)	-0.10267
δ_r (rad)	0
δ_{th} (%)	38.425

Table 1.4 aFltSim WL1g trimmed control output data for Example 1.1

Trimmed control output name	Trimmed control output value
V_T (ft/s)	609.57
β (rad)	0
α (rad)	0.11197
\dot{h} (ft/s)	0
$h = \text{Alt}$ (ft)	25,000
A_x (g)	0.11174
A_y (g)	0
A_z (g)	-0.99374

Linearization and Primary Flight Modes

In order to formally analyze the aircraft 6-DoF dynamics (1.22), we need to develop linear time-invariant (LTI) models with respect to selected trim conditions. These LTI systems quantify essential components and mode shapes that drive the open-loop aircraft dynamics near equilibrium. Later on, we would use linear open-loop models to design robust and adaptive flight control systems.

Linearization of the nonlinear 6-DoF flight dynamics (1.22) is often performed numerically. Deriving analytical open-loop linear models is uncommon and impossible for most if not for all practical aircraft applications where aerodynamic and propulsion models as well as other subsystems are represented by nonlinear lookup tables. Aircraft data are derived from theoretical predictions, wind tunnel, and laboratory testing. Resulting database would often have inherently nonlinear trends with respect to flight conditions and vehicle configurations.

Linearization of a generic nonlinear dynamical system relies on Taylor series expansions and numerical differentiation of the system right-hand side. Our preferred numerical differentiation method is the central difference approximation [7].

$$\begin{aligned}
 \text{System} & : \dot{x} = f(x, u), \quad x \in R^n, \quad u \in R^m \\
 \text{Trim} & : \dot{x}_{\text{eq}} = f(x_{\text{eq}}, u_{\text{eq}}) \\
 \text{Linearization} & : \dot{x} = f(x_{\text{eq}} + \Delta x, u_{\text{eq}} + \Delta u) \approx \underbrace{f(x_{\text{eq}}, u_{\text{eq}})}_{\dot{x}_{\text{eq}}} + A \Delta x + B \Delta u \\
 \text{Jacobian w.r.t } x & : A = \frac{\partial f(x_{\text{eq}}, u_{\text{eq}})}{\partial x} = \left[\frac{\partial f_i(x_{\text{eq}}, u_{\text{eq}})}{\partial x_j} \right]_{\substack{i=1:n \\ j=1:n}} \\
 & \approx \left[\frac{f_i(x_{\text{eq} \ 1}, \dots, (x_{\text{eq} \ j} + \Delta x_j), \dots, x_{\text{eq} \ n}, u_{\text{eq}}) - f_i(x_{\text{eq} \ 1}, \dots, (x_{\text{eq} \ j} - \Delta x_j), \dots, x_{\text{eq} \ n}, u_{\text{eq}})}}{2 \Delta x_j} \right] \\
 \text{Jacobian w.r.t } u & : B = \frac{\partial f(x_{\text{eq}}, u_{\text{eq}})}{\partial u} = \left[\frac{\partial f_i(x_{\text{eq}}, u_{\text{eq}})}{\partial u_j} \right]_{\substack{i=1:n \\ j=1:m}} \\
 & \approx \left[\frac{f_i(x_{\text{eq}}, u_{\text{eq} \ 1}, \dots, (u_{\text{eq} \ j} + \Delta u_j), \dots, u_{\text{eq} \ m}) - f_i(x_{\text{eq}}, u_{\text{eq} \ 1}, \dots, (u_{\text{eq} \ j} - \Delta u_j), \dots, u_{\text{eq} \ m})}}{2 \Delta u_j} \right] \quad (1.36)
 \end{aligned}$$

In (1.36), a nonlinear system $\dot{x} = f(x, u)$ is considered, and its trajectory deviations $\Delta \dot{x} = \dot{x} - \dot{x}_{\text{eq}}$ from equilibrium are approximated by the LTI dynamics via Taylor series expansions to the first order [7].

$$\Delta \dot{x} = A \Delta x + B \Delta u \quad (1.37)$$

The resulting Jacobian matrices,

$$A = \frac{\partial f(x_{\text{eq}}, u_{\text{eq}})}{\partial x} \in R^{n \times n}, \quad B = \frac{\partial f(x_{\text{eq}}, u_{\text{eq}})}{\partial u} \in R^{n \times m} \quad (1.38)$$

are computed numerically by the central difference method as indicated in (1.36). Derivative approximations are performed iteratively, one state component at a time, while holding the others at their corresponding equilibrium values.

By definition, the central difference method uses two neighbored points to approximate the function derivative at the center point. It is equivalent to approximating the function at the three points by a parabola and then computing its exact derivative at the center point [7].

Example 1.2 aFltSim Linearization around WL1g Trim Conditions We use trimmed data from Example 1.1 and linearization equations (1.36)–(1.38) to construct linear models of the simulated aircraft dynamics. Linearization process is conducted numerically. As a result, we build a linear time-invariant (LTI) model.

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{1.39}$$

The system state and output dynamics are completely defined by the four matrices (A , B , C , D). When compared to (1.37), we dropped the symbol Δ for brevity. However, the model signals should be understood as incremental with respect to the selected trim conditions. With that in mind in (1.39), $x \in R^9$ is the system state vector, $u \in R^4$ is the control input, and $y \in R^8$ is the system output.

$$\begin{aligned}x &= (u_b \ v_b \ w_b \ p_b \ q_b \ r_b \ \varphi \ \theta \ z)^T \\ u &= (\delta_a \ \delta_e \ \delta_r \ \delta_T)^T \\ y &= (V_T \ \beta \ \alpha \ \dot{h} \ h \ A_x \ A_y \ A_z)^T\end{aligned}$$

The state components are selected as a minimal subset from Table 1.2 to represent the aircraft dynamics as an LTI system. The control inputs are the same as defined in Table 1.3. The model outputs are defined from a control design perspective. These are the signals we will use throughout the book to demonstrate various control design methods.

The linear system A -matrix data are shown in Table 1.5.

Table 1.6 defines the system B -matrix.

The state-to-output C -matrix data are given in Table 1.7.

The system direct control-to-output coefficients (the D -matrix) are listed in Table 1.8.

The system linear data show a clear separation between the aircraft longitudinal and lateral-directional dynamics (1.32). In other words, the longitudinal degrees-of-freedom depend only on itself and on longitudinal control inputs. They are not influenced by the lateral-directional degrees-of-freedom nor by the lateral-directional control inputs. This decoupling can be better seen if the system state is converted into the output via an appropriate state similarity transformation. We will study these techniques later in the book.

Finally, it is worth mentioning that the validity of the LTI dynamics can be verified in the 6-DoF aFltSim environment. We can use sufficiently small control inputs to drive the open-loop simulation and then compare its output to that of the linear model. ■

Table 1.5 aFltSim WL1g A-matrix linear data for Example 1.2

u_b	v_b	w_b	p_b	q_b	r_b	φ	θ	ζ
-0.022387	0	0.024784	0	-67.773	0	0	-31.895	0
0	-0.10141	0	68.477	0	-603.86	31.895	0	0
-0.050794	0	-0.48483	0	593.47	0	0	-3.5864	-0.00111
0	-0.009085	0	-1.256	0	0.1170	0	0	0
0.000665	0	-0.005917	0	-0.66653	0	0	0	0
0	0.0069445	0	-0.016256	0	-0.33911	0	0	0
0	0	0	1	0	0.11244	0	0	0
0	0	0	0	1	0	0	0	0
-0.11174	0	0.99374	0	0	0	0	-609.57	0

Table 1.6 aFltSim WL1g B -matrix linear data for Example 1.2

δ_a	δ_e	δ_r	δ_{th}
0	2.071	0	0.1294
0	0	7.946	0
0	- 23.58	0	0
- 5.87	0	2.106	0
0	- 5.732	0	0
- 0.01938	0	- 1.57	0
0	0	0	0
0	0	0	0
0	0	0	0

Table 1.7 aFltSim WL1g C -matrix linear data for Example 1.2

u_b	v_b	w_b	p_b	q_b	r_b	φ	θ	z
0.9937	0	0.1117	0	0	0	0	0	0
0	0.001651	0	0	0	0	0	0	0
- 0.0001833	0	0.00163	0	0	0	0	0	0
0.1117	0	- 0.9937	0	0	0	0	609.6	0
0	0	0	0	0	0	0	0	- 1
- 0.0006975	0	0.0007722	0	0.01055	0	0	0	0
0	- 0.00316	0	0.01138	0	0.05912	0	0	0
- 0.001583	0	- 0.01511	0	- 0.3827	0	0	0	0

Table 1.8 aFltSim WL1g D -matrix linear data for Example 1.2

δ_a	δ_e	δ_r	δ_{th}
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0.06453	0	0.004031
0	0	0.2476	0
0	- 0.7346	0	0

It turns out that an attempt to use the fully coupled aircraft 6-DoF model (1.22) for control design would most likely result in an impractical control solution of unnecessary complexity and with an undesirable high sensitivity due to model

parameters. This phenomenon immediately presents a modeling-for-control challenge: How detailed does a control-oriented model need to be so that the resulting control solution is simple, robust, and effective and works per design specifications, when applied to the original process dynamics? The answer to this question of course depends on the application of interest. In the next section, we will construct simplified flight dynamics models for control design purposes.

1.5 Simplified Flight Dynamics for Control Design

The aircraft 6-DoF motion (1.22) can be decomposed into a mean or a steady-state component around an operating point (trim) and perturbation dynamics around the trim condition of interest. Such a decomposition allows one to reduce the overall nonlinear fully coupled 6-DoF aircraft dynamics into a tractable form, suitable for control design and analysis. As previously discussed, the notion of “trimming an aircraft” refers to finding a balance or an equilibrium between aerodynamic, propulsive, and gravitational forces and moments that are acting on the vehicle. In flight, an aircraft is trimmed by setting its primary controls to values that would result in the desired steady-state flight conditions. The trim function would be performed by a pilot or by an automatic flight control system.

For example, in order to achieve WL1g trim conditions for (1.22), we need to find a set of equilibrium states and controls such that the three translational and three angular accelerations are zeroed out,

$$\begin{aligned} \text{Translational DoF: } 0 &= - \left[\begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} u \\ v \\ w \end{pmatrix} \right] + \frac{1}{m} \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} + \vec{g} \\ \text{Rotational DoF: } 0 &= - \left[\begin{pmatrix} p \\ q \\ r \end{pmatrix} \times J \begin{pmatrix} p \\ q \\ r \end{pmatrix} \right] + \begin{pmatrix} \bar{L} \\ M \\ N \end{pmatrix} \end{aligned} \quad (1.40)$$

An aircraft would have many distinct equilibriums throughout the vehicle flight operational envelope (Fig. 1.10).

These trim points depend first hand on altitude and airspeed.

Example 1.3 aFltSim WL1g Trim Data We use the numerical trim subroutine within Simulink® to generate a set of WL1g equilibrium flight conditions at 25,000 ft altitude at various Mach numbers. The five selected trim points with corresponding dynamic pressure lines are shown in Fig. 1.11.

Figure 1.12 shows trimmed flight data versus dynamic pressure.

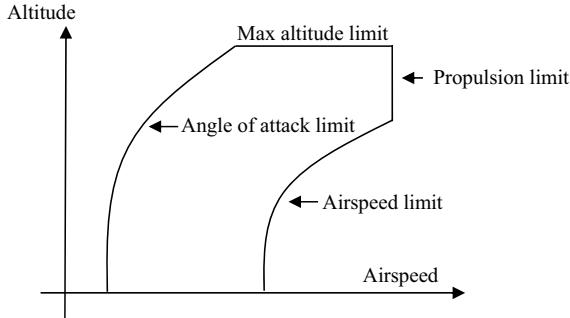


Fig. 1.10 Aircraft operational flight envelope, as a function of altitude and airspeed

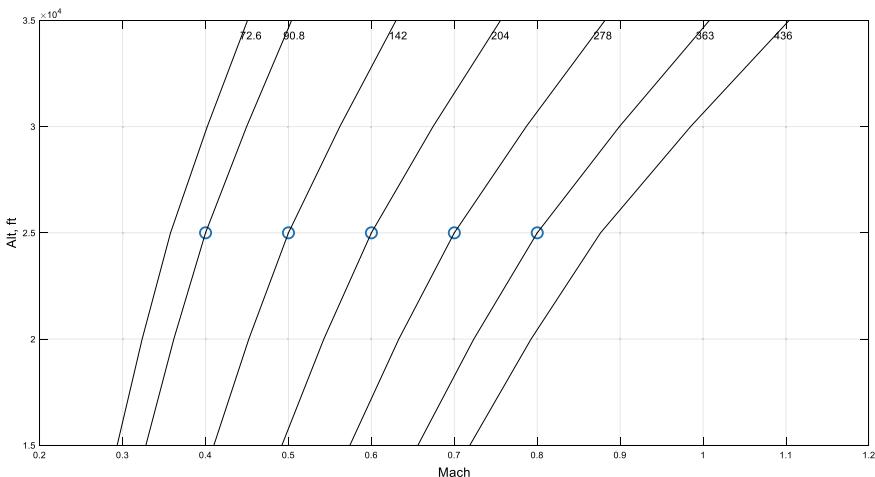


Fig. 1.11 Selected trimmed flight conditions (five circles) at 25,000 ft altitude, in Example 1.3

We can see from the data that as the airspeed increases, the angle of attack (AOA) decreases and absolute values of the elevator (Dele) decrease as well. This is the expected trend in fixed-wing aircraft flight dynamics. At higher airspeeds (large dynamic pressure values), a smaller AOA is required in order to generate a sufficiently large lift force to counter the aircraft gross weight. At the same time, the elevator deflection moves trailing edge up (decreasing absolute value of Dele) to trim the vehicle pitching moment (Fig. 1.12). ■

Based on available trim flight conditions, the main idea behind constructing control-oriented models and then performing flight control design consists of several distinct steps. They are:

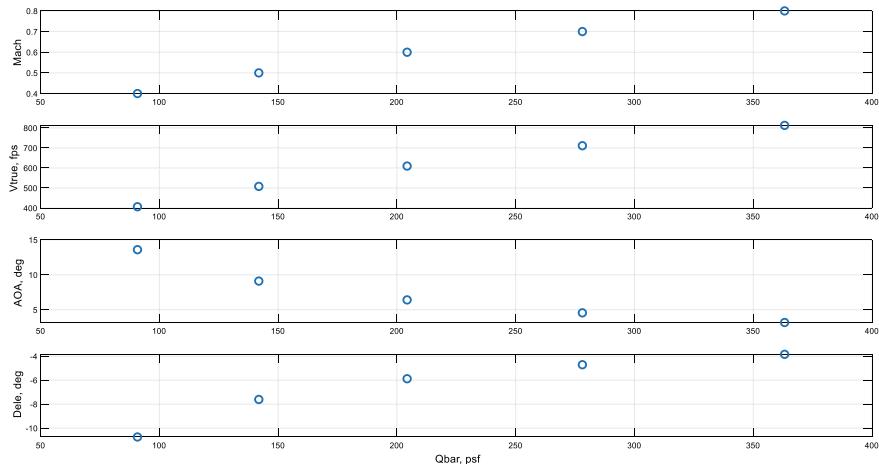


Fig. 1.12 Selected trimmed flight conditions versus dynamic pressure, in Example 1.3

1. Cover the flight envelope with a dense set of trim points.
2. Derive simplified linear models around each of the trim point.
3. Use these dynamics to design fixed-point flight controllers per point.
4. Interpolate (i.e., gain schedule based on flight conditions) to combine linear controllers.

The result is a gain-scheduled flight control system that would be valid for the entire flight operational envelope. In what follows, we will concentrate on Step 2 and derive open-loop time-invariant linear models to define incremental dynamics with respect to a selected trim point.

When a conventional aircraft is trimmed wings-level, the vehicle dynamics naturally decouple into longitudinal and lateral-directional modes. We are going to derive each of these separately.

Longitudinal Dynamics

The aircraft longitudinal dynamics describe changes in forward, vertical, and pitching motion of the vehicle. These dynamics can be further decomposed into fast and slow components, or modes. The former is called the short-period, and the latter is the phugoid. Typically, there would be a timescale separation between the two modes. The short-period describes fast coupling between the aircraft angle of attack and the pitch rate. On the other hand, the phugoid represents a much slower (when compared to the short-period) dynamic interchange between the vehicle altitude and the airspeed, or equivalently, between the aircraft potential and kinetic energy levels.

The short-period and the phugoid modes can be revealed after the aircraft model is linearized around a trim point (an equilibrium) [1–4]. For clarity of presentation,

we assume that the thrust line is aligned with the vehicle x -axis. Then, the aircraft longitudinal equations of motion are

$$\underbrace{\begin{pmatrix} \dot{V}_T \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} X_V & X_\alpha & 0 & -g \cos \gamma_0 \\ \frac{Z_V}{V_0} & \frac{Z_\alpha}{V_0} & 1 + \frac{Z_q}{V_0} & -\frac{g \sin \gamma_0}{V_0} \\ M_V & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_A \underbrace{\begin{pmatrix} V_T \\ \alpha \\ q \\ \theta \end{pmatrix}}_x + \underbrace{\begin{pmatrix} X_{\delta_{th}} \cos \alpha_0 & X_{\delta_e} \\ -X_{\delta_{th}} \sin \alpha_0 & \frac{Z_{\delta_e}}{V_0} \\ M_{\delta_{th}} & M_{\delta_e} \\ 0 & 0 \end{pmatrix}}_B \underbrace{\begin{pmatrix} \delta_{th} \\ \delta_e \end{pmatrix}}_u \quad (1.41)$$

where V_0 is the trimmed airspeed, α_0 is trimmed angle of attack, $\gamma_0 = \theta_0 - \alpha_0$ is the trimmed flight path angle, and θ_0 is the trimmed pitch angle. The model states (V_T, α, q, θ) and the control inputs (δ_{th}, δ_e) are incremental due to their trimmed values.

In order to shorten notations, we often use $p = p_b, q = q_b, r = r_b$ for body axis angular rates. In (1.41), the matrix components represent constant (for fixed flight conditions) stability and control derivatives of the aircraft forces and moments, with respect to the longitudinal states and control inputs. When aircraft specific values of these derivatives are substituted into the model (1.41), most often the open-loop system eigenvalues will consist of a fast (short-period) and a slow (phugoid) pairs of complex-conjugate numbers. Modal decompositions explain the timescale separation in the longitudinal dynamics of an aircraft, such as (1.41).

The short-period mode of an aircraft is defined by angle of attack α and pitch rate q . Extracting those from the model (1.41) yields

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V_0} & 1 + \frac{Z_q}{V_0} \\ M_\alpha & M_q \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} \frac{Z_{\delta_e}}{V_0} \\ M_{\delta_e} \end{pmatrix} \delta_e \quad (1.42)$$

These dynamics describe aircraft motion on a short interval of time, due to elevator input. Throughout the book, we shall utilize the short-period system quite often in our exploration of robust and adaptive control design and analysis methods.

The aircraft phugoid motion can be derived by setting $\dot{\alpha} = \dot{q} = 0$ in (1.41), solving for the corresponding “fast steady-state” values (α, q) and substituting them into the remaining dynamic equations for V_T and θ .

We leave the phugoid derivations to the reader and turn our attention back to the short-period dynamics (1.42). Let us now introduce the flight path angle γ . This is the angle between the aircraft airspeed vector and the horizon. For small angles, the following relationship links angle of attack α , pitch angle θ , and flight path angle γ .

$$\alpha = \theta - \gamma \quad (1.43)$$

It is depicted in Fig. 1.13.

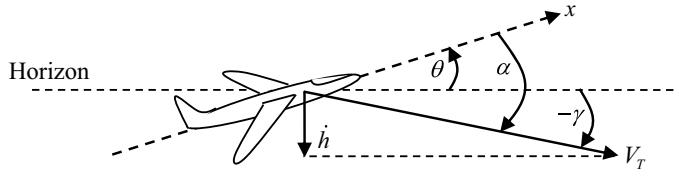


Fig. 1.13 Aircraft longitudinal motion and related angles

Multiplying both sides of (1.43) by true airspeed V_T gives the vertical speed in inertial space.

$$\dot{h} = V_T \gamma = V_T (\theta - \alpha) \quad (1.44)$$

For small angles and assuming that the true airspeed is constant, we can differentiate (1.44) with respect to time, use α -dynamics from (1.42), and finally compute (approximately) the vertical acceleration a_z in body axes.

$$A_z \approx -\ddot{h} = -V (\dot{\theta} - \dot{\alpha}) = V (\dot{\alpha} - q) = Z_\alpha \alpha + Z_\delta \delta_e \quad (1.45)$$

In several of our upcoming design studies and examples, we shall treat this signal as the regulated output of the aircraft longitudinal dynamics (1.42).

$$A_z = (Z_\alpha \ 0) \begin{pmatrix} \alpha \\ q \end{pmatrix} + Z_\delta \delta_e \quad (1.46)$$

Sometimes, we choose to utilize A_z , rather than α , as the preferred state component. Differentiating (1.45) and solving for α in (1.46),

$$\alpha = \frac{A_z - Z_\delta \delta_e}{Z_\alpha} \quad (1.47)$$

gives,

$$\begin{aligned} \dot{A}_z &= Z_\alpha \dot{\alpha} + Z_\delta \dot{\delta}_e = \frac{Z_\alpha}{V} \underbrace{(Z_\alpha \alpha + Z_\delta \delta_e)}_{a_z} + Z_\alpha q + Z_\delta \dot{\delta}_e \\ &= \frac{Z_\alpha}{V} A_z + Z_\alpha q + Z_\delta \dot{\delta}_e \end{aligned} \quad (1.48)$$

and then the pitch dynamics become

$$\begin{aligned} \dot{q} &= M_\alpha \alpha + M_q q + M_\delta \delta_e = M_\alpha \left(\frac{A_z - Z_\delta \delta_e}{Z_\alpha} \right) + M_q q + M_\delta \delta_e \\ &= \frac{M_\alpha}{Z_\alpha} A_z + M_q q + \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha} \right) \delta_e \end{aligned} \quad (1.49)$$

Collecting (1.48) and (1.49), we can rewrite the short-period dynamics (1.42) in terms of the new state components (A_z , q).

$$\begin{pmatrix} \dot{A}_z \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V} & Z_\alpha \\ \frac{M_\alpha}{Z_\alpha} & M_q \end{pmatrix} \begin{pmatrix} A_z \\ q \end{pmatrix} + \begin{pmatrix} 0 \\ M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha} \end{pmatrix} \delta_e + \begin{pmatrix} Z_\delta \\ 0 \end{pmatrix} \dot{\delta}_e \quad (1.50)$$

We immediately note that presence of the control rate $\dot{\delta}_e$ in (1.50) requires the addition of an actuator model. The latter can be modeled by a second-order ordinary differential equation, with a specified natural frequency ω and a damping ratio ξ . The actuator model is driven by the elevator command δ_e^{cmd} , and its dynamics are

$$\ddot{\delta}_e = -2\xi\omega\dot{\delta}_e + \omega^2(\delta_e^{\text{cmd}} - \delta_e) \quad (1.51)$$

Combining (1.50) with (1.51), we arrive at the following four-dimensional system,

$$\begin{pmatrix} \dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V} & Z_\alpha & 0 & Z_\delta \\ \frac{M_\alpha}{Z_\alpha} & M_q & M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & -2\xi\omega \end{pmatrix} \begin{pmatrix} A_z \\ q \\ \delta_e \\ \dot{\delta}_e \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \omega^2 \end{pmatrix} \delta_e^{\text{cmd}} \quad (1.52)$$

that describes the short-period dynamics of an aircraft, driven by an elevator command through actuation. Such a model is very helpful in flight control applications whereby angle-of-attack measurements are not available (or deemed unreliable). The tradeoff here is that the model order has increased twice. In addition, the actuator position and the rate may not be available as measurements.

Lateral-Directional Dynamics

We begin with the kinematics of the Euler roll equation from (1.11)

$$\dot{\phi} = p + \tan \theta (q \sin \varphi + r \cos \varphi) \quad (1.53)$$

Let θ_0 denote the trimmed pitch angle. Then a linear approximation of (1.53) around $\varphi_0 = p_0 = q_0 = r_0 = 0$ can be written as

$$\dot{\phi} = p + r \tan \theta_0 \quad (1.54)$$

We now defined stability axis roll and yaw rates (p_s, r_s). They are related to the body axis roll and yaw rates (p, r) in the following way:

$$\begin{aligned} p_s &= p \cos \alpha + r \sin \alpha \\ r_s &= r \cos \alpha - p \sin \alpha \end{aligned} \quad (1.55)$$

Let α_0 denote the trimmed angle of attack (AOA). Then a linear approximation of (1.55) is of the form

$$\begin{aligned} p_s &= p \cos \alpha_0 + r \sin \alpha_0 \\ r_s &= r \cos \alpha_0 - p \sin \alpha_0 \end{aligned} \quad (1.56)$$

Solving (1.56) for (p, r) yields

$$\begin{aligned} p &= p_s \cos \alpha_0 - r_s \sin \alpha_0 \\ r &= r_s \cos \alpha_0 + p_s \sin \alpha_0 \end{aligned} \quad (1.57)$$

Substituting (1.57) into (1.54) results in

$$\begin{aligned} \dot{\varphi} &= p_s \cos \alpha_0 - r_s \sin \alpha_0 + (r_s \cos \alpha_0 + p_s \sin \alpha_0) \tan \theta_0 \\ &= (\cos \alpha_0 + \sin \alpha_0 \tan \theta_0) p_s + (\cos \alpha_0 \tan \theta_0 - \sin \alpha_0) r_s \end{aligned} \quad (1.58)$$

As we have previously noted, the following relation exists between the flight path angle, the pitch angle, and the angle of attack (at zero bank and sideslip angles):

$$\alpha_0 = \theta_0 - \gamma_0 \quad (1.59)$$

Substituting (1.59) into (1.58) gives

$$\begin{aligned} \dot{\varphi} &= \underbrace{(\cos \alpha_0 + \sin \alpha_0 \tan \theta_0)}_{\frac{\cos \gamma_0}{\cos \theta_0}} p_s + \underbrace{(\cos \alpha_0 \tan \theta_0 - \sin \alpha_0)}_{\frac{\sin \gamma_0}{\cos \theta_0}} r_s \\ &= \frac{\cos \gamma_0}{\cos \theta_0} p_s + \frac{\sin \gamma_0}{\cos \theta_0} r_s \end{aligned} \quad (1.60)$$

Assuming small angles, the angle of sideslip dynamics can be written as,

$$\dot{\beta} = \frac{1}{V_0} (Y_\beta \beta + Y_p p_s + Y_r r_s + Y_{\delta_{ail}} \delta_{ail} + Y_{\delta_{rud}} \delta_{rud}) + \left(\frac{g \cos \theta_0}{V_0} \right) \varphi - r_s \quad (1.61)$$

where the right-hand side of the equation depends on the derivatives of the side force Y , computed with respect to the lateral-directional states $(\beta, p_s, r_s, \varphi)$ and the control inputs $(\delta_{ail}, \delta_{rud})$. Using (1.60) and (1.61), the aircraft lateral-directional linearized dynamics are

$$\begin{aligned} \dot{\beta} &= \frac{g \cos \theta_0}{V} \varphi + \frac{Y_\beta}{V} \beta + \frac{Y_p}{V} p_s + \left(\frac{Y_r}{V} - 1 \right) r_s + \frac{Y_{\delta_{ail}}}{V} \delta_{ail} + \frac{Y_{\delta_{rud}}}{V} \delta_{rud} \\ \dot{p}_s &= L_\beta \beta + L_p p_s + L_r r_s + L_{\delta_{ail}} \delta_{ail} + L_{\delta_{rud}} \delta_{rud} \\ \dot{r}_s &= N_\beta \beta + N_p p_s + N_r r_s + N_{\delta_{ail}} \delta_{ail} + N_{\delta_{rud}} \delta_{rud} \end{aligned}$$

$$\dot{\phi} = \frac{\cos \gamma_0}{\cos \theta_0} p_s + \frac{\sin \gamma_0}{\cos \theta_0} r_s \quad (1.62)$$

We can easily rewrite (1.62), in matrix form.

$$\underbrace{\begin{pmatrix} \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \\ \dot{\phi} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} \frac{Y_\beta}{V_0} & \frac{Y_p}{V_0} & \frac{Y_r}{V_0} - 1 & \frac{g \cos \theta_0}{V_0} \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & \frac{\cos \gamma_0}{\cos \theta_0} & \frac{\sin \gamma_0}{\cos \theta_0} & 0 \end{pmatrix}}_A \underbrace{\begin{pmatrix} \beta \\ p_s \\ r_s \\ \varphi \end{pmatrix}}_x + \underbrace{\begin{pmatrix} \frac{Y_{\delta_{ail}}}{V_0} & \frac{Y_{\delta_{rud}}}{V_0} \\ L_{\delta_{ail}} & L_{\delta_{rud}} \\ N_{\delta_{ail}} & N_{\delta_{rud}} \\ 0 & 0 \end{pmatrix}}_B \underbrace{\begin{pmatrix} \delta_{ail} \\ \delta_{rud} \end{pmatrix}}_u \quad (1.63)$$

When the airspeed is sufficiently high, the gravity term in (1.63) becomes negligible: $\frac{g \cos \theta_0}{V_0} \approx 0$. In this case, the bank dynamics can be eliminated.

$$\underbrace{\begin{pmatrix} \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} \frac{Y_\beta}{V_0} & \frac{Y_p}{V_0} & \frac{Y_r}{V_0} - 1 \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{pmatrix}}_A \underbrace{\begin{pmatrix} \beta \\ p_s \\ r_s \end{pmatrix}}_x + \underbrace{\begin{pmatrix} \frac{Y_{\delta_{ail}}}{V_0} & \frac{Y_{\delta_{rud}}}{V_0} \\ L_{\delta_{ail}} & L_{\delta_{rud}} \\ N_{\delta_{ail}} & N_{\delta_{rud}} \end{pmatrix}}_B \underbrace{\begin{pmatrix} \delta_{ail} \\ \delta_{rud} \end{pmatrix}}_u \quad (1.64)$$

The resulting third-order lateral-directional linear model would be suitable for a control design where the goal is to regulate the vehicle roll and yaw rates, as well as the angle of sideslip.

Example 1.4 aFltSim Linearization Based on the WL1g trim data (Fig. 1.11), we can numerically compute open-loop linear time-invariant models for longitudinal (pitch) (1.41) and lateral-directional (roll–yaw) (1.63) dynamics. Figure 1.14 shows the modal data. These are natural frequencies and damping ratios of the open-loop system eigenvalues.

The same data can be plotted in a root locus format (Fig. 1.15).

There is a clear separation between fast and slow modes in both dynamics. Also, as the vehicle airspeed increases, both pitch and roll–yaw modes become faster.

The fast pitch modes define the short-period dynamics (1.42) and the slow pitch modes form the phugoid dynamics. From the control perspective, the short-period mode is of primary interest. Its natural frequency and damping ratio would often be modified via feedback control.

There are two sets of fast roll–yaw modes. They are called “Roll” and “Dutch-roll”. Roll is usually the fastest of the two. It defines the aircraft rolling motion around the longitudinal x -axis, with respect to body or stability axes. The latter is preferred for flight control applications. The Dutch-roll motion defines the dynamic coupling between the aircraft rolling and yawing degrees-of-freedom. Both roll and Dutch-roll are actively controlled to enforce desired closed-loop characteristics. The slow roll–yaw dynamics define the aircraft spiral mode.

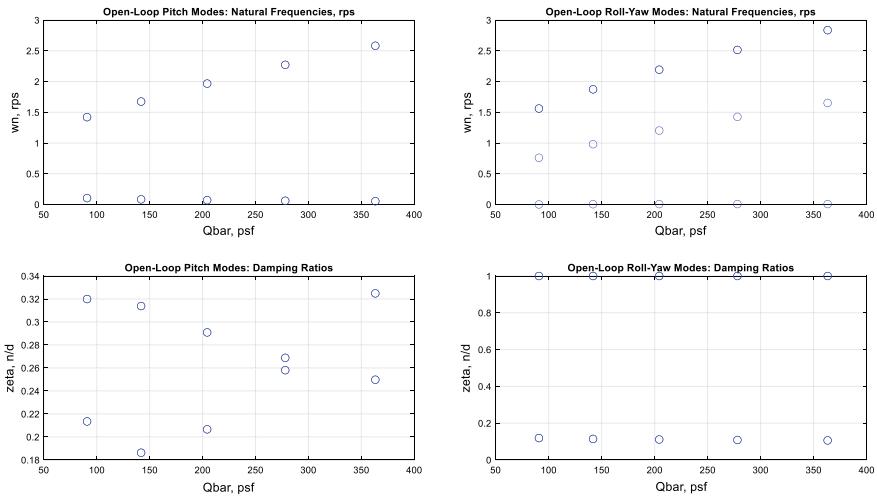


Fig. 1.14 Pitch and roll–yaw modes versus dynamic pressure in Example 1.4

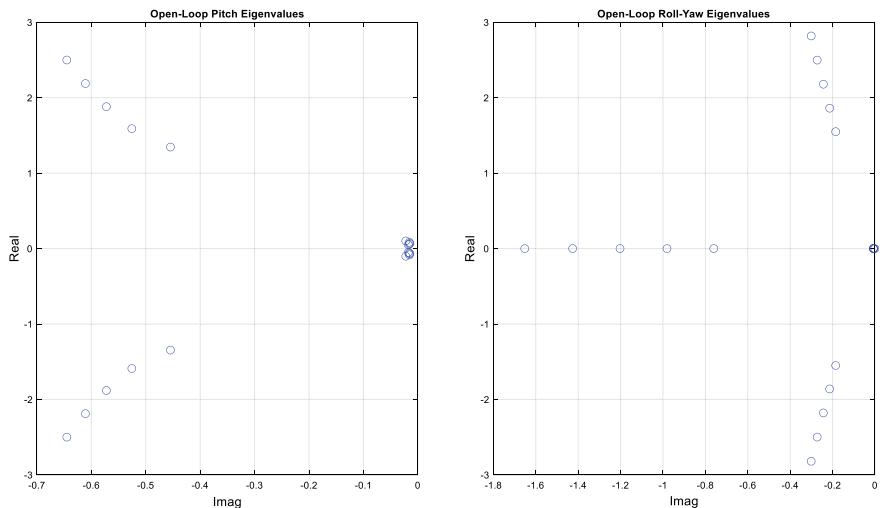


Fig. 1.15 Pitch and roll–yaw eigenvalues root locus versus dynamic pressure in Example 1.4

Figures 1.16, 1.17, 1.18, and 1.19 show the aircraft linear model data: The A - and B -matrices, as defined in (1.41) and (1.63), for longitudinal and lateral-directional dynamics, respectively. The data are plotted versus dynamic pressure to show observed trends with respect to airspeed.

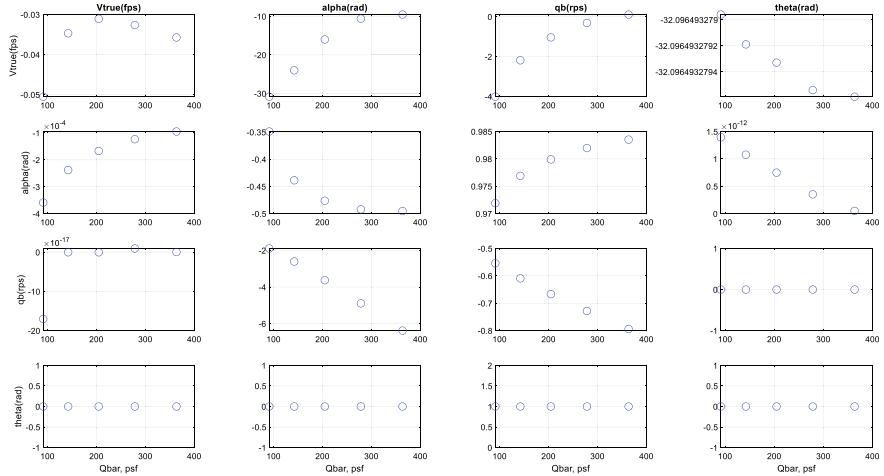


Fig. 1.16 Longitudinal A-matrix versus dynamic pressure in Example 1.4

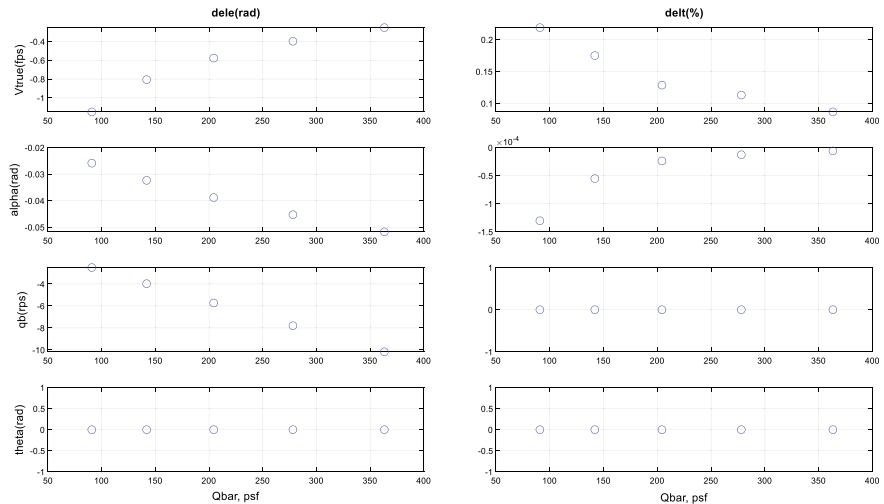


Fig. 1.17 Longitudinal B-matrix versus dynamic pressure in Example 1.4

The reader should correlate these data with the linear models (1.41) and (1.63) to better understand the influence of each matrix element to the aircraft dynamics. This constitutes the essential knowledge that would be carried over to support aircraft flight control design and analysis processes. ■

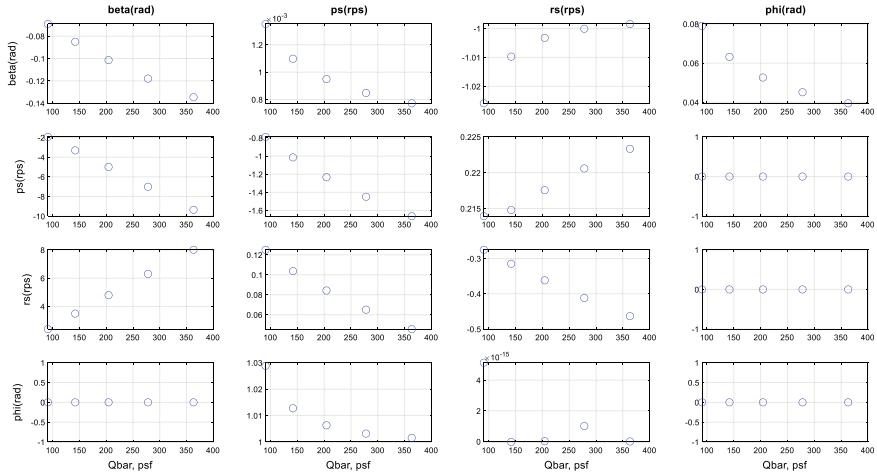


Fig. 1.18 Lateral-directional A -matrix versus dynamic pressure in Example 1.4

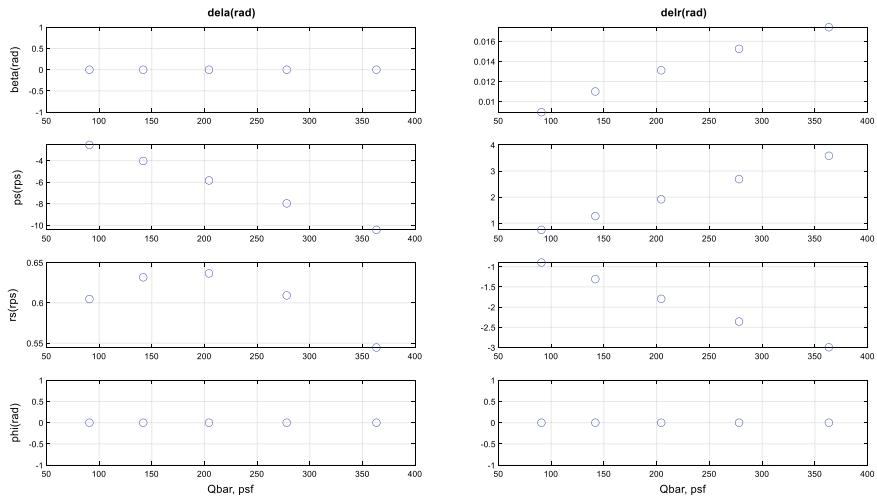


Fig. 1.19 Lateral-directional B -matrix versus dynamic pressure in Example 1.4

Model Generalizations for Adaptive Control Design

The aircraft short-period dynamics (1.42), as well as the lateral-directional models (1.63) and (1.64), represent linear time-invariant controllable systems,

$$\dot{x} = A x + B u \quad (1.65)$$

with the n -dimensional state x , the m -dimensional control u , the p -dimensional output,

$$y = C x + D u \quad (1.66)$$

and with the matrices (A, B, C, D) of the corresponding dimensions. In Part I, we will discuss various methods to design robust linear controllers and to analyze their performance.

Subsequently in Part II, we will focus our attention on adaptive control techniques, with the goal of maintaining closed-loop stability and robustness in the presence of unexpected events. Specifically, we shall insert uncertainties into (1.65) and (1.66) and consider a class of dynamical systems in the form,

$$\dot{x} = A x + B \Lambda (u + f(x)) \quad (1.67)$$

where the $(m \times m)$ -matrix Λ models control actuation failures and the m -dimensional vector function $f(x)$ represents all other “unknown unknowns” in the system dynamics. The uncertain model (1.67) is our attempt to embed an extra realism into the “ideal” system (1.65). The uncertainties in (1.67) are called “matched”, in the sense that they enter the system dynamics through control channels. As long as Λ is invertible, the system (1.67) remains controllable. It so happens that the matched uncertainty assumption implies existence of at least one control solution, capable of steering the system state along the desired trajectories.

We shall also consider regulation problems with non-matched but bounded uncertainties, such as time-dependent noise and environmental disturbances, represented by an n -dimensional uniformly bounded piece-wise continuous vector function $\xi(t)$.

$$\dot{x} = A x + B \Lambda (u + f(x)) + \xi(t) \quad (1.68)$$

Again, we would like to point out that the assumed boundedness of $\xi(t)$ does not destroy the system controllability. So, the unwanted effects caused by bounded noise and disturbances can be mitigated through proper control synthesis. In Part II, we will explore robust and adaptive methods to control uncertain systems, such as (1.67) and (1.68).

Readers who are interested in adaptive control may find the matched uncertainty assumption to be quite restrictive. Some may even argue that there are many dynamical systems, arising from realistic applications, that do not satisfy the matching conditions. Be as it may, in aerospace applications, matched uncertainties are of primary concern, and that explains our interest in the control of uncertain systems such as (1.68). Finally, we would like to note that most of the adaptive control methods presented in this book can be extended to handle systems with non-matched uncertain dynamics, but these extensions are outside of the book scope.

1.6 Summary

Robust and adaptive control of continuous dynamical systems is the focus of this book. We have presented a concise self-contained introduction into the underlying theory and methods, while emphasizing how to design and analyze practical control systems for multi-input–multi-output systems with nonlinear and uncertain dynamics. Our true inspiration comes from aerospace applications. During our professional careers, we have been fortunate to have had the opportunity to design control systems for various types of aerial platforms, most of which were tested in flight and others went into production. In this chapter, we have added introduction and examples of flight dynamics models to later demonstrate a variety of robust and adaptive control technologies. We hope that readers would find these dynamics interesting and ever challenging.

1.7 Exercises

Exercise 1.1 Use the aircraft longitudinal system of Eq. (1.41) to derive the vehicle short-period dynamics (1.42). Define the assumptions required to perform the short-period dynamics derivations. Use similar assumptions to derive the aircraft phugoid dynamics for the vehicle airspeed and pitch angle coupling, while holding angle of attack and pitch rate deviations at zero values. Comment on your results.

Exercise 1.2 Consider the longitudinal dynamics,

$$\begin{pmatrix} \dot{V} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -0.038 & 18.984 & 0 & -32.174 \\ -0.001 & -0.632 & 1 & 0 \\ 0 & -0.759 & -0.518 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} V \\ \alpha \\ q \\ \theta \end{pmatrix} + \begin{pmatrix} 10.1 & 0 \\ 0 & -0.0086 \\ 0.025 & -0.011 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_{\text{th}} \\ \delta_e \end{pmatrix}$$

representative of a transport aircraft, trimmed at $V_0 = 250$ ft/s, and flying at a low altitude. In the model, all angles and angular rates are in radians, airspeed is in ft/s, throttle is in lbs, and elevator deflections are in radians. Compute open-loop system eigenvalues. Extract the aircraft short-period dynamics. Compute and compare the approximated short-period modes to the original system eigenvalues. Simulate open-loop system responses due to elevator and thrust step-inputs. Identify (numerically) a timescale separation between the short-period and the phugoid modes. Introduce vertical acceleration A_z , as defined in (1.46), and simulate its response due to a negative step-input in the elevator (trailing edge up). Observe the initial tendency of A_z . When the elevator is deflected trailing edge up to pitch the vehicle nose

up, there is a small instant decrease in the vertical acceleration. Then, A_z starts to increase, resulting in the aircraft pitch up motion. This transient is caused by the elevator deflecting upward and creating a small negative lift increment. As a result, the vertical acceleration momentarily goes into the “wrong” direction, before it reverses and builds up. These dynamics can also be explained by the fact that there is a non-minimum phase zero (with a positive real part) in the transfer function from δ_e to A_z . It is important to understand that all tail-driven aerial vehicles have similar characteristics. This phenomenon becomes very important during control design.

Exercise 1.3 Use (1.63) to derive the reduced-order lateral-directional dynamics (1.64). Analytically compute the system modal characteristics: The eigenvalues that correspond to the aircraft roll and Dutch-roll dynamics. Identify primary elements of the A -matrix for each of the modes. Comment on your results.

Exercise 1.4 The lateral-directional dynamics of a passenger aircraft, in a cruise configuration, are given below,

$$\begin{pmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} -0.0829 & 0 & -1 & 0.0487 \\ -4.546 & -1.699 & 0.1717 & 0 \\ 3.382 & -0.0654 & -0.0893 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \beta \\ p \\ r \\ \phi \end{pmatrix} + \begin{pmatrix} 0 & 0.0116 \\ 27.276 & 0.5758 \\ 0.3952 & -1.362 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_a \\ \delta_r \end{pmatrix}$$

where the roll and sideslip angles are in radians, the angular rates are in rad/s, and the aileron and rudder deflections are in radians. Compute open-loop system eigenvalues and compare the data to the analytical predictions from Exercise 1.3. Simulate open-loop system response due to aileron and rudder step-inputs. Observe the roll-rate response due to aileron and the coupling between the roll and yaw rates (called the “Dutch-roll” mode). These dynamics are fast, when compared to the much slower changes in the roll angle (called the “roll subsidence” mode). Similar to short-period, the roll rate and the Dutch-roll modes are the main quantities for stabilization and regulation. This task is often accomplished during the so-called inner-loop control design phase, where the angular rates are stabilized via feedback connections, driving the aileron and the rudder. For the inner-loop design, the bank dynamics are ignored and the three-dimensional lateral-directional model (1.64) is utilized. Extract these dynamics from the model data, and simulate responses of the simplified model due to the same step-inputs in aileron and rudder. Compare and discuss simulation results.

Exercise 1.5 Install 6-DoF aircraft flight simulation (aFltSim) software. Select a flight condition. Perform wings-level 1g trim and linearization. Compute the corresponding decoupled longitudinal and lateral-directional systems. Extract reduced-order systems (1.42) and (1.64). Compare step responses between the outputs from aFltSim with the responses of the corresponding open-loop linear full-order and reduced-order models. Comment on your results.

References

1. Etkin B (1982) Dynamics of flight. Stability and control, 2nd edn. Wiley, New York
2. Blakelock JH (1990) Automatic control of aircraft and missiles, 2nd edn. Wiley, New York
3. McRuer D, Ashkenas I, Graham D (1990) Aircraft dynamics and automatic control. Princeton University Press, Princeton, NJ
4. Stevens BL, Lewis FL (1992) Aircraft control and simulation. Wiley, New York
5. U.S. Standard Atmosphere (1976) National Oceanic and Atmospheric Administration, National Aeronautics and Space Administration, United States Air Force, Washington, DC
6. Klein V, Morelli EA (2006) Aircraft system identification. Theory and practice. AIAA Educational Series
7. Stoer J, Bulirsch R (1992) Introduction to numerical analysis, 2nd edn. Springer
8. Grauer JA, Morelli EA (2015) Generic global aerodynamic model for aircraft. J Aircraft 52(1)



Linear Time-Invariant Systems and Control

2

This chapter presents key classical control concepts and introduces state-space modeling and analysis of control systems as applied to the control of linear time-invariant (LTI) dynamical systems. This material is an integral part of Linear Systems Theory. It also represents the “backbone” of our book and becomes a prerequisite for understanding the book’s more advanced concepts. We assume that the reader has a basic knowledge of classical control, modeling, and analysis of LTI systems and matrix algebra. However, for clarity and completeness purposes, we shall devote this chapter to a review of these basics toward modeling and control of LTI dynamical systems.

2.1 Model-Based Control Engineering

Control system design and analysis methods today are predominantly model based, and as such they all start with a model of the system to be controlled. A model-based controller must be robust and work well for the real process represented by the model. The art of control design starts with the formulation of a “simplified” reduced-order linear model that is representative of the process. That enables applications of formal linear control design and analysis methods with stability and robustness guarantees. A linear controller designed using a linear model will adequately work for the real process due to robustness properties of the underlying linear control design methods. Throughout the book, we will introduce and demonstrate capabilities of model-based linear control design and analysis methods, along with their applications to control real systems.

For example, in Chap. 1, we defined the equations of motion for flight dynamics and a high-fidelity flight simulation model for a fixed-wing aircraft. That model

contains many degrees-of-freedom as well as inherent nonlinearities due to aerodynamics, propulsion, and other effects. Using such a model directly for control design could produce a very complicated control system that would be sensitive to model parameter variations and system nonlinearities. At the same time, we also derived reduced-order linear models that are suitable for flight control design and have quantifiable guarantees to operate the real vehicle. The linearization of the nonlinear model at an equilibrium or trim condition, and the subsequent design and analysis of a controller for that model is the standard method used in industry. Key stability, performance, and robustness metrics are computed from the linear models and are used in program/vehicle design reviews to proceed to flight. These metrics are also used to show that both time domain performance requirements and frequency domain robustness requirements are met and satisfied. This process has shown to produce safe and reliable flight critical systems in operational environments.

The gain design and analysis process can occur over thousands of equilibrium design points. It is usually automated in design scripts in order to reduce time design cycles and to enable rapid vehicle prototyping. The gain design process is repeated as needed during a development program when model databases are updated or revised, or when the installed performance of the vehicle equipment is measured and shown to be significantly different from the assumed models. The design process of linearizing the nonlinear model at an equilibrium and designing the gains based upon the linear model creates a *baseline control* for the system. Equilibrium points are computed over the flight envelope of the vehicle, and the subsequent controllers/gains are tabulated and implemented using gain scheduling with flight parameters. Interpolation between design points is used as needed. Later in this book we add *adaptive control* to the baseline control to further extend the baseline controller's performance and robustness in the presence of system uncertainties.

Figure 2.1 illustrates a three-step design process for a single equilibrium design point. The process begins with the linearization at an equilibrium as defined in Eq. 1.36. This populates the feedback control design model in Step 1 with trim condition model data. The feedback gain design model is the lowest-order model (as discussed in Sect. 1.5) required to adequately represent the dynamics and variables to be controlled. Keeping this model in low order reduces the number of feedback variables needed for implementation and reduces control system complexity. In Step 1 of the process the feedback gains are designed (this will be discussed at length in subsequent chapters) and then analyzed in the frequency domain Step 2 and via the time domain in Step 3.

The frequency domain analysis includes a stability margin assessment, “gang of four” Bode analysis (see Chap. 3) which includes analysis of sensor noise and load disturbance transfer functions, the overall loop attenuation at high frequencies, as well as robustness analysis predictions. In Fig. 2.1 the analysis models used in this step include everything known in the system that can change gains and/or increase phase lags. Inclusion of other known dynamics within the system produces a control design that meets requirements. Early in a program, details

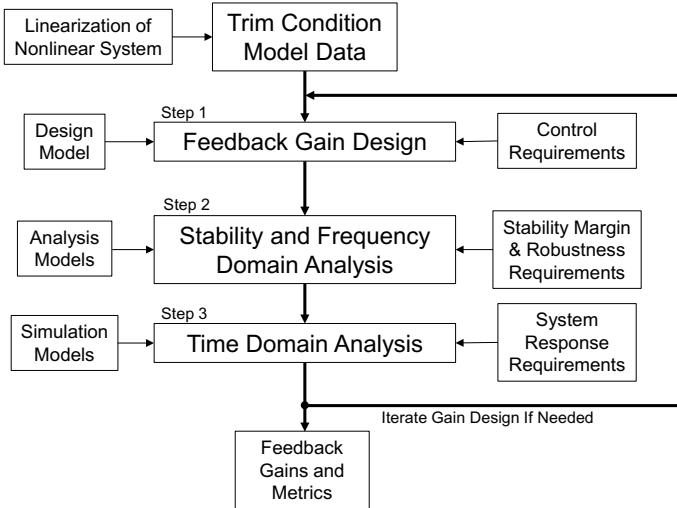


Fig. 2.1 Model-based control system design

about system models may be approximate. As models are improved it is important to repeat this analysis step to assess stability and robustness and repeat the gain design step if needed if metrics are not satisfied.

The time domain analysis in Step 3 yields performance metrics that show if time-based maneuver requirements are satisfied. This step can sometimes use simulation models that differ from the frequency domain analysis models used in Step 2. For example, it may be important to implement nonlinear effects like rate limiting in the actuators to assess the impact on maneuver performance. As models mature, it is important to repeat this analysis step to show that the time domain metrics are satisfied and decide if a gain redesign is warranted.

Figure 2.2 shows a more detailed block diagram for the frequency domain analysis step in Fig. 2.1. It is assumed in this model that a computer is used to implement the control system. The plant block represents a model of the process to be controlled. It has inputs of u which models the command to the actuators and d , the load disturbance. The sensors block models the measurements devices that produce feedback signals into the controller. These sensor measurements model assumes additive sensor noise n . At both the plant input and plant output we have time delay blocks. These blocks model time delays for the computer commanding the actuators (input time delay), and for the computer sampling the measurements (output time delay). The filters block in the controller model represents any filtering used to attenuate sensor noise and/or structural vibrations measured by the sensors. The controller model block implements the control system algorithms, processing the filtered measurements and command to the control system. The

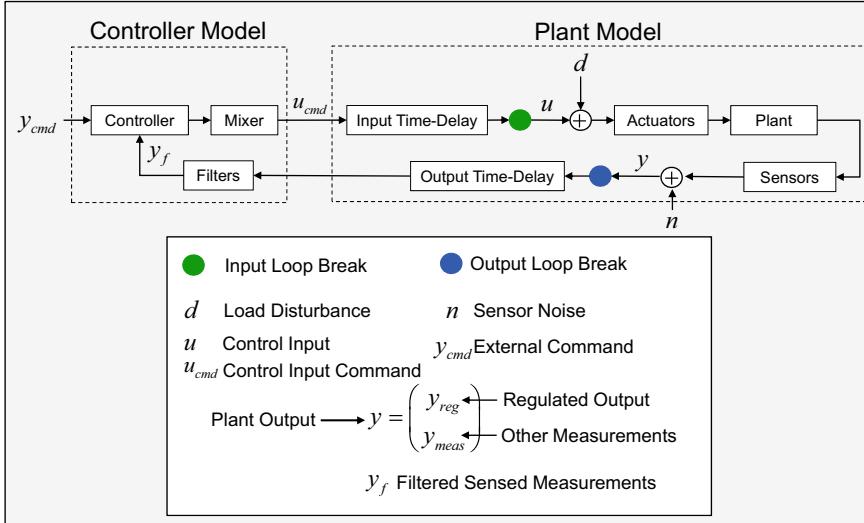


Fig. 2.2 Generic control system block diagram

mixer block shown in this model represents algorithms that distribute control commands to multiple surfaces. For example, a pitch-axis elevator command could be distributed to six surfaces along the wing, with the roll-axis aileron command distributed to just the outboard wing surfaces (ailerons). This block allows for modeling and analysis of axis prioritization and other important features in the flight control system design.

The control system design and analysis should account for all known subsystems as shown in Fig. 2.2. Even though the gains are designed using linear model approximations, the analysis should be performed using the highest-fidelity models along with all other known processing blocks that exist within the corresponding closed-loop system.

The control system analysis should include artifacts that demonstrate: (a) stability margins; (b) command tracking performance metrics; and (c) robustness measures with respect to the plant model parameters. These three essential “must-have” components within the control analysis process provide formal guarantees and numerical evidence that a gain-scheduled control system designed based on a set of simplified linear models will adequately perform while controlling the dynamics of the real process. In short, enforcing and balancing robustness and performance within the linear control design framework is the key to successful applications of linear control methods to real processes and systems.

2.2 Control System Design Goals and Objectives

There are four overarching goals (requirements) for any control system to achieve and demonstrate. They are:

- Closed-loop stability.
- Relative stability, such as gain and phase margins at the plant inputs and at the control-critical sensors.
- Robustness to model parameters and other uncertainties.
- Command tracking performance, such as rise and settling time and percent undershoot and overshoot.

Initially, verification and validation (V&V) of these four requirements is conducted via formal design and analysis methods from linear systems and control. After that, V&V tests are repeated in a high-fidelity simulation and a hardware-in-the-loop environments. These tests are essential to reduce risk in fielding any control system.

Other process-specific objectives are often imposed on the control system performance such as:

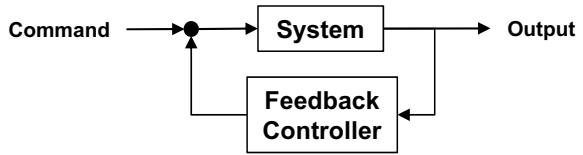
- Regulated output definition to achieve project specific closed-loop system requirements.
- Adequate closed-loop performance for medium-to-large control inputs.
- Turn coordination constraints.
- Pilot controls and haptic interfaces.
- Feedforward control to achieve flying qualities targets, (a pilot perception of closed-loop vehicle behavior).
- Plant nonlinearities accommodation.
- Mitigation of control position and rate saturations.
- Control failure management.

Most of the above tasks are attained using nonlinear control methods, signal processing, and extensive simulation studies within a Monte Carlo context. We will address methods for nonlinear control design and analysis in Part II.

2.3 Feedback and Feedforward Control

In 1909, Nobel Laureate Karl Ferdinand Braun introduced the term “feedback” as a noun referencing undesired coupling between components of an electronic circuit [1]. However, the use of feedback as a mechanical device to control processes dates many years back. In the literature today, we use the word “feedback” to define interconnections between systems such as the closed-loop block diagram in Fig. 2.3.

Fig. 2.3 Closed-loop system block diagram with feedback controller



This system generates its command based on the system response. In contrast, a feedforward control as shown in Fig. 2.4 sends commands into a system without any knowledge or interaction of its output response.

In practice, a controller would include both feedback and feedforward connections as shown in Fig. 2.5 to ensure stability, robustness, and performance. The first two properties (stability and robustness) are provided by feedback, while performance is achieved using the feedforward command connection.

Such a closed-loop system would be designed to force the output signal to track external bounded, possibly time-varying commands. Throughout the book we will discuss formal yet practical design and analysis methods for controllers that serve this purpose. They are called “servo-controllers” to indicate their primary goal of command tracking with stability guarantees.

Figure 2.6 illustrates three command-following control architectures that are common in flight control applications; they are: (a) stability augmentation system (SAS); (b) control augmentation system (CAS); and (c) autopilot (AP).



Fig. 2.4 Open-loop system block diagram with feedforward controller

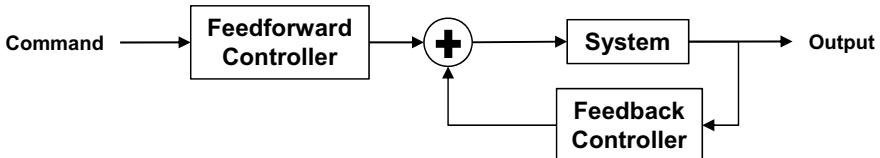


Fig. 2.5 Closed-loop system block diagram with (feedback + feedforward) controller

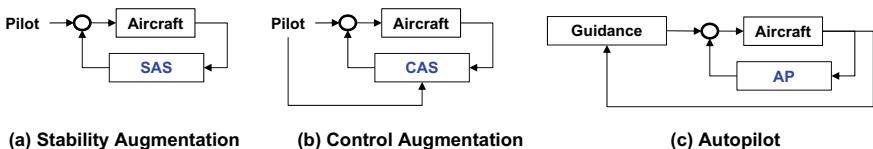


Fig. 2.6 Three flight controllers

The first SAS and CAS architectures are designed to support the pilot with enhanced stability and control of the aircraft. A SAS typically contains mechanism like dampers and stabilizers that use feedback proportional to sensed angular rates of the aircraft. The SAS augments aircraft damping, and during pilot commanded maneuvers opposes the maneuver, often slowing the response. The McDonnell Douglas F-4 is the well-known jet aircraft that employed a SAS. CAS designs as shown in Fig. 2.6 were introduced into aircraft to eliminate the response characteristics associated with the control system opposing the pilots command. CASs significantly improve aircraft handling qualities and are used in today's fly-by-wire aircraft control systems. Autopilots provide "automatic" command tracking and stabilization of flight. The first uses of autopilots centered on reducing pilot workload. They track commands generated from guidance and navigation algorithms which steer the aircraft along a route. Autopilot typically closes feedback loops on angular rates and accelerations, providing an *inner-loop* control. Guidance algorithms can be thought of as *outer-loop* closures surrounding the inner-loop autopilot. In autonomous aircraft, the guidance and autopilot architecture automate the control of the aircraft through all phases of flight, including ground operations. There is a hierarchical nature to control system architectures used in autonomous aircraft that is worthy of further discussion.

Figure 2.7 represents a block diagram of an autonomous aircraft altitude control system. The inner-loop controller K_1 regulates the vertical acceleration A_z of the aircraft, receiving its acceleration command $A_{z,\text{cmd}}$ from an outer-loop guidance controller, K_2 , which uses measurements of velocity and position to generate acceleration commands. This closes loops around the inner loop. Also shown in Fig. 2.7 is a flight management (mgmt.) loop closure with controller K_3 . It uses a route plan and measurements of altitude to generate commands to the guidance loop as the aircraft flies its route. This controller closes another outer loop around the guidance loop closures. Understanding the concept of closing a control loop around another control loop is an important one to discuss.

Figure 2.8 shows a block diagram of a classical proportional-plus-integral (PI) control acceleration command inner-loop controller for K_1 . From the diagram we see an acceleration loop is closed around an inner pitch rate loop. The second-order actuator block in Fig. 2.8 models the actuator control system which controls

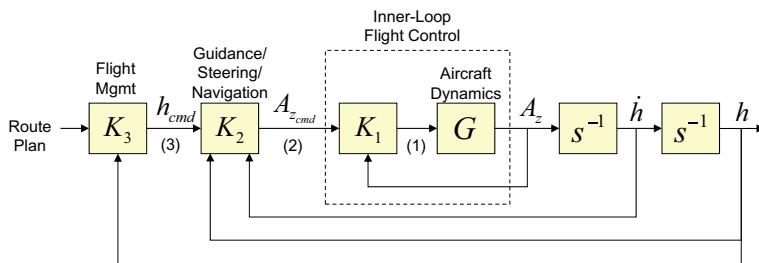


Fig. 2.7 Autonomous aircraft altitude tracking control system

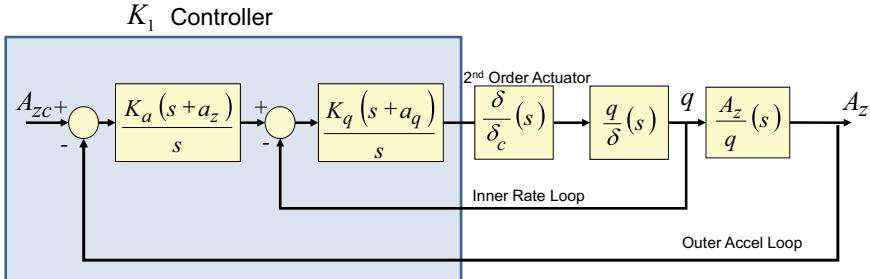


Fig. 2.8 Inner-loop acceleration controller K_1

the position of the pitch-axis control surface δ . The actuator model models loop closures within the physical actuator and represents the most innerloop of the aircraft dynamics in Fig. 2.7. We see from the models in Figs. 2.7 and 2.8 that in both piloted and autonomous aircraft the control systems have loop closures closed around inner loops.

Each outer-loop closure must be designed to be slower than the loops it contains. We say that these loops have frequency separation. We will explain this as follows. In the block diagram in Fig. 2.7 we list three loop breakpoints listed as (1), (2), and (3). When the loop is opened at point (1) for stability analysis, the loop transfer function (LTF) is $L_1 = K_1 G$. If we compute a Bode plot of L_1 , it will have a loop gain-crossover frequency ω_c (this is the frequency where the loop gain crosses 0 dB) and where the associated gain and phase margins define the system relative stability.

If we open the guidance loop at point (2), compute the LTF L_2 for this loop, and subsequently compute a Bode plot for L_2 , the LGCF for L_2 needs to be lower in frequency than the LGCF of L_1 . The same occurs for the flight mgmt. loop at point (3). The LGCF for L_3 needs to be lower in frequency than the LGCF of L_2 . In other words, there needs to be a frequency separation in all these LGCFs. If gains are increased in the outer-loop controllers to try and achieve faster performance, it may destabilize one or several of the inner loops.

2.4 State-Space Systems

For a dynamical system, the concept of an internal state was originally introduced by Rudolf Emil Kalman 1, 3. In 3, Kalman writes: “*The fundamental concept is the notion of the state. By this is meant, intuitively, some quantitative information (a set of numbers, a function, etc.) which is the least amount of data one has to know about the past behavior of the system in order to predict its future behavior. The dynamics is then described in terms of state transitions, i.e., one must specify how one state is transformed into another as time passes*”.

The ground-breaking notion of state of a system allowed mathematicians and engineers to define a dynamic mapping from the system input to its output via a set of ordinary differential equations and provided tractable time domain solutions for them. For linear dynamics, this concept allowed high-order sets of differential equations to be solved very easily and subsequently led to the development of what was called modern control theory, which was built based on these state-space modeling techniques.

We begin with a generic definition. A dynamical system may be thought of as a collection of finite number of interconnected possibly time-dependent components.

$$\dot{x} = f(t, x, u), \quad x(t_0) = x_0 \quad (2.1)$$

In (2.1), $t \in R^+$ denotes time and $f : R \times R^{n_x} \times R^{n_u} \rightarrow R^{n_x}$ is a vector function. We call (2.1) the system dynamics, refer to $x \in R^{n_x}$ as the n_x -dimensional system state at time t , and define $u \in R^{n_u}$ as the n_u -dimensional control input, (an externally supplied signal). The number of the state components n_x is called the order of the system. The system initial conditions $x(t_0) = x_0$ define a starting point for the process evolution in time, with an initial state value $x_0 \in R^{n_x}$ specified at some initial time $t_0 \geq 0$.

Subsystem components (a.k.a. degrees-of-freedom) form the internal state of the system $x(t)$. It allows to describe evolution of the system input-to-output map, which defines the system dynamic response due to initial conditions $x(t_0) = x_0$ and control input $u(t)$.

The system dynamics are driven by an environment where the system operates. When subjected to an external control input $u(t)$, the system generates an output $y(t)$, which in turn may explicitly depend on the system states $x(t)$ and on the control input $u(t)$,

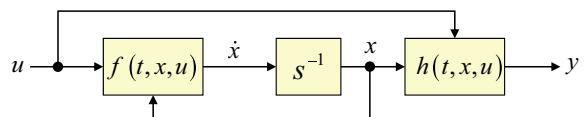
$$y = h(t, x, u) \quad (2.2)$$

where $h : R \times R^{n_x} \times R^{n_u} \rightarrow R^{n_y}$ and $y \in R^{n_y}$.

Equations (2.1)–(2.2) define the state-space model (SSM) for the dynamics of interest. A sketch of the system block diagram is shown in Fig. 2.9.

A solution $x(t)$ of (2.1) (if one exists) corresponds to a curve in the system state space R^{n_x} , as t varies from an initial time t_0 to infinity. This curve is often referred to as the system state trajectory. Later in the book (Part II), we will formulate sufficient conditions guaranteeing existence and uniqueness of solutions for generic dynamical systems such as (2.1), starting from a given set of initial conditions $x(t_0) = x_0$.

Fig. 2.9 State-space model block diagram



2.4.1 Time and Frequency Domain Modeling of State-Space Systems

We now turn our attention to a special class of systems (2.1) whose dynamics are time invariant and linear in the system state and control variables 4, 5, 6. These processes can be modeled by a finite number of first-order linear time-invariant (LTI)-coupled scalar ordinary differential equations.

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{2.3}$$

The LTI system dynamics (2.3) are completely characterized by the matrix quadruple (A, B, C, D) . In (2.3), $x \in R^{n_x}$ is the system state, $u \in R^{n_u}$ is the control input, and $y \in R^{n_y}$ is the system output. In addition, we make a standard assumption that the dimensions of controls and outputs are no greater than the number of the system state components.

$$\dim(u) \leq \dim(x), \quad \dim(y) \leq \dim(x)\tag{2.4}$$

From a historical perspective, it is interesting to note that the original notation for LTI dynamics (2.3) was different. In his papers, Kalman used $(F, G, H, 0)$ and $(F, D, M, 0)$ to define LTI systems. He also argued against the direct feed-through term in the output equation, setting $D = 0$ and writing that “... such a “feed-through” term is not dynamic, and consequently spoils the elegance of the algebraic theory”. Contrary to that statement, many aerospace systems have direct feed-through terms in their outputs, especially when vehicle accelerations are used as sensed measurements.

Throughout the book, we shall continue using the notation (2.3) to model LTI processes, since it became standard in linear systems and control. Figure 2.10 shows a block diagram representation of an LTI system.

As we previously mentioned, the LTI system (2.3) is completely characterized by the corresponding (A, B, C, D) matrices.

$$\left. \begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\right\} \Leftrightarrow \boxed{\begin{pmatrix} A & B \\ C & D \end{pmatrix} \in R^{(n_x+n_y) \times (n_x+n_u)}}\tag{2.5}$$

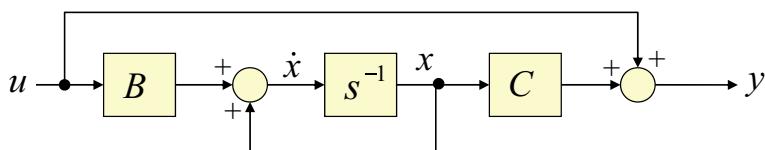


Fig. 2.10 LTI system block diagram

Due to linearity property of the system dynamics, it is sufficient to consider $t_0 = 0$. Then starting at any initial condition x_0 and using a time-dependent control input $u(t)$, the corresponding analytic solution for the system state trajectory of (2.5) can be written explicitly as a sum of the initial condition response and the forced response,

$$x(t) = \underbrace{e^{At}x_0}_{\text{Initial Condition Response}} + \underbrace{\int_0^t e^{A(t-\tau)}B u(\tau)d\tau}_{\text{Forced Response}} \quad (2.6)$$

where the matrix exponential $e^{At} \in R^{n_x \times n_x}$ is defined as the infinite series.

$$e^{At} = I_{n_x} + At + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \cdots + \frac{(At)^n}{n!} + \cdots = \sum_{k=0}^{\infty} \frac{(At)^k}{k!} \quad (2.7)$$

It is known that this series converges for all square matrices, which in turn implies that e^{At} is a finite matrix for any given t . Moreover, the output of (2.5) can also be written analytically.

$$y(t) = C \left(e^{At}x_0 + \int_0^t e^{A(t-\tau)}B u(\tau)d\tau \right) + Du \quad (2.8)$$

The matrix exponential e^{At} is called the *state transition matrix* and is commonly written as

$$e^{A(t-t_0)} = \Phi(t, t_0) \quad (2.9)$$

which describes the transition of the state x from time $t = t_0$ to time t . Using this notation, (2.6) and (2.8) can be rewritten as

$$\begin{aligned} x(t) &= \Phi(t, t_0)x_0 + \int_{t_0}^t \Phi(t, \tau)Bu(\tau)d\tau \\ y(t) &= C \left(\Phi(t, t_0)x_0 + \int_{t_0}^t \Phi(t, \tau)Bu(\tau)d\tau \right) + Du \end{aligned} \quad (2.10)$$

2.4.1.1 Time Domain Modeling of LTI Systems

Solutions (2.6) and (2.8) form the basis for numerical implementation and simulation of the LTI dynamics (2.5) in the time domain. Let $k = 0, 1, 2, \dots$ be a positive integer. Starting at $t = k \Delta t$, let Δt denote a time-sampling interval and assume that the control input $u(t)$ is constant over Δt seconds, $u(t) = u(k \Delta t)$, for $k \Delta t \leq t < (k + 1) \Delta t$. For simplicity, we define

$$x(k) = x(k \Delta t), \quad u(k) = u(k \Delta t) \quad (2.11)$$

Then starting from $x(k)$, the state evolution (2.6) can be written as,

$$\begin{aligned} x(k+1) &= \underbrace{e^{A \Delta t}}_{A_d} x(k) + \int_{k \Delta t}^{(k+1) \Delta t} e^{A((k+1)\Delta t - \tau)} B \underbrace{u(\tau)}_{u(k)} d\tau \\ &= A_d x(k) + \underbrace{\left(\int_0^{\Delta t} e^{A \tau} d\tau B \right)}_{B_d} u(k) = A_d x(k) + B_d u(k) \end{aligned} \quad (2.12)$$

where

$$A_d = e^{A \Delta t}, \quad B_d = \left(\int_0^{\Delta t} e^{A \tau} d\tau \right) B \quad (2.13)$$

are the system discrete matrices. Also, the output Eq. (2.8) becomes,

$$y(k) = C x(k) + D u(k) \quad (2.14)$$

where the matrices C and D remain the same. The resulting discrete LTI system

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k), \quad x(0) = x_0, \quad k = 0, 1, 2, \dots \\ y(k) &= C x(k) + D u(k) \end{aligned} \quad (2.15)$$

can be simulated in the time domain by iterating on k . Similarly to (2.5), the discrete LTI dynamics (2.15) are well defined by the matrix quadruple (A_d, B_d, C, D) .

$$\left. \begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C x(k) + D u(k) \end{aligned} \right\} \Leftrightarrow \boxed{\begin{pmatrix} A_d & B_d \\ C & D \end{pmatrix} \in R^{(n_x+n_y) \times (n_x+n_u)}} \quad (2.16)$$

The process of transforming a continuous LTI system (2.5) into a discrete counterpart (2.16) is called discretization. So far, we discussed the exact analytical discretization method (2.13). It requires computing matrix exponentials. In order

to avoid that and thus improve numerical efficiency, an approximate discretization is often utilized. For small time steps $0 < \Delta t \ll 1$, the matrix exponential can be approximated by the first two terms in the Taylor series (2.7) to compute A_d and by the unity matrix to compute B_d .

$$\boxed{e^{At} \approx I_{n \times n} + At} \Rightarrow A_d$$

$$\boxed{e^{At} \approx I_{n \times n}} \Rightarrow B_d \quad (2.17)$$

Substituting (2.17) into (2.13) results in

$$A_d = I_{n \times n} + \Delta t A, \quad B_d = \Delta t B \quad (2.18)$$

and gives the corresponding discrete LTI system.

$$x(k+1) = x(k) + \Delta t(Ax(k) + Bu(k)) \quad (2.19)$$

Note that the same discrete LTI dynamics can be derived via forward Euler integration method by approximating the state derivative with the first difference quotient.

$$\dot{x}(t) \approx \frac{x(t + \Delta t) - x(t)}{\Delta t} \quad (2.20)$$

2.4.1.2 Frequency Domain Modeling of LTI Systems

So far, we discussed modeling LTI processes in the time domain. Another insightful method for understanding LTI dynamic characteristics is modeling these systems in the frequency domain. This leads to the fundamental notion of a transfer function. It defines a map between an external input to the system and its corresponding output.

We shall follow derivations from 6. For simplicity, assume that the system control input is a scalar and thus $m = 1$. Consider the response of a continuous LTI process (2.5) due to a complex exponential input.

$$\boxed{u = e^{st}} = e^{(\sigma + j\omega)t} = e^{\sigma t} e^{j\omega t} = e^{\sigma t} (\cos \omega t + j \sin \omega t) \quad (2.21)$$

Let's assume that the complex number s is not an eigenvalue of A . In this case, the system state response can be computed from (2.6).

$$\begin{aligned}
x(t) &= e^{At} x(0) + \int_0^t e^{A(t-\tau)} B e^{s\tau} d\tau = e^{At} \left(x(0) + \int_0^t e^{(sI_{n_x} - A)\tau} B d\tau \right) \\
&= e^{At} \left(x(0) + (sI_{n_x} - A)^{-1} (e^{(sI_{n_x} - A)t} - I_{n_x}) B \right) \\
&= e^{At} x(0) + \underbrace{e^{At} (sI_{n_x} - A)^{-1} e^{(sI_{n_x} - A)t} B}_{(sI_{n \times n} - A)^{-1} B e^{st}} - e^{At} (sI_{n_x} - A)^{-1} B \\
&= \boxed{e^{At} \left(x(0) - (sI_{n_x} - A)^{-1} B \right)} + \boxed{(sI_{n_x} - A)^{-1} B e^{st}} \\
&\quad \text{Transient due to Initial Condition} \qquad \qquad \qquad \text{“Steady State” Forced Response} \tag{2.22}
\end{aligned}$$

The process output (2.8) defines the system transfer function $G(s)$,

$$\begin{aligned}
y(t) &= Cx + Du = C \left((sI_{n_x} - A)^{-1} B e^{st} \right) + D e^{st} \\
&= \underbrace{\left(C(sI_{n_x} - A)^{-1} B + D \right) e^{st}}_{G(s): \text{Transfer Function}} = G(s)u(t) \tag{2.23}
\end{aligned}$$

and consequently

$$y(t) = G(s)u(t) \tag{2.24}$$

It is important to emphasize that in (2.24), both input and output signals are functions of time, while the argument of the transfer function is complex. We immediately note that (2.23) and (2.24) do not constitute a “standard” textbook classical way to define transfer functions for LTI systems. Nevertheless, we feel it is important to demonstrate multiple approaches.

Toward that end, we now present the classical way to derive a transfer function for an LTI system 4. It is based on the Laplace transform (LT) of a time-varying integrable vector function $f(t)$.

$$\boxed{F(s) = \int_0^\infty f(t) e^{-st} dt} \Rightarrow \boxed{\text{LT} : f(t) \rightarrow F(s)} \tag{2.25}$$

The LT operation (2.25) maps time domain functions $f(t)$ into their frequency domain images $F(s)$. It is defined only for those complex variables $s \in \mathbb{C}$ where the Laplace integral converges, (i.e., finite). In essence for LT to exist, $f(t)$ must be integrable and grow no faster than the complex exponential $e^{s_0 t}$ with a fixed s_0 and a large t .

$$F(s) = \int_0^\infty f(t)e^{-st} dt, \quad \operatorname{Re} s > s_0 \quad (2.26)$$

The LT mapping is understood component-wise. The transform has many useful properties. Among those are the two that are of interest: For zero initial conditions, differentiation and integration operations in the time domain are mapped into multiplication and division in the frequency domain, correspondingly.

$$\begin{aligned} \text{Differentiation : } & \operatorname{LT}\{\dot{x}\} = s \operatorname{LT}\{x(t)\} = s X(s) \\ \text{Integration : } & \operatorname{LT}\left\{\int x dt\right\} = \frac{\operatorname{LT}\{x(t)\}}{s} = \frac{X(s)}{s} \end{aligned} \quad (2.27)$$

Example 2.1 Consider the step function $u_s(t)$ given as

$$u_s(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (2.28)$$

The LT of a step function with amplitude A is given by

$$\operatorname{LT}\{Au_s(t)\} = \int_0^\infty Au_s(t)e^{-st} dt = A \int_0^\infty e^{-st} dt = \frac{Ae^{-st}}{-s} \Big|_0^\infty = A \left[\frac{e^{-\infty}}{-s} + \frac{e^0}{s} \right] = \frac{A}{s} \quad (2.29)$$

Example 2.2 Consider the exponential function e^{-at} . Its LT is given by

$$\operatorname{LT}\{e^{-at}\} = \int_0^\infty e^{-at} e^{-st} dt = \int_0^\infty e^{-(s+a)t} dt = \frac{e^{-(s+a)t}}{-(s+a)} \Big|_0^\infty = \frac{1}{s+a} \quad (2.30)$$

This result reflects the importance of the state-space modeling in control theory. Specifically, it allows to map a scalar linear differential equation in the time domain into an algebraic expression in the frequency domain and then write its solution to connect the two domains.

$$\dot{x} = -ax \Leftrightarrow X(s) = \frac{1}{s+a}x(0) \Leftrightarrow x(t) = e^{-at}x(0) \quad (2.31)$$

Example 2.3 Consider the scalar transfer function in (2.24) having distinct poles with input $u(t) = A \sin(\omega_0 t) u_s(t)$ then

$$\begin{aligned} U(s) &= \frac{A\omega_0}{s^2 + \omega_0^2} \\ Y(s) &= G(s)U(s) = G(s)\frac{A\omega_0}{s^2 + \omega_0^2} \end{aligned} \quad (2.32)$$

If we perform a partial fraction expansion on $Y(s)$ we have

$$Y(s) = G(s)\frac{A\omega_0}{s^2 + \omega_0^2} = \frac{r_1}{s + p_1} + \frac{r_2}{s + p_2} + \cdots + \frac{r_{n_x}}{s + p_{n_x}} + \frac{\gamma_0}{s + j\omega_0} + \frac{\gamma_0^*}{s - j\omega_0} \quad (2.33)$$

Then the inverse LT gives

$$\begin{aligned} y(t) &= r_1 e^{-p_1 t} + r_2 e^{-p_2 t} + \cdots + r_{n_x} e^{-p_{n_x} t} + 2|\gamma_0| \sin(\omega_0 t + \phi) \\ \phi &= \tan^{-1}\left(\frac{\text{Im}(\gamma_0)}{\text{Re}(\gamma_0)}\right) \end{aligned} \quad (2.34)$$

The result in (2.34) shows that for stable LTI systems the steady-state response to a sine wave in is a sine wave out that has an amplitude and a phase shift. The coefficients r_i in (2.34) are the residues associated with each pole and describe the contribution of that pole to the transient response of the system.

Applying LT (2.26) to the multi-input–multi-output (MIMO) LTI dynamics (2.3) with zero initial conditions, while using the first relation in (2.27), results in

$$\boxed{\begin{aligned} \dot{x} = A x + B u, \quad x(0) = 0 \Rightarrow & s X(s) - \underbrace{x(0)}_0 = A X(s) + B U(s) \\ \Rightarrow & X(s) = (s I_{n_x} - A)^{-1} B U(s) \end{aligned}} \quad (2.35)$$

In addition, the system output can be written as

$$Y(s) = \underbrace{\left(C(s I_{n_x} - A)^{-1} B + D\right)}_{G(s)} U(s) = G(s)U(s) \quad (2.36)$$

where

$$G(s) = C(s I_{n_x} - A)^{-1} B + D \quad (2.37)$$

is the system *transfer function matrix*. It represents a map from the LT input $U(s)$ to the LT output $Y(s)$, with both vector signals defined as complex functions of a complex argument, in the frequency domain.

$$G(s) \in R^{n_y \times n_u} : U(s) \rightarrow Y(s) \quad (2.38)$$

Note that the LT state $X(s)$ (2.35) is eliminated from the input–output definition (2.37). It accentuates the internal nature of the system state [1–3].

State definition is not unique. Any appropriate definition suffices as described in the next section on similarity transformations. At the same time in the frequency domain, only input and output are required to completely characterize the system forced steady-state response with zero initial conditions.

2.4.1.3 Transfer Functions, Block Diagrams, and the Root Locus

The use of the Laplace transform (2.25) allows us to solve, analyze, and understand complicated, high-order linear time-invariant differential equations driven by signals of interest. This is fundamental to what is called classical control theory, in which engineers use transfer functions and block diagrams to model, design, and analyze complicated control systems.

Block diagrams, and the algebra associated with them, allow for detailed modeling and analysis of the control system within the process dynamics. Block diagrams are used to represent complex physical systems and are the primary tool for showing and forming process and control system models. They decompose the system down into distinct elements and models that support all aspects of the design, analysis, and simulation.

Consider the single-input–single-output (SISO) system model shown in Fig. 2.11. There are several important transfer functions that are used to understand and evaluate the control system design. These transfer functions are defined and analyzed in Chap. 3 for both SISO and MIMO systems and are presented here without derivation. They include the open-loop transfer function (the loop gain) at the plant input given by

$$L(s) = K(s)G(s) = G(s)K(s) \quad (2.39)$$

and four closed-loop transfer functions we will call the “gang of four” [4] given by

$$\begin{aligned} \frac{e}{y_{\text{cmd}}} &= S(s) = \frac{1}{1 + L(s)}, \quad \frac{y}{y_{\text{cmd}}} = T(s) = \frac{L(s)}{1 + L(s)}, \\ \frac{u}{n} &= K(s)S(s), \quad \frac{y}{d} = G(s)S(s) \end{aligned} \quad (2.40)$$

In (2.40) $S(s)$ is called the sensitivity, $T(s)$ is the complementary sensitivity, $\frac{u}{n} = K(s)S(s)$ is the control-to-noise transfer function, and $\frac{y}{d} = G(s)S(s)$ is the load disturbance transfer function.

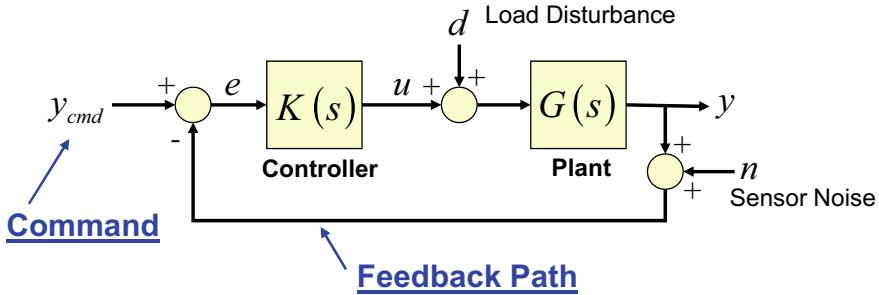


Fig. 2.11 SISO KG block diagram

Control engineers examine these transfer functions in both the frequency domain and the time domain to understand relative stability, performance, noise, and disturbance sensitivity. We will save the frequency domain discussion of these transfer functions for the next chapter where we extend and apply these SISO transfer function concepts to MIMO systems. The time domain aspect will be discussed here.

Each of the closed-loop SISO transfer functions in (2.40) has a numerator and denominator polynomial. The denominator polynomial roots are the poles of the closed-loop system. We will call this polynomial $\phi_{cl}(s)$. From our previous example, (2.34), we know that the time response of a linear system to an input is determined by the locations of the poles of the transfer function (the exponentials in (2.34)). Figure 2.12 shows a plot of the s -plane pole location and the corresponding time response for that pole. The poles that are closest to the origin in the s -plane will dominate the response because they are the slowest. Those furthest from the origin will rapidly decay. Thus, if the time response of the system does not meet requirements, the closed-loop poles must be moved so that requirements are met.

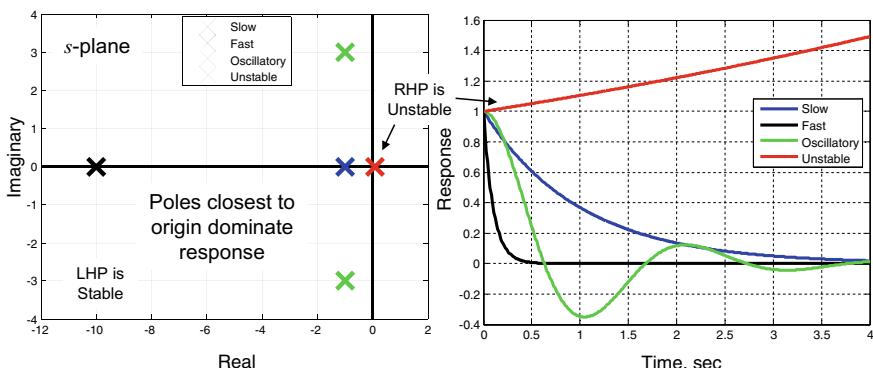


Fig. 2.12 Time response of stable and unstable poles

The root locus plot from classical control theory gives insight into relating the control gain to closed-loop poles location in the s -plane. In addition to analyzing poles, it is critically important in the design of any control system to know where the zeros of the plant are located and more importantly whether they are in the open left half plane (LHP) or in the right half plane (RHP). When feedback loops are closed in a control system, the poles move toward the zeros. The location of the closed-loop poles will determine the speed of response for the system and whether the response will have oscillatory components (complex poles). At the same time, the zeros define regions of attraction for the poles to go to, if the feedback gains are increased to improve closed-loop tracking performance.

In Fig. 2.11, consider a feedback controller with a constant gain K and a plant model with a transfer function $G(s) = n(s)/\phi_{ol}(s)$. The roots of the numerator $n(s)$ are the plant zeros, and the roots of $\phi_{ol}(s)$ are the open-loop plant poles. The return difference transfer function $1+L(s)$ (see Chap. 3 for definition), which is the denominator in the sensitivity and complementary sensitivity transfer functions, gives an expression for the system closed-loop characteristic polynomial $\phi_{cl}(s)$ divided by the open-loop characteristic polynomial $\phi_{ol}(s)$.

$$1 + L(s) = 1 + K \frac{n(s)}{\phi_{ol}(s)} = \frac{\phi_{ol}(s) + Kn(s)}{\phi_{ol}(s)} = \frac{\phi_{cl}(s)}{\phi_{ol}(s)} \quad (2.41)$$

Equating the numerator of (2.41) to zero defines the classical root locus problem: Where do closed-loop poles move as the feedback gain is increased? The root locus analysis is based on the following algebraic equation.

$$\phi_{ol}(s) + Kn(s) = 0 \quad (2.42)$$

Figure 2.13 shows three root locus examples using different transfer functions where the feedback gain K is varied from $K = 0$ to $K = \infty$. For zero gain, the poles of the system are at the open-loop poles, the roots of $\phi_{ol}(s)$. As the gain is increased, the poles move either to finite zeros, roots of $n(s)$, or along asymptotes to infinity. In Fig. 2.13a the transfer function has two LHP zeros, and we see that the entire root locus is contained in the LHP. For different values of gain K the closed-loop poles can be made real or complex. As the feedback gain increases, the poles move further into the LHP. Their absolute values increase, and as a result the closed-loop system response becomes faster.

We see in (a) that a pole approaches the zero at $s = -5$. The zero attracting this pole creates a limitation for this system speed of response. As the gain is increased, the pole absolute value does not increase and the system response does not get any faster. Thus, low frequency zeros in the plant transfer function can limit achievable closed-loop performance.

In Fig. 2.13b the transfer function has a stable LHP zero and an unstable RHP zero. The description of stable versus unstable relates to the type of response as discussed in Fig. 2.12. The root locus plot for this transfer function shows that a constant positive gain K does not produce a stable closed-loop system.

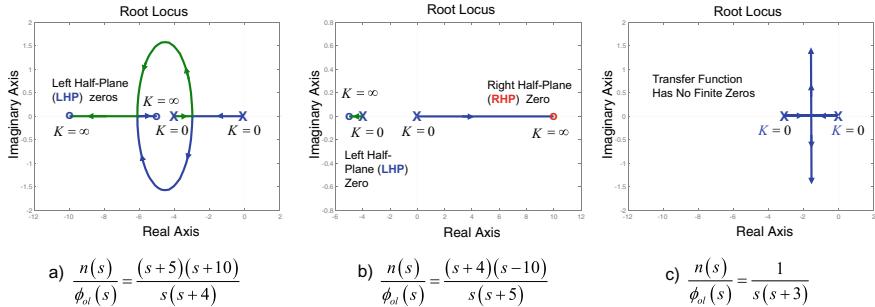


Fig. 2.13 Root locus examples

In Fig. 2.13c we have a root locus plot for a system whose transfer function has no finite zeros. As the gain K is increased, the closed-loop poles break away from the real axis and move toward zeros at infinity. As the poles move out along the asymptotes, the damping in the pair of closed-loop poles decreases, and for a large enough gain, the system response will look like an oscillator, exhibiting low-to-no damping characteristics.

These root locus examples in Fig. 2.13 show that as feedback gains in the system are varied, the closed-loop poles will change in a prescribed way, and their location in the s -plane will be influenced by the zero dynamics of the plant and the controller.

When using state-space models to describe the plant and controller dynamics, it is not directly evident what kind of zero dynamics are present in the model. Control system designers should always form the corresponding transfer functions in order to better understand the plant zero dynamics and its influence on the control design.

2.4.1.4 Common Transfer Functions for Building Controllers

In this section we will discuss the gain and phase characteristics of some common transfer functions used in building controllers and modeling elements of an LTI system. Figure 2.2 presents a generic control system block diagram whose blocks represent mathematical or functional relationships describing elements of the system model. Each block may contain transfer functions, state-space models, or combinations of either. If and when these elements are known to the designer, they allow to predict and analyze the system performance under closed-loop control operation.

Some blocks may model control algorithms, some represent filters, and others could characterize hardware components such as control actuators and measurement sensors. Every dynamic block in the diagram introduces gain and/or phase changes, which in turn impact system stability and performance.

Table 2.1 Common transfer functions used in building controllers

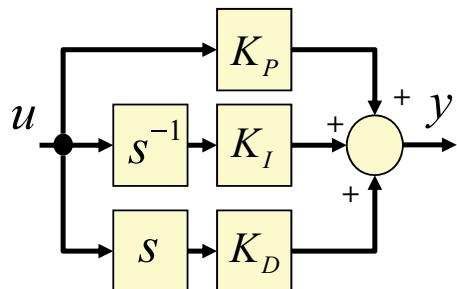
Type	ODE	Transfer function
(1) Integrator	$\dot{y} = u$	$\frac{1}{s}$
(2) Differentiator	$y = \dot{u}$	s
(3) PID controller	$y = K_P u + K_D \dot{u} + K_I \int u$	$\frac{K_D s^2 + K_P s + K_I}{s}$
(4) Low pass filter	$\tau \dot{y} = y + u$	$\frac{1}{\tau s + 1}$
(5) Lead or lag filter (lead $a > 1$) (lag $a < 1$)	$a\tau \dot{y} + y = \tau \dot{u} + u$	$\frac{1+a\tau s}{1+\tau s}$
(6) Damped oscillator	$\ddot{y} + 2\xi\omega_n \dot{y} + \omega_n^2 y = \omega_n^2 u$	$\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$
(7) Notch filter	$\ddot{y} + 2\xi\omega_n \dot{y} + \omega_n^2 y = \ddot{u} + \omega_n^2 u$	$\frac{s^2 + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$
(8) Third-order Butterworth filter	$\ddot{y} + a\ddot{y} + a\dot{y} + y = u$	$\frac{1}{s^3 + as^2 + as + 1}$

Table 2.1 lists several common transfer functions, their related ordinary differential equation (ODE), and the corresponding LT representations. The table contains generic elements that are used to build feedback controllers, filters, and other elements for modeling dynamic subsystems.

The first three rows in Table 2.1 define elements of proportional, integral, and derivative (PID) controllers. Many operational systems today use PID controllers. Figure 2.14 shows a block diagram implementation of a PID control system.

The P-control path amplifies the input with a gain K_P and does not introduce any phase shift to the input signal. The I-control path integrates the input, adding a negative 90-degree phase lag to the input signal. The integrator must “sum” the input before influencing the response, thus the lag. This summing action smooths the response and can reduce any high frequency noise on the input signal. The integrator transfer function has a pole $s = 0$ in the s -plane.

The D -control path differentiates the input and adds a positive 90-degree phase lead to the input. The D -control component and approximations of it (lead compensator, row 6) are very attractive elements because they introduce a phase lead. In position control systems, this element can be used to increase damping in the

Fig. 2.14 Proportional-integral-derivative control

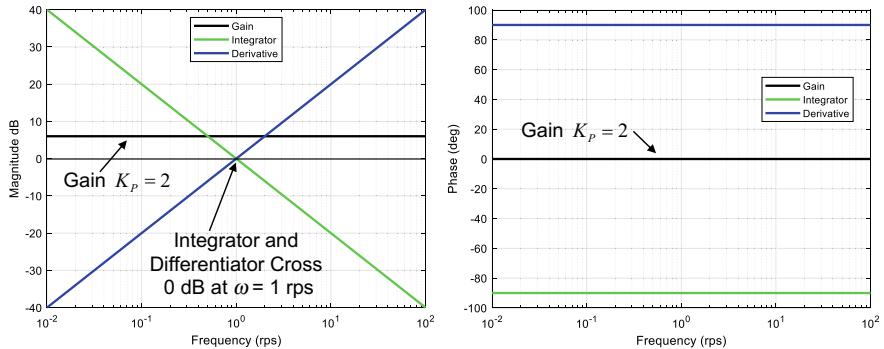


Fig. 2.15 Bode plot of PID control elements

system response. However, if there is a noise on the input signal, the differentiation process can lead to large undesirable responses in the system output.

Figure 2.15 shows a Bode magnitude and phase frequency response plot of the PID control elements. We selected a proportional gain $K_P = 2$ which converts to 6 dB magnitude on the Bode plot. The integrator frequency response has a slope of $-20 \text{ dB per decade}$ of frequency (crossing 0 dB at 1 rps), with -90° degree of phase lag at all frequencies.

The differentiator has a slope of $+20 \text{ dB per decade}$ (also crossing 0 dB at 1 rps), with $+90^\circ$ degree of phase lead at all frequencies.

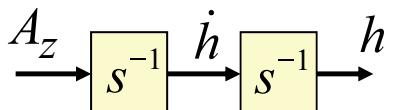
Not shown in the table is a double integrator. This would be described by

$$\ddot{y} = u \xleftrightarrow{\text{LT}} \frac{Y(s)}{U(s)} = \frac{1}{s^2} \quad (2.43)$$

A Bode frequency response for the double integrator would show a slope of $-40 \text{ dB per decade}$, crossing 0 dB at 1 rps, and with -180° degree of phase lag at all frequencies. Double integrators are seldom used for command tracking due to the excessive phase lag. It is very difficult to stabilize a system in the presence of this much lag. Double integrators are used when we integrate vehicle acceleration into its velocity and subsequently into the vehicle position. Figure 2.16 shows an example where vertical acceleration is integrated twice to form altitude rate position signals.

Row 4 of Table 2.1 shows a low pass filter. This is probably the most common transfer function used in control for command tracking, filtering, and modeling subsystems. The parameter τ is called the *time constant* of the filter and defines

Fig. 2.16 Double integrator example



what is called the *corner frequency* of the filter, $\omega_\tau = 1/\tau$. This is the frequency where the filter gain attenuation begins. It passes the frequency content of the input signal at frequencies below ω_τ and attenuates the frequency content above ω_τ . This is best displayed on a Bode plot. Figure 2.17 shows a Bode plot of three low pass filters with different time constants. Since this filter model has unity DC gain (unit magnitude at $s = 0$), it has a magnitude of 0 dB at low frequencies and provides no attenuation. At the corner frequency, the filter produces -3 dB of attenuation. At frequencies higher than the corner frequency the filter produces attenuation at a rate of -20 dB per decade. You will notice that this is the same slope as the integrator shown in Fig. 2.15. Unlike the integrator which has a phase lag that is constant with frequency, the low pass filter introduces phase lag that ranges from 0° at low frequencies to -90° at high frequencies, as shown in Fig. 2.17. The lag starts at a frequency of two decades before the corner frequency and has its maximum value two decades after. At the corner frequency, the filter is producing 45 degree of phase lag. The low pass filter is often combined with other filters in Table 2.1 to provide the required attenuation levels needed in a control system. We will discuss this useful feature in details later in the book.

Example 2.4 Low pass filters are often employed to design or improve command tracking controllers. Consider the following implementation:

$$\frac{u}{e} = \frac{K_P}{\tau s + 1} \quad (2.44)$$

where the tracking error signal e is the input to the filter. It produces the control input u . The transfer function in (2.44) is a low pass filter with a gain. It is commonly referred to as a *leaky integrator*. This type of a transfer function is often used in the

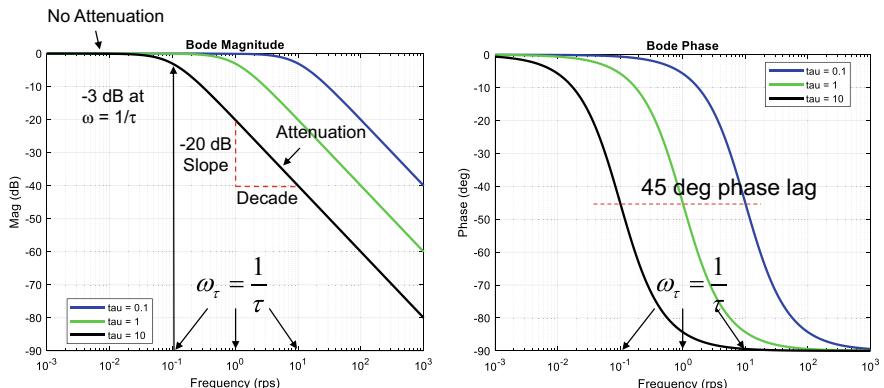


Fig. 2.17 Bode plot of low pass filter

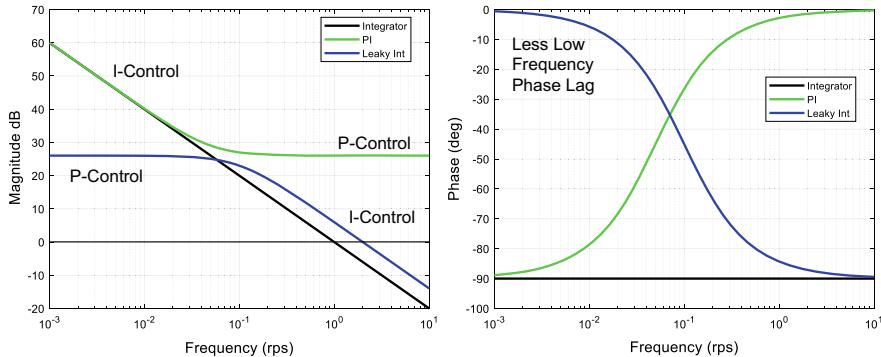


Fig. 2.18 Bode plot comparing leaky integrator with PI and I-control

control of hydraulics. Consider the leaky integrator (2.44) and a PI controller.

$$\text{Leaky integrator : } \frac{u}{e} = \frac{20}{10s + 1}; \quad \text{PI-control : } \frac{u}{e} = \frac{20s + 1}{s} \quad (2.45)$$

Figure 2.18 shows a Bode plot comparison of these two transfer functions along with a pure integrator. Both the I-control and the PI-control provide a *Type 1* control action that will track constant commands with zero errors. The leaky integrator having only proportional gain at low frequency will not do that. There will be a steady-state error for tracking constant commands. It can be made smaller by increasing the numerator gain K_P . In controllers for actuation systems where commands are changing often, the leaky integrator can be a useful alternative to integral control. A clear advantage is in the phase response of the transfer function. A leaky integrator can have less phase lag than a PI controller.

Row 5 of Table 2.1 shows a first-order transfer function that adds a numerator (real zero) to the low pass filter. If the zero is lower in frequency than the low pass corner frequency, $a < 1$, the filter will produce phase lead and is called a *lead filter*. If the zero is higher in frequency than the low pass corner frequency, $a > 1$, the filter will produce a phase lag and because of that it is called a *lag filter*. The pole-zero arrangement in these filters will create two corner frequencies, one at $\omega = 1/aT$ and the other at $\omega = 1/T$.

Lead filters differentiate the signal over a frequency region defined by the zero and the pole. When the slope of the Bode magnitude is positive, it is indicative of a differentiation (see Fig. 2.15) and gives a phase lead. When using a lead filter,

the input signal should not contain noise in the frequency region where differentiation occurs. Lead filters are employed in many control systems to improve phase margins (see Chap. 3). The maximum value of the phase, ϕ_m , produced by a lead filter and the frequency it occurs at, ω_m , can be computed from the magnitude and phase of the transfer function [5].

$$\omega_m = \frac{1}{\sqrt{a}T}; \quad \sin \phi_m = \frac{a-1}{a+1} \quad (2.46)$$

Example 2.5 Consider the lead filter given by

$$\frac{y_f}{y} = \frac{K(10s + 1)}{0.1s + 1} \quad (2.47)$$

with the gain $K = 1$. Figure 2.19 illustrates the pole-zero locations for this filter in the s -plane, and Fig. 2.20 shows its Bode plot data. In the frequency region between 10^{-1} and 10^2 rps, the filter differentiates its input signal and provides a significant amount of phase lead. Note that this filter has a large proportional gain at high frequencies. The gain K in (2.47) can be used to reduce the high frequency gain of the filter ($K < 1$) with no impact to the phase lead added by the filter. In practice, the lead filter would be combined with a low pass filter to roll off the high frequency gain at high frequencies.

Example 2.6 Consider the lag filter given by

$$\frac{y_f}{y} = \frac{K(0.1s + 1)}{10s + 1} \quad (2.48)$$

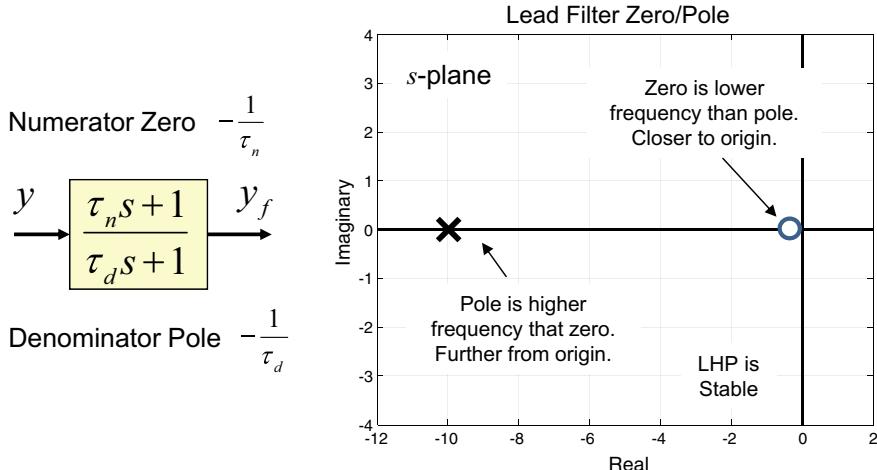


Fig. 2.19 Lead filter pole-zero location in s -plane

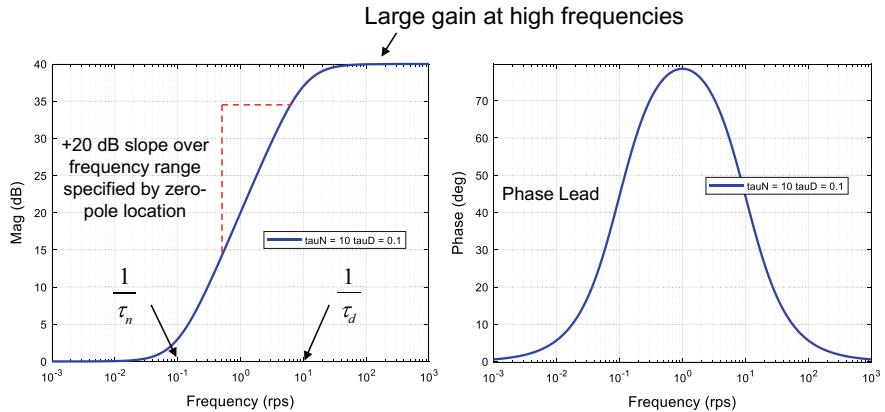


Fig. 2.20 Lead filter Bode plot

with the gain $K = 1$. Figure 2.21 illustrates the pole-zero locations for this filter in the s -plane and Fig. 2.22 shows its Bode plot data. In the frequency region between 10^{-1} and 10^2 rps, this filter attenuates (integrates) its input signal and introduces a phase lag.

Row 6 of Table 2.1 shows a second-order transfer function that has no numerator dynamics. This transfer function can be used as a second-order filter (same as using two low pass filter) to provide increased attenuation with the penalty of increased phase lag. This transfer function is also a model of a spring-mass-damper

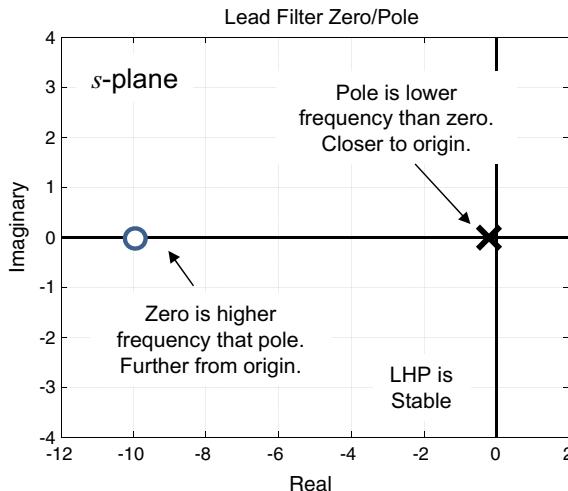


Fig. 2.21 Lag filter pole-zero location in s -plane

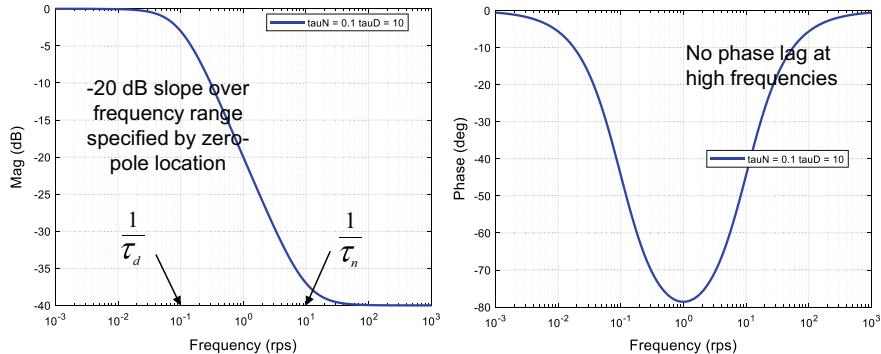


Fig. 2.22 Lag filter Bode plot

mechanical system. If the damping factor ζ in the denominator is less than one, the poles will be complex. The filter frequency response (discussed in the next section) can have a peak resonance at a corner frequency whose peak depends upon the damping factor. In the next section we will discuss using this transfer function in modeling for certain aircraft subsystems.

Row 7 of Table 2.1 shows another second-order transfer function that is called a *notch filter*. This is a very important transfer function widely used in aerospace control systems. It is typically designed to remove flexible body dynamics (“modes”) from sensor measurements to prevent feedback propagation of undesirable dynamics. The addition of a pair of complex zeros to the numerator creates the “notch” and filters out the frequency ω_n from its input signal, passing those frequencies outside the notch on both sides. The damping factor in the denominator makes the notch narrow or wide and impacts its depth (how much attenuation is achieved at the center frequency ω_n). Figure 2.23 shows a Bode plot of a notch filter in which the damping factor in the denominator polynomial is varied to show the widening of the notch, at the expense of more phase lag. If the notch center frequency is low and/or close to the control system loop gain-crossover frequency (see Chap. 3), the associated phase lag from the notch can significantly decrease the phase margin.

If a damping term is added to the notch numerator polynomial (creating complex zeros not on the $j\omega$ axis), the phase lag from the filter can be reduced. This is given by

$$\frac{y_f}{y} = \frac{s^2 + 2\zeta_n\omega_n s + \omega_n^2}{s^2 + 2\zeta_d\omega_n s + \omega_n^2} \quad (2.49)$$

Figure 2.24 shows a Bode plot comparison of (2.49) varying the numerator damping coefficient ζ_n . Larger values of ζ_n form a notch that is shallow. This design modification significantly reduces the phase lag associated with the filter but decreases the attenuation at the notch center frequency (depth of the notch).

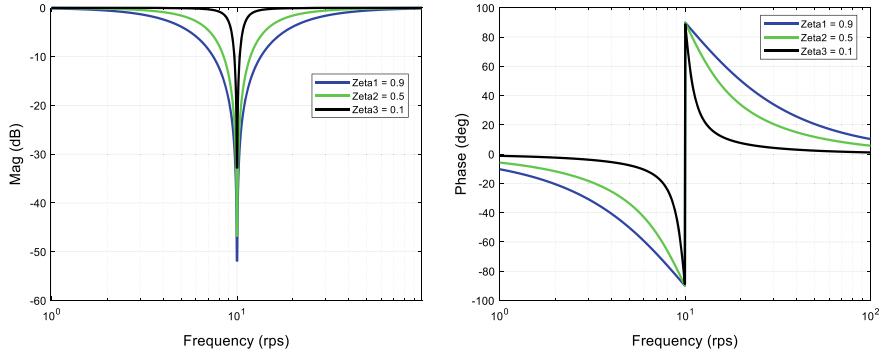


Fig. 2.23 Bode plot of notch filter

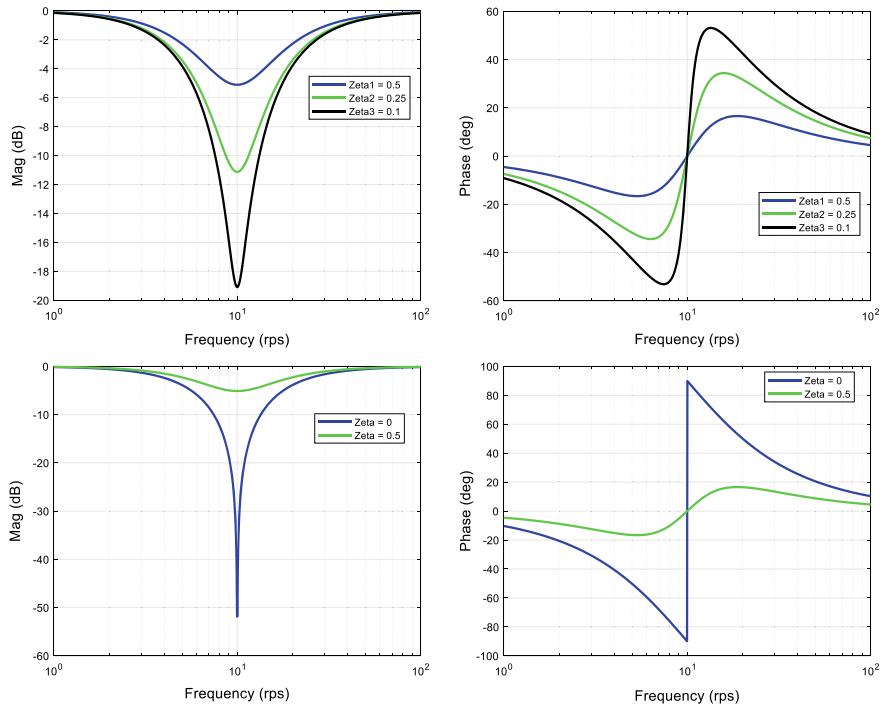


Fig. 2.24 Bode plot of notch filter with numerator damping

Row 8 of Table 2.1 presents a third-order Butterworth filter. Butterworth filters are used to attenuate frequencies in a signal above a specific frequency range, similar to what a low pass filter would do. Butterworth filters come in various orders. These filters can roll off the system loop gain, providing significant attenuation at frequencies past their center frequency, with less phase lag than a filter composed

of low pass filters of the same order. The poles of the Butterworth filter have a unique angular displacement pattern in the s -plane. They are always stable, and the complex roots fall on equally spaced angles between each other.

Example 2.7 Consider the following Butterworth filter given by

$$\frac{y_f}{y} = \frac{1}{1 + 2s + 2s^2 + s^3} \quad (2.50)$$

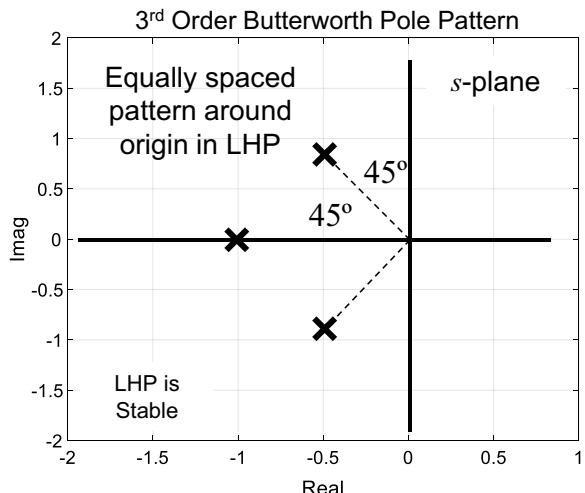
and compare it to a third-order low pass filter given by

$$\frac{y_f}{y} = \frac{1}{(s + 1)^3} \quad (2.51)$$

Figure 2.25 shows the pole location in the s -plane of (2.50). Note the equally spaced angles between the complex-conjugate poles. Figure 2.26 shows a Bode plot comparison between (2.50) and (2.51). Since this is a third-order filter it produces attenuation at a rate of -60 dB per decade. We see from the phase plot that the Butterworth filter produces less phase lag than the first-order low pass cubed filter.

Our discussion of the transfer functions in Table 2.1 focused on the gain and phase characteristics and how they can be used in building a controller. When using any of these control elements it is important to understand both the magnitude and the phase contributions they provide. Of particular importance is selecting components that minimize phase lag so as to improve stability margins of the overall design. As we will discuss later in Chap. 3, it is important to roll off the loop gain to create a control system that is robust to sensor noise and high frequency dynamics.

Fig. 2.25 Third-order Butterworth filter pole pattern in s -plane



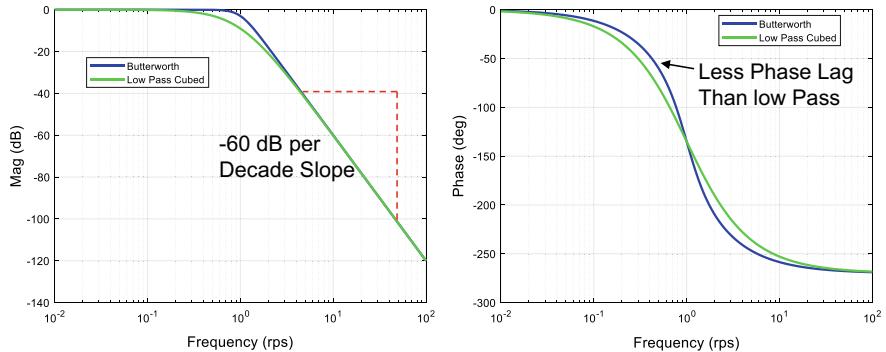


Fig. 2.26 Butterworth filter Bode plot

2.4.1.5 Modeling Systems and Subsystems with Transfer Functions

In aerospace control engineering, it is often important to model a system or a subsystem using a low-order linear model described by a transfer function. This is done in order to capture both the time response dynamics and the gain and phase characteristics of the subsystem for stability analysis. The block diagram in Fig. 2.2 lists common elements used in modeling a flight control system.

Actuators are typically modeled with the second-order model, given by

$$\frac{\delta}{\delta_c} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2.52)$$

where δ is the actuator surface position in radians and δ_c is the commanded position, also in radians. These dynamics are generally low frequency. It is important to select the damping factor ξ and the natural frequency ω_n to model the true physical hardware. Figure 2.27 shows a step response and a frequency response of a second-order transfer function for various damping levels and with the natural frequency normalized to 1. When underdamped, this second-order model will have an oscillatory response. It will introduce gain amplification at certain frequencies and contribute up to -180 degree of phase lag. When analyzed in a flight control system this will degrade stability margins.

Sensor models are usually high frequency in nature and are typically modeled using a first-order transfer function given by

$$\frac{y_m}{y} = \frac{1}{\tau s + 1} \quad (2.53)$$

where y_m is the measured signal, y is the plant-output variable, and τ is the transfer function time constant in seconds. This model does not produce gain amplification and will contribute up to -90 degree of phase lag.

Time delays as shown in Fig. 2.2 are included to model the phase lag associated with using a computer to implement the control system. Time delays at both inputs-to and outputs-from the computer are modeled. Time delays can also be inserted

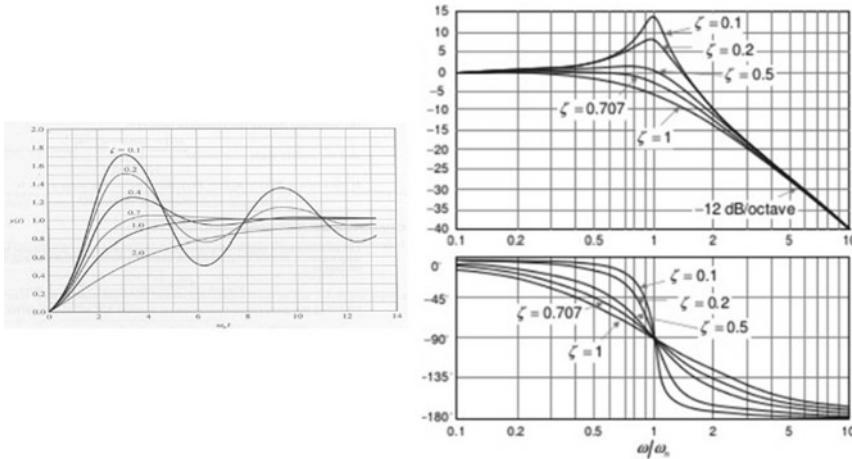


Fig. 2.27 Second-order transfer function frequency response

to represent lag effects associated with unknown subsystems. Dynamics of a time delay is best modeled using the complex exponential e^{-Ts} directly in the analysis, where T is the delay in seconds. When evaluated in the frequency domain, the complex exponential has magnitude and phase given by

$$e^{-Tj\omega} = 1 \angle -\omega T \quad (2.54)$$

When a transfer function or a state-space model of the time delay is needed, a series approximation such as

$$e^{-Ts} \cong \frac{1}{[1 + (Ts/n)]^n} \quad n \rightarrow \infty \quad (2.55)$$

or a Pade' approximation in the form

$$e^{-Ts} \cong \frac{1 - Ts/2}{1 + Ts/2} \quad (2.56)$$

can be used.

Example 2.8 Figure 2.28 shows a phase lag comparison between the complex exponential, a third-order series approximation, and a Pade' approximation for two different time delays. It is important when selecting a time delay model that the model phase lag be accurate at or near the systems loop gain-crossover frequency.

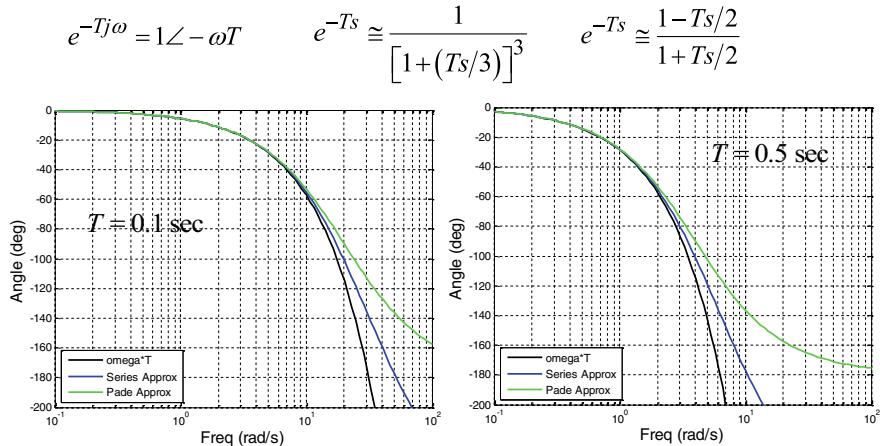


Fig. 2.28 Comparison of time delay models

Example 2.9 Consider the loop gain transfer function

$$L(s) = \frac{e^{-Ts}}{s(s+1)(s+2)} \quad (2.57)$$

Figure 2.29 shows a Nyquist plot of (2.57) for a time delay of $T = 0.5$ and $T = 1.0$ s using a third-order series approximation ($n = 3$) in (2.55), along with Pade' approximation (2.56). We see that larger time delays introduce more phase lag, which in turn degrades the system phase margin.

Figure 2.30 shows an example of using transfer functions to model the acceleration response characteristics of a flight control system with a first-order transfer function. The time constant in the transfer function is chosen as the 63% rise time

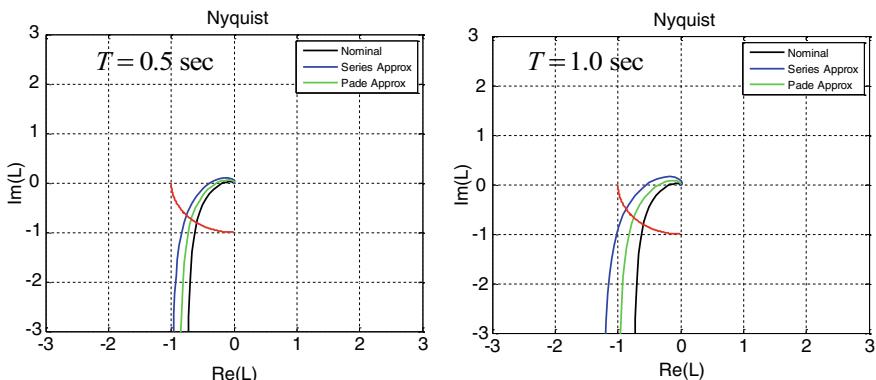


Fig. 2.29 Nyquist analysis of time delays

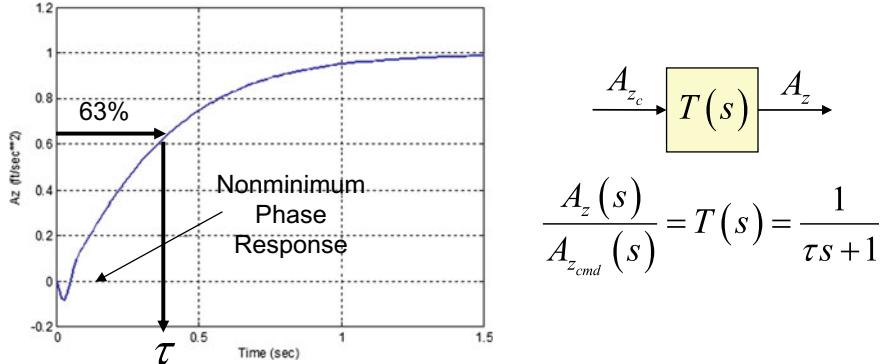


Fig. 2.30 Approximation of an acceleration command inner-loop control system with a first-order model

in the acceleration step response. This model will produce a simple exponential response with no overshoot and does not capture the non-minimum phase response shown in the time history.

2.4.2 Control-Oriented Models for Linear Time-Invariant Systems

In this section we present state-space LTI plant and controller models for control design and analysis that will be used in subsequent chapters. These models are defined for the controller and the plant in Fig. 2.2 and will be used to form closed-loop simulation and open-loop frequency domain analysis models at both the plant input and the plant output.

The plant model is

$$\begin{aligned}\dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u\end{aligned}\quad (2.58)$$

where $x \in R^{n_x}$ is the state, $u \in R^{n_u}$ is the control input, and $y \in R^{n_y}$ is the system output. The real matrices (A_p, B_p, C_p, D_p) are of appropriate dimension and describe the dynamics of the plant.

The controller model is

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} y_{cmd} \\ u &= C_c x_c + D_{c1} y + D_{c2} y_{cmd}\end{aligned}\quad (2.59)$$

where $x_c \in R^{n_{xc}}$ is the controller state vector, u the control input from (2.58), y the output from (2.58), and $y_{\text{cmd}} \in R^{n_u}$ represents the external, possibly time-varying, command vector. The real matrices $(A_c, B_{c1}, B_{c2}, C_c, D_{c1}, D_{c2})$ are of appropriate dimensions and describe the controller dynamics (2.59).

Example 2.10 In order to cast a commonly used static proportional state feedback controller such as,

$$u = -K x \quad (2.60)$$

into the form of (2.59), we first choose $C_p = I_{n_x}$, $D_p = 0_{n_x \times n_u}$ in (2.58). This gives $y = x$. Then, we define $C_c = -K$ in (2.59) and set the rest of the matrices in that equation to zero.

Next we connect the generic controller (2.59) to the plant model (2.58), derive state-space models for the closed-loop system, and then derive the loop gain at the plant input and output breakpoints. When building these models, one should simulate the closed-loop system to make sure that the model is correctly connected with the minus signs inserted where appropriate (to represent negative feedback connections). After that, we would use the system loop gain models to compute the necessary frequency responses. Note that both the plant and the controller may have feedforward connections, with nonzero D -matrices. The feedforward terms must be properly handled, when forming the closed-loop system dynamics.

We will start with substituting the plant-output equation into the control law.

$$\begin{aligned} u &= C_c x_c + D_{c1} y + D_{c2} y_{\text{cmd}} = C_c x_c + D_{c1} (C_p x + D_p u) \\ &\quad + D_{c2} y_{\text{cmd}} \\ \underbrace{(I_{n_u} - D_{c1} D_p)}_Z u &= C_c x_c + D_{c1} C_p x + D_{c2} y_{\text{cmd}} \\ u &= Z^{-1} (C_c x_c + D_{c1} C_p x + D_{c2} y_{\text{cmd}}) \end{aligned} \quad (2.61)$$

We must assume that matrix Z in (2.61) is invertible. This makes the overall problem formulation well posed. Substituting (2.61) into the plant model (2.58) yields

$$\begin{aligned} \dot{x} &= A_p x + B_p Z^{-1} (C_c x_c + D_{c1} C_p x + D_{c2} y_{\text{cmd}}) = (A_p + B_p Z^{-1} D_{c1} C_p) x \\ &\quad + B_p Z^{-1} C_c x_c + B_p Z^{-1} D_{c2} y_{\text{cmd}} \end{aligned} \quad (2.62)$$

We can also substitute the system output y into the controller.

$$\begin{aligned}
\dot{x}_c &= A_c x_c + B_{c1} (C_p x + D_p u) + B_{c2} y_{cmd} \\
&= A_c x_c + B_{c1} (C_p x + D_p Z^{-1} (C_c x_c + D_{c1} C_p x + D_{c2} r)) + B_{c2} y_{cmd} \\
&= (A_c + B_{c1} D_p Z^{-1} C_c) x_c + B_{c1} (I_{n_y} + D_p Z^{-1} D_{c1}) C_p x \\
&\quad + (B_{c2} + B_{c1} D_p Z^{-1} D_{c2}) y_{cmd}
\end{aligned} \tag{2.63}$$

Let us define an augmented state vector in the form

$$x_a = (x^T \ x_c^T)^T \tag{2.64}$$

Then the closed-loop system dynamics can be written as

$$\begin{pmatrix} \dot{x} \\ \dot{x}_c \end{pmatrix} = \underbrace{\begin{pmatrix} A_p + B_p Z^{-1} D_{c1} C_p & B_p Z^{-1} C_c \\ B_{c1} (I_{n_y} + D_p Z^{-1} D_{c1}) C_p & A_c + B_{c1} D_p Z^{-1} C_c \end{pmatrix}}_{A_{cl}} \begin{pmatrix} x \\ x_c \end{pmatrix} + \underbrace{\begin{pmatrix} B_p Z^{-1} D_{c2} \\ B_{c2} + B_{c1} D_p Z^{-1} D_{c2} \end{pmatrix}}_{B_{cl}} y_{cmd} \tag{2.65}$$

or, equivalently

$$\dot{x}_a = A_{cl} x_a + B_{cl} y_{cmd} \tag{2.66}$$

This is the closed-loop model. Its output can be easily defined as follows:

$$\begin{aligned}
y &= C_p x + D_p u = C_p x + D_p Z^{-1} (C_c x_c + D_{c1} C_p x + D_{c2} y_{cmd}) \\
&= (C_p + D_p Z^{-1} D_{c1} C_p \ D_p Z^{-1} C_c) \begin{pmatrix} x \\ x_c \end{pmatrix} + D_p Z^{-1} D_{c2} y_{cmd} \\
&= \underbrace{\begin{pmatrix} (I_{n_y} + D_p Z^{-1} D_{c1}) C_p & D_p Z^{-1} C_c \end{pmatrix}}_{C_{cl}} \begin{pmatrix} x \\ x_c \end{pmatrix} + \underbrace{D_p Z^{-1} D_{c2}}_{D_{cl}} y_{cmd}
\end{aligned} \tag{2.67}$$

and so, the closed-loop system output becomes

$$y = C_{cl} x_a + D_{cl} y_{cmd} \tag{2.68}$$

Equations (2.65) and (2.68) give the state-space model $(A_{cl}, B_{cl}, C_{cl}, D_{cl})$ for the closed-loop system.

The loop gain model at the plant input, $L_u(s)$, is formed to support frequency domain analysis of the design at the plant-input loop breakpoint. In this model, we treat the control input to the plant as the model input u_{in} . The control output

from the controller becomes the model output u_{out} . Also, we neglect the command vector y_{cmd} . In this case, the plant and the controller models are

$$\begin{aligned}\dot{x} &= A_p x + B_p u_{\text{in}} \\ y &= C_p x + D_p u_{\text{in}}\end{aligned}\quad (2.69)$$

and

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} y \\ u_{\text{out}} &= C_c x_c + D_{c1} y\end{aligned}\quad (2.70)$$

We can connect these two systems with u_{in} as the input and u_{out} as the output,

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} (C_p x + D_p u_{\text{in}}) = A_c x_c + B_{c1} C_p x + B_{c1} D_p u_{\text{in}} \\ u_{\text{out}} &= C_c x_c + D_{c1} (C_p x + D_p u_{\text{in}}) + D_{c2} r = C_c x_c + D_{c1} C_p x + D_{c1} D_p u_{\text{in}}\end{aligned}\quad (2.71)$$

and rewrite these relations in matrix form.

$$\begin{aligned}\begin{pmatrix} \dot{x} \\ \dot{x}_c \end{pmatrix} &= \underbrace{\begin{pmatrix} A_p & 0 \\ B_{c1} C_p & A_c \end{pmatrix}}_{A_{L_u}} \begin{pmatrix} x \\ x_c \end{pmatrix} + \underbrace{\begin{pmatrix} B_p \\ B_{c1} D_p \end{pmatrix}}_{B_{L_u}} u_{\text{in}} \\ u_{\text{out}} &= \underbrace{\begin{pmatrix} D_{c1} C_p & C_c \end{pmatrix}}_{C_{L_u}} \begin{pmatrix} x \\ x_c \end{pmatrix} + \underbrace{\begin{pmatrix} D_{c1} D_p \end{pmatrix}}_{D_{L_u}} u_{\text{in}}\end{aligned}\quad (2.72)$$

The system loop gain at the plant input is

$$L_u(s) = C_{L_u} (s I_{n_x+n_{x_c}} - A_{L_u})^{-1} B_{L_u} + D_{L_u} \quad (2.73)$$

and it is completely defined by the matrix quadruple $(A_{L_u}, B_{L_u}, C_{L_u}, D_{L_u})$.

Similarly, the loop gain model at the plant output, $L_y(s)$, is formed to support frequency domain analysis of the design at the plant-output loop breakpoint. In this model, we treat the plant output feeding the controller as the model input, y_{in} , the plant output from the plant as the model output, y_{out} , and neglect the command vector r . The plant and controller models are

$$\begin{aligned}\dot{x} &= A_p x + B_p u \\ y_{\text{out}} &= C_p x + D_p u\end{aligned}\quad (2.74)$$

and

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c_1} y_{\text{in}} \\ u &= C_c x_c + D_{c_1} y_{\text{in}}\end{aligned}\quad (2.75)$$

Connecting these two systems with y_{in} as the input and y_{out} as the output yields

$$\boxed{\begin{aligned}\dot{x} &= A_p x + B_p (C_c x_c + D_{c_1} y_{\text{in}}) = A_p x + B_p C_c x_c + B_p D_{c_1} y_{\text{in}} \\ y_{\text{out}} &= C_p x + D_p (C_c x_c + D_{c_1} y_{\text{in}}) = C_p x + D_p C_c x_c + D_p D_{c_1} y_{\text{in}}\end{aligned}} \quad \Downarrow \quad \boxed{\begin{aligned}\left(\begin{array}{l} \dot{x} \\ \dot{x}_c \end{array}\right) &= \underbrace{\begin{pmatrix} A_p & B_p C_c \\ 0_p & A_c \end{pmatrix}}_{A_{L_y}} \left(\begin{array}{l} x \\ x_c \end{array}\right) + \underbrace{\begin{pmatrix} B_p D_{c_1} \\ B_{c_1} \end{pmatrix}}_{B_{L_y}} y_{\text{in}} \\ y_{\text{out}} &= \underbrace{\begin{pmatrix} C_p & D_p C_c \\ C_{L_y} & \end{pmatrix}}_{C_{L_y}} \left(\begin{array}{l} x \\ x_c \end{array}\right) + \underbrace{\begin{pmatrix} D_p D_{c_1} \\ D_{L_y} \end{pmatrix}}_{D_{L_y}} y_{\text{in}}\end{aligned}} \quad (2.76)}$$

So, the loop gain at the plant output is defined by $(A_{L_y}, B_{L_y}, C_{L_y}, D_{L_y})$.

$$L_y(s) = C_{L_y} (s I_{n_x+n_{x_c}} - A_{L_y})^{-1} B_{L_y} + D_{L_y} \quad (2.77)$$

The derived loop gains, (2.73) and (2.77), become essential tools to analyze relative stability properties of closed-loop linear time-invariant systems in the frequency domain.

2.4.3 State-Space Similarity Transformations

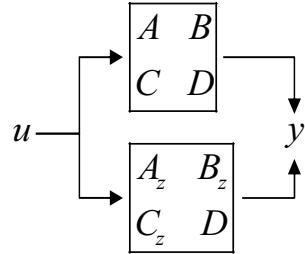
The state definition for an LTI system is not unique. Consider a non-singular matrix $T \in R^{n \times n}$, and define the state-space similarity transformation.

$$z = T x \quad (2.78)$$

This transformation is essentially a change of basis vectors in the system state space. Then the state-space representation of the LTI dynamics (2.5) can be written in terms of the new state z .

$$\left. \begin{aligned}\dot{z} &= \underbrace{T A T^{-1}}_{A_z} z + \underbrace{T B}_{B_z} u \\ y &= \underbrace{C T^{-1}}_{C_z} z + D u\end{aligned} \right\} \Leftrightarrow \boxed{\begin{pmatrix} A_z & B_z \\ C_z & D \end{pmatrix} \in R^{(n_n+n_y) \times (n_x+n_u)}} \quad (2.79)$$

Fig. 2.31 Similarity transformation preserves LTI input–output dynamics



This proves an equivalence (i.e., similarity) between the original and the transformed system.

$$\begin{aligned} \dot{x} = Ax + Bu \\ y = Cx + Du \end{aligned} \Leftrightarrow \boxed{\begin{pmatrix} A & B \\ C & D \end{pmatrix} \in R^{(n_n+n_y) \times (n_x+n_u)}} \\ \Leftrightarrow \boxed{\begin{pmatrix} A_z & B_z \\ C_z & D \end{pmatrix} \in R^{(n_n+n_y) \times (n_x+n_u)}} \Leftrightarrow \begin{cases} \dot{z} = A_z z + B_z u \\ y = C_z z + Du \end{cases} \quad (2.80)$$

At the same time, it is not so difficult to show that the system input-to-output relationship (2.36) remains the same. In other words, the LTI system transfer function matrix (2.37) is invariant with respect to the state similarity transformation (Fig. 2.31).

The state similarity transformation is one of the important concepts in linear systems and control. It allows to express LTI dynamics in special forms to support analysis of the system fundamental properties such as stability, controllability, and observability.

2.4.4 Eigenvalues and Eigenvectors

The *characteristic equation* (CE) of an LTI system can be formed from taking the LT of the corresponding differential equations or using the denominator of the transfer function for the system. CE can also be calculated from the system state-space model. The roots of the CE are the poles of the transfer function. The sign of real part of the roots determines stability (LHP), marginal stability (on $j\omega$ axis), or instability (RHP) for the dynamics being modeled.

Consider the calculation of the transfer function given in (2.37).

$$G(s) = C(s I_{n_x} - A)^{-1} B + D = C \frac{\text{adj}(s I_{n_x} - A)}{\det(s I_{n_x} - A)} B + D \quad (2.81)$$

By setting the denominator polynomial of the transfer function matrix to zero, we get the CE.

$$\det(sI_{n_x} - A) = 0 \quad (2.82)$$

Definition 2.1 The variable λ_i is an *eigenvalue* of a square matrix A if there exists a nonzero vector v^i satisfying the equation

$$Av^i = \lambda_i v^i \quad (2.83)$$

The determinant expansion of the matrix $(\lambda I_{n_x} - A)$ forms an n_x -order polynomial in λ whose roots are the eigenvalues of A . Thus, $\det(\lambda I_{n_x} - A) = 0$ gives the eigenvalues $\lambda_1, \dots, \lambda_{n_x}$. The nonzero vector v^i in (2.83) is called the *eigenvector* of A associated with the eigenvalue λ_i . Equation (2.83) can be written as

$$(A - \lambda_i I_{n_x})v^i = 0 \quad \text{or} \quad (\lambda_i I_{n_x} - A)v^i = 0 \quad (2.84)$$

When the eigenvalue of A is inserted into the matrix $(\lambda I_{n_x} - A)$, the matrix drops rank creating a null space. In (2.84) we see that the eigenvector v^i lies on the right-hand side of the matrix $(\lambda_i I_{n_x} - A)$. For that reason, it is called the right eigenvector. For each eigenvalue λ_i we have a corresponding eigenvector v^i . As a set, $[v^1 \dots v^{n_x}]$, forms a matrix of the *right eigenvectors* of A .

The vectors w^i that solve

$$(w^i)^T (\lambda_i I_{n_x} - A) = 0 \quad (2.85)$$

are called the *left eigenvectors* of A .

If the eigenvalues are distinct, then the corresponding n_x eigenvectors are linearly independent and they form a basis for the state space.

Consider a change of basis (the similarity transformation) as in (2.78) in which T^{-1} contains right eigenvectors and T is comprised of the left eigenvectors of A . That is,

$$\begin{aligned} \dot{z} &= \underbrace{T A T^{-1}}_{\Lambda} z + \underbrace{T B}_{\Gamma} u = \Lambda z + \Gamma u \\ y &= \underbrace{C T^{-1}}_{C_z} z + D u = C_z z + D u \end{aligned} \quad (2.86)$$

The resulting transformed model in (2.86) is called the *modal form*. If eigenvalues are distinct then Λ is diagonal and its diagonal elements are the eigenvalues of A . In (2.86)

$$\Lambda = T A T^{-1} \quad \text{and} \quad A = T^{-1} \Lambda T \quad (2.87)$$

Example 2.11 Consider the LTI system with state distribution matrix A and $(\lambda I_{n_x} - A)$ given by

$$A = \begin{bmatrix} -1 & 1 \\ 1/2 & -3/2 \end{bmatrix}; \quad (\lambda I_{n_x} - A) = \begin{bmatrix} \lambda + 1 & -1 \\ -1/2 & \lambda + 3/2 \end{bmatrix} \quad (2.88)$$

The CE is $\det(\lambda I - A) = \lambda^2 + 5/2\lambda + 1 = (\lambda + 2)(\lambda + 1/2)$ indicating the eigenvalues at $-1/2$ and -2 . The eigenvector associated with $\lambda_1 = -1/2$ is computed from

$$\begin{bmatrix} -1/2 + 1 & -1 \\ -1/2 & -1/2 + 3/2 \end{bmatrix} v^1 = \begin{bmatrix} 1/2 & -1 \\ -1/2 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \Rightarrow 1/2v_1 + v_2 = 0 \\ \Rightarrow v^1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad (2.89)$$

The eigenvector associated with $\lambda_1 = -2$ is computed from

$$\begin{bmatrix} -2 + 1 & -1 \\ -1/2 & -2 + 3/2 \end{bmatrix} v^2 = \begin{bmatrix} -1 & -1 \\ -1/2 & -1/2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \Rightarrow -v_1 - v_2 = 0 \\ \Rightarrow v^2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (2.90)$$

Thus from (2.87)

$$T^{-1} = \begin{bmatrix} 2 & 1 \\ 1 & -1 \end{bmatrix} \quad T = \begin{bmatrix} 1/3 & 1/3 \\ 1/3 & -2/3 \end{bmatrix} \\ \Lambda = T A T^{-1} = \begin{bmatrix} 1/3 & 1/3 \\ 1/3 & -2/3 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1/2 & -3/2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} -1/2 & 0 \\ 0 & -2 \end{bmatrix} \\ A = T^{-1} \Lambda T = \begin{bmatrix} 2 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} -1/2 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} 1/3 & 1/3 \\ 1/3 & -2/3 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1/2 & -3/2 \end{bmatrix} \quad (2.91)$$

2.4.5 Computing the State Transition Matrix

As defined in (2.9), the state transition matrix $\Phi(t, t_0) = e^{A(t-t_0)}$ describes the homogenous solution to the unforced LTI state equation, starting at an initial condition x_0 .

$$\boxed{\dot{x}(t) = A x(t), \quad x(t_0) = x_0} \Rightarrow \boxed{x(t) = \underbrace{e^{A(t-t_0)}}_{\Phi(t,t_0)} x_0 = \Phi(t, t_0)x_0} \quad (2.92)$$

Equation (2.7) shows that $\Phi(t, t_0)$ results from an infinite series expansion that converges. We will present two approaches for analytically forming $\Phi(t, t_0)$ when one needs to compute this matrix. The first method uses a similarity transformation whose transformation matrix is made of the eigenvectors of the A -matrix, (see previous section for discussion on eigenvectors). When A is diagonal, $A = \text{diag}(a_{ii})$, $i = 1, \dots, n_x$, the matrix exponential is easily computed as

$$e^{At} = \begin{pmatrix} e^{a_{11}t} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & e^{a_{n_x n_x} t} \end{pmatrix} = \text{diag}(e^{a_{ii}t}) \quad (2.93)$$

Consider the infinite series expansion in (2.7) where $A = T^{-1}\Lambda T$ from (2.87), and this yields

$$\begin{aligned} e^{T^{-1}\Lambda T} &= I_{n_x} + T^{-1}\Lambda T t + \frac{(T^{-1}\Lambda T t)^2}{2!} + \dots = \sum_{k=0}^{\infty} \frac{(T^{-1}\Lambda T t)^k}{k!} = T^{-1} \left(\sum_{k=0}^{\infty} \frac{(\Lambda t)^k}{k!} \right) T \\ &= T^{-1}(e^{\Lambda t})T \end{aligned} \quad (2.94)$$

Using (2.93),

$$e^{At} = e^{T^{-1}\Lambda T} = T^{-1}(e^{\Lambda t})T = T^{-1}\text{diag}(e^{\lambda_i t})T \quad (2.95)$$

where T^{-1} contains right eigenvectors and T the left eigenvectors of A .

The state transition matrix in (2.92) satisfies the following properties [4]:

- (1) $\Phi(t, t) = I_{n_x}$
 - (2) $\Phi^{-1}(t, t_0) = \Phi(t_0, t)$
 - (3) $\Phi(t_2, t_0) = \Phi(t_2, t_1)\Phi(t_1, t_0)$
 - (4) $\dot{\Phi}(t, t_0) = A(t)\Phi(t, t_0)$
- (2.96)

Example 2.12 Consider the LTI system with diagonal state distribution matrix A given by

$$A = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (2.97)$$

Since A is diagonal, the diagonal elements are its eigenvalues. The state transition matrix for diagonal matrices with distinct eigenvalues ($T = I_{n_x}$ in (2.95)) is a diagonal matrix with exponentials given by

$$\Phi(t, t_0) = e^{A(t-t_0)} = \begin{bmatrix} e^{\lambda_1(t-t_0)} & 0 \\ 0 & e^{\lambda_2(t-t_0)} \end{bmatrix} \quad (2.98)$$

Now consider a matrix with repeated eigenvalues, given by

$$A = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix} \quad (2.99)$$

For this matrix, the state transition matrix is given by

$$\Phi(t, t_0) = e^{A(t-t_0)} = \begin{bmatrix} e^{\lambda(t-t_0)} & te^{\lambda(t-t_0)} \\ 0 & e^{\lambda(t-t_0)} \end{bmatrix} \quad (2.100)$$

Example 2.13 Consider the LTI system with state distribution matrix A from Example 2.11. From (2.95) e^{At} can be computed as

$$\begin{aligned} e^{At} &= e^{T^{-1}\Lambda T} = T^{-1}(e^{\Lambda t})T \\ &= \begin{bmatrix} 2 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} e^{-1/2t} & 0 \\ 0 & e^{-2t} \end{bmatrix} \begin{bmatrix} 1/3 & 1/3 \\ 1/3 & -2/3 \end{bmatrix} \\ &= \begin{bmatrix} 1/3e^{-2t} + 2/3e^{-1/2t} & 2/3e^{-1/2t} - 2/3e^{-2t} \\ 1/3e^{-1/2t} - 1/3e^{-2t} & 2/3e^{-2t} + 1/3e^{-1/2t} \end{bmatrix} \end{aligned} \quad (2.101)$$

The second method for computing the state transition matrix $\Phi(t, t_0)$ uses the Laplace transform (LT) approach, (2.25), from classical control theory. Here we have

$$\dot{x}(t) = Ax(t), \quad x(t_0) = x_0 \quad (2.102)$$

Taking the LT of (2.102) yields

$$\begin{aligned} sX(s) - x_0 &= AX(s) \\ sX(s) - AX(s) &= x_0 \\ (sI_{n_x} - A)X(s) &= x_0 \\ X(s) &= (sI_{n_x} - A)^{-1}x_0 \end{aligned} \quad (2.103)$$

Taking the inverse LT yields

$$x(t) = \text{LT}^{-1}\left\{(sI_{n_x} - A)^{-1}\right\}x_0 \quad (2.104)$$

Equating this to (2.92) gives

$$\text{LT}^{-1}\left\{(sI_{n_x} - A)^{-1}\right\} = \Phi(t, t_0) \quad (2.105)$$

The matrix $(sI_{n_x} - A)^{-1}$ will be a matrix of transfer functions in which the inverse LT will produce exponentials as in (2.95).

Example 2.14 Consider the diagonal matrix A from Example 2.12. The matrix $(sI_{n_x} - A)$ is

$$(sI_{n_x} - A) = \begin{bmatrix} s - \lambda_1 & 0 \\ 0 & s - \lambda_2 \end{bmatrix} \quad (2.106)$$

The matrix $(sI_{n_x} - A)^{-1}$ is computed as

$$(sI_{n_x} - A)^{-1} = \frac{1}{(s - \lambda_1)(s - \lambda_2)} \begin{bmatrix} s - \lambda_2 & 0 \\ 0 & s - \lambda_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{(s - \lambda_1)} & 0 \\ 0 & \frac{1}{(s - \lambda_2)} \end{bmatrix} \quad (2.107)$$

The state transition matrix is formed using (2.105) as

$$\text{LT}^{-1}\left\{(sI_{n_x} - A)^{-1}\right\} = \text{LT}^{-1}\left\{\begin{bmatrix} \frac{1}{(s - \lambda_1)} & 0 \\ 0 & \frac{1}{(s - \lambda_2)} \end{bmatrix}\right\} = \begin{bmatrix} e^{\lambda_1(t-t_0)} & 0 \\ 0 & e^{\lambda_2(t-t_0)} \end{bmatrix} \quad (2.108)$$

Example 2.15 Consider the matrix A from Example 2.11. The matrix $(sI_{n_x} - A)$ is

$$(sI_{n_x} - A) = \begin{bmatrix} s + 1 & -1 \\ -1/2 & s + 3/2 \end{bmatrix} \quad (2.109)$$

The matrix $(sI_{n_x} - A)^{-1}$ is computed as

$$(sI_{n_x} - A)^{-1} = \begin{bmatrix} s + 1 & -1 \\ -1/2 & s + 3/2 \end{bmatrix}^{-1} = \frac{1}{(s+2)(s+1/2)} \begin{bmatrix} s + 3/2 & 1 \\ 1/2 & s + 1 \end{bmatrix} \quad (2.110)$$

The state transition matrix is formed using (2.110) as

$$\text{LT}^{-1}\left\{\begin{bmatrix} \frac{s+3/2}{(s+2)(s+1/2)} & \frac{1}{(s+2)(s+1/2)} \\ \frac{1/2}{(s+2)(s+1/2)} & \frac{s+1}{(s+2)(s+1/2)} \end{bmatrix}\right\} = \begin{bmatrix} 1/3e^{-2t} + 2/3e^{-1/2t} & 2/3e^{-1/2t} - 2/3e^{-2t} \\ 1/3e^{-1/2t} - 1/3e^{-2t} & 2/3e^{-2t} + 1/3e^{-1/2t} \end{bmatrix} \quad (2.111)$$

2.5 Stability, Controllability, and Observability

The concepts of stability, controllability, and observability are central in systems and control theory. For LTI dynamics these three properties are well understood and can be directly analyzed.

2.5.1 Stability of LTI Systems

We begin with the notion of stability for the unforced LTI dynamics without a control input, that is, $u = 0$.

$$\dot{x} = A x, \quad x(0) = x_0 \quad (2.112)$$

The stability of the system trajectories (2.112) is defined with respect to the system equilibrium point $x_{\text{eq}} \in R^n$, which in turn is not necessarily unique and is defined by the algebraic equations.

$$A x_{\text{eq}} = 0 \quad (2.113)$$

Clearly, if A is non-singular then the system equilibrium is unique and is located at the origin, $x_{\text{eq}} = 0$. Any nonzero equilibrium can be transformed into the origin. Consequently, it is sufficient to study stability of the system solutions with respect to the origin.

The stability of a solution can be interpreted as a continuity of the system trajectories, with respect to initial conditions, over infinite time interval. The keywords here are “over infinite time interval”. They highlight the difference between the notions of the system stability and continuity on initial conditions. It is well known that the solutions of (2.112) continuously depend on the system initial conditions. However, the notion of stability requires that this continuity property holds infinitely in time. In Chap. 8, we will study stability of nonlinear systems. Here, we shall focus on the LTI dynamics. For these systems, stability definitions are global in nature; that is, they are independent of and valid uniformly for all initial conditions.

Definition 2.2 (*Stability in the Sense of Lyapunov*) The equilibrium point $x_{\text{eq}} = 0$ of the LTI unforced dynamics (2.112) is stable if for any $\varepsilon > 0$ there exists $\delta(\varepsilon) > 0$ such that for all initial conditions x_0 , and for all $t \geq 0$, the system trajectories are uniformly bounded, $\|x(t)\| \leq \varepsilon$.

Definition 2.3 (*Instability*) The equilibrium $x_{\text{eq}} = 0$ of the LTI system (2.112) is unstable if it is not stable.

Definition 2.4 (*Asymptotic Stability*) The equilibrium $x_{\text{eq}} = 0$ of the LTI system (2.112) is asymptotically stable if it is stable and $\lim_{t \rightarrow \infty} x(t) = 0$.

Definition 2.5 (*Exponential Stability*) The equilibrium $x_{\text{eq}} = 0$ of the LTI system (2.112) is exponentially stable if it is asymptotically stable and there are exist positive constants C and η such that for all initial conditions x_0 and for all $t \geq 0$, $\|x(t)\| \leq C e^{-\eta t} \|x_0\|$.

For LTI systems, the notions of asymptotic and exponential stability are equivalent. This fact directly follows from (2.6) with $u = 0$ and assuming that A has distinct eigenvalues 4, 5, 6.

$$x(t) = \underbrace{e^{At} x_0}_{\text{Initial Condition Response}} = \sum_{k=1}^n c_k e^{\lambda_k t} e_k \quad (2.114)$$

In this case, the unforced solution (2.114) is governed by the system eigenvalues λ_k , the corresponding eigenvectors e_k , and the coefficients c_k that define individual contributions of the eigenvectors. This is a decomposition of the system solution onto the eigenvectors (mode shapes) of the A -matrix. When A has repeated eigenvalues, the system solution contains time-dependent polynomials. However, its asymptotic (and thus exponential) stability solely depends the system eigenvalues located in the open left-hand complex plane \mathbb{C}^- .

Theorem 2.1 *The LTI system (2.112) is globally exponentially stable if and only if all of the eigenvalues of A have strictly negative real parts. If at least one eigenvalue has a positive real part then the system is unstable. If there are distinct eigenvalues with a zero real part then the system is stable but not exponentially stable.* ■

Often in controls literature, the eigenvalues of A with strictly negative real parts are called the stable poles of the system, and the remaining ones are referred to as the unstable poles.

The eigenvalues with the zero real parts (located on the imaginary $j\omega$ -axis) require special treatment. If they are distinct, that is if the system has the unique eigenvalue whose real part is zero, then the system is stable but not asymptotically stable. Repeated eigenvalues on the $j\omega$ -axis will cause the system to become unstable.

2.5.2 Controllability of LTI Systems

In order to control an n_x -dimensional LTI system such as,

$$\dot{x} = A x + B u \quad (2.115)$$

we need to first define the meaning of controllability.

Definition 2.6 The LTI system (2.115) is controllable if an n_u -dimensional control input $u(t)$ can be found to steer the n_x -dimensional system state x from any initial condition $x(0) = x_0$ to any final value $x(t_f) = x_f$, in a finite time t_f .

For LTI systems, the controllability criterion is well known and can be found in any textbook on linear systems and control.

Theorem 2.2 (Controllability Criterion) *The n_x -dimensional LTI system with n_u -dimensional control input is controllable if and only if the controllability matrix W_c has full row rank.*

$$\text{rank}(W_c) = \text{rank}(\begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix})_{n_x \times (n_x \times n_u)} = n_x \quad (2.116)$$

Clearly, the system controllability depends only on the system (A, B) pair of matrices. Another criterion for controllability is given by the Popov–Belevitch–Hautus (PBH) tests [6].

Theorem 2.3 (PBH Eigenvector Test for Controllability) *A pair (A, B) is controllable if and only if there is no left row eigenvector of A that is orthogonal to B .* ■

This theorem can be applied to systems to understand controllability by transforming the system to modal coordinates using (2.86) and determining if the transformed control distribution matrix $\Gamma = T B$ has a row of zeros. If there is a row of zeros in Γ , it indicates that the mode associated with that row is not controllable. We shall demonstrate this in the following example:

Example 2.16 Consider the LTI state-space model $\dot{x} = Ax + Bu$ with state and control distribution matrices

$$A = \begin{bmatrix} -7 & -2 & 6 \\ 2 & -3 & -2 \\ -2 & -2 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 0 \end{bmatrix} \quad (2.117)$$

It is not evident examining the numerical entries in these matrices if the system is controllable or not. The CE for the system is $\det(sI - A) = (s + 1)(s + 3)(s + 5)$ which has distinct roots. Using (2.86), the eigenvalues and right eigenvectors are

$$\begin{aligned} \lambda_1 &= -1 \\ \lambda_2 &= -3 \\ \lambda_3 &= -5 \end{aligned} \quad \begin{bmatrix} v^1 & v^2 & v^3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{3} & 1/\sqrt{2} \\ 0 & -1/\sqrt{3} & -1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{3} & 0 \end{bmatrix}}_{T^{-1}} \quad (2.118)$$

with the resulting transformed system (2.86) given as

$$\dot{z} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & -5 \end{bmatrix} z + \begin{bmatrix} 0 & 0 \\ \sqrt{3} & 0 \\ 0 & \sqrt{2} \end{bmatrix} u \quad (2.119)$$

We see from (2.119) that in the transformed control distribution matrix Γ the first mode, $\lambda_1 = -1$, has a row of zeros. Therefore, the eigenvector for the first mode is orthogonal to each column of B , and the controls will not influence this mode. It is uncontrollable.

Theorem 2.4 (PBH Rank Test for Controllability) *A pair (A, B) is controllable if and only if*

$$\text{rank}([sI_{n_x} - A \ B]) = n_x = \dim(x), \quad \text{for all } s \in \mathbb{C} \quad (2.120)$$

The above controllability Theorems 2.2–2.4 are all equivalent. Their proofs are based on the direct application of Definition 2.6 and the analytic solution (2.6) for the LTI system trajectories.

Example 2.17 Consider the following LTI system

$$\begin{aligned}\dot{x}_1 &= -2x_1 + u \\ \dot{x}_2 &= -x_1 - x_2 + u\end{aligned} \quad (2.121)$$

The state and control distribution matrices are

$$A = \begin{bmatrix} -2 & 0 \\ -1 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (2.122)$$

We will evaluate controllability of (2.121) by examining the rank of the controllability matrix W_c , the PBH eigenvector test, and the PBH rank test. Note that each method will produce the same conclusion regarding model controllability. Some methods are a yes/no answer, and some methods give more insight as to which mode or modes in the system dynamics may become controllable.

Controllability Matrix W_c : Using (2.116) the controllability matrix W_c is computed as

$$W_c = [B \ AB] = \begin{bmatrix} 1 & -2 \\ 1 & -2 \end{bmatrix}_{rk=1} \quad (2.123)$$

which also shows the system is not controllable.

PBH Eigenvector Test: The CE for the system is $\det(sI - A) = (s + 1)(s + 2)$ which has distinct roots. The eigenvalues and right eigenvectors are

$$\Lambda = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \quad T^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \quad (2.124)$$

and in modal coordinates the transformed system is

$$\dot{z} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (2.125)$$

From (2.125) we see that in the transformed control distribution matrix Γ the first mode, $\lambda_1 = -1$, has a row of zeros. The control will not influence this mode. It is uncontrollable so the system is not controllable.

PBH Rank Test: For this problem the PBH rank test is

$$\text{rank}([sI - A \ B]) = \text{rank}\left(\begin{bmatrix} s+2 & 0 & 1 \\ 1 & s+1 & 1 \end{bmatrix}\right) \quad (2.126)$$

and needs to be evaluated at each eigenvalue of A . For $\lambda_1 = -1$

$$\text{rank}\left(\begin{bmatrix} \lambda_1 + 2 & 0 & 1 \\ 1 & \lambda_1 + 1 & 1 \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}\right) = 1 \quad (2.127)$$

which is rank deficient indicating lack of controllability for this mode. For $\lambda_2 = -2$

$$\text{rank}\left(\begin{bmatrix} \lambda_2 + 2 & 0 & 1 \\ 1 & \lambda_2 + 1 & 1 \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} 0 & 0 & 1 \\ 1 & -1 & 1 \end{bmatrix}\right) = 2 \quad (2.128)$$

which is full rank indicating that this mode is controllable.

2.5.3 Observability of LTI Systems

Often, we will encounter control applications for LTI systems (2.3) with a partially known state vector. Here, the system control input and the measured output are the only signals available for control design. For these systems, a control strategy that requires measurements of the system full state would not be directly applicable. In order to control a system with incomplete state information and based on its input–output measurements only, in Chap. 6 we will introduce dynamic state observers. Those are mathematical algorithms that operate on the system input and output signals to reconstruct and estimate the system state vector. Essentially, a state observer is an LTI system on its own. It takes available control input and the system measured output to compute the system full state estimate in real time. Reconstruction of the system state using the process control input and output signals is based on the system observability property. Our first step is to formally define the notation of observability for LTI dynamics [4].

Definition 2.7 The unforced LTI system with n_x -dimensional state x and a n_y -dimensional output y

$$\dot{x} = A x, \quad y = C x \quad (2.129)$$

is observable if for any $T > 0$ it is possible to determine the state of the system $x(T)$ through measurement of $y(t)$ on the interval $[0, T]$. In this case, the pair (A, C) is called observable.

Theorem 2.5 (Observability Criterion) *The LTI system (2.129) with an n_x -dimensional state vector x and a n_y -dimensional output y is observable if and only if the observability matrix W_o has full column rank.*

$$\text{rank}(W_o) = \text{rank} \left(\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \right)_{(p \times n) \times n} = n_x = \dim(x) \quad (2.130)$$

■

The observability property is independent of the system control input. The original LTI dynamics (2.3) are observable if and only if the pair (A, C) is observable. In this case, the system state can be reconstructed using the control input and the measured output. Formally speaking, the observability property for a forced LTI system means that for any $T > 0$ it is possible to determine the system state $x(T)$ through measurements of $u(t)$ and $y(t)$ on the interval $[0, T]$.

Comparing the controllability and observability criterions, (2.116) and (2.130), an equivalence relation between the two notions clearly emerges. The pair (A, B) is controllable if and only if the pair (A^T, B^T) is observable. This is called the *Duality Principle* in linear systems [1, 2, 7]. Duality is prevalent in mathematics. In control of LTI systems it takes on a special form. The principle of duality allows to analysis and design controllers and state observers using a unified theory and numerically efficient methods developed for control or state observation. The design and analysis methods are basically the same, and their application requires a transposition of the system respective matrices to go between the two problems. Using the principle of duality we state the PBH eigenvector test and rank test for observability theorems as follows and then demonstrate their use with example.

Theorem 2.6 (PBH Eigenvector Test for Observability) *A pair (A, C) is observable if and only if there is no right column eigenvector of A that is orthogonal to C .* ■

Theorem 2.7 (PBH Rank Test for Observability) A pair (A, C) is observable if and only if

$$\text{rank}\left(\begin{bmatrix} sI_{n_x} - A \\ C \end{bmatrix}\right) = n_x = \dim(x), \quad \text{for all } s \in \mathbb{C} \quad (2.131)$$

■

Example 2.18 Consider the LTI system from (2.117) with output distribution matrix C given by

$$C = \begin{bmatrix} -2 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \quad (2.132)$$

Let us investigate observability of this linear system using the rank of the observability matrix W_o , the PBH eigenvector test, and the PBH rank test.

Observability Matrix W_o : Using (2.130) the observability matrix W_o is computed as

$$\text{rank}(W_o) = \text{rank}\left(\begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} -2 & 1 & 1 \\ 0 & 1 & -1 \\ 14 & -1 & -13 \\ 4 & -1 & -3 \\ -74 & 1 & 73 \\ -24 & 1 & 23 \end{bmatrix}\right) = 2 \quad (2.133)$$

which also shows the system is not observable.

PBH Eigenvector Test: The transformed system output distribution matrix C_z from (2.86) is

$$C_z = CT^{-1} = \begin{bmatrix} -\sqrt{2} & 0 & -2.1213 \\ -\sqrt{2} & 0 & -\sqrt{2} \end{bmatrix} \quad (2.134)$$

From (2.134) we see that in the transformed output distribution matrix C_z the second mode, $\lambda_2 = -3$, has a column of zeros. Thus the output vector will not contain this mode. It is not observable.

PBH Rank Test: For this problem the PBH rank test is

$$\text{rank}\left(\begin{bmatrix} sI_{n_x} - A \\ C \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} s+7 & 2 & -6 \\ -2 & s+3 & 2 \\ 2 & 2 & s-1 \\ -2 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}\right) \quad (2.135)$$

and needs to be evaluated at each eigenvalue of A . For $\lambda_1 = -1$,

$$\text{rank} \begin{pmatrix} \lambda_1 + 7 & 2 & -6 \\ -2 & \lambda_1 + 3 & 2 \\ 2 & 2 & \lambda_1 - 1 \\ -2 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix} = \text{rank} \begin{pmatrix} 6 & 2 & -6 \\ -2 & 2 & 2 \\ 2 & 2 & -2 \\ -2 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix} = 3 \quad (2.136)$$

we have a full column rank indicating this mode is observable. For $\lambda_2 = -2$,

$$\text{rank} \begin{pmatrix} \lambda_2 + 7 & 2 & -6 \\ -2 & \lambda_2 + 3 & 2 \\ 2 & 2 & \lambda_2 - 1 \\ -2 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix} = \text{rank} \begin{pmatrix} 4 & 2 & -6 \\ -2 & 0 & 2 \\ 2 & 2 & -4 \\ -2 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix} = 2 \quad (2.137)$$

and so it is column rank deficient indicating this mode is not observable. For $\lambda_3 = -5$,

$$\text{rank} \begin{pmatrix} \lambda_3 + 7 & 2 & -6 \\ -2 & \lambda_3 + 3 & 2 \\ 2 & 2 & \lambda_3 - 1 \\ -2 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix} = \text{rank} \begin{pmatrix} 2 & 2 & -6 \\ -2 & -2 & 2 \\ 2 & 2 & -6 \\ -2 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix} = 3 \quad (2.138)$$

the matrix has full column rank indicating this mode is observable.

2.6 Norms of Vectors and Matrices in Euclidean Spaces

This chapter presents a brief overview of norms for vectors and matrices. We shall use these concepts very often throughout the book.

The n -dimensional Euclidean space R^n is a collection of all n -dimensional vectors $x = (x_1 \dots x_n)^T$, whose components x_i are real numbers, where the upper script “T” denotes the transposition operator, which turns a vector row into a vector column and vice versa. For $n = 1$, we get the one-dimensional Euclidean space of real numbers $R = R^1$.

The set of all $(n \times m)$ real matrices, with n rows and m columns, defines the $(n \times m)$ -dimensional Euclidean space $R^{n \times m}$. Elements of a Euclidean space can be added, subtracted, and multiplied by a scalar.

The inner product of two vectors x and y from R^n equals the sum of products of their corresponding components: $x^T y = \sum_{i=1}^n x_i y_i$. The product of two matrices $A = R^{n \times m}$ and $B = R^{m \times p}$ is the matrix $C = R^{n \times p}$, whose (i, j) th element is the inner product of the i th row of A and j th column of B .

For a vector $x \in R^n$, its length (or magnitude) is given by the norm $\|x\|$ —a real-valued function from R^n to R , with the following properties:

1. For any $x \in R^n$: $\|x\| \geq 0$.
2. $\|x\| = 0$ if and only if x is the zero vector in R^n .
3. For any two vectors x and y from R^n , the triangular inequality holds: $\|x + y\| \leq \|x\| + \|y\|$.
4. For any real constant $\lambda \in R$ and any vector $x \in R^n$: $\|\lambda x\| = |\lambda| \|x\|$.

In the forthcoming design and analysis of adaptive controllers, we will encounter the class of vector p -norms.

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty \quad (2.139)$$

For notational sake, we would often drop the lower script “ p ” and write $\|x\|$. Given a vector p -norm $\|x\|$, the induced matrix norm

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\| \quad (2.140)$$

clearly depends on the selected vector p -norm.

For a matrix $A = [a_{ij}] \in R^{n \times m}$, the Frobenius norm is defined by,

$$\|A\|_F = \sqrt{\text{tr}(A^T A)} = \sqrt{\sum_{i,j} a_{ij}^2} \quad (2.141)$$

with $\text{tr}()$ denoting the trace of a matrix, which is equal to the sum of the matrix diagonal elements.

The following statements are well known and are listed here without proofs.

- For the vector 1-norm $\|x\|_1 = \sum_{i=1}^n |x_i|$, the corresponding induced matrix norm is equal to the maximum absolute column sum, that is, $\|A\|_1 = \max_{1 \leq j \leq m} \left| \sum_{i=1}^n a_{ij} \right|$.
- For the vector 2-norm $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$, the induced matrix norm is equal to the maximum singular value of A , that is, $\|A\|_2 = \sigma_{\max}(A)$.
- For the vector ∞ -norm $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$, the induced matrix norm is equal to the maximum absolute row sum, that is, $\|A\|_\infty = \max_{1 \leq i \leq n} \left| \sum_{j=1}^m a_{ij} \right|$.
- The induced matrix norm satisfies $\|Ax\|_p \leq \|A\|_p \|x\|_p$, and for any two compatibly dimensioned matrices, A and B , one also has $\|AB\|_p \leq \|A\|_p \|B\|_p$.
- The Frobenius norm is not an induced norm of any vector norm, but it is compatible with the 2-norm in the sense that $\|Ax\|_2 \leq \|A\|_F \|x\|_2$.
- For any two compatibly dimensioned matrices A and B , the Frobenius inner product is defined as $\langle A, B \rangle_F = \text{trace}(A^T B)$.

- According to the Schwartz inequality

$$|\text{trace}(A^T B)| = |\langle A, B \rangle_F| \leq \|A\|_F \|B\|_F \quad (2.142)$$

- For any two codimensional vectors a and b , the trace identity relation is

$$a^T b = \text{tr}(b a^T) \quad (2.143)$$

2.7 Summary

In order to make the book material as self-contained as possible, in this chapter we reviewed and summarized basic definitions from linear systems and control, such as state-space modeling in the time domain and system input–output representation via transfer functions in the frequency domain. This included top-level definitions of controllability and observability that need to be discussed in later chapters of the text where we will present optimal, robust, and adaptive control theories. We have also addressed fundamental concepts in feedback control and model-based controls engineering, with a brief discussion on control goals, objectives, requirements, and V&V methods for reducing risk in the development and fielding of control systems.

2.8 Exercises

Exercise 2.1 Derive (2.6).

Exercise 2.2 Derive matrices (2.13) for the discrete LTI dynamics (2.12). Assume that A is invertible. Compute B_d in (2.13). When A is singular, use the Taylor series expansion (2.7) to compute (A_d, B_d) in (2.13) for (A_d, B_d) in (2.13) that are valid for sufficiently small time step $0 < \Delta t \ll 1$. Connect your discretized approximation data to the forward Euler integration formula: $x(k+1) = x(k) + \Delta t \underbrace{(A x(k) + B u(k))}_{\dot{x}(k)} = x(k) + \Delta t \dot{x}(k)$.

Exercise 2.3 Apply (2.20) to (2.3) and derive the discrete LTI dynamics (2.19).

Exercise 2.4 Prove (2.27).

Exercise 2.5 Derive (2.79) and show that the transformed system transfer function matrix remains the same as in (2.37).

Exercise 2.6 Use the system state similarity transformation (2.78) to prove that the controllability matrices (2.116) for the original and the transformed system satisfy the following equation:

$$T W_c = T(B \ A \ B \ A^2 B \ \dots \ A^{n-1} B) = \underbrace{(B_z \ A_z \ B_z \ A_z^2 B_z \ \dots \ A_z^{n-1} B_z)}_{\tilde{W}_c} = \tilde{W}_c$$

Exercise 2.7 Consider a controllable n -dimensional LTI system with a single input.

$$\dot{x} = Ax + bu, \quad x \in R^n, \quad u \in R$$

Let $\lambda(s) = s^n + a_n s^{n-1} + \dots + a_2 s + a_1$ be the system characteristic polynomial. Consider the system representation in the controllable canonical form.

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_{n-1} \\ \dot{z}_n \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 1 \\ -a_1 & -a_2 & \dots & -a_{n-1} & -a_n \end{pmatrix}}_{A_z} \underbrace{\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-1} \\ z_n \end{pmatrix}}_z + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}}_{b_z} u$$

$\Leftrightarrow \boxed{\dot{z} = A_z z + b_z u, \quad z \in R^n, \quad u \in R}$

Find the state-space similarity transformation matrix $T \in R^{n \times n}$ such that $z = Tx$. Hint: Compute controllability matrices for the original and the canonical systems. Then use Exercise 2.5 to compute T .

Exercise 2.8 Form the Laplace transform for $g(t) = (2e^{-2t} - t) \sin(2t)$, $t \geq 0$.

Exercise 2.9 Sketch the root locus with respect to K for the equation $1 + KL(s) = 0$ for

$$L(s) = \frac{(s+2)(s+4)}{s(s+1)(s+5)(s+10)}$$

Exercise 2.10

(a) Convert this transfer function into a state-space model

$$\frac{Y}{U} = \frac{s^2 + \omega_1^2}{s^2 + 2\xi\omega_1 s + \omega_1^2}$$

(b) Convert this transfer function into a state-space model

$$\frac{Y}{U} = \frac{1}{s^2}$$

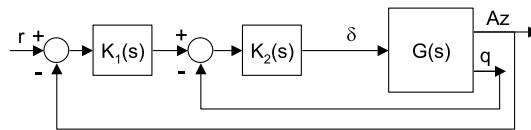
(c) Convert this state-space model to transfer function

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}u \\ y &= [1 \ 0 \ 0]x + [0]u\end{aligned}$$

(d) Convert this state-space model to a transfer function matrix. What is this a model of? What are the outputs?

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 1 \\ -\omega_1^2 & -2\zeta\omega_1 \end{bmatrix}x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u \\ y &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}x + \begin{bmatrix} 0 \\ 0 \end{bmatrix}u\end{aligned}$$

Exercise 2.11 Consider the flight control system shown in the block diagram



where

$$K_1(s) = \frac{K_a(s + 10)}{s} \quad K_2(s) = K_q$$

and the UAV dynamics $G(s)$ are described using the model $\dot{x} = Ax + Bu$, $y = Cx$, $u = \delta$, $y = [A_z \ q]^T$, with $G(s) = C(sI_{n_x} - A)^{-1}B$.

(a) Derive a state-space model for the controller

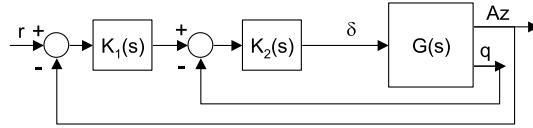
$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r,\end{aligned}$$

with y as the input to the controller and $u = \delta$ as the output from the controller.

(b) Describe what effect the transfer function $K_1(s)$ has on the system response.

- (c) Describe what effect the transfer function $K_2(s)$ has on the system response.

Exercise 2.12 Consider the flight control system shown in the block diagram



where

$$K_1(s) = \frac{K_1(\tau_n s + 1)}{s} \quad K_2(s) = \frac{K_q(s^2 + \omega_n^2)}{(s^2 + 2\zeta\omega_n s + \omega_n^2)}$$

and the UAV dynamics $G(s)$ are described using the model $\dot{x} = Ax + Bu$, $y = Cx$, $u = \delta$, $y = [A_z \ q]^T$, with $G(s) = C(sI_{n_x} - A)^{-1}B$.

- (a) Derive a state-space model for the controller

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r,\end{aligned}$$

with y as the input to the controller and $u = \delta$ as the output from the controller.

- (b) Describe what effect the transfer function $K_1(s)$ has on the system response.
(c) Describe what effect the transfer function $K_2(s)$ has on the system response.

Exercise 2.13 Find the eigenvalues and eigenvectors for $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix}$.

Exercise 2.14 Find e^{At} if $A = \begin{bmatrix} 1 & -1 \\ 2 & 4 \end{bmatrix}$.

Exercise 2.15 Find e^{At} if $A = \begin{bmatrix} -1 & 1/2 \\ 0 & 1 \end{bmatrix}$.

Exercise 2.16 Find the state transition matrix $\Phi(t)$ for the system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u$$

Exercise 2.17 Consider the system model $\dot{x} = Ax + Bu$, $y = Cx$ where

$$[A \ B \ C] = \left[\begin{bmatrix} 3 & 6 & 4 \\ 9 & 6 & 10 \\ -7 & -7 & -9 \end{bmatrix} \ \begin{bmatrix} -2/3 & 1/3 \\ 1/3 & -2/3 \\ 1/3 & 1/3 \end{bmatrix} \ \begin{bmatrix} 2 & 3 & 4 \\ 2 & 1 & 3 \end{bmatrix} \right]$$

- (a) Check controllability for this system.
- (b) Check observability for this system.

Exercise 2.18 Consider the system model $\dot{x} = Ax + Bu$, $y = Cx$. Check controllability and observability for

$$(a) [A \ B \ C] = \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \right]$$

$$(b) [A \ B \ C] = \left[\begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & 0 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \right]$$

Exercise 2.19 Find the state transition matrix $\Phi(t)$ for the system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u$$

Exercise 2.20 The short-period dynamics of an aircraft were derived in Chap. 1. When the vehicle airspeed V_0 is sufficiently large, $\frac{Z_q}{V_0}$ and $\frac{Z_{\delta_e}}{V_0}$ become negligible and the resulting simplified short-period dynamics become

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V_0} & 1 \\ M_\alpha & M_q \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} 0 \\ M_{\delta_e} \end{pmatrix} \delta_e$$

Apply the result from Example 2.7 to explicitly compute the state-space controllable canonical form for this system. Rewrite the canonical form as a second-order differential equation in terms of α . Comment on your derivations and their relevance to flight control.

References

1. Braun KF (1909) Electrical oscillations and wireless telegraphy. Nobel lecture, 11 Dec
2. Kalman RE (1960) A new approach to linear filtering and prediction problems. Trans ASME J Basic Eng 82:35–45
3. Kalman RE (1960) Contributions to the theory of optimal control. Bol Soc Mat Mex 5:102–119.
In: Proceedings of the symposium on ordinary differential equations, Mexico City

4. Åström KJ, Murray RM (2008) Feedback systems: an introduction for scientists and engineers. Princeton University Press, Princeton, p 08540
5. Kuo BC (1991) Automatic control systems, 6th edn. Prentice Hall, Englewood Cliffs, p 07632
6. Kailath T (1980) Linear systems. Prentice-Hall Inc., Englewood Cliffs
7. Kwakernaak H, Sivan R (1972) Linear optimal control systems. Wiley, New York



Frequency Domain Analysis

3

In this chapter we present frequency domain analysis methods for both single-input-single-output and multi-input-multi-output control systems. Transfer function matrices, Nyquist theory for multivariable systems, and singular value frequency response methods are discussed in detail. Modeling techniques for robust stability analysis are discussed in which both complex and real parametric uncertainties are covered. Theory and examples for the structured singular value μ and the real stability margin are presented. Flight control systems designed using both classical and optimal control theories are analyzed to demonstrate how to determine their robust stability.

3.1 Introduction

Frequency domain analysis methods are among the most useful tools available for the development of control systems. When designing a control system, it is very important to understand the system response, as well as stability and robustness properties of the design. For linear time-invariant (LTI) systems, these properties are best analyzed, displayed, and understood in the frequency domain. For linear single-input-single-output (SISO) systems, frequency domain methods for analysis, as well as techniques for synthesis of a controller, have been developed and used in industry since the 1950s. These analysis and design methods, often referred to as the classical methods, include techniques like root locus, Bode and Nyquist diagrams, as well as Nichols charts. For multi-input-multi-output (MIMO) systems, the analysis methods used are typically extensions of the SISO methods. In order to understand the methods for MIMO analysis, one should have a good grasp of the fundamental for the classical SISO frequency domain methods.

Since the early 1980s, control system analysts have been focused upon determining the stability and robustness of MIMO feedback designs in the presence of uncertainties. In particular, this focus has been upon frequency domain techniques using methods which employ singular value frequency responses. These singular value-based methods of analysis join, and in some cases replace, the classical Bode and Nyquist techniques with multivariable generalizations, and extend many modeling uncertainty capabilities. They have become widespread in industry, as today's systems require MIMO analysis.

This chapter presents an overview of the theory and methods available, connecting the classical and multivariable analysis methods and tools, and highlights aerospace control applications and analyses in the frequency domain.

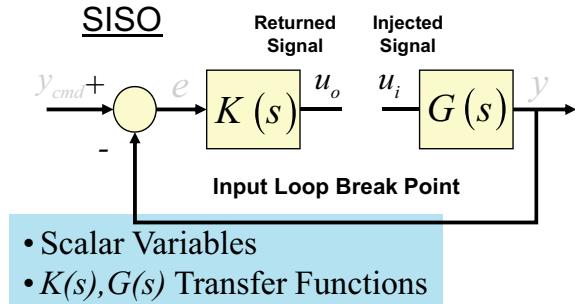
3.2 Transfer Functions and Transfer Function Matrices

Many of the frequency domain analysis models for MIMO systems are natural extensions of scalar transfer functions used to analyze SISO systems. However, unlike these transfer functions, MIMO analysis models have different sizes depending upon where the loop is broken for analysis. Consider the SISO system shown in the block diagram of Fig. 3.1.

In LTI systems, the external command into the system does not impact closed-loop stability of the system. To determine relative stability computed as a gain and phase margin, the loop gain transfer function is analyzed. The loop gain for this system can be calculated by opening the loop at the control generation point (plant input) and injecting a signal u_i . The returned signal is

$$u_o = -\underbrace{K(s)G(s)}_{L(s)} u_i \quad (3.1)$$

Fig. 3.1
Single-input-single-output
 KG block diagram



in which $L(s)$ is the *loop gain transfer function*. Differencing the injected signal u_i and the returned signal u_o results in

$$u_i - u_o = u_i + K(s)G(s)u_i = (1 + K(s)G(s))u_i = (1 + L(s))u_i \quad (3.2)$$

which is the *return difference* for the loop. We will find later in this chapter that the *return difference matrix*, $(I + L(s))$, plays a very important role in the development of stability robustness analysis tests for MIMO systems. The error transfer function for this system is

$$\frac{e(s)}{y_{\text{cmd}}(s)} = \frac{1}{1 + K(s)G(s)} = S(s) \quad (3.3)$$

where $S(s)$ is the *sensitivity function*, which describes the error dynamics. Note that the sensitivity is the inverse of the return difference. The closed-loop response to an external command input is,

$$\frac{y(s)}{y_{\text{cmd}}(s)} = \frac{K(s)G(s)}{1 + K(s)G(s)} = T(s) \quad (3.4)$$

where $T(s)$ is the *closed-loop transfer function*. The transfer function $T(s)$ is also called the *complementary sensitivity*, since $S(s)$ and $T(s)$ satisfy the identity

$$S(s) + T(s) = 1 \quad (3.5)$$

Now, consider the multivariable equivalent of Fig. 3.1 as shown in Fig. 3.2. In Fig. 3.2 the variables $u_i, u_o \in R^{n_u}$ and $y_{\text{cmd}}, e, y, y_i, y_o \in R^{n_y}$, with the controller $K(s)$ a $n_u \times n_y$ matrix and the plant $G(s)$ a $n_y \times n_u$ matrix. The figure shows the loop broken at the plant-input breakpoint and the plant-output breakpoint. We shall introduce a subscript on the MIMO transfer function matrices that indicate at which loop breakpoint they are formed. The loop gain $L_u(s)$ is formed using the same procedure as in (3.1) where $L_u(s) = K(s)G(s)$ is a $n_u \times n_u$ matrix. Forming the return difference matrix yields

$$u_i - u_o = (I_{n_u} + K(s)G(s))u_i = (I_{n_u} + L_u(s))u_i \quad (3.6)$$

where $I_{n_u} + L(s)$ is also a $n_u \times n_u$ matrix. If this same procedure for calculating the loop gain is applied at the output of the plant, as shown in Fig. 3.2, the return difference dynamics are

$$y_i - y_o = (I_{n_y} + G(s)K(s))y_i \quad (3.7)$$

which produces a loop gain and return difference matrix that are $n_y \times n_y$ in dimension. It is very important to learn that for MIMO systems the loop gain is different at the plant-input and plant-output loop breakpoints, which is unlike SISO systems. This dissimilarity is caused by the fact that matrices do not commute, but

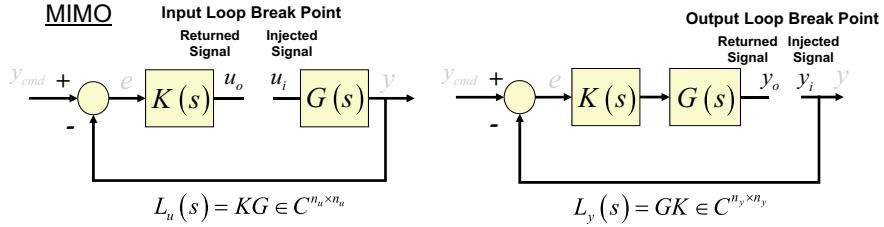


Fig. 3.2 Multi-input–multi-output KG block diagram

Table 3.1 Summary of transfer functions and transfer function matrices used in frequency domain analysis

Function	SISO (Fig. 3.1)	MIMO (Fig. 3.2) Plant input	MIMO (Fig. 3.2) Plant output
Loop gain	$L(s) = K(s)G(s)$ $= G(s)K(s)$	$L_u(s) = K(s)G(s)$	$L_y(s) = G(s)K(s)$
Return difference	$1 + L(s)$	$I_{n_s} + L_u(s)$	$I_{n_y} + L_y(s)$
Sensitivity	$S(s) = \frac{1}{1+L(s)}$	$S_u(s) = (I_{n_v} + L_u(s))^{-1}$	$S_y(s) = (I_{n_y} + L_y(s))^{-1}$
Complementary sensitivity	$T(s) = \frac{L(s)}{1+L(s)}$	$T_u(s) = (I_{n_v} + L_u(s))^{-1}L_u(s)$	$T_y(s) = (I_{n_y} + L_y(s))^{-1}L_y(s)$

scalars do. Table 3.1 summarizes the loop gain, return difference, sensitivity, and complementary sensitivity transfer functions and matrices for the SISO and MIMO systems shown in Figs. 3.1 and 3.2. In Table 3.1 for the MIMO system matrices we have introduced a subscript “ u ” and “ y ” to indicate the loop breakpoint.

Table 3.1 lists the various matrices used to analyze MIMO control systems. We will use the subscript u and y on the MIMO system matrices so the reader understands the matrix dimension and loop breakpoint. For notational convenience, we will occasionally drop the dimension on the identity matrix. In that case, the subscript on the MIMO system matrices it is being added to will indicate the dimension.

Example 3.1 Consider the LTI pitch-plane dynamics of an unpiloted aircraft shown in Fig. 3.3, controlled using a classical proportional-plus-integral control architecture. The pitch-plane short-period dynamics are given by (A, B, C, D) and can be written as

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{Z_\alpha}{V} & 1 \\ M_\alpha & 0 \end{bmatrix}}_A \begin{bmatrix} \alpha \\ q \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{Z_\delta}{V} \\ M_\delta \end{bmatrix}}_B \delta_e$$

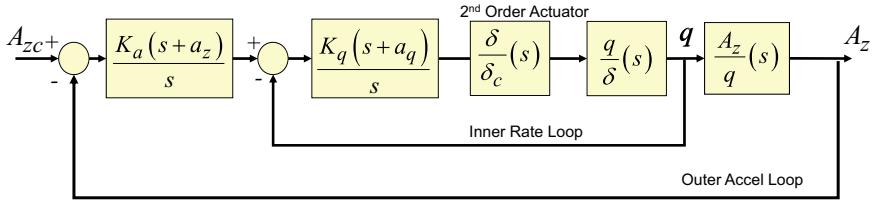


Fig. 3.3 Pitch-plane dynamics and autopilot controller

$$\begin{bmatrix} A_z \\ q \end{bmatrix} = \underbrace{\begin{bmatrix} Z_\alpha & 0 \\ 0 & 1 \end{bmatrix}}_C \begin{bmatrix} \alpha \\ q \end{bmatrix} + \underbrace{\begin{bmatrix} Z_\delta \\ 0 \end{bmatrix}}_D \delta_e \quad (3.8)$$

These dynamics form a SISO system. The transfer function matrix for the plant dynamics is,

$$G(s) = C(sI - A)^{-1}B + D = \begin{bmatrix} \frac{A_z}{\delta_e} \\ \frac{q}{\delta_e} \end{bmatrix} \quad (3.9)$$

which is a 2×1 matrix. The autopilot (controller) for this plant contains proportional-plus-integral control elements in the inner-rate loop closure and outer-acceleration loop closure, given by

$$K_{A_z}(s) = \frac{K_a(s + a_z)}{s} \quad (3.10)$$

and

$$K_q(s) = \frac{K_q(s + a_q)}{s} \quad (3.11)$$

with the controller transfer function matrix given by,

$$K(s) = [K_{A_z}(s) \ K_q(s) \ K_q(s)] \quad (3.12)$$

which is a 1×2 matrix. A state-space model for this controller is

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \quad (3.13)$$

with matrices given as

$$\begin{aligned} A_c &= \begin{bmatrix} 0 & 0 \\ K_q a_q & 0 \end{bmatrix}; \quad B_{c1} = \begin{bmatrix} -K_a a_z & 0 \\ -K_a K_q a_q & -K_q a_q \end{bmatrix}; \quad B_{c2} = \begin{bmatrix} K_a a_z \\ K_a K_q a_q \end{bmatrix} \\ C_c &= [K_q \ 1]; \quad D_{c1} = [-K_a K_q \ -K_q]; \quad D_{c2} = [K_a K_q] \end{aligned} \quad (3.14)$$

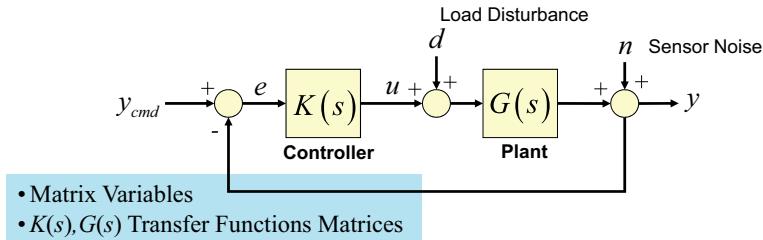


Fig. 3.4 MIMO system with disturbance and measurement noise

The loop gain at the input to the plant is,

$$L_u(s) = K(s)G(s) = K_{A_z}(s)K_q(s)\frac{A_z}{\delta_e} + K_q(s)\frac{q}{\delta_e} \quad (3.15)$$

which is a scalar transfer function. To analyze stability for this system any SISO analysis technique can be applied. If we examine the loop gain at the plant output then,

$$L_y(s) = G(s)K(s) = \begin{bmatrix} \frac{A_z}{\delta_e} K_{A_z}(s)K_q(s) & \frac{A_z}{\delta_e} K_q(s) \\ \frac{q}{\delta_e} K_{A_z}(s)K_q(s) & \frac{q}{\delta_e} K_q(s) \end{bmatrix} \quad (3.16)$$

which is a 2×2 matrix, and is a singular matrix since it is the product of matrices that are $(2 \times 1) \times (1 \times 2)$ in dimension. It is typical in most aerospace applications that the plant and controller matrices are non-square. In this case, stability analysis should be conducted at the loop breakpoint of minimum dimension.

Figure 3.4 illustrates an LTI MIMO system with command $y_{cmd}(t) \in R^{n_y}$, plant disturbance $d(t) \in R^{n_d}$, and measurement noise $n(t) \in R^{n_n}$. For MIMO systems, just like in the analysis of SISO systems, there are several important transfer functions (matrices) that are used to understand and evaluate a control system design. They include the loop transfer function matrix given in Table 3.1 and the “gang of four” [1], which includes (1) sensitivity, (2) complementary sensitivity, (3) control-to-noise transfer function matrix, and (4) the system response to the load disturbance, given by

$$\frac{e}{y_{cmd}} = S = \frac{1}{1 + L} \quad \frac{y}{y_{cmd}} = T = \frac{L}{1 + L} \quad \frac{u}{n} = KS \quad \frac{y}{d} = GS \quad (3.17)$$

Table 3.1 defines the loop transfer function matrices. Using Fig. 3.4, we will derive expressions for the gang of 4 for MIMO systems.

The output response matrices from the command, load disturbance, and sensor noise shown in Fig. 3.4 can be formed by writing the following loop equations:

$$\begin{aligned}
 u &= Ke \\
 \eta &= G(u + d) = GK e + Gd \\
 y &= \eta + n = GK(y_{\text{cmd}} - y) + Gd + n \\
 (I_{n_y} + GK)y &= GK y_{\text{cmd}} + Gd + n \\
 y &= (I_{n_y} + GK)^{-1}(GK y_{\text{cmd}} + Gd + n) \\
 Y(s) &= T_y y_{\text{cmd}} + S_y Gd + S_y n
 \end{aligned} \tag{3.18}$$

This equation shows how the output response directly depends upon the sensitivity and complementary sensitivity functions. At frequencies $s = j\omega$ where commands are to be followed we want $T_y(s) \rightarrow I_{n_y}$, which shows that sensor noise is also passed through the system into the output. It is not possible to reject sensor noise and track commands at the same frequencies. At frequencies where plant disturbances are to be rejected, we want $S_y(s) \rightarrow 0$.

The error response $E(s)$ matrices can be formed by writing the following loop equations:

$$\begin{aligned}
 u &= Ke \\
 \eta &= G(u + d) = GK e + Gd \\
 y &= \eta + n = GK e + Gd + n \\
 e &= y_{\text{cmd}} - y = y_{\text{cmd}} - GK e - Gd - n \\
 (I_{n_y} + GK)e &= y_{\text{cmd}} - Gd - n \\
 e &= (I_{n_y} + GK)^{-1}(y_{\text{cmd}} - Gd - n) \\
 E(s) &= S_y(s)y_{\text{cmd}} - S_y(s)Gd - S_y(s)n
 \end{aligned} \tag{3.19}$$

which shows that to make errors in tracking commands small, we want $S_y(s) \rightarrow 0$. Equations (3.18) and (3.19) illustrate the control design dilemma faced by engineers, that is to make $S_y(s) \rightarrow 0$ at low frequencies for command tracking and disturbance rejection, and $T_y(s) \rightarrow 0$ at high frequencies for sensor noise rejection and robustness to high-frequency unmodeled dynamics. The dilemma is that $S(s) + T(s) = I_{n_y}$ at all frequencies and at both loop breakpoints. So as the sensitivity is made small, the complementary sensitivity tends to unity, and vice versa.

The control response to sensor noise transfer function matrix can be formed by writing the following loop equations:

$$\begin{aligned}
 u &= Ke = K(y_{\text{cmd}} - y) \\
 y &= G(u + d) + n \\
 u &= K(y_{\text{cmd}} - G(u + d) - n) \\
 (I_{n_u} + KG)u &= Ky_{\text{cmd}} - KGd - Kn \\
 u &= (I_{n_u} + KG)^{-1}(Ky_{\text{cmd}} - KGd - Kn) \\
 U(s) &= S_u Ky_{\text{cmd}} - T_u d - S_u Kn
 \end{aligned} \tag{3.20}$$

Thus, for MIMO systems, the loop transfer function and gang of four matrices are

$$\begin{aligned}
 L_u &= KG & L_y &= GK \\
 \frac{y}{y_{\text{cmd}}} &= T_y & \frac{e}{y_{\text{cmd}}} &= S_y & \frac{u}{n} &= S_u K & \frac{y}{d} &= S_y G
 \end{aligned} \tag{3.21}$$

In later sections in this chapter, and through our examples on frequency response analysis, we will discuss the information that can be obtained from examining these transfer function matrices.

3.3 Multivariable Stability Margins

Classical stability margin analyses use frequency response methods (such as Bode and Nyquist diagrams) in determining the relative stability of SISO systems. These methods manipulate the loop transfer function of the system to derive gain and phase margins, which in turn define typical measures of relative stability. In multivariable systems (MIMO systems), the loop transfer function of the system is a complex-valued matrix, making it more difficult to apply the same SISO methods to determine relative stability. The question of stability is easily answered by examining the poles of the closed-loop transfer function or the eigenvalues of the closed-loop matrix A_{cl} . It is the relative stability question, that is the determination of gain and phase margins for MIMO systems that is difficult.

In SISO systems, the gain of the loop transfer function is determined by computing the magnitude of the complex-valued transfer function versus frequency. For MIMO systems, the notion of gain or magnitude for the loop transfer function matrix becomes a question of determining the “magnitude” of a matrix versus frequency. To accomplish this task, the singular values of the matrix can be computed versus frequency and used as a measure of its magnitude.

In this section, we are concerned with deriving stability margins for MIMO systems. The robust stability analysis tests and stability margin formulas developed here are derived from application of the multivariable Nyquist theorem. These tests and formulas are natural extensions of the SISO tests reviewed in the previous section.

3.3.1 Singular Values

The singular value decomposition of a matrix $A \in R^{n \times m}$ of dimension $n \times m$ is defined as

$$A = U \Sigma V^* \quad (3.22)$$

where $*$ denotes complex conjugate transpose, and where $\Sigma \in R^{n \times m}$, $U \in C^{n \times n}$ and $V \in C^{m \times m}$ are unitary matrices, whose columns denote left and right singular vectors of the matrix A , respectively. Note the similarity to an eigenvalue decomposition of a matrix. Assuming that the matrix is of rank k , the nonzero portion of the singular value matrix is

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \tilde{\Sigma} \end{bmatrix}; \quad \Sigma_1 = \text{diag}[\sigma_1 \dots \sigma_k] \quad (3.23)$$

with the singular values ordered in size, where $\bar{\sigma} = \sigma_1$ is the largest and $\underline{\sigma} = \sigma_k$ is the smallest. The use of singular values plays an important role in analyzing the near singularity of matrices. If A is a square singular matrix, then $\underline{\sigma} = 0$.

The maximum and minimum singular values of the matrix A can be defined as

$$\bar{\sigma}(A) = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \|A\|_2, \quad \underline{\sigma}(A) = \min_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} \quad (3.24)$$

The maximum singular value of the matrix A (its 2-norm) represents how “big” the matrix is, or how large the “gain” of the matrix is. The minimum singular value represents how nearly singular the matrix is. The condition number for a matrix, $\kappa(A)$, is the ratio of the maximum and minimum singular values, given by

$$\kappa(A) = \frac{\bar{\sigma}(A)}{\underline{\sigma}(A)} \quad (3.25)$$

and it is used by numerical analyst to gain insight into how invertible a matrix is.

Associated with each singular value are singular vectors that describe the “direction” of the singular value. Consider the matrix $A \in C^{n \times m}$ with rank $k = \min(n, m)$. The k nonzero singular values of A , denoted as $\sigma_i(A)$, are the strictly positive square roots of the k nonzero eigenvalues of A^*A (or equivalently AA^*). This is expressed as

$$\sigma_i(A) = \sqrt{\lambda_i(A^*A)} = \sqrt{\lambda_i(AA^*)} > 0 \quad (3.26)$$

Each singular value has an input and output direction which can be determined by examining the singular vectors associated with the singular value decomposition (SVD) of the matrix. The SVD of a complex matrix $A \in \mathbb{C}^{n \times m}$ has the same form as (3.22),

$$A = U \Sigma V^* \quad (3.27)$$

where U is an $n \times n$ unitary matrix (i.e., $U^* = U^{-1}$) consisting of orthonormal column vectors u_i ,

$$U = [u_1 \dots u_n] \quad (3.28)$$

which are referred to as the left singular vectors of the matrix, V is a unitary matrix consisting of orthonormal column vectors v_i ,

$$V = [v_1 \dots v_m] \quad (3.29)$$

which are referred to as the right singular vectors of the matrix, and Σ is a real $n \times m$ matrix given by:

$$\Sigma = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ & & \sigma_k \\ \hline & 0 & 0 \end{bmatrix} \quad (3.30)$$

The σ_i in (3.30) is the i th singular value of the matrix A , with a corresponding left singular vector u_i (3.28) and right singular vector v_i (3.29). It is easy to show that

$$\begin{aligned} Av_i &= \sigma_i u_i \\ A^* u_i &= \sigma_i v_i \end{aligned} \quad (3.31)$$

The above equations can also be written as,

$$\begin{aligned} A^* A v_i &= \sigma_i^2 v_i \\ A A^* u_i &= \sigma_i^2 u_i \end{aligned} \quad (3.32)$$

which shows that σ_i^2 is an eigenvalue of AA^* or A^*A , and u_i is an eigenvector of AA^* , and v_i is an eigenvector of A^*A .

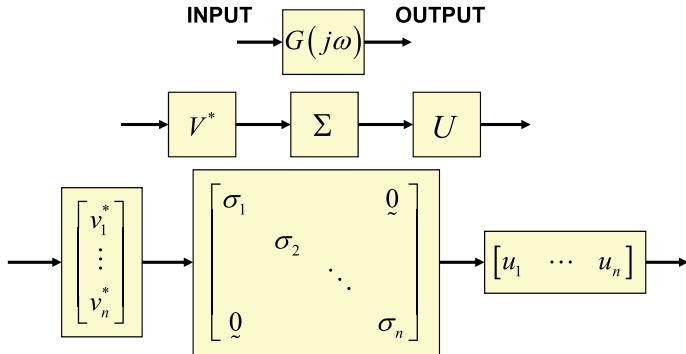


Fig. 3.5 Singular value decomposition of a transfer function matrix

Consider a square $(n \times n)$ -matrix $A \in \mathbb{C}^{n \times n}$ of rank k . Then using an SVD, the matrix A can be represented using a dyadic expansion.

$$A = \sigma_1 u_1 v_1^* + \sigma_2 u_2 v_2^* + \dots + \sigma_k u_k v_k^* = \sum_{i=1}^k \sigma_i u_i v_i^* \quad (3.33)$$

The SVD of a matrix describes the gain through the matrix, with the maximum gain equal to the 2-norm of the matrix: $\|A\|_2 = \sigma_1(A) = \bar{\sigma}(A)$. In addition to the gain, the SVD describes the direction associated with the gain. The dyadic expansion in (3.33) indicates that the left and right singular vectors describe the direction of the gain. The maximum gain through the matrix occurs with the input direction from v_1 and output direction u_1 .

Figure 3.5 illustrates the input-to-output mapping for a general transfer function matrix $G(j\omega) \in \mathbb{C}^{n \times n}$. Here the singular value expansion provides insight into the relative gain between input-to-output channels for a transfer function matrix.

Singular Value Properties

If the matrix A is invertible, that is if A^{-1} exists, then

$$\begin{aligned} \bar{\sigma}(A^{-1}) &= \frac{1}{\underline{\sigma}(A)} \text{ and } \underline{\sigma}(A^{-1}) = \frac{1}{\bar{\sigma}(A)}. \\ \|A\|_2 &= \bar{\sigma}(A), \quad \|A\|_F^2 = \sum_{i=1}^n \sigma_i^2(A) \end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. If the matrices U and V are unitary, then

$$\sigma_i(UA) = \sigma_i(A), \quad \sigma_i(AV) = \sigma_i(A)$$

which says that unitary matrices preserve the singular values and $\|\cdot\|_2$ of a matrix.

3.3.2 Multivariable Nyquist Theory

The multivariable Nyquist criterion gives a “yes or no” answer to the stability question. Other methods such as computing the eigenvalues of the system A matrix, examining the poles of the closed-loop transfer function, or solving a Lyapunov equation, can also be used to answer the stability question. However, understanding the multivariable Nyquist criterion leads to important understanding of robustness analysis tests used to evaluate model uncertainties. In addition, time delays in the frequency domain, e^{sT} , are easily modeled and incorporated into the analysis of MIMO systems with time delays.

The multivariable Nyquist criterion is derived from an application of the principle of variation of the argument, from the complex variable theory.

Theorem 3.1 (Principle of Variation of the Argument) *Let Γ be a closed counter-clockwise contour in the s -plane. Let $f(s)$ be a complex-valued function. Suppose that*

- (i) $f(s)$ is analytic on and inside Γ except at a finite number of its zeros and poles.
- (ii) $f(s)$ has Z zeros inside Γ .
- (iii) $f(s)$ has P poles inside Γ .

Then as s traverses Γ in the counterclockwise direction, $f(s)$ will encircle the origin $Z - P$ times. ■

Let $N(p, f(s), \Gamma)$ denote the number of encirclements of the point p made by the function $f(s)$ as s traverses the closed contour Γ in the clockwise direction, which is standard in controls theory. In that case, Γ is the standard Nyquist D contour (D_R) encircling the closed right half complex plane and $N(0, f(s), D_R) = Z - P$.

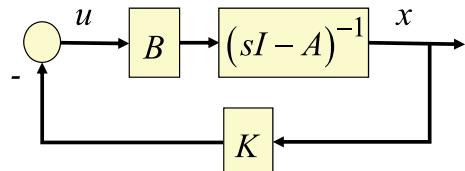
Consider the feedback system shown in Fig. 3.6. The state equations for this system are

$$\boxed{\dot{x} = Ax + Bu, \quad u = -K_x x} \Rightarrow \boxed{\dot{x} = (A - BK_x)x} \quad (3.34)$$

Let $L_u(s)$ denotes the loop gain transfer function matrix for this system.

$$L_u(s) = K_x(sI - A)^{-1}B$$

Fig. 3.6 State feedback block diagram



We can show that the determinant of the return difference matrix is equal the closed-loop characteristic polynomial $\phi_{\text{cl}}(s)$ divided by the open-loop characteristic polynomial $\phi_{\text{ol}}(s)$.

$$f(s) = \det[I + L_u(s)] = \frac{\phi_{\text{cl}}(s)}{\phi_{\text{ol}}(s)} \quad (3.35)$$

Indeed, starting with

$$\phi_{\text{cl}}(s) = \det[sI - A + BK_x] = \underbrace{\det[sI - A]}_{\phi_{\text{ol}}(s)} \det[I + (sI - A)^{-1}BK_x] \quad (3.36)$$

and using the identity relation,

$$\det \left[I_n + \underbrace{\begin{matrix} F \\ n \times m \end{matrix}}_{m \times n} \underbrace{\begin{matrix} G \\ m \times n \end{matrix}}_n \right] = \det \left[I_m + \underbrace{\begin{matrix} G \\ m \times n \end{matrix}}_{n \times m} \underbrace{\begin{matrix} F \\ n \times m \end{matrix}}_m \right]$$

proves (3.35).

$$\begin{aligned} \phi_{\text{cl}}(s) &= \phi_{\text{ol}}(s) \det[I + (sI - A)^{-1}BK_x] = \phi_{\text{ol}}(s) \det \left[I + \underbrace{\begin{matrix} K_x(sI - A)^{-1}B \\ L_u(s) \end{matrix}}_n \right] \\ &= \phi_{\text{ol}}(s) \underbrace{\det[I + L(s)]}_{f(s)} \end{aligned} \quad (3.37)$$

Now we can apply Theorem 3.1 to the function $f(s)$ in (3.35). The function has P_{cl} zeros (which are the closed-loop poles of the system) and P_{ol} poles which are the open-loop poles of the system. According to Theorem 3.1,

$$N(0, f(s), D_R) = P_{\text{cl}} - P_{\text{ol}} \quad (3.38)$$

where we reversed the encirclement direction to counterclockwise. If we assume that the closed-loop system (3.34) is stable then $P_{\text{cl}} = 0$. Moreover, if the open-loop system has P_{ol} unstable poles then (3.38) takes the standard form of the Nyquist criterion.

$$N(0, f(s), D_R) = -P_{\text{ol}} \quad (3.39)$$

With this understanding we can state the multivariable Nyquist theorem.

Theorem 3.2 (Multivariable Nyquist Theorem) *The feedback control system shown in Fig. 3.6 is closed-loop stable in the sense that $\phi_{cl}(s)$ has no closed right half plane zeros if and only if for all R sufficiently large (radius of the D contour)*

$$N(0, \det[I + L(s)], D_R) = -P_{ol} \quad (3.40)$$

or equivalently,

$$N(-1, -1 + \det[I + L(s)], D_R) = -P_{ol}$$

where $P_{ol} = N(0, \phi_{ol}(s), D_R)$ equals the number of open-loop right half plane poles. ■

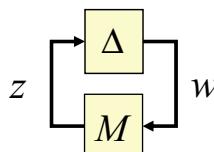
The multivariable Nyquist theorem (MNT) states that closed-loop stability requires the number of encirclements made by the determinant of the return difference matrix locus to be equal to the number of unstable open-loop poles. Encirclements can be counted relative to the origin $(0, j_0)$ or as in classical Nyquist diagrams about $(-1, j_0)$.

Stability margins for multivariable systems can be derived using the MNT by assuming that the controller $K(s)$ stabilizes the nominal plant $G(s)$, and that gain and phase uncertainties are large enough to change the number of encirclements made by the determinant of the return difference matrix locus. The assumption that the nominal plant is stabilized by the controller tells us that the return difference matrix encircles the origin P_{ol} times in the proper sense.

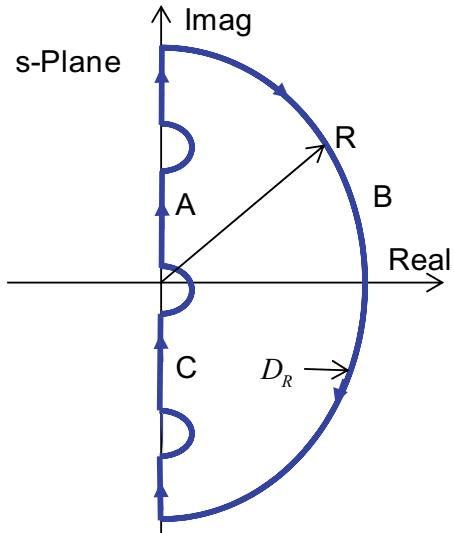
Gain and phase margins can be computed by inserting a gain and a phase variation $ke^{j\phi}$ in between the controller $K(s)$ and the plant $G(s)$, and then solving for the gain k (with $\phi = 0$) and the phase ϕ (with $k = 1$) that destabilizes the system. To proceed in a more general manner, we consider the stability analysis model shown in Fig. 3.7 where the uncertainties in the system (gain and phase uncertainties) are represented in a block matrix $\Delta(s)$ and the nominal plant and controller are represented in a matrix $M(s)$. Techniques for deriving this model will be presented in the next section.

The stability analysis question is “how large can the uncertainties $\Delta(s)$ become before the system becomes unstable?” The loop transfer function matrix for this system is $L(s) = \Delta(s)M(s)$, with the return difference matrix given by $I + L(s) =$

Fig. 3.7 ΔM analysis model



- z, w Vector Variables
- $M(s), \Delta(s)$ Transfer Functions Matrices

Fig. 3.8 Nyquist D_R contour

$I - \Delta(s)M(s)$. Using the MNT, for the system to become unstable the uncertainties $\Delta(s)$ must change the number of encirclements made by the $\det[I + L(s)]$ locus. Note that the block diagram in Fig. 3.7 has no summing node with negative sign in most block diagrams. Thus, the return difference matrix for this system is written $I - \Delta M$.

As long as the return difference matrix $I + L(s)$ is non-singular (for $s = j\omega$ along the D contour) the number of encirclements made by the locus of $\det[I + L(s)]$ will not change. This is best explained by examining the $\det[I + L(s)]$ locus as s traverses the D_R contour. Fundamental to this approach is the assumption that the nominal closed-loop system is stable, that is the control design stabilizes the open-loop system.

Assume that the nominal closed-loop system is stable, $\phi_{cl}(s)$ is a stable polynomial. Then it has no right half plane zeros. Define $f(s) = \det[I + L(s)]$, and let us represent $f(j\omega)$ via its magnitude and phase,

$$f(j\omega) = |f(j\omega)|e^{j\phi(\omega)} \quad (3.41)$$

as s traverses the D_R contour in the s -plane.

Consider the $j\omega$ axis path A shown in Fig. 3.8, where $0 \leq \omega \leq +\infty$. The section A of the locus of $f(j\omega)$ is shown in Fig. 3.9a. At low frequencies, the magnitude of $f(j\omega)$ is large due to the magnitude of $L(j\omega)$. As $\omega \rightarrow \infty$, the loop transfer matrix $L(j\omega) \rightarrow 0$, resulting in the $\det[I + L(j\omega)] = 1$ ($1, j_0$). Along the infinite radius path B, $s = e^{j\psi} R$, with $R \rightarrow \infty$ and $-\pi/2 \leq \psi \leq \pi/2$. When $R \rightarrow \infty$, $L(j\omega) \rightarrow 0$. This results in encirclements of the point $(1, j_0)$. Section C is the complex conjugate of the section A path. Figure 3.9b shows the entire locus and the number of encirclements N . Figure 3.9b shows two clockwise encirclements of the origin.

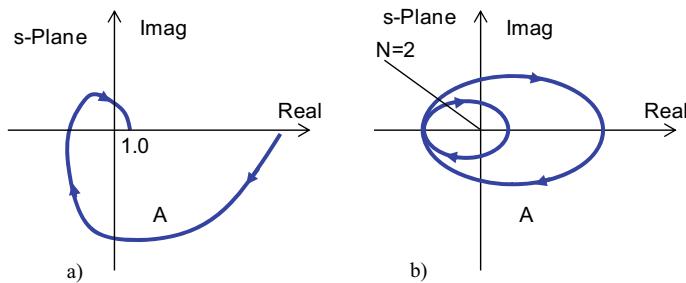


Fig. 3.9 Nyquist examples

The number of encirclements N of the $\det[I + L(s)]$ locus must be equal to the number of open-loop unstable poles, P_{ol} , if the closed-loop system is to be stable. If $\det[I + L(s)]$ is equal to zero then the number of encirclements would be indeterminate, or at least not equal to P_{ol} . This is shown in Fig. 3.10. In order for the number of encirclements to change, the $\det[I + L(s)]$ must equal zero at some frequency.

If $\phi_{\text{ol}}(s)$ is a stable polynomial, then $P_{\text{ol}} = 0$. An example $\det[I + L(s)]$ locus for this condition is shown in Fig. 3.11. In order for stable system to be destabilized by uncertainties Δ , the origin must be encircled.

Fig. 3.10 Counting encirclements

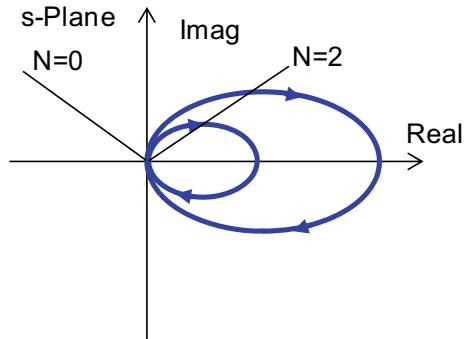
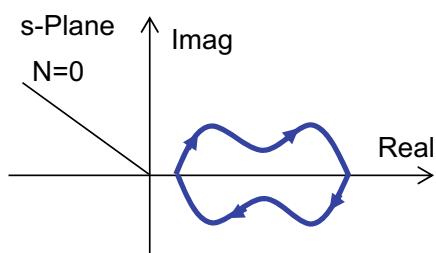


Fig. 3.11 Counting encirclements



3.3.3 Singular Value-Based Stability Margins for MIMO Systems

Uncertainty models used for stability analysis may be categorized as unstructured or structured. If the system uncertainty is modeled as a full single-block matrix, the uncertainty is unstructured. If the uncertainty is modeled as a block diagonal matrix, the uncertainty is structured. Both unstructured and structured uncertainty analysis procedures use singular value theory to measure the size of complex-valued matrices.

The following robustness theorems which are used to define stability margins for multivariable systems are derived from an application of the multivariable Nyquist theorem. Consider the state feedback control system shown in Fig. 3.6. The basic problem is to determine the robustness of the design in the presence of uncertainties. This design has the state-space realization using the triple (A, B, K) , with the loop transfer matrix (LTM) given by

$$L_u(s) = K_x(sI - A)^{-1}B \quad (3.42)$$

We wish to determine to what extent gain and phase uncertainty within the LTM can vary without compromising the stability of the closed-loop system. From the previous section (Eq. 3.37) we know that

$$\det[I + L(s)] = \frac{\phi_{cl}(s)}{\phi_{ol}(s)} \quad (3.43)$$

where

$$\begin{aligned} \phi_{ol}(s) &= \det[sI - A] && \text{open-loop characteristic polynomial} \\ \phi_{cl}(s) &= \det[sI - A + BK_x] && \text{closed-loop characteristic polynomial.} \end{aligned}$$

Using the multivariable Nyquist theorem, stability for this system can be asserted. In this case, the system of Fig. 3.6 is closed-loop stable in the sense that $\phi_{cl}(s)$ has no closed right half plane zeros if and only if for all R sufficiently large

$$N(0, \det[I + L(s)], D_R) = -P_{ol} \quad (3.44)$$

or equivalently

$$N(-1, -1 + \det[I + L(s)], D_R) = -P_{ol} \quad (3.45)$$

where D_R is the standard Nyquist D contour, which encloses all P_{ol} closed right half plane zeros of $\phi_{ol}(s)$. Note that $N(b_1, f(s), D)$ is indeterminate if $\phi(s_0) = b_1$ for some s_0 on the contour D .

The stability robustness of a multivariable system can be observed by the near singularity of its return difference matrix, $I + L(s)$, at some frequency $s = j\omega_0$. If $I + L(s)$ is nearly singular then a small change in $L(s)$ could make $I + L(s)$ singular. From the SISO systems viewpoint, this is the distance from the $(-1, j_0)$

point in the complex plane made by the gain loci $L(j\omega)$. If the gain loci then encircles the $(-1, j_0)$ point, instability results. The robustness theory discussed here gives an analogous distance measure for MIMO systems.

Application of the multivariable Nyquist theorem above is of little interest as a robustness indicator because the $\det[I + L(s)]$ does not indicate the near singularity of $I + L(s)$. The multivariable Nyquist theorem only determines absolute stability. To determine the degree of robustness for a multivariable system, we determine *how nearly singular the return difference matrix is* by computing its singular values versus frequency.

Examining the magnitude of the singular values of the return difference matrix will indicate how close the matrix is to being singular. This measure of closeness to singularity is used in forming a multivariable gain and phase margins, similar to the classical SISO margins. However, as with many matrix norms, there is a restriction on the applicability of the singular value analysis. This restriction states that the compensated system described using the nominal $L(s)$ is closed-loop stable. This means that the controller stabilizes the system under the nominal model.

Classical SISO Stability Margins

Classical *gain, phase, and vector margins* are used to measure relative stability of SISO systems to perturbations in model parameters within the feedback loop. They are widely employed in practice and are often required when certifying flight control systems. Stability margins provide an easily understood measure of how sensitive the control system is to unknown gain and phase characteristics of models used to describe the system. We will review these classical stability margins next so we can compare and contrast them to singular value-based margins for MIMO systems.

Consider the SISO LTI system shown in Fig. 3.12. In the block diagram, we have inserted a block containing $ke^{j\phi}$ which is used to model gain and phase changes in the $L(j\omega) = K(j\omega)G(j\omega)$ caused by perturbation in model parameters. To proceed, define the following quantities:

Definition 3.1 (Phase-Crossover Point) A *phase-crossover point* is the point where $\angle L(j\omega) = -180^\circ$.

In a Nyquist plot this is where the $L(j\omega)$ locus intersects the negative real axis. In a Bode plot of $L(j\omega)$, it is the point where the phase angle, $\angle L(j\omega)$, crosses -180° .

Definition 3.2 (Phase-Crossover Frequency) A *phase-crossover frequency* is the frequency ω_c at the phase-crossover point.

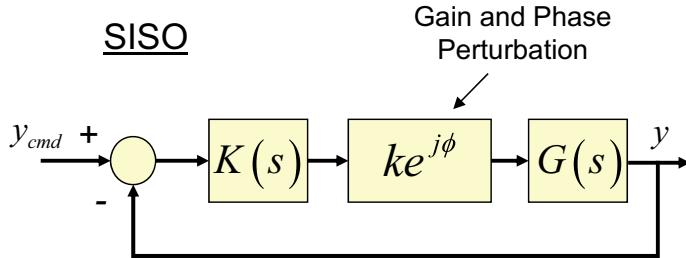


Fig. 3.12 SISO LTI system with gain and phase perturbation

This occurs in either a Nyquist plot or Bode plot when $\angle L(j\omega) = -180^\circ$.

Definition 3.3 (*Loop Gain-Crossover Point*) A *loop gain-crossover point* is the point where the magnitude of $L(j\omega)$ is equal to 1, $|L(j\omega)| = 1$.

On a Nyquist plot $L(j\omega)$ the gain-crossover points are where the locus of $L(j\omega)$ crosses a unit circle centered at the origin. On a Bode plot of $L(j\omega)$ this is where $|L(j\omega)|$ crosses 0 dB.

Definition 3.4 (*Gain-Crossover Frequency*) A *loop gain-crossover frequency* (LGCF) is the frequency ω_g at the gain-crossover point.

Consider the Nyquist plot in Fig. 3.13 for the system in Fig. 3.12. Gain Margin (GM) describes how close the locus of $L(j\omega)$ is to the *critical point* $(-1, j_0)$ in the Nyquist plane. It is a one-dimensional measure of the relative stability of the closed-loop system. Formally, GM is defined as the smallest increase or decrease of the value of k in decibels (dB) that makes the system unstable. GM is computed using the value of $L(j\omega_c)$ on the negative real axis, $-1/g_m$. The point $-1/g_m$ occurs at the phase-crossover frequency ω_c . In principle, a system with large GM would be more stable than a system with a small GM. However, GM by itself is inadequate at measuring relative stability when parameters in $L(j\omega)$ are subject to variation. Consider the Nyquist plot shown in Fig. 3.14. These two systems appear to have the same GM but $L_A(j\omega)$ is more stable than $L_B(j\omega)$. The reason is that if parameters in the plant model affect the phase it is much easier for $L_B(j\omega)$ to pass through -1 than $L_A(j\omega)$. This fact leads to our discussion on phase margins for a system which include the effect of phase shift on stability.

Fig. 3.13 Nyquist plot showing gain, phase, and vector margin

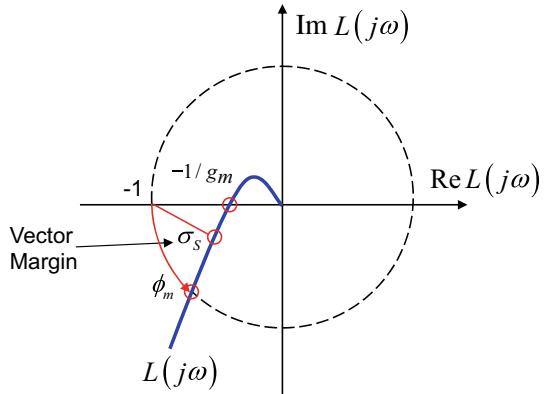
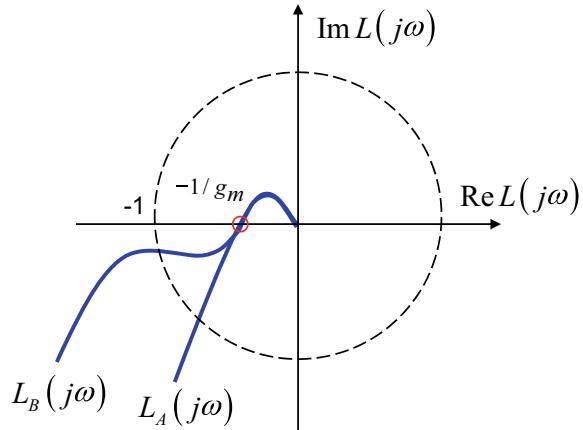


Fig. 3.14 Nyquist plot showing relative stability of two SISO systems



Phase Margin (PM) is the smallest amount of phase lag $e^{j\phi}$ that makes the system unstable. It is the angle in degrees that the locus of $L(j\omega)$ (see Fig. 3.13) must be rotated about the origin in order for the gain-crossover point to pass through the critical point $(-1, j_0)$. The PM indicates the effect on stability due to parameter perturbations in the plant model which theoretically alter the phase of $L(j\omega)$ by an equal amount at all frequencies. We see in Fig. 3.14 that $L_B(j\omega)$ lacks sufficient phase margin.

A formula for computing SISO margins (Fig. 3.12) is as follows.

To compute gain margin:

1. Set $\phi = 0 (e^{j\phi} = 1)$
2. Increase gain k until system is unstable. +GM:
3. Decrease gain k until system is unstable. -GM:

To compute phase margin:

1. Set $k = 1$
2. Increase gain ϕ until system is unstable. +PM:
3. Decrease gain ϕ until system is unstable. -PM:

Gain Margin

$$GM = 20 \log_{10} \frac{1}{|L(j\omega_c)|} \text{ dB} \quad (3.46)$$

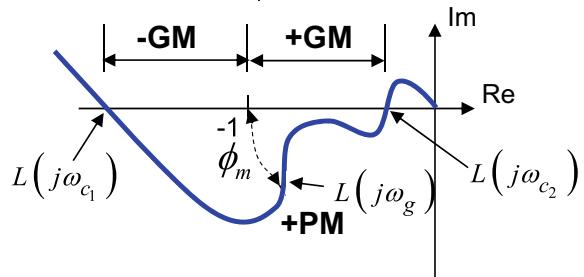
Phase Margin

$$PM = \angle L(j\omega_g) - 180^\circ \quad (3.47)$$

In aerospace applications, it is particularly important to discuss relative stability in the context of systems that are open-loop unstable. The formula stated above for computing GM and PM outlines a procedure for +GM and -GM. For an open-loop stable system, the +GM measures the gain increase that causes the system to go unstable. In Fig. 3.13 as the gain is increased the red circle on the real axis would move out radially to the point $(-1, j_0)$ (no phase shift). The -GM (the amount of gain decrease) for Fig. 3.13 would be infinite. The gain can decrease to zero, and the system would remain stable. Figure 3.15 shows an example Nyquist plot for open-loop system unstable which has a single pole in the RHP. For this system the locus of $L(j\omega)$ will encircle $(-1, j_0)$ $N = -P_{ol} = -1$ times. This creates two phase-crossover points at frequencies ω_{c1}, ω_{c2} , respectively. For the +GM, the gain k in Fig. 3.12 is increased until the phase-crossover point $L(j\omega_{c2})$ touches the critical point $(-1, j_0)$. For -GM, k is decreased until the phase-crossover point $L(j\omega_{c1})$ touches the critical point $(-1, j_0)$.

We see from our discussion of classical SISO gain and phase margins that both of these are one-dimensional measures of the relative stability of the closed-loop system. A margin that considers both gain and phase occurring together is called the *vector margin*. The vector margin is the radius of the smallest circle centered

Fig. 3.15 Nyquist plot for unstable open-loop system



at $(-1, j_0)$ tangent to $L(j\omega)$. In Fig. 3.13 this is shown as σ_S . Numerically it is computed as follows.

Vector Margin

$$\sigma_S = \min_{\omega} |1 + L(j\omega)| \quad (3.48)$$

The locus of $L_B(j\omega)$ in Fig. 3.14 and $L(j\omega)$ in Fig. 3.15 both show low (small) vector margins. This indicates that both of these systems are sensitive to simultaneous gain and phase perturbations.

Singular Value-Based Stability Margins for MIMO Systems

As previously discussed, singular values are used in measuring the robustness of multivariable systems by determining the near singularity of the return difference matrix. Let $L'(s)$ denote the perturbed loop transfer matrix (LTM), which represents the actual system and differs from the nominal LTM $L(s)$ because of uncertainties in the open-loop plant model. Assume that $L'(s)$ has the state-space realization (A', B', K') and that both open-loop and closed-loop characteristic polynomials are given by,

$$\phi'_{\text{ol}}(s) = \det[sI - A'], \quad \phi'_{\text{cl}}(s) = \det[sI - A' + B'K'] \quad (3.49)$$

respectively. Define $\tilde{L}(s, \varepsilon)$ as a matrix of rational transfer functions with real coefficients which are continuous in ε for all ε such that $0 \leq \varepsilon \leq 1$ and for all $s \in D_R$, which satisfies $\tilde{L}(s, 0) = L(s)$ and $\tilde{L}(s, 1) = L'(s)$. Using these definitions of the perturbed model we are ready to state the following Fundamental Robustness Theorem.

Theorem 3.3 (Fundamental Robustness Theorem) *The polynomial $\phi'_{\text{cl}}(s)$ has no zeros in the closed right half plane, and the perturbed feedback system is stable if the following hold:*

1. (a) $\phi_{\text{ol}}(s)$ and $\phi'_{\text{ol}}(s)$ have the same number of zeros in the closed right half plane.
 (b) $\phi_{\text{cl}}(s)$ has no zeros in the closed right half plane.
2. $\det[I + L(s, \varepsilon)] \neq 0$ for all (s, ε) in $D_R \times [0, 1]$ and for all R sufficiently large. ■

These are sufficient conditions for closed-loop stability of the perturbed system. If by continuously deforming the Nyquist loci for the nominal system into that of the perturbed system $I + \tilde{L}(s, \varepsilon)$, the number of encirclements of the critical point $(-1, j_0)$ is the same for $L'(s)$ and $L(s)$, then no closed right half plane zeros were introduced into $\phi'_{\text{cl}}(s)$, resulting in a stable closed-loop system.

This theorem is used to develop simple tests for different types of model error characterizations. Just as there is no unique representation for dynamic systems there are many different forms for describing their modeling errors. The

most common model error characterizations are *additive* errors and *multiplicative* errors (also described as relative or absolute errors). The classical gain and phase margins are associated with multiplicative error models since these margins are multiplicative in nature, (see [2], Table 1, for representative types of uncertainty characterizations.)

Let $\Delta(s)$ denotes the modeling error under consideration. The additive model error is given by

$$\Delta_a(s) = L'(s) - L(s) \quad L'(s) = L(s) + \Delta_a(s) \quad (3.50)$$

and the multiplicative model error is given by

$$\Delta_m(s) = [L'(s) - L(s)]L^{-1}(s) \quad L'(s) = L(s)(I + \Delta_m(s)) \quad (3.51)$$

The perturbed LTM can be constructed using Eqs. (3.50) and (3.51). For the additive error model we have

$$\tilde{L}(s, \varepsilon) = L(s) + \varepsilon \Delta_a(s) \quad (3.52)$$

and for the multiplicative error model we have

$$\tilde{L}(s, \varepsilon) = [I + \varepsilon \Delta_m(s)]L(s) \quad \tilde{L}(s, \varepsilon) = L(s)(I + \varepsilon \Delta_m(s)) \quad (3.53)$$

Both Eqs. (3.50) and (3.51) define the same $\tilde{L}(s, \varepsilon)$ using different model error characterizations, given by

$$\tilde{L}(s, \varepsilon) = (1 - \varepsilon)L(s) + \varepsilon L'(s) \quad (3.54)$$

showing that $\tilde{L}(s, \varepsilon)$ is continuous in ε for $\varepsilon \in [0, 1]$ and for all $s \in D_R$.

We have now defined the true perturbed plant model in terms of its nominal design model and the uncertainty matrix. The Fundamental Robustness Theorem uses the return difference matrix $I + \tilde{L}(s, \varepsilon)$ to determine if the number of encirclements of the critical point will change with the uncertainties. This happens when $I + \tilde{L}(s, \varepsilon)$ becomes singular, in which case: $\det[I + \tilde{L}(s, \varepsilon)] = 0$.

Of particular interest to this discussion is the uncertain MIMO LTI system depicted in Fig. 3.16 in which we define a model Δ for the gain and phase uncertainties. In Fig. 3.16 the matrix $\Delta = \text{diag}[k_i e^{j\phi_i}]$ models simultaneous independent gain and phase perturbation in each input channel that could cause the system to lose stability. Our interest is in determining the smallest set (in each channel independently) that causes the perturbed return difference matrix to become singular, $\det[I + \tilde{L}(s, \varepsilon)] = 0$. The return difference matrix for this system model is

$$I_u + \tilde{L}_u = I_u + K G \Delta \quad (3.55)$$

To investigate the near singularity of the return difference matrices (3.55), we will use the following lemma.

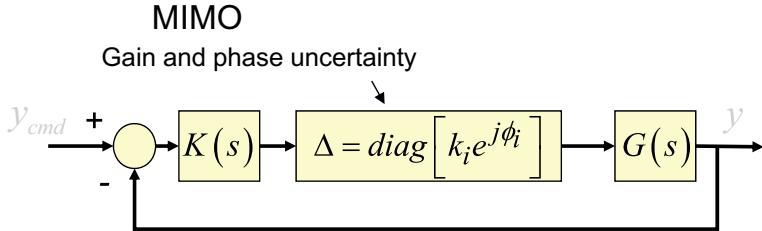


Fig. 3.16 MIMO LTI system with gain and phase perturbation

Lemma 3.1 (The $(A + B)$ Argument) Consider a non-singular matrix A and a matrix B of the same dimension. If $\underline{\sigma}(A) > \bar{\sigma}(B)$ then $(A + B)$ is non-singular.

Proof of Lemma 3.1 Suppose that the sum $(A + B)$ is singular. Then $(A + B)$ is rank deficient and it has a null space, that is there exists a nonzero unit vector x such that

$$(A + B)x = 0 \Rightarrow Ax = -Bx \Rightarrow \|Ax\|_2 = \|Bx\|_2 \quad (3.56)$$

Consequently,

$$\underline{\sigma}(A) \leq \|Ax\|_2 = \|Bx\|_2 \leq \bar{\sigma}(B) \Rightarrow \underline{\sigma}(A) \leq \bar{\sigma}(B) \quad (3.57)$$

We proved that if $(A + B)$ is singular then $\underline{\sigma}(A) \leq \bar{\sigma}(B)$. Negating the argument proves the lemma. ■

Consider the system in Fig. 3.16 in which the nominal system is assumed to be closed-loop stable. Thus, $\phi_{cl}(s)$ for the nominal system has no zeros in the closed right half plane and $I_u + L_u$ is non-singular. From (3.55),

$$\begin{aligned}
 I_u + L_u \Delta [\pm \Delta] &= I_u - \underbrace{\Delta}_{S_u^{-1}} + (I_u + L_u) \Delta \\
 &= S_u^{-1} (S_u (I_u - \Delta) + \Delta) = S_u^{-1} \underbrace{\left(S_u (\Delta^{-1} - I_u) + I_u \right)}_{S_\Delta} \Delta \\
 &= S_u^{-1} S_\Delta \Delta
 \end{aligned} \quad (3.58)$$

Suppose in (3.58) the perturbation Δ causes $S_u^{-1} S_\Delta \Delta$ to become singular. We know that S_u^{-1} is non-singular since the nominal system is closed-loop stable. We know Δ is non-singular since it is a diagonal matrix of nonzero elements. For the matrix product $S_u^{-1} S_\Delta \Delta$ to be singular, the matrix $S_\Delta = S_u (\Delta^{-1} - I_u) + I_u$ must

be singular. Let $A = I_u$, $B = S_u(\Delta^{-1} - I_u)$, and apply the $(A + B)$ argument. The matrix S_Δ will be non-singular if $\underline{\sigma}(A) > \bar{\sigma}(B)$. Substituting, yields:

$$\bar{\sigma}(S_u(\Delta^{-1} - I_u)) < \underbrace{\underline{\sigma}(I_u)}_1 \Rightarrow \det S_\Delta(j\omega) \neq 0 \quad (3.59)$$

Now use $\bar{\sigma}(AB) \leq \bar{\sigma}(A)\bar{\sigma}(B)$ to produce

$$\bar{\sigma}(S_u(\Delta^{-1} - I_u)) \leq \bar{\sigma}(S_u)\bar{\sigma}(\Delta^{-1} - I_u) < 1 \quad (3.60)$$

The condition $\bar{\sigma}(S_u)\bar{\sigma}(\Delta^{-1} - I_u) < 1$ in (3.60) is a key result. It gives a sufficient condition for closed-loop stability using the sensitivity matrix S_u that implies the return difference matrix in (3.55) is non-singular. Next, rewrite (3.58) as

$$\begin{aligned} I_u + L_u \Delta [\pm L_u] &= \underbrace{(I_u + L_u)}_{S_u^{-1}} + L_u(\Delta - I_u) \\ &= S_u^{-1} \left(I_u + \underbrace{S_u L_u}_{T_u} (\Delta - I_u) \right) = S_u^{-1} \underbrace{[(I_u + T_u(\Delta - I_u))]}_{T_\Delta} \\ &= S_u^{-1} T_\Delta \end{aligned} \quad (3.61)$$

We know that S_u^{-1} is non-singular. For the matrix product $S_u^{-1} T_\Delta$ to be singular the matrix $T_\Delta = I_u + T_u(\Delta - I_u)$ must be singular. Let $A = I_u$, $B = T_u(\Delta - I_u)$, and apply the $(A + B)$ Argument. The matrix T_Δ will be non-singular if $\underline{\sigma}(A) > \bar{\sigma}(B)$. Substituting yields

$$\bar{\sigma}(T_u(\Delta - I_u)) < \underbrace{\underline{\sigma}(I_u)}_1 \Rightarrow \det T_\Delta(j\omega) \neq 0 \quad (3.62)$$

Now use $\bar{\sigma}(AB) \leq \bar{\sigma}(A)\bar{\sigma}(B)$ to produce:

$$\bar{\sigma}(T_u(\Delta - I_u)) \leq \bar{\sigma}(T_u)\bar{\sigma}(\Delta - I_u) < 1 \quad (3.63)$$

The condition $\bar{\sigma}(T_u)\bar{\sigma}(\Delta - I_u) < 1$ is also a key result. It also provides a sufficient condition for closed-loop stability, but it uses the complementary sensitivity matrix T_u . Combining (3.60) and (3.63), we can state the following two Stability Robustness Theorems.

Theorem 3.4 (Stability Robustness Theorem—Sensitivity Matrix) *The polynomial $\phi'_{\text{cl}}(s)$ has no closed right half plane zeros, and the perturbed feedback system is stable if the following hold:*

1. $\phi_{\text{cl}}(s)$ has no zeros in the closed right half plane.
2. $\bar{\sigma}(\Delta^{-1} - I_u) < 1/\bar{\sigma}(S_u) \forall s \in D_R$ and for all R sufficiently large, with Δ given in (3.55). ■

Theorem 3.5 (Stability Robustness Theorem—Complementary Sensitivity Matrix)
The polynomial $\phi'_{\text{cl}}(s)$ has no zeros in the closed right half plane, and the perturbed feedback system is stable if the following hold:

1. $\phi_{\text{cl}}(s)$ has no zeros in the closed right half plane.
2. $\bar{\sigma}(\Delta - I_u) < 1/\bar{\sigma}(T_u) \forall s \in D_R$ and for all R sufficiently large, with Δ given in (3.55). ■

These two theorems give sufficient conditions for the return difference matrix in (3.55) to be non-singular. As long as the singular value frequency responses do not overlap, stability is guaranteed. Using (3.60) and (3.63) bounds on the uncertainty Δ can be formed as:

$$\begin{aligned} \bar{\sigma}(\Delta^{-1}(j\omega) - I_u) &< \max_{\omega} \frac{1}{\bar{\sigma}(S_u(j\omega))} = \sigma_S \\ \text{or} \\ \bar{\sigma}(\Delta(j\omega) - I_u) &< \max_{\omega} \frac{1}{\bar{\sigma}(T_u(j\omega))} = \sigma_T \end{aligned} \quad (3.64)$$

where the bounds σ_S and σ_T are both computed from the nominal sensitivity and complementary sensitivity matrices, respectively. Thus

$$\boxed{\bar{\sigma}(\Delta^{-1} - I_u) < \sigma_S} \vee \boxed{\bar{\sigma}(\Delta - I_u) < \sigma_T} \Rightarrow \boxed{\det[I_u + L_u \Delta(j\omega)] \neq 0} \quad (3.65)$$

Closed-Loop Stability

Consider the uncertainty model in Fig. 3.16.

$$\Delta = \text{diag}\left[k_i e^{j\phi_i}\right] = \text{diag}[\Delta_{ii}]. \quad (3.66)$$

The singular values of a diagonal matrix are equal to the absolute value of the diagonal elements. Thus $\sigma_i(\Delta^{-1} - I_u) = |\Delta_{ii}^{-1} - 1|$, where Δ_{ii} is the i th diagonal element of Δ . Using the sensitivity and complementary sensitivity matrix bounds in (3.64) we have

$$\boxed{\bar{\sigma}(\Delta^{-1} - I_u) = \max_i |\Delta_{ii}^{-1} - 1| < \sigma_S \Leftrightarrow 1 - \sigma_S \leq \max_i \Delta_{ii}^{-1} \leq 1 + \sigma_S} \quad (3.67)$$

and

$$\boxed{\bar{\sigma}(\Delta - I_u) = \max_i |\Delta_{ii} - 1| < \sigma_T \Leftrightarrow 1 - \sigma_T \leq \max_i \Delta_{ii} \leq 1 + \sigma_T} \quad (3.68)$$

Using the classical definition of the gain margin, $\Delta_{ii} = k_i$, $1 - \sigma_S \leq 1/k_i \leq 1 + \sigma_S$ and $1 - \sigma_T \leq k_i \leq 1 + \sigma_T$ yields a gain margin described by the union of two intervals.

$$\left[\frac{1}{1 + \sigma_S}, \frac{1}{1 - \sigma_S} \right] \cup [1 - \sigma_T, 1 + \sigma_T] \quad (3.69)$$

Consider $|e^{j\phi_i} - 1| < \sigma$. Then

$$\begin{aligned} |e^{j\phi_i(\omega)} - 1| &= |\cos(\phi_i(\omega)) - 1 + j \sin(\phi_i(\omega))| \\ &= (\cos^2(\phi_i(\omega)) - 2 \cos(\phi_i(\omega)) + 1 + \sin^2(\phi_i(\omega)))^{\frac{1}{2}} \\ &= (2(1 - \cos(\phi_i(\omega))))^{\frac{1}{2}} = \left(4 \sin^2\left(\frac{\phi_i(\omega)}{2}\right) \right)^{\frac{1}{2}} \leq \sigma \end{aligned}$$

Using the classical phase margin model, $\Delta_{ii} = e^{j\phi_i}$, the sensitivity and complementary sensitivity matrix bounds in (3.64) produce:

$$\bar{\sigma}(\Delta^{-1} - I_u) = \max_i |e^{-j\phi_i} - 1| = \max_i \left| 2 \sin \frac{\phi_i}{2} \right| < \sigma_S \quad (3.70)$$

and

$$\bar{\sigma}(\Delta - I_u) = \max_i |e^{j\phi_i} - 1| = \max_i \left| 2 \sin \frac{\phi_i}{2} \right| < \sigma_T \quad (3.71)$$

which gives a phase margin prediction for the system given as a union of the two intervals.

$$\left(2 \arcsin\left(\frac{\sigma_S}{2}\right) [-1, 1] \right) \cup \left(2 \arcsin\left(\frac{\sigma_T}{2}\right) [-1, 1] \right) \quad (3.72)$$

Singular Value Stability Margins

Let $\sigma_S = 1/\|S_u\|_\infty$ and $\sigma_T = 1/\|T_u\|_\infty$. Both σ_S and σ_T are not in dB. The final GM and PM are converted to dB and degrees after substitution. Then

$$\text{GM}_S = \left[\frac{1}{1 + \sigma_S}, \frac{1}{1 - \sigma_S} \right] \text{dB}; \quad \text{PM}_S = \pm 2 \sin^{-1} \frac{\sigma_S}{2} \text{ degree} \quad (3.73)$$

$$\text{GM}_T = [1 - \sigma_T, 1 + \sigma_T] \text{dB}; \quad \text{PM}_T = \pm 2 \sin^{-1} \frac{\sigma_T}{2} \text{ degree} \quad (3.74)$$

Then, the $[-\text{GM}, +\text{GM}]$ and PM for the system are

$$\text{GM} = \left[1 - \sigma_T, \frac{1}{1 - \sigma_S} \right] \text{dB}, \quad \text{PM} = \max(\text{PM}_S, \text{PM}_T) \text{ degree} \quad (3.75)$$

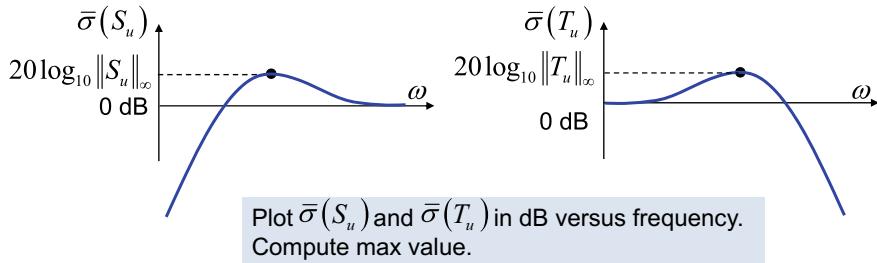
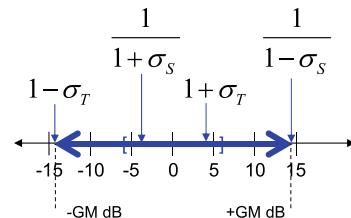


Fig. 3.17 Magnitude plot of S_u and T_u versus frequency for stability margin analysis

Consider the frequency response plots of S_u and T_u in Fig. 3.17. The smallest (best) value of $\|S_u\|_\infty$ is $\|S_u\|_\infty = 1$ (all $\sigma_i(S_u) = 1$ at high frequency where $L_u = 0_{n_u \times n_u}$). Substituting this into (3.73) produces a gain margin interval of $GM_S = [1/2, +\infty]$, and converting to decibels produces $GM_S = [-6, +\infty]$ dB. Similarly, the smallest (best) value of $\|T_u\|_\infty$ is $\|T_u\|_\infty = 1$ (all $\sigma_i(T_u) = 1$ at low frequency where L_u is large). Substituting this into (3.74) produces a gain margin interval of $GM_T = [0, 2]$, and converting to decibels produces $GM_T = [-\infty, +6]$ dB. We see that analysis of the sensitivity matrix S_u gives a GM range that covers a positive GM (an increasing gain k_i in (3.66)), with a lower limit of -6 dB. The complementary sensitivity matrix T_u gives a GM range that covers a negative GM (a decreasing gain k_i in (3.66)), with an upper limit of 6 dB. The union of these GM ranges is the total GM (and PM) for the system as shown in (3.75). Using (3.73), (3.74), and (3.75) to compute a GM results in a $-GM$ and $+GM$ as shown in Fig. 3.18. The resulting GM for the system will be $[1 - \sigma_T, 1/1 + \sigma_S]$ dB with a PM equal to the $\max(PM_S, PM_T)$ degree.

Example 3.2 Gain and Phase Margins Using Singular Values Consider the acceleration control of an unpiloted aircraft presented in Example 3.1 which is controlled using the classical proportional-plus-integral control architecture shown in Fig. 3.3.

Fig. 3.18 Singular value-based gain margins



Closed loop system will remain stable for gain variations within this range

Since the pitch-plane dynamics has a single input, we can use this example to compute both classical and singular value stability margins and relate them to each other. This will provide insight into how the singular value margins can be interpreted.

We will use a high-speed open-loop unstable flight condition and will add a second-order actuator model on the elevator. The dynamics are:

$$\begin{aligned}\dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta_e + q \\ \dot{q} &= M_\alpha\alpha + M_\delta\delta_e \\ \ddot{\delta}_e &= -2\zeta\omega_n\dot{\delta}_e - \omega_n^2(\delta_e - \delta_{\text{cmd}})\end{aligned}\quad (3.76)$$

In this example the pitch rate damping stability derivative $M_q = 0$. The actuator natural frequency is $\omega_n = 113$ rad/s with a damping factor $\zeta = 0.6$. The feedback variables used in this PI controller come from the Inertial Measurement Unit (IMU) and represent the vehicle acceleration A_z and pitch rate q , where $A_z = Z_\alpha\alpha + Z_\delta\delta_e$. For analysis and plotting purposes we will include the angle of attack (AOA) α and the actuator states $(\delta_e, \dot{\delta}_e)$ in modeling the plant output. Note that these variables are not used by the controller. The open-loop plant model (from Eq. 2.58) is

$$\begin{aligned}\dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u\end{aligned}\quad (3.77)$$

where $x_p = [\alpha \ q \ \delta_e \ \dot{\delta}_e]^T$, $u = \delta_{\text{cmd}}$, and $y = [A_z \ \alpha \ q \ \delta_e \ \dot{\delta}_e]^T$. The numerical values for the matrices are:

$$\begin{bmatrix} A_p & B_p \\ C_p & D_p \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} -1.3046 & 1.0 & -0.21420 & 0 \\ 47.71 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 1.0 \\ 0 & 0 & -12769.0 & -135.6 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 12769.0 \end{bmatrix} \\ \begin{bmatrix} -1156.9 & 0 & -189.95 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} \quad (3.78)$$

The controller model (from Eq. 2.59) is

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r\end{aligned}\quad (3.79)$$

with the matrices defined in (3.14). The gains are $K_a = -0.0015$, $K_q = -0.32$, $a_z = 2.0$, and $a_q = 6.0$. Substituting these values into (3.14) yields

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ -1.92 & 0 \\ -0.32 & 1.0 \end{bmatrix} & \begin{bmatrix} 0.0030 & 0 & 0 & 0 \\ -0.0029 & 0 & 1.92 & 0 \\ -0.0005 & 0 & 0.32 & 0 \end{bmatrix} & \begin{bmatrix} -0.0030 \\ 0.0029 \\ 0.00048 \end{bmatrix} \\ \begin{bmatrix} x_{c1} & x_{c2} \end{bmatrix} & \begin{bmatrix} A_z & \alpha & q & \dot{\delta}_e \end{bmatrix} & \begin{bmatrix} A_{z_{cmd}} \end{bmatrix} \end{bmatrix} \quad (3.80)$$

Next, we connect the controller and the plant state-space models (see Eqs. 2.65 and 2.67) and simulate the closed-loop system (A_{cl} , B_{cl} , C_{cl} , D_{cl}) to verify that the model is correctly connected. Then we use the loop gain models (A_{L_u} , B_{L_u} , C_{L_u} , D_{L_u}) and (A_{L_y} , B_{L_y} , C_{L_y} , D_{L_y}) (see Eqs. 2.72 and 2.76) to compute the necessary frequency responses and the gang of 4. Figure 3.19 shows a step response of the closed-loop system indicating that the plant and the controller are connected properly.

For this SIMO system we will compute a Nyquist, Bode, S_u , and T_u plots at the plant input using L_u . The Nyquist and Bode plots are open-loop analysis plots at the plant input. The S_u , and T_u plots are “closed-loop” analysis plots at the plant input. The gang of 4 analysis for this example computes the following:

$$\frac{y}{y_{cmd}} = T_y \quad \frac{e}{y_{cmd}} = S_y \quad \frac{u}{n} = S_u K \quad \frac{y}{d} = S_y G \quad (3.81)$$

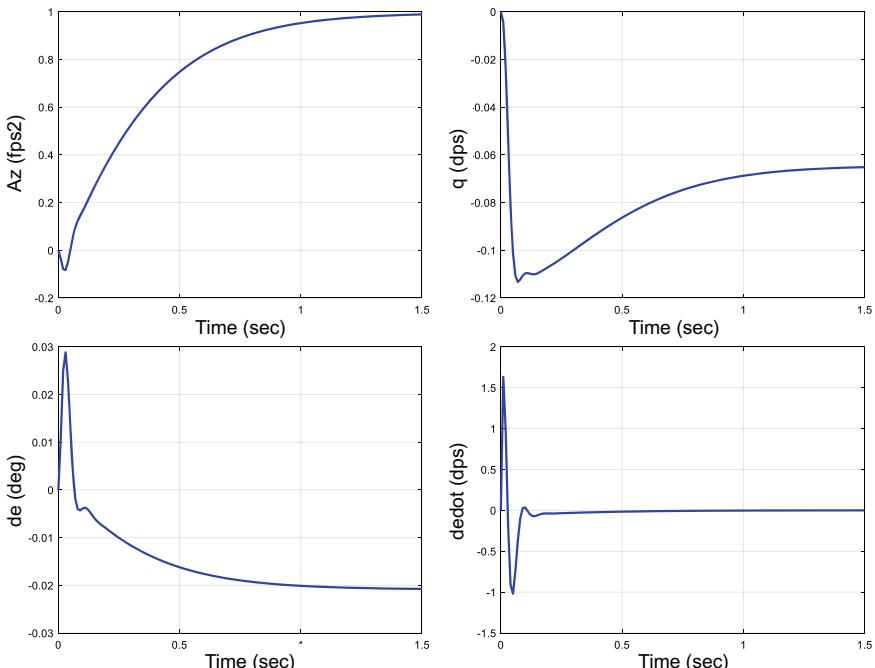


Fig. 3.19 Acceleration step response for Example 3.3

where K represents the controller and G the plant. Note that $y/y_{\text{cmd}} = A_z/A_{z_{\text{cmd}}}$ and $e/y_{\text{cmd}} = e_{A_z}/A_{z_{\text{cmd}}}$ are scalars. These plots give insight into the frequency response of the regulated variable A_z . The noise-to-control frequency response u/n is a 1×2 matrix, $u/n = [\delta_e/A_{z_{\text{noise}}} \quad \delta_e/q_{\text{noise}}]$. We will also examine $\dot{u}/n = [\dot{\delta}_e/A_{z_{\text{noise}}} \quad \dot{\delta}_e/q_{\text{noise}}]$. The load disturbance plot y/d is also a scalar.

Figure 3.20 through Fig. 3.23 show the plant-input frequency response curves (Nyquist, Bode, $|S_u|$, and $|T_u|$). On the Nyquist plot in Fig. 3.20 we have drawn a circle centered at $(-1, j_0)$ that has radius equal to the maximum of $|S_u|$ (from Fig. 3.16). We also locate on the figure the phase and loop gain crossovers. The classical gain and phase margins from Fig. 3.20 are 8.8 dB (3.7536) and 50° . These are also easily extracted from the Bode plot in Fig. 3.21. From Figs. 3.22 and 3.23, we have

$$\sigma_S = 1/\|S_u\|_\infty = 0.5676; \quad \sigma_T = 1/\|T_u\|_\infty = 0.7305 \quad (3.82)$$

These values are substituted into the singular value gain and phase margins in (3.73) and (3.74):

$$\begin{aligned} \text{GM}_S &= \left[\frac{1}{1+\sigma_S}, \frac{1}{1-\sigma_S} \right] \text{dB} \\ &= [0.6379 \quad 2.3127] \\ &= [-3.9 \quad 7.28] \text{ dB} \end{aligned} \quad \begin{aligned} \text{PM}_S &= \pm 2 \sin^{-1} \frac{\sigma_S}{2} \\ &= \pm 32.97 \text{ degree} \end{aligned} \quad (3.83)$$

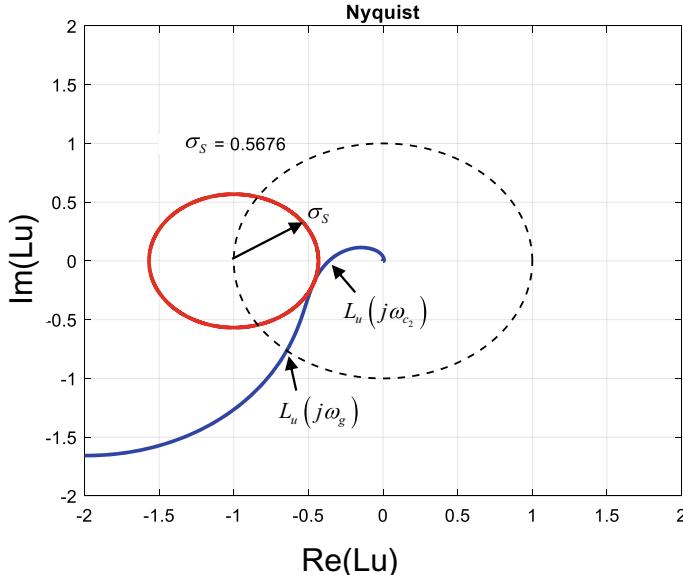


Fig. 3.20 Nyquist plot at the plant-input loop breakpoint

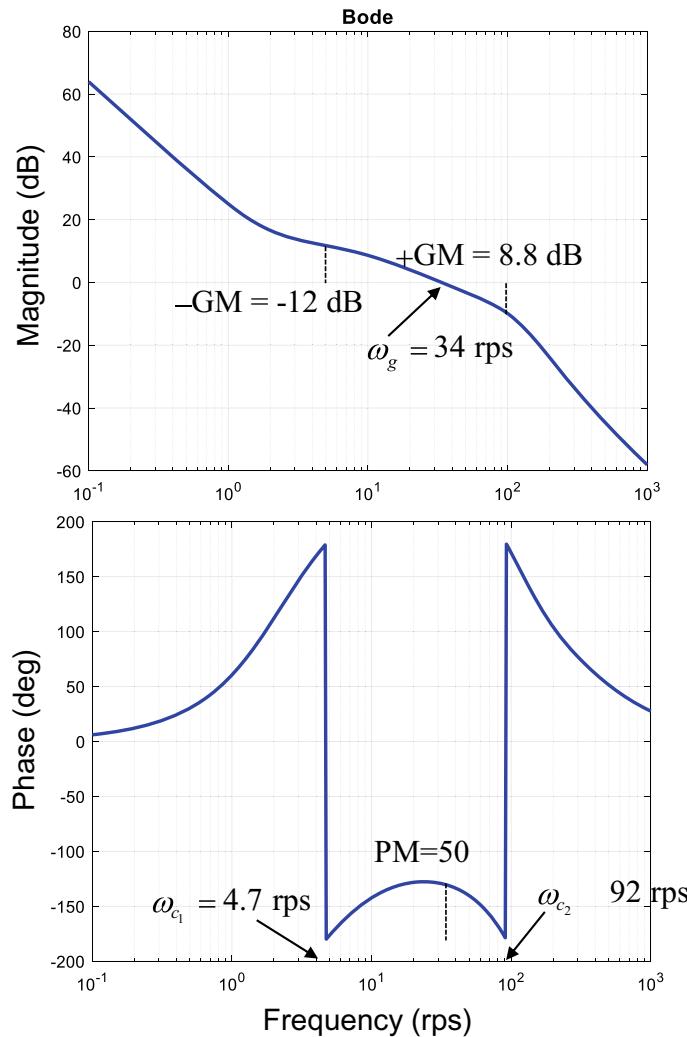


Fig. 3.21 Bode plot at the plant-input loop breakpoint

$$\begin{aligned} GM_T &= [1 - \sigma_T, 1 + \sigma_T] \text{ dB} \\ &= [0.2695 \ 1.7305] \\ &= [-11.4 \ 4.7] \text{ dB} \end{aligned} \quad \begin{aligned} PM_T &= \pm 2 \sin^{-1} \frac{\sigma_T}{2} \text{ degree} \\ &= \pm 42.84 \text{ degree} \end{aligned} \quad (3.84)$$

With the final result from (3.75) computed as:

$$GM = [-11.4 \ 7.28] \text{ dB}; \ PM = \pm 42.84 \text{ degree} \quad (3.85)$$

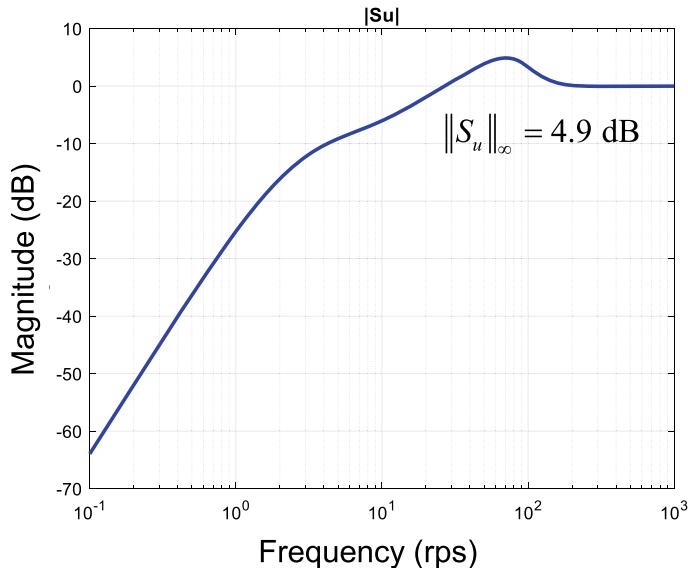


Fig. 3.22 $\bar{\sigma}(S_u)$ versus frequency at the plant-input loop breakpoint

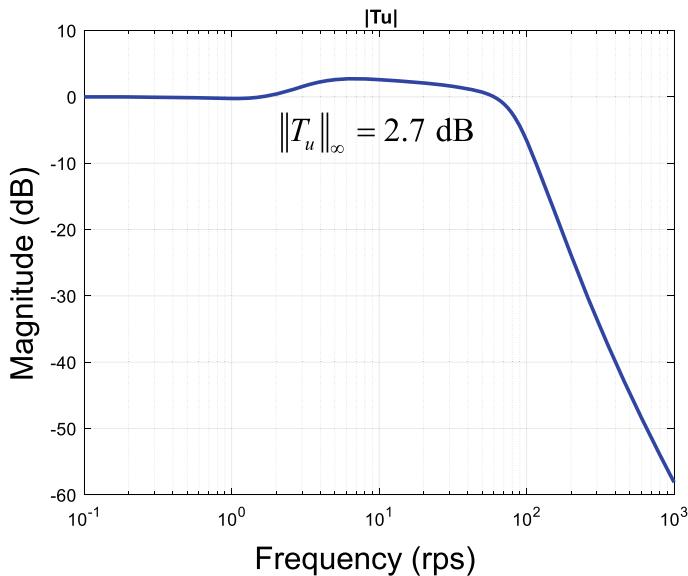


Fig. 3.23 $\bar{\sigma}(T_u)$ versus frequency at the plant-input loop breakpoint

Note that the classical margins from Figs. 3.20 and 3.21 are larger. The singular value stability margins are always more conservative than the single-loop classical margins.

Figures 3.24 and 3.25 show the sensitivity S_y and the complementary sensitivity T_y output frequency response curves formed at the output loop breakpoint for the acceleration signal, (the (1,1) element of S_y and T_y matrices). The sensitivity is the inverse of the return difference at the plant output. The infinity norm $\|S_y\|_\infty$ is equivalent to the minimum of $\underline{\sigma}(I + L_y)$ (they are inversely related). From Fig. 3.24 $\|S\|_\infty = 1.0257$ or 0.2 dB. This is a very small peak indicating good margins at the plant output. Figure 3.25 shows the complementary sensitivity $|T|$ which is the acceleration closed-loop transfer function. The $\|T\|_\infty = 1.07$ (0.6 dB) is a measure of the peak resonance in the acceleration loop. This is a small value also indicating good margins in this loop. If either $\|S\|_\infty$ or $\|T\|_\infty$ were large, this could indicate a potential sensitivity problem in the overall design. In some multivariable systems, the margins at the plant input will be adequate but at the plant output they are low. It is always prudent to check margins at all loop breakpoints to make sure no sensitivity problems exist.

Figure 3.26 shows Nyquist plots computed at the plant output for the acceleration and pitch rate feedback loops. For the acceleration loop analysis, the pitch rate loop is closed. For the pitch rate loop analysis, the acceleration loop is closed. Both plots show excellent stability margins. This directly relates to the excellent values of $\|S\|_\infty$ and $\|T\|_\infty$ in Figs. 3.24 and 3.25.

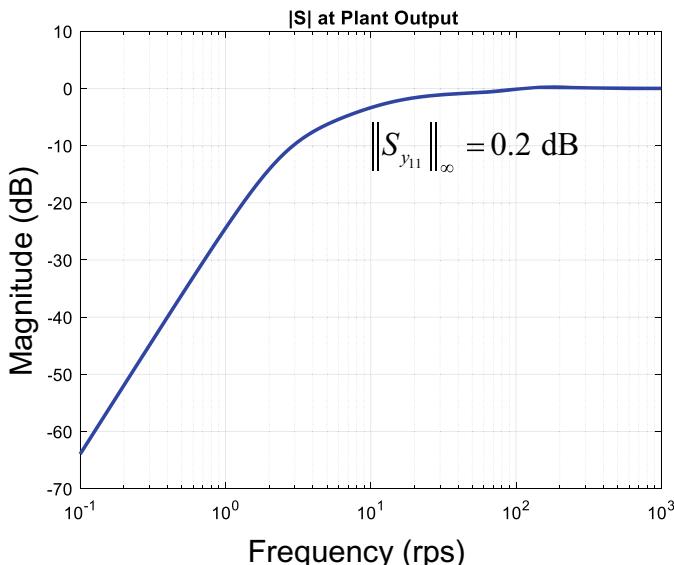


Fig. 3.24 $|S_{y11}|$ versus frequency at the plant-output loop breakpoint for this acceleration command system

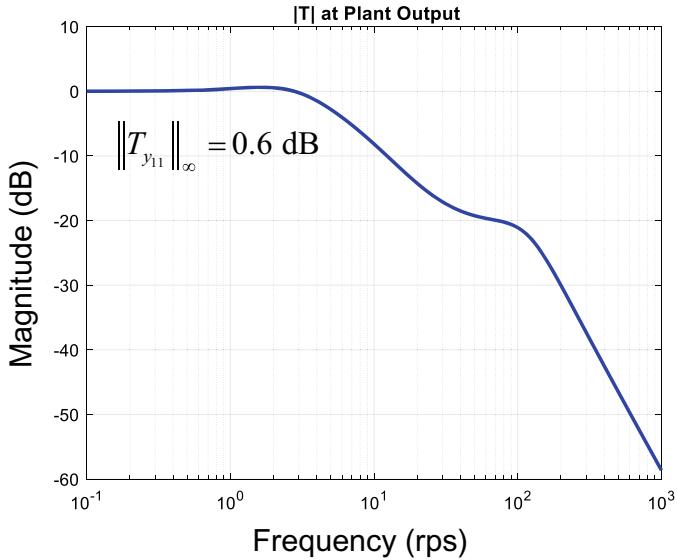


Fig. 3.25 $|T_{y_{11}}|$ versus frequency at the plant-output loop breakpoint for this acceleration command system

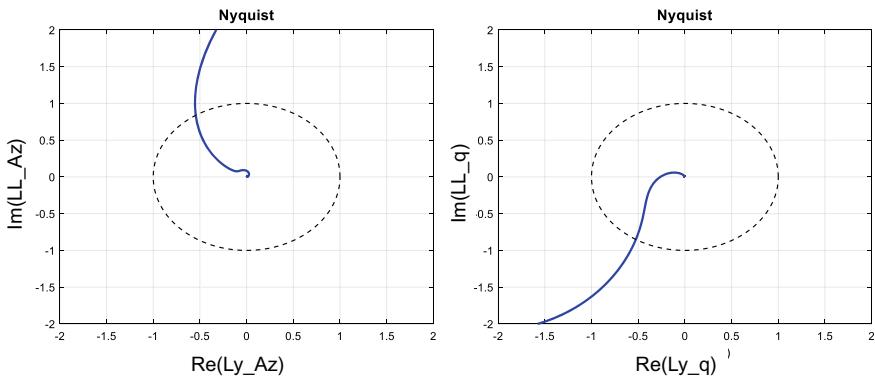


Fig. 3.26 Nyquist plots for the acceleration and pitch rate loops at the plant output

Figure 3.27 shows the noise-to-control and noise-to-control rate frequency responses. This figure indicates the amplification, or attenuation, of sensor noise through the controller/plant. Although not directly related to stability margins, this frequency response should be examined to make sure the bandwidth of the controller is not too high and that high-frequency noise is not adversely amplified. The controller in this example is a simple PI control. The noise-to-control plot for the accelerometer shows that the acceleration noise is greatly attenuated at

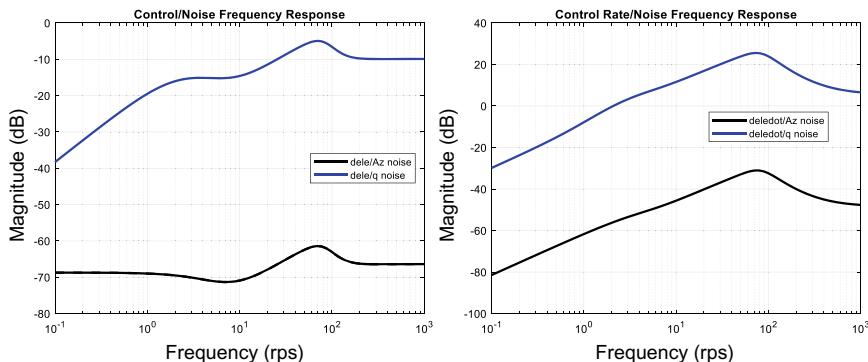


Fig. 3.27 u/v and \dot{u}/v magnitude versus frequency

all frequencies. No additional filtering would be needed. The noise-to-control rate for the accelerometer also shows minimal activity due accelerometer noise. The noise-to-control and noise-to-control-rate curves show a different result. The gyro noise (for pitch rate) response has a peak at -6 dB at 60 rps. This indicates an attenuation gain of $1/2$ (-6 dB) for the gyro noise. The noise-to-control-rate curve shows significant rate activity in the control signal due to gyro noise. This result suggests additional filtering might be required.

When signal noise amplification is a problem in the system analysis, additional filtering using low-pass filters and/or higher-order filters would be needed. These additional filters can be problematic as the associated phase lag causes issues for systems with low stability margins. Each integrator in the filter (each state) adds an additional phase lag and can make the stabilization of unstable systems very difficult. This may cause the need to detune the control gains to provide a lower loop gain-crossover frequency with improved phase margins. It may take several iterations to achieve a stable design, balancing stability margins with noise attenuation.

In the next section we will explore the use of the structured singular value (SSV) μ in computing the robustness of control systems. The SSV is a very powerful analysis tool used to evaluate stability robustness of systems with respect to a large variety of uncertainties and disturbances.

3.4 Control System Robustness Analysis

In this section we present modeling techniques and analysis methods for control system robustness analysis. Developing methods for analyzing control system sensitivity to uncertainties in dynamics has been a major focus for many years. In the past, the most widely used measure of stability robustness has been single-loop gain and phase margins derived from classical frequency response calculations.

These methods and their singular value counterparts were discussed in the previous section.

Gain and phase margins provide significant insight into the robustness characteristics of the system. It has been proven many times in real-world applications that systems that have poor stability margins do not perform as desired. Requirements are typically levied onto the control design to provide 6 dB gain margin and at least 45° phase margin. Whether these margins be classical or singular value based, having adequate gain and phase margins is an important aspect of control system design.

In the 1980s, significant research was performed on analyzing the robustness of control systems to neglected and mismodeled dynamics and real parameters uncertainties. Analysis methods were developed to further analyze linear models to gain more insight into controllers' sensitivity to unmodeled dynamics, gain and phase uncertainties at different loop breakpoints, and the sensitivity to parameter variations within the model. All of these methods aim to determine bounds on how large the uncertainties can be before the system would go unstable.

Many methods exist for solving the control parameters sensitivity problem, all having differing amounts of conservatism in computing robustness bounds. This conservatism exists due to the model of the uncertainties and how the uncertainties enter into the problem structure. Over the years, polynomial methods, singular-value-based methods, and other frequency domain techniques were developed.

Table 3.2 illustrates several methods applied to a pitch autopilot analysis problem [3] investigating how much uncertainty can be tolerated in the aerodynamic parameters before the system goes unstable. The results in [3] use the same system model presented in Example 3.2. In this problem, each of the aerodynamic stability derivatives (Z_α/V , Z_δ/V , M_α , M_δ) were allowed to vary simultaneously using

$$\left(\frac{Z_\alpha}{V} (1 \pm \delta_1), \frac{Z_\delta}{V} (1 \pm \delta_2), M_\alpha (1 \pm \delta_3), M_\delta (1 \pm \delta_4) \right) \quad (3.86)$$

Table 3.2 Comparison of robustness analysis methods analyzing sensitivity to real parameter variations

Robustness theory	% Perturbation
Small gain theorem	13.8%
SSV_μ	49.0%
Stability hypersphere $Xp = \delta$	0.1%
Stability hypersphere $p = A\alpha + b$	20.3%
Stability hypersphere (Lyapunov unscaled)	0.007%
Stability hypersphere (Lyapunov scaled)	0.02%
Karitonov's theorem	13.3%
De Gaston–Safonov real margin	60.44%
Monte Carlo eigenanalysis	60–61%

It was desired to compute the smallest variation in these parameters that would cause the system to become unstable.

Table 3.2 shows various algorithms and modeling techniques applied to this analysis problem. The exact answer is 60.44%. This answer was derived and verified by a Monte Carlo analysis, as well as by inserting the predicted uncertainties back into the system model to show that the closed-loop system indeed had poles on the $j\omega$ -axis. As shown in the figure, some of the methods were found to be quite conservative. The small gain theorem, the structured singular value (SSV), and the De Gaston and Safanov [4] real stability margin all produced reasonable results when applied to this aerospace problem.

3.4.1 Analysis Models for Uncertain Systems

Stability analysis models for multivariable systems can be formed to analyze gain and phase uncertainties, neglected and/or mismodeled dynamics, real parameter uncertainties, and combinations thereof using methods identical to forming models for single-input-single-output systems. These models can be easily formed using block diagram algebra, signal flow graph methods, or algebraic manipulation of loop equations. The resulting models will have a “structure” associated with them depending upon the specific problem, and the analysis will depend upon the structure. Within these models the uncertainties in the system are usually isolated from the system models of the dynamics.

Figure 3.28 shows a block diagram similar to Fig. 3.16 where gain and phase uncertainties are modeled using the matrix $\Delta \in \mathbb{C}^{n_u \times n_u}$. This block diagram model using $\Delta = \text{diag}[k_i e^{j\phi_i}]$ was used in the previous section to derive singular value-based stability margins. The loop transfer function at plant input is $L_u = KG\Delta$. Figure 3.29 shows two alternate models that can be used to replace the Δ -block model using an additive and feedback-type block diagram structure. When using the additive model in Fig. 3.29, the loop transfer function at plant input L_u is

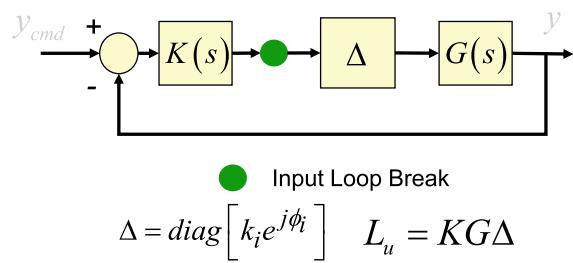
$$L_u = KG(I + \Delta_u) \quad (3.87)$$

and when the system is nominal (no uncertainty), this model uses $\Delta_u = 0_{n_u \times n_u}$. When using the feedback model in Fig. 3.29, the loop transfer function at plant input L_u is

$$L_u = KG(I + \Delta_f)^{-1} \quad (3.88)$$

and when the system is nominal (no uncertainty), this model uses $\Delta_f = 0_{n_u \times n_u}$. These models give alternate means of introducing uncertainties, and when analyzed can provide varying amounts of conservatism. Figure 3.30 shows a block diagram with plant-input, Δ_1 , plant-additive, Δ_b , and plant-output, Δ_2 , uncertainty models have been introduced. At the plant input this model can be used to model control law digital implementation effects (digital-to-analog), time delays,

Fig. 3.28 MIMO system with plant-input uncertainty



and actuation subsystem gain and phase uncertainties. We can see from (3.87) that this uncertainty model Δ_1 is dimensionless. This is important as it can impact singular value analyses.

The additive uncertainty model Δ_b in Fig. 3.30 can be used to model uncertainties in the plant model G , and/or neglected and/or uncertain flexible body dynamics. The elements of this matrix have units of the plant model from input to

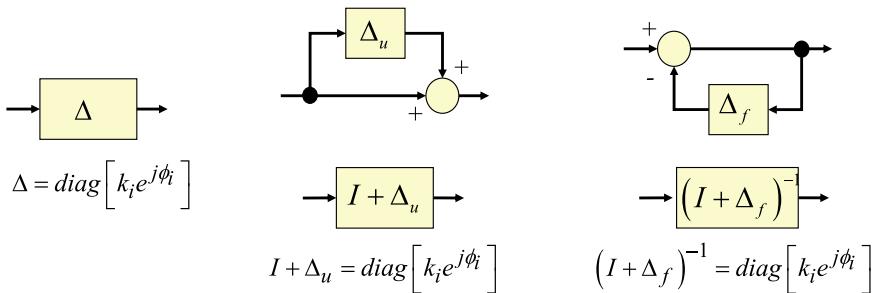


Fig. 3.29 Additive and feedback uncertainty models

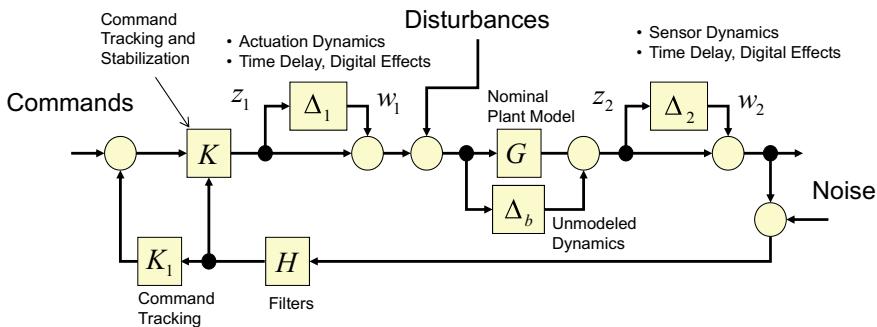


Fig. 3.30 MIMO system with input, plant, and output uncertainty models

sensor. Uncertainties in the plant model G can arise from uncertainties in the aerodynamic model, or from uncertainties in air data system parameters, i.e., errors in measured or estimated Mach number, angle of attack or sideslip, air temperature, dynamic pressure. This additive model also can capture neglected and/or uncertain flexible body dynamics. Figure 3.31 shows an actuator installation model. The elevon drive rib has a center of mass that lies off its center of rotation, and when rotated by the actuator, the angular acceleration of this mass creates forces that are reacted back into structure exciting flexible body modes. The gyros and accelerometers sense the flexible body motion and if the feedback control system amplifies this motion instability can result.

The plant-output uncertainty Δ_2 can also be used to model control law digital implementation effects at the plant-output (analog-to-digital), time delays, and sensor subsystem gain and phase uncertainties. This model is also dimensionless.

Example 3.3 Actuator Uncertainty Model In the design of a control system the actuator (assumed to be modeled using a second-order transfer function) was neglected in the design of the control law. Consider using the plant-input additive uncertainty model shown in Fig. 3.30 to model the neglected second-order actuator. The actuator model is

$$\frac{\delta}{\delta_c} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.89)$$

Equating this second-order transfer function to the additive uncertainty model at the plant input we have

$$I + \Delta_1 = \frac{\delta}{\delta_c} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.90)$$

Solving for Δ_1 yields

$$\Delta_1 = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} - 1 = \frac{-s^2 - 2\xi\omega_n s}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.91)$$

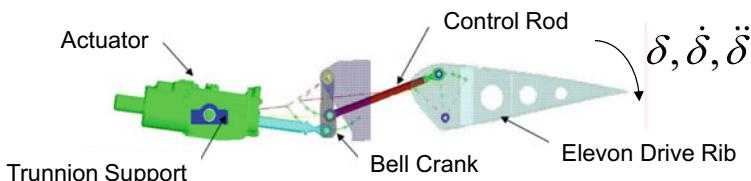


Fig. 3.31 Actuator installation model

We will show how to use this model in the next section to the control system is robust to the neglected actuator model.

Figure 3.7 illustrated a general ΔM control system analysis model in which the matrix $\Delta(s)$ models uncertainties and $M(s)$ is a transfer function matrix modeling the dynamics between the output from the uncertainties to its input. We will use this ΔM representation of the dynamics for many of our stability analysis problems. The matrix $\Delta(s)$ will model gain and phase uncertainties in the system, i.e., neglected and/or mismodeled dynamics, real parameter uncertainties, or any combinations thereof. The matrix $M(s)$ will model the dynamics in the system that are assumed to be known.

For a control system under no uncertainty, the controller stabilizes the plant, and the return difference matrix is non-singular at all frequencies. Stability of the nominal system implies

$$\det[I + L(s)] \neq 0 \quad \forall s \in D_R \quad (3.92)$$

Using the ΔM analysis model shown in Fig. 3.7, (3.92) is equivalent to

$$\det[I - \Delta M(s)] \neq 0 \quad \forall s \in D_R \quad (3.93)$$

Under no uncertainty, $\Delta = 0$, this condition is clearly satisfied. The analysis question is to determine how large can Δ be while the system remains stable.

Example 3.4 Consider the ΔM analysis model shown in Fig. 3.7, and a stability analysis problem for a system as depicted in Fig. 3.32. The control system block diagram in Fig. 3.32 shows uncertainties Δ_1 at the input to the plant and uncertainties Δ_2 at the output of the plant. The uncertainties Δ_1 and Δ_2 can be constructed to model any type of uncertainty, depending upon the analysis question at hand. Δ_1 could model actuator uncertainties, unmodeled dynamics, time delays, or any plant-input uncertainty. While Δ_2 could model sensor uncertainties, unmodeled dynamics, time delays, or any plant-output uncertainty. Block diagram algebra is used to transform the system shown in Fig. 3.32 into the ΔM analysis model. The matrix $\Delta(s)$ will be a block diagonal matrix, given as

$$\Delta = \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix} \quad (3.94)$$

with each matrix or scalar uncertainty in the system located on the diagonal of $\Delta(s)$. The matrix $M(s)$ is a block matrix where the ij th block is the transfer function matrix between the output of the j th uncertainty $\Delta_j(s)$ to the input of the i th uncertainty $\Delta_i(s)$.

Consider the loop equations from Fig. 3.32 written as

$$z_1 = K(z_2 + w_2), \quad z_2 = G(z_1 + w_1)$$

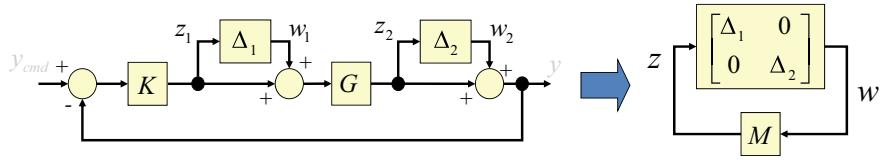


Fig. 3.32 Control system with simultaneous uncertainty at input and output

Substituting the z_2 expression into the z_1 equation and manipulating yields

$$\begin{aligned} z_1 &= -K(G(z_1 + w_1) + w_2) \\ z_1 &= -KG(z_1 + w_1) - Kw_2 \\ (I + KG)z_1 &= KGw_1 + Kw_2 \\ z_1 &= -(I + KG)^{-1}KGw_1 - (I + KG)^{-1}Kw_2 \end{aligned}$$

Substituting the z_1 expression into the z_2 equation and manipulating yields

$$\begin{aligned} z_2 &= G(-K(z_2 + w_2) + w_1) \\ z_2 &= -GK(z_2 + w_2) + Gw_1 \\ (I + GK)z_2 &= GKw_2 + Gw_1 \\ z_2 &= (I + GK)^{-1}Gw_1 - (I + GK)^{-1}GKw_2 \end{aligned}$$

Combining these two expressions and writing in matrix form yields

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -(I - KG)^{-1}KG & -(I - KG)^{-1}K \\ (I - GK)^{-1}G & -(I - GK)^{-1}GK \end{bmatrix}}_{M(s)} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (3.95)$$

We see from (3.95) that the diagonal elements of the M matrix are the negatives of the closed-loop transfer function formed at the plant input and plant output, respectively.

The loop equations for the uncertainty matrix Δ modeled in the system can be written as

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix}}_{\Delta(s)} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad (3.96)$$

From this example we see that the Δ matrix in (3.96) is block diagonal and has structure. This structure is important when performing stability analysis. If Δ is a block diagonal matrix as in (3.96), then the variables z_1 and z_2 affect the system stability at the input and output independently. If Δ is a full matrix, then

the cross-elements in the matrix (zero in (3.96)) couple at both the plant input and plant output in affecting the stability of the system.

Real Parameter Uncertainties

The above example essentially uses block diagram algebra to form the analysis model isolating the uncertainties from the nominal closed-loop system. When considering uncertainties in the real parameters contained in the state-space matrix A , working through the algebra can be tedious and error prone. This problem is easily solved for system linear in the uncertain parameters using a method [5, 6] that factors out the uncertain parameters in the closed-loop matrix A_{cl} , and decomposes the matrices using a singular value decomposition to form a state-space model for $M(s)$, $M(s) = C_M(sI_n - A_M)^{-1}B_M$. The n uncertain parameters in A_{cl} are modeled as

$$p_i = \bar{p}_i(1 \pm \delta_i) \quad (3.97)$$

where p_i is the uncertain parameter with \bar{p}_i its nominal value, and δ_i the dimensionless uncertainty. The closed-loop system is written isolating the uncertainties as follows:

$$A_{\text{cl}} = A_0 + \sum_{i=1}^n E_i \delta_i \quad (3.98)$$

where A_0 is the nominal closed-loop matrix (stable matrix) and the coefficient matrices E_i factor out the uncertainties. The matrices E_i are the structural definitions for each of the parameter perturbations δ_i and show how they enter into the system model, with the rank of the matrix used to describe the parameter uncertainty. Using this model, we decompose each $n \times n$ matrix E_i using a singular value decomposition. This gives

$$E_i = U \Sigma V^* \quad (3.99)$$

The matrix Σ will have k nonzero singular values, where k is equal to the rank of the matrix, with the remaining $n - k$ singular values equal to zero. Discard the zero singular values and make Σ a $k \times k$ diagonal matrix containing only the nonzero singular values. We can write (3.99) with this new Σ as

$$E_i = b_i c_i \quad (3.100)$$

where $b_i = U \Sigma^{1/2}$ and $c_i = \Sigma^{1/2} V^*$. The matrices b_i and c_i in (3.100) depend only on the magnitude of the i th nominal parameter. By using the decomposition described in (3.100), we can replace $E_i \delta_i$ in (3.98) with $b_i c_i \delta_i$. By using this modeling approach, we can separate out the parameter variations δ_i , form $\Delta = \text{diag}[\delta_i]$ and create the nominally stable $M(s) = C_M(sI - A_M)^{-1}B_M$. The state-space triple (A_M, B_M, C_M) for $M(s)$ is formed as follows.

Consider rank 1 perturbation only. b_i is $n \times 1$ and c_i is $1 \times n$. Then, (3.98) is

$$A_{\text{cl}} = A_0 + \sum_{i=1}^n b_i c_i \delta_i \quad (3.101)$$

with $-1 < \delta_i < 1$. Write the closed-loop system as

$$\dot{x} = A_0 x + \sum_{i=1}^n b_i u_i \quad (3.102)$$

where u_i are input variables. Let the output y for this system be defined as

$$y = \sum_{i=1}^n c_i x = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} x \quad (3.103)$$

Then, $u_i = \delta_i y_i$ and we can close the loop with

$$u_i = \delta_i y_i = \delta_i c_i x \quad (3.104)$$

Substituting (3.104) into (3.102) yields

$$\begin{aligned} \dot{x} &= A_0 x + \sum_{i=1}^n b_i \delta_i c_i x \\ &= \left(A_0 + \sum_{i=1}^n b_i \delta_i c_i \right) x = A_{\text{cl}} x \end{aligned} \quad (3.105)$$

which is the closed-loop system model. We can write a state-space triple (A_M, B_M, C_M) for this system as

$$A_M = A_0; \quad B_M = [b_1 \dots b_n]; \quad C_M = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (3.106)$$

This triple describes the M matrix in the ΔM analysis model (Fig. 3.7).

Example 3.5 Consider the longitudinal dynamics from Example 3.3. An angle-of-attack controller is designed, and we desire to form an analysis model to analyze the uncertain aerodynamic parameters using the above method. The α control law is designed using the robust servomechanism approach from this chapter. This

analysis will determine how sensitive the LQR control law is to knowing the aerodynamic stability derivatives used in the design model. The longitudinal dynamics with second-order actuator model are

$$\begin{aligned}\dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta_e + q \\ \dot{q} &= M_\alpha\alpha + M_\delta\delta_e \\ \ddot{\delta}_e &= -2\zeta\omega_n\delta_e - \omega_n^2(\delta_e - \delta_c)\end{aligned}\quad (3.107)$$

The robust servomechanism model $\dot{z} = \tilde{A}z + \tilde{B}\mu$ from this chapter is

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{Z_\alpha}{V} & 1 & \frac{Z_\delta}{V} & 0 \\ 0 & M_\alpha & 0 & M_\delta & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_a^2 & -2\zeta_a\omega_a \end{bmatrix} = \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 \\ 0 & -1.3046 & 1.0 & -0.21420 & 0 \\ 0 & 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0 & -4624.0 & -81.6 \end{bmatrix} \quad (3.108)$$

$$\tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \omega_n^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 4624.0 \end{bmatrix} \quad (3.109)$$

The LQR penalty matrices used to design the controller are:

$$Q = \text{diag}[464.16 \ 0 \ 0 \ 0 \ 0]; \quad R = 1 \quad (3.110)$$

with the resulting state feedback gain matrix given as

$$K_c = [-21.544 \ -3.8421 \ -0.29392 \ 0.32045 \ 0.0021463] \quad (3.111)$$

The nominal closed-loop system matrix is

$$A_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{Z_\alpha}{V} & 1 & \frac{Z_\delta}{V} & 0 \\ 0 & M_\alpha & 0 & M_\delta & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\omega_a^2 k_1 & -\omega_a^2 k_2 & -\omega_a^2 k_3 & -\omega_a^2(1 + \omega_a^2 k_4) & -2\zeta_a\omega_a - \omega_a^2 k_5 \end{bmatrix}; \quad (3.112)$$

Consider real parameter uncertainties in the four aerodynamic coefficients ($Z_\alpha/V, Z_\delta/V, M_\alpha, M_\delta$) in (3.112). The parameter uncertainty model is $p_i =$

$\bar{p}_i(1 \pm \delta_i)$. Using the closed-loop system uncertainty model from (3.98), we factor out each δ_i and form the matrices E_i :

$$E_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{Z_\alpha}{V} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad E_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{Z_\delta}{V} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$E_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & M_\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad E_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & M_\delta & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad (3.113)$$

The singular value decomposition for E_1 is used to form the first column in B_M and first row in C_M . Substituting the numerical values into (3.113) we have

$$E_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1.3046 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = U\Sigma V = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} [1.3046] \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} 0 \\ 1.1422 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{b_1} \underbrace{\begin{bmatrix} 0 & -1.1422 & 0 & 0 & 0 \end{bmatrix}}_{c_1} \quad (3.114)$$

Using this same approach for each E_i , the columns and rows of the matrices B_M and C_M are populated. The state-space triple (A_M, B_M, C_M) is then

$$\begin{aligned}
 A_M &= \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 \\ 0 & -1.3046 & 1.0 & -0.2142 & 0 \\ 0 & 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 275,100 & 49,059 & 3753 & -16,861 & -163 \end{bmatrix} \\
 B_M &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.1422 & 0.4628 & 0 & 0 \\ 0 & 0 & -6.9073 & -10.2389 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \\
 C_M &= \begin{bmatrix} 0 & -1.1422 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.4628 & 0 \\ 0 & -6.9073 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10.2389 & 0 \end{bmatrix} \quad (3.115)
 \end{aligned}$$

Now, using the parameter uncertainty models presented in this section we will investigate analysis methods to determine the robust stability.

3.4.2 Singular Value Robustness Tests

A very quick and useful analysis method for analyzing stability robustness is to apply the small gain theorem. This method is accurate when the uncertainty modeling matrix $\Delta(s)$ is a full complex-valued matrix. That is, when the matrix has no structure and is complex. When the matrix has structure, as in (3.96), the small gain theorem can be quite conservative. The more the structure deviates from a full complex-valued matrix, the more conservative is the result.

The structured singular value (SSV), denoted as μ , was developed to reduce the conservatism of evaluating stability robustness for problems like those in (3.96) that have structure. By structuring the uncertainty model into a block diagonal matrix form, and applying the SSV μ -test, a less conservative estimate of stability robustness is obtained.

Stability under the presence of uncertainty, assuming the nominal system is stable, requires the return difference matrix to become singular under the uncertainty. The stability test is described in (3.93). The following singular value robustness tests are designed to examine the return difference matrix and determine when it becomes singular.

3.4.2.1 The Small Gain Theorem

Consider the stability robustness analysis problem for the ΔM analysis model shown in Fig. 3.7. The return difference matrix for this system is $I + L = I - \Delta M$. The analysis problem is to determine the “size” of the matrix $\Delta(s)$ such that the system transitions from stable to unstable. This says that the return difference matrix transitions from non-singular to singular under the uncertainty.

The matrix $\Delta(s)$ can be a block diagonal (BD) matrix, with each matrix entry $\Delta_i(s)$ on the diagonal corresponding to a matrix of perturbations occurring in the system. The matrix $(-M(s))$ is the transfer function between the output of the perturbation to its input. It depends upon the controller $K(s)$, the plant model $G(s)$, and the structure of the perturbations. Matrix $M(s)$ is a block matrix where the (i, j) th block $M_{i,j}(s)$ is the negative of the transfer function from the output of $\Delta_i(s)$ to the input of $\Delta_j(s)$.

We can intuitively define the bound on the norm of $\Delta(s)$ by using the $(A + B)$ argument of the preceding section. Specifically, we have a sufficient condition for closed-loop stability of the perturbed system.

$$\underline{\sigma}[I] > \bar{\sigma}[\Delta M] \Rightarrow \det[I - \Delta M(s)] \neq 0 \quad (3.116)$$

Using $\bar{\sigma}[\Delta M] \leq \bar{\sigma}[\Delta]\bar{\sigma}[M]$, and the fact that $\underline{\sigma}[I] = 1$, we can derive a more conservative bound on the system uncertainty

$$\bar{\sigma}[\Delta] < 1/\bar{\sigma}[M] \quad (3.117)$$

which is referred to as the small gain theorem (SGT). This is a *sufficient* test for stability. If it is violated, the system may still be closed-loop stable. The conservatism is introduced in the step where $\bar{\sigma}[\Delta M]$ is bounded above by the product $\bar{\sigma}[\Delta]\bar{\sigma}[M]$. This step loses all structural information inherent in the matrices. It models a worst-case scenario in which $\Delta(s)$ is a full complex-valued matrix.

3.4.2.2 The Structured Singular Value μ

The structured singular value (SSV) μ analysis was developed by Doyle [7] to reduce the conservatism of evaluating stability robustness using unstructured singular value computations SGT. Consider the control system with input and output uncertainties as shown in Fig. 3.32. Stability for the perturbed closed-loop system is guaranteed only when the return difference dynamics remains non-singular, that is

$$\det[I - K(I - \Delta_2)G(I - \Delta_1)] \neq 0 \quad \forall \Delta_1, \Delta_2, \text{ and } s \in D_R \quad (3.118)$$

which is equivalent to,

$$\det[I - \Delta M] \neq 0 \quad \forall \Delta = \text{diag}[\Delta_1, \Delta_2], \text{ and } s \in D_R \quad (3.119)$$

The definition of the SSV μ is

$$\mu_\Delta(M) = \begin{cases} \frac{1}{\min\{\bar{\sigma}(\Delta) : \Delta \in \Delta, \det[I - \Delta M] = 0\}} & M \in \mathbb{C}^{n \times n} \\ 0, & \text{if no } \Delta \in \Delta \text{ makes } I - \Delta M \text{ singular} \end{cases} \quad (3.120)$$

The computation of the SSV μ in (3.120) utilizes the structure for $\Delta(s)$ to develop a less conservative answer to the bound on the destabilizing uncertainty.

Consider the simplest structure for the uncertainty Δ , that is a diagonal matrix whose diagonal is a complex scalar, i.e.,

$$\Delta = \{\delta I_n : \delta \in C\} \quad (3.121)$$

Substitute this Δ model into (3.119). Assuming the uncertainty destabilizes the system, the return difference matrix is singular and can be written as

$$(I - \Delta M)w = (I - \delta I_n M)w = \delta \left(\frac{1}{\delta} I - M \right)w = 0 \quad (3.122)$$

for arbitrary vector w . This simplest structure defines an eigenvalue problem, with the SSV μ from (3.120) given as

$$\mu_\Delta(M) = \bar{\rho}(M) \quad (3.123)$$

where $\bar{\rho}(M)$ is the maximum spectral radius of the matrix M . When the uncertainty is a full complex matrix, as shown in the previous section, the small gain theorem produces an accurate bound on the uncertainty, with the SSV μ given as

$$\mu_\Delta(M) = \bar{\sigma}(M) \quad (3.124)$$

So, for problems of arbitrary structure, that is for a block diagonal Δ , the SSV μ will be bounded above and below by

$$\bar{\rho}(M) \leq \mu_\Delta(M) \leq \bar{\sigma}(M) \quad (3.125)$$

Commercial software is available for computing the SSV μ in MATLAB®. This software bounds the SSV μ through optimization given by

$$\max_Q |\lambda_{\max}(QM)| < \mu < \inf_D \bar{\sigma}(DMD^{-1}) \quad (3.126)$$

In analysis problems where real uncertainties are analyzed, or problems where real and complex (dynamic) uncertainties are analyzed together, there can exist a spread between the bounds in (3.126). This introduces some conservatism in bounding the uncertainty in some problems. A method discussed in the next section exactly computes the bound on Δ when Δ contains only real parameters.

Example 3.6 The SSV μ is a very powerful analysis tool used by researchers, scientists, and engineers. In this example we will demonstrate its use as an analysis tool to understand the robustness of a MIMO flight control system, in which uncertainty is modeled at the plant input and output. When tools like the small gain theorem are used to predict the robustness to uncertainty, conservative robustness bounds are produced. By using the SSV μ , more accurate predictions of robust stability can be obtained. In this example we will model the uncertainties varying the *structure* to illustrate and contrast the conservatism when using these tools.

Consider the following lateral-directional flight control system. It is desired to track a stability axis roll-rate command p_{sc} while keeping sideslip angle β small. In this example we will compare two controllers and their associated robustness properties.

The state-space model $\dot{x} = Ax + Bu$ for the lateral-directional dynamics is

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} Y_\beta & s(\alpha) & -c(\alpha) \\ L_\beta & 0 & 0 \\ N_\beta & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (3.127)$$

where $s(\alpha) = \sin(\alpha)$, $c(\alpha) = \cos(\alpha)$, the state $x = [\beta \ p \ r]^T$ contains the sideslip angle, roll rate, and yaw rate, and the control $u = [\delta_a \ \delta_r]^T$ contains the aileron and rudder commands. The system and control distribution matrices are

$$A = \begin{bmatrix} -0.0251 & 0.10453 & -0.99452 \\ 574.7 & 0 & 0 \\ 16.2 & 0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0.1228 & -0.2763 \\ 195.5 & -529.4 \\ -53.61 & 33.25 \end{bmatrix} \quad (3.128)$$

We will use the robust servomechanism infinite-time LQR control from Chap. 4 to design the controllers. The first controller uses a single integrator to track stability axis roll-rate commands. The LQR design model is

$$\dot{z} = \begin{bmatrix} 0 & C_c \\ 0 & A \end{bmatrix} z + \begin{bmatrix} 0 \\ B \end{bmatrix} u \quad (3.129)$$

where $z = [\int e_{p_s} \ \beta \ p \ r]^T$ and $C_c = [0 \ c(\alpha) \ s(\alpha) \ 0]$. The LQR penalty matrices are

$$Q = \begin{bmatrix} 1.7 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.130)$$

where the (1,1) element of Q penalizes the error in tracking the stability axis roll-rate command, and the (2,2) element of Q penalizes the sideslip angle. Solving the Algebraic Riccati Equation the resulting state feedback gain matrix is

$$K_c = \begin{bmatrix} -0.7852 & -2.0536 & -0.0797 & 0.0458 \\ 1.0409 & 3.8238 & 0.1280 & -0.1020 \end{bmatrix} \quad (3.131)$$

The closed-loop system matrix, $A_{cl} = \tilde{A} - \tilde{B}K_c$ is

$$A_{cl} = \begin{bmatrix} 0 & 0 & 0.99452 & 0.10453 \\ 0.38402 & 1.2836 & 0.14968 & -1.0283 \\ -76.704 & 337.46 & -8.5278 & 5.8489 \\ 704.55 & 2442.0 & 83.351 & -62.971 \end{bmatrix} \quad (3.132)$$

with eigenvalues $\lambda_{1,2} = -12.1814 \pm 22.1215j$ $\lambda_{3,4} = -22.9261 \pm 11.8584j$. Figure 3.33 shows a step response commanding a stability axis roll rate $p_s = p \cos(\alpha) + r \sin(\alpha)$, as well as the response of the sideslip β , roll rate p and yaw rate r . Figure 3.34 shows the frequency response analysis at the plant input where $\bar{\sigma}(L_u)$, $\bar{\sigma}(S_u)$, and $\bar{\sigma}(T_u)$ are plotted versus frequency, and the loop gain-crossover frequency ω_c and singular-value stability margins are computed using (3.75). As discussed Chap. 4, the LQR state feedback design has excellent stability margins at the plant input. The peak in $\bar{\sigma}(S_u)$ has a minimum value of 1 and $\bar{\sigma}(T_u) = 0.5815$. A typical requirement is to keep this minimum above 0.5.

We desire to investigate robustness of this control system to simultaneous uncertainties at the plant input and plant output, and demonstrate how the structure of the uncertainties impacts the analysis. We will use the ΔM analysis model for the

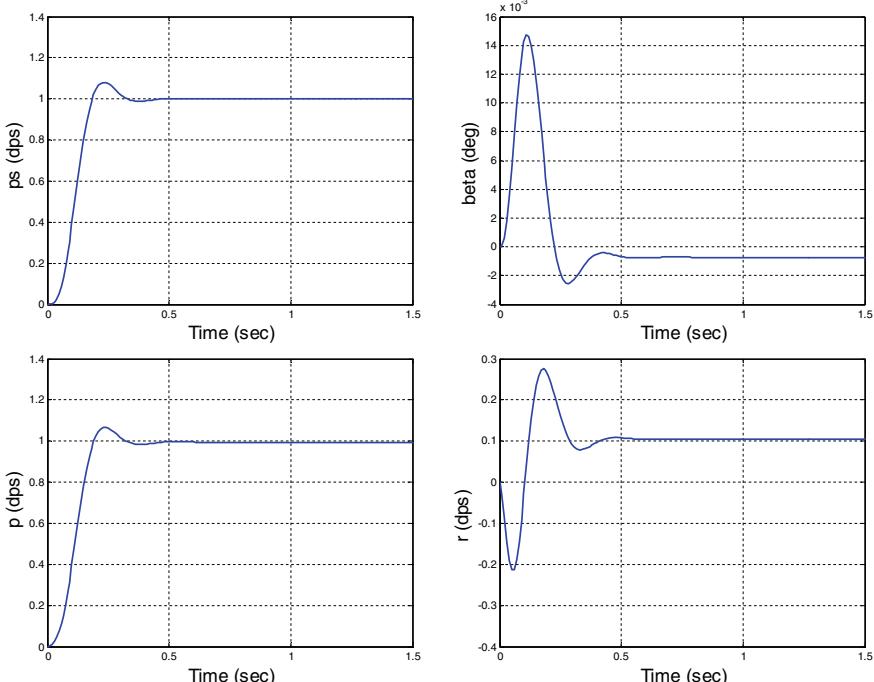


Fig. 3.33 Example 3.6 stability axis roll-rate step response time histories

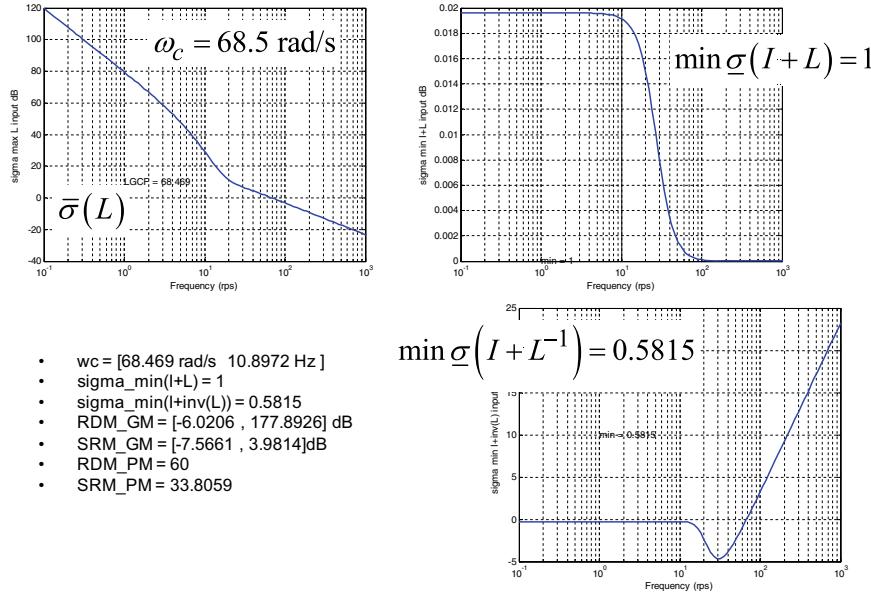


Fig. 3.34 Example 3.6 plant-input frequency response analysis and stability margins

block diagram shown in Fig. 3.7. The plant model is

$$\begin{aligned}\dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u\end{aligned}\quad (3.133)$$

where

$$\begin{bmatrix} A_p & B_p \\ C_p & D_p \end{bmatrix} = \begin{bmatrix} -0.0251 & 0.10453 & -0.99452 \\ 574.7 & 0 & 0 \\ 16.2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.1228 & -0.2763 \\ 195.5 & -529.4 \\ -53.61 & 33.25 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.134)$$

The RSLQR controller is modeled as

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r\end{aligned}\quad (3.135)$$

where

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & [0.9945 \ 0.1045] & [-1] \\ [0.7852] & [2.0536 \ 0.0797 \ -0.0458] & [0] \\ [-1.0409] & [-3.8238 \ -0.1280 \ 0.1020] & [0] \end{bmatrix} \quad (3.136)$$

When forming analysis models, it is important to check them for correctness. The plant (3.133) and controller (3.135) are connected to form a closed-loop state-space model of M using the linear fractional transformation (lft) command in MATLAB®. The closed-loop system has $n_x = 4$ states. The eigenvalues of the system are then compared to the eigenvalues of (3.132). This is a partial demonstration that the system is connected properly. Frequency domain models can also be checked (at the input and output) by evaluating the controller K and plant G at a given frequency, and forming M as in (3.95). This matrix can be compared to the state-space model frequency response created using the (lft) command.

To begin the analysis, we will examine robustness of this MIMO control system to the input uncertainties Δ_1 . The M matrix at this loop breakpoint has dimensions 2×2 and is the (1,1) element of M in (3.95). We will compute the SSV μ along with the SGT bound versus frequency. Figure 3.35 shows the SSV μ results for Δ_1 a full matrix and $\Delta_1 = \text{diag}[\delta_1 \ \delta_2]$, and the SGT bound $1/\bar{\sigma}(M)$. We do not see any difference at the plant input from the SSV μ and SGT bounds, and by restricting the uncertainty to be diagonal. Further insight into this result can be gained by examining the M matrix elements.

$$\begin{array}{cc} \delta_a & \delta_r \\ \delta_a & \left[\begin{array}{cc} M_{11} & M_{12} \\ M_{21} & M_{22} \end{array} \right] \\ \delta_r & \end{array} \quad (3.137)$$

Figure 3.36 shows the magnitude of the elements of $M(j\omega)$. We see that the matrix is dominated by the (2,2) element. This implies that the directional β dynamics (versus the roll dynamics) driven by the rudder is establishing the bound. We see from Fig. 3.36 that by diagonalizing $\Delta_1 = \text{diag}[\delta_1 \ \delta_2]$ does not alter the bound.

Next, we perform robustness analysis at the plant output using just Δ_2 . The M matrix at this loop breakpoint has dimensions 3×3 . Figure 3.37 shows the SSV μ results for Δ_2 a full matrix and $\Delta_2 = \text{diag}[\delta_1 \ \delta_2 \ \delta_3]$, and the SGT bound $1/\bar{\sigma}(M)$. Here we see a much reduced bound as compared to the plant-input loop breakpoint. At the input, $\min(1/\mu) = 0.5815$ and was the same for both Δ_1 a full matrix and $\Delta_1 = \text{diag}[\delta_1 \ \delta_2]$. Here, $\min(1/\mu) = 0.01389$ for Δ_2 a full matrix and $\min(1/\mu) = 0.18855$ when $\Delta_2 = \text{diag}[\delta_1 \ \delta_2 \ \delta_3]$. Restricting the Δ_2

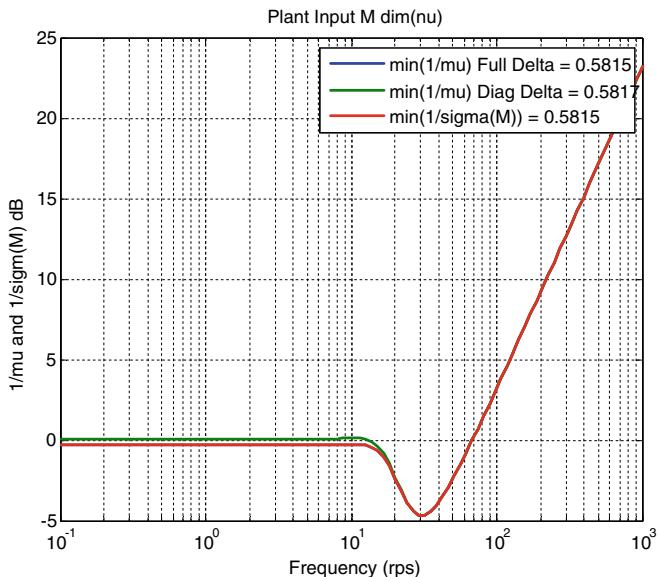


Fig. 3.35 Example 3.6 SSV μ and small gain theorem bounds at the plant input

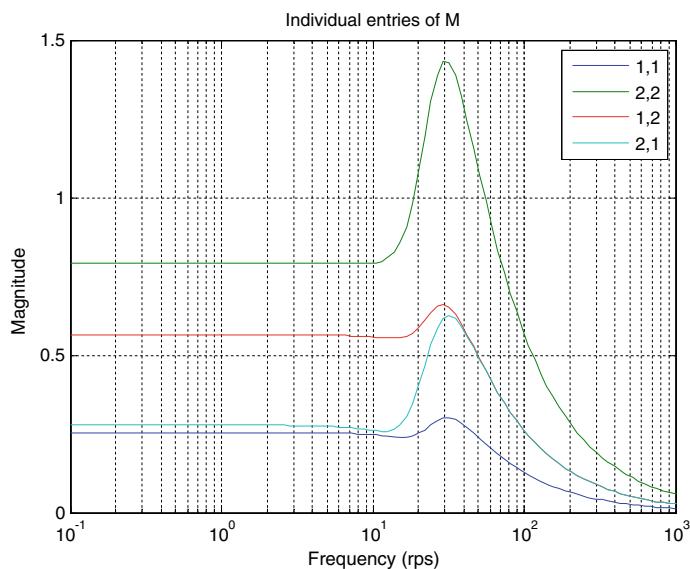


Fig. 3.36 Example 3.6 magnitudes of $M(j\omega)$ frequency response for plant-input uncertainty

matrix to be diagonal has a large impact. We would expect that the off-diagonal elements of $M(j\omega)$ are influencing the bound. Figure 3.38 shows the magnitude of the elements of $M(j\omega)$ for the diagonal element and off-diagonal elements and confirms this result. The M matrix at the plant output is

$$\begin{array}{c} \beta \ p \ r \\ \beta \left[\begin{array}{ccc} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{array} \right] \\ p \\ r \end{array} \quad (3.138)$$

We see from Fig. 3.38 that the 2,1 and 3,1 elements of $M(j\omega)$ are dominating the matrix. This also implies sensitivity in the directional β -dynamics.

The last analysis is the combined plant-input and plant-output case, as shown in Fig. 3.32. The M matrix has dimensions 5×5 . Figure 3.39 shows the SSV μ results for a full matrix Δ , a block diagonal matrix $\Delta = \text{diag}[\Delta_1 \ \Delta_2]$, a diagonal matrix $\Delta = \text{diag}[\delta_1 \dots \delta_5]$, and the SGT bound $1/\bar{\sigma}(M)$. The curves for the SSV μ analysis of a full matrix Δ_2 and the SGT bound are identical. Introducing structure in Δ only slightly improves the bound. To better understand this, we examine the frequency response $M(j\omega)$. Figure 3.40 shows the magnitude of the elements of $M(j\omega)$ for the diagonal elements and off-diagonal elements. We see that the dominant entry in $M(j\omega)$ is the (3,3) entry. This corresponds to the directional β dynamics, which are open-loop unstable. The rudder ($\delta_r \Rightarrow \beta$) off-axis

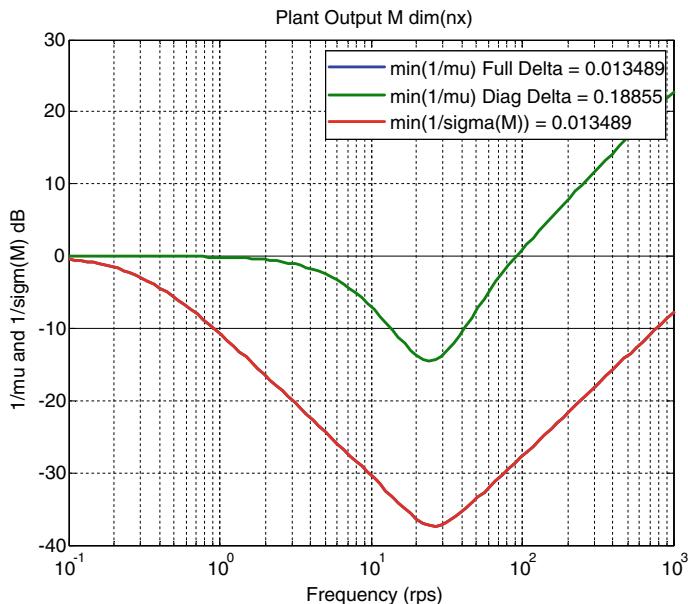


Fig. 3.37 Example 3.6 SSV μ and small gain theorem bounds at the plant output

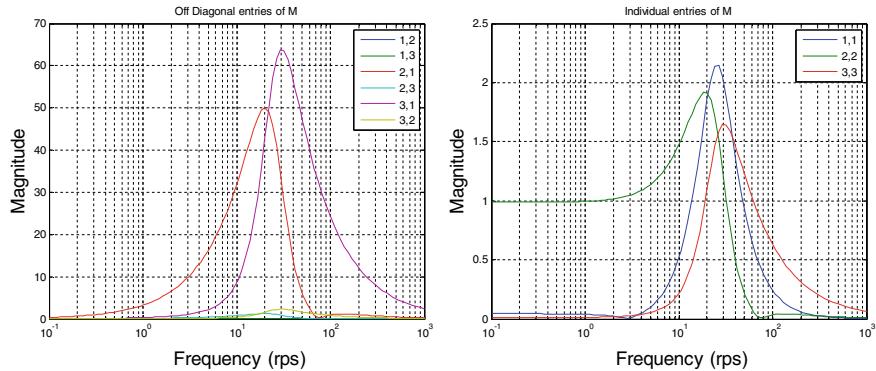


Fig. 3.38 Example 3.6 magnitudes of $M(j\omega)$ frequency response for plant-output uncertainty

elements are the next largest elements. It is the rudder δ_r that primarily stabilizes the directional axis. This large (3,3) entry in $M(j\omega)$ is dominating the stability analysis. This is why the block diagonal and diagonal Δ matrices did not produce larger bounds. Examination of $M(j\omega)$ confirms the robustness analysis indicating that the open-loop unstable vehicle is sensitive to uncertainties in the directional axis dynamics. Uncertainties in the air data system used to produce the feedback signal β should be examined in detail, and in simulation.

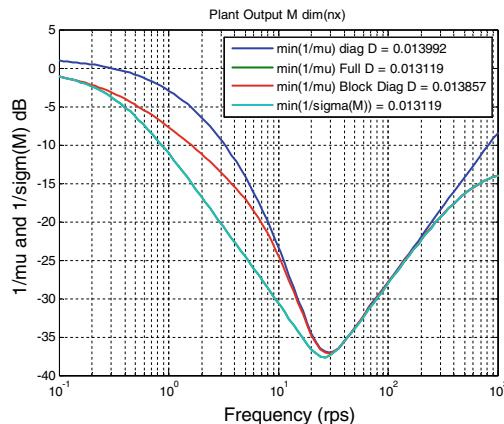


Fig. 3.39 Example 3.6 SSV μ and small gain theorem bounds for combined plant-input and output uncertainties

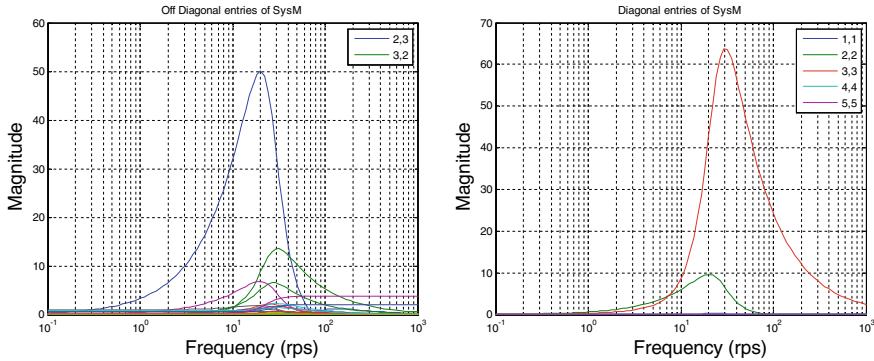


Fig. 3.40 Example 3.6 magnitudes of $M(j\omega)$ frequency response for combined plant-input and output uncertainties

3.4.3 Real Stability Margin

An important question in the design of a control system is one that asks how well one must know the parameters in the model of the dynamics? This question arises due to the fact that the coefficients in the differential equations are seldom known exactly. A large gain margin gives some comfort that the system is robust, but it does not accurately predict the sensitivity of the controller to knowing the parameters.

The SSV μ discussed in the previous section is a method for measuring the sensitivity of the system to uncertainties. An additional measure called the real multiloop stability margin, or real margin, as defined by De Gaston and Safonov in [4], is a scalar quantity also interpreted as a gain margin. Its calculation gives a non-conservative measure of control system stability robustness to real parameter variations modeled in the system dynamics (see Table 3.2).

In classical control system analyses, root locus plots can be used to graphically analyze system sensitivity to a change in gain. This same method can be used to analyze real parameter variations varying a single parameter at a time. For a single parameter this is a very easy and useful analysis to perform. In Kuo [8], the root locus approach is extended for multiple parameters. This approach has not been widely used due to its complexity and difficult graphical interpretation.

Consider a control system whose plant dynamics have uncertain dynamics. These uncertainties can arise from uncertain real parameter variations, neglected/modeled dynamics, or combinations of both. In this section we will focus on real parameter uncertainty. To analyze the control system, we will transform the model into the ΔM analysis model Fig. 3.7. The real parameter uncertainties in the system are isolated and placed into a diagonal matrix Δ . The transfer matrix M describes nominal system characteristics that are stabilized by the controller. We assume that the system has adequate stability margins. Thus, for $\Delta = 0$ (no uncertainty), the system is stable and the performance meets requirements.

Consider n uncertain real parameters represented by $\delta_i \in D_i \subseteq \mathbb{R}$ where D_i is the domain of the i th parameter. Let

$$\Delta = \text{diag}[\delta_1, \dots, \delta_n] \quad (3.139)$$

and define the parameter space D as

$$D = D_1 \times D_2 \times \cdots \times D_n \quad (3.140)$$

This parameter space describes the uncertain real parameters modeled in Fig. 3.7.

Using the multivariable Nyquist Theorem 3.2, the stability of the system described by Fig. 3.7 is implied by $\det[I - \Delta M] \neq 0$. The analysis problem is to find the largest parameter space D such that the system remains stable. This can be interpreted as finding the smallest uncertainty Δ that destabilizes the system. Consider the scalar stability margin k_m defined as

$$k_m = \min\{k \in [0, \infty) | \det[I - k\Delta M] = 0\} \quad (3.141)$$

If

$$(1/k_m)\delta_i \in D_i \quad \forall i \quad (3.142)$$

then ΔM remains stable for $\Delta \subseteq D$. This defines k_m as a multiloop stability margin.

A numerical algorithm for computing k_m can be constructed to converge by iterating lower and upper bounds on k_m , which are determined when either the convex hulls or interior points, respectively, of certain image sets first intercept the origin. This development is made possible by the use of a mapping theorem taken from Zadeh and Desoer [9]. Then the multiloop stability margin is computed by finding the smallest k for which there exists $\Delta = \text{diag}[\delta_1, \dots, \delta_n] \in D$ such that $\det[I - k\Delta M] = 0$.

A step-by-step approach to computing an estimate of the smallest value of k can be defined as follows:

1. Select an initial value of k .
2. Map the parameter space D into the complex plane via $\det[I - k\Delta M]$. If this region does not include the origin, then k is a lower bound on the stability margin k_m .
3. Increment k positively until the origin is just included in the map. This would yield k_m exactly. However, computing the true image of D under the mapping $\det[I - k\Delta M]$ is computationally prohibitive. To circumvent this problem, the convex hull of the image of D can be used.

In general, the parameter space D will be an n -dimensional polytope having 2^n vertices. By scaling the parameter uncertainties and incorporating the scaling into M , a hypercube describing the parameter space may be used rather than a polytope. Define V_i as a vertex of the hypercube D , where $i = 1, \dots, 2^n$.

The vertex V_i represents a corner of the hypercube. Let

$$V = \{V_1, V_2, \dots, V_m\}, \quad m = 2^n \quad (3.143)$$

denote the set of all hypercube vertices $V \subset D$. For example, consider three uncertain parameters. There are then $2^3 = 8$ vertices in the hypercube. They are

$$V = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{bmatrix} \quad (3.144)$$

Let Δ_{V_i} be a matrix of parameter uncertainties made up of the vertex points $v_{i,j}$ as j is varied from 1 to n . This is described as

$$\Delta_{V_i} = \text{diag}[v_{i,j}, j = 1, \dots, n] \quad (3.145)$$

From (3.144), $\Delta_{V_2} = \text{diag}[1 \ 1 \ -1]$. Define

$$\det[I - kDM] = \left\{ z \in C | z = \det[I - k\Delta M] \quad \forall \delta_i \in D_i, \right. \quad \left. i = 1, \dots, n, \text{ with } k, M \text{ fixed} \right\} \quad (3.146)$$

This is a set of points that represents the entire hypercube solid being mapped into the complex plane through the determinant function. It describes the entire image of D (the image of the parameter uncertainties) in the Nyquist plane. Next, define

$$\det[I - kVM] = \{y_i \in C | y_i = \det[I - k\Delta_{V_i}M], \quad i = 1, \dots, n\} \quad (3.147)$$

Equation (3.147) describes the set of points mapped into the complex plane made by the hypercube vertices. Let $F_i = \det[I - k\Delta_{V_i}M]$ be the mapping of the i th vertex. F_i is a single point in the set $\det[I - kVM]$. With these definitions, we are now ready to state the following theorem.

Theorem 3.6 [4] *Let $k, M, D, \det[I - kDM]$, and $\det[I - kVM]$ be defined as above. Fix k . Then*

$$\det[I - kDM] \subset \text{co}\{\det[I - kVM]\} \quad (3.148)$$

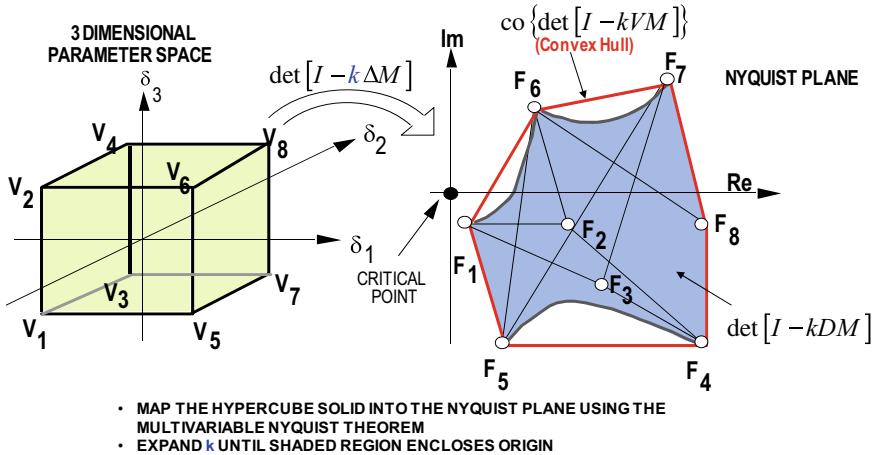


Fig. 3.41 Mapping the uncertain parameter space into the Nyquist plane

This theorem states that the true image of the hypercube is contained in the convex hull created from the vertices. By mapping the 2^n vertices of D into the Nyquist plane and then constructing a convex hull about the 2^n points, a polygon is created that encompasses the $\det[I - kDM]$. Figure 3.41 illustrates this for a 3-dimensional hypercube.

The hypercube is in Fig. 3.41 is the yellow box with the $2^3 = 8$ vertices arbitrarily labeled. Thus, $\Delta \subset D = D_1 \times D_2 \times D_3$ with $D_i = [\delta_{i,\min}, \delta_{i,\max}]$. The parameters $\delta_{i,\min}$ and $\delta_{i,\max}$ describe the lower and upper bounds of the parameter δ_i . By scaling these uncertainties and incorporating the scaling into M , we can model each parameter δ_i with limits of ± 1 .

Figure 3.41 shows the mapping of this parameter-space hypercube into the complex plane using the determinant mapping function. The blue shaded region depicts the true image of the cube solid mapped into the Nyquist plane. If the origin was contained in the shaded region, then the system would be unstable. Since the origin is not in the shaded region, the gain margin k used in $\det[I - kDM]$ is smaller than the true stability margin and should be increased in magnitude until the origin is included.

The value of k such that the origin is just included in the shaded region is the exact stability margin k_m we seek to compute. In Fig. 3.41, the vertex points V_i are mapped into the F_i points. The convex hull containing the image of the hypercube is denoted as $\text{co}\{\det[I - kVM]\}$ and is shown as a heavy red border around the $\det[I - kDM]$ image. We see from the figure that if $\text{co}\{\det[I - kVM]\}$ were used to determine k , conservatism would be present since the $\text{co}\{\det[I - kVM]\}$ contains more points than the true image of $\det[I - kDM]$. This fact is used to define a lower bound on k_m , resulting in the following lemma.

Corollary 3.1 Let M , D , and $\det[I - kDM]$ be defined as given in Theorem 3.3. Then, for $k > k_0$,

$$\det[I - k_0 DM] \subset \det[I - kDM] \quad (3.149)$$

$$\text{co}\{\det[I - k_0 DM]\} \subset \text{co}\{\det[I - kDM]\} \quad (3.150)$$

This corollary states that the image of the hypercube solid under the determinant mapping function, for k_0 , is a subset of the image mapped using a larger k . Thus, the convex hull containing $\det[I - k_0 DM]$ is contained in the convex hull $\text{co}\{\det[I - kDM]\}$.

Lower Bound on the Stability Margin k_m

Application of Corollary 3.1 allows us to expand the $\text{co}\{\det[I - kDM]\}$ until the origin is enclosed. We show this graphically in Fig. 3.42. The solid lines represent $\text{co}\{\det[I - k_i VM]\}$ for k_1, k_2 , and k_3 . As shown in the figure, for all $k < k_3$, the origin is not enclosed by $\text{co}\{\det[I - kVM]\}$. Thus, $k_3 < k_m$ is a lower bound for the stability margin k_m . If k increases without $\text{co}\{\det[I - kVM]\}$ intercepting the origin, then $k_m = \infty$.

Upper Bound on the Stability Margin k_m

To compute the upper bound on k_m , the path between the vertices whose line segment intercepts the origin must be examined more closely. Define the following critical vertices:

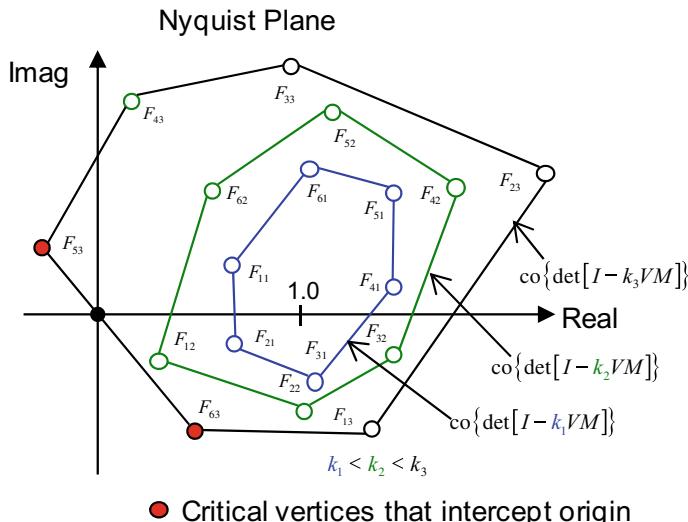


Fig. 3.42 Expanding the convex hulls until the origin is intercepted

- Critical vertices: $F_i = \det[I - k\Delta_{V_i}M]$, $F_j = \det[I - k\Delta_{V_j}M]$, $i \neq j$, $\beta \in [0, 1]$ such that $(1 - \beta)F_i + \beta F_j = 0$.
- Isolated critical vertex (ICV): F_i is isolated if $F_i \neq F_j$, $i \neq j$.
- Coincident critical vertex (CCV): F_i is coincident if $F_i = F_j$, $i \neq j$.

Critical vertices are defined as the two vertices whose line segment intercepts the origin. This is shown in Fig. 3.42 as the convex hull line segment $F_{53} - F_{63}$. These critical vertices are isolated if $F_i \neq F_j$. They are coincident if $F_i = F_j$. Let $m(i, j)$ be equal to the number of differing coordinates of the two vertices V_i and V_j that are mapped by $\det[I - k\Delta_{V_i}M]$ to F_i and F_j . In Fig. 3.42 the critical vertices are F_{53} and F_{63} . The $m(i, j)$ is the minimum number of edges on the hypercube D from vertex V_i to V_j . In Fig. 3.42, $m(i, j) = 1$ (examine the cube in Fig. 3.41 to see that vertex V_5 connects directly to V_6). The following corollary will aid in the calculation of the upper bound on k_m .

Corollary 3.2 *Any path along a single coordinate in D is mapped by $\det[I - kDM]$ to a straight line in the complex plane.* ■

For fixed M , the $\det[I - k\Delta M]$ for $\Delta = \text{diag}[\delta_1, \dots, \delta_n] \in D$ is a polynomial in the variables δ_i and is affine with respect to each of the δ_i . This is true only for a diagonal Δ and is obtained by definition of the determinant. This affine relationship proves this lemma.

Corollary 3.2 guarantees that any point on the face of the hypercube D mapped into the Nyquist plane will be contained in the convex hull formed by the mapped vertices. This is true only for real parameters. If the parameters under variation were complex, any path along a single coordinate would trace an arc in the Nyquist plane. Thus, points contained on the face of a complex-parameter hypercube mapped into the Nyquist plane need not be contained in the convex hull formed by the hypercube vertices. This fact precludes the use this parameter-space method in analyzing complex-parameter variations.

Define a vertex path as any path between critical vertices F_i and F_j , consisting of $m(i, j)$ straight-line segments, defined by $\det[I - k\Delta_x M]$ as x progresses from V_i to V_j along the edges of the hypercube D . The first such vertex path to touch the origin defines a point in $\det[I - kDM]$ and the associated k is an upper bound on the stability margin k_m . The vertex path will determine the region in the parameter space that causes instability.

Convergence to k_m

The actual stability margin is computed by an iterative algorithm. It begins by examining the vertex paths between critical vertices (ICVs or CCVs) that intercept the origin. This defines the edges of the hypercube closest to the origin. The domain D is then split along this vertex path, creating subdomains. Convex hulls around smaller and smaller subdomains are computed. As the subdomains become small, the union of all of the convex hulls for all of the subdomains gets close to the actual image of the domain D . The accuracy in the computation of k_m is then

dependent on how small the subdomains are made. The following three lemmas are used to prove the convergence theorem that computes the exact multi-loop real stability margin.

Lemma 3.2 *On a hypercube of dimension n with two vertices that differ by m coordinates, there are $m!$ paths between these two vertices along the edges of the hypercube. Each path between these two vertices will have $m + 1$ vertices along the path (including the original vertices). ■*

Lemma 3.3 *Let $k, M, D, \det[I - kDM]$, and $\det[I - kVM]$ be defined as previously given. Let F_i and F_j be isolated critical vertices with $m(i, j) > 2$ and F_k denote the first vertex along a vertex path emerging from F_i . Define a point along the line segment between F_i and F_k as F_x , exclusive of the end points, i.e., $F_x = (1 - \beta)F_i + \beta F_k, \beta \in (0, 1)$. Let V_x be the associated point on the hypercube edge defined between V_i and V_k . Cut the domain D at V_x orthogonally to this edge to create two subdomains D_1 and D_2 , where $V_i \in D_1$ and $V_j \in D_2$. Then neither $\text{co}\{\det[I - kD_1M]\}$ nor $\text{co}\{\det[I - kD_2M]\}$ includes the origin. ■*

Lemma 3.4 *Let $k, M, D, \det[I - kDM]$, and $\det[I - kVM]$ be defined as previously given. Then there is at least one Δ_{V_i} associated with the stability margin k_m that assumes an extremal value. ■*

Lemma 3.2 is used to determine the number of vertex paths between critical vertices. These vertex paths define the coordinate direction in which the parameter-space domain D is split into subdomains.

Lemma 3.3 is the heart of the convergence theorem used to obtain k_m . It is employed when $m(i, j) > 2$. The utility of this lemma is best explained by an example. In Fig. 3.41, let vertex images F_1 and F_6 be isolated critical vertices, with k_l determined such that $\text{co}\{\det[I - k_l VM]\}$ intercepts the origin. For this case, $m(i, j) = m(1, 6) = 2$. The convex hull enclosing the $\det[I - kDM]$ image has a larger area than the true image of the hypercube solid (shaded area). The area contained in $\text{co}\{\det[I - k_l VM]\}$ that is not contained in $\det[I - kDM]$ makes k_l a conservative estimate, i.e., $k_l < k_m$. Lemma 3.4 says that if we split the parameter space into two subdomains along one of the two vertex paths ($V_1 - V_2 - V_6$) or ($V_1 - V_5 - V_6$), and compute convex hulls about each of the images of the two subdomains, then the origin will not be contained in either convex hull. This guarantees that we can converge to the true stability margin k_m by splitting the parameter space into subdomains. As the subdomains become smaller, we approach the true image of $\det[I - kDM]$.

Lemma 3.3 states that $k < k_m$ will not destabilize the system. Geometrically, this places the Δ_{V_i} on the boundary of D and guarantees a unique stability margin k_m . By using these Lemmas 3.1 through 3.3, the convergence theorem [4] follows.

Theorem 3.7 *Let $k, M, D, \det[I - kDM]$, and $\det[I - kVM]$ be defined as previously given; then an iterative algorithm can be constructed that converges to k_m .*

If k_m is finite, then this procedure identifies the parameters $\delta_i \in D$ at which k_m is determined. There are three steps involved in determining k_m :

- (1) Determine the lower bound on k_m ,
- (2) Determine the upper bound on k_m , and
- (3) Iterate lower and upper bounds and converge on k_m .

The actual procedure involved in each step is very problem dependent. As may be expected, there are several special cases concerning the critical vertices that vary the algorithm. For example, let $\text{co}\{\det[I - kVM]\}$ intercept the origin between two critical vertices F_i and F_j , one or both of which are coincident. For this case, different logic is required when splitting the domain into subdomains. De Gaston and Safonov [4] present an excellent exposition of these special cases. They are briefly summarized here.

Special Case 1

The $\text{co}\{\det[I - kVM]\}$ intercepts the origin at a single isolated critical vertex (ICV) $F_i = \det[I - k_l \Delta_{V_i} M]$. Then $m(i, j) = 0$, $k_l = k_u = k_m$, and the algorithm stops. The parameters that cause instability are at the vertex V_i .

Special Case 2

The $\text{co}\{\det[I - kVM]\}$ intercepts the origin between two ICVs F_i and F_j where $m(i, j) = 1$. Both points F_i and F_j are contained in the mapped hypercube image $\det[I - kDM]$. With Corollary 3.2, the line segment connecting these two vertices is also contained in the mapped hypercube image $\det[I - kDM]$. Thus, $k_l = k_u = k_m$ and the algorithm stops. The Δ along this edge of the hypercube that is destabilizing is given by

$$\Delta_\beta = \text{diag}[(1 - \beta)V_i + \beta V_j]$$

where

$$\beta \in (0, 1) \text{ such that } \det[I - k\Delta_\beta M] = 0 \quad (3.151)$$

If either of the above special cases is true, the application of step 1 determines k_m . Only if $m(i, j) \geq 2$ does the algorithm progress further.

Consider ICVs F_i and F_j with $m(i, j) \geq 2$. The upper bound on k_m is determined by examining the $m(i, j)!$ vertex paths between F_i and F_j . The upper bound k_u is determined by the largest k along one of the $m(i, j)!$ vertex paths that intercepts the origin. If k is increased and the origin is not intercepted, then $k_u = \infty$. Once the lower and upper bounds k_l and k_u have been determined, Lemmas 3.1–3.3 are used to converge to k_m .

Special Case 3

The $\text{co}\{\det[I - kVM]\}$ intercepts the origin between two critical vertices F_i and F_j in which one or both are coincident.

Special Case 3a

Consider the problem where F_i and F_j both intercept the origin, i.e., $\det[I - k_l \Delta_{V_i} M] = \det[I - k_l \Delta_{V_j} M] = 0$. Then $k_l = k_m$ and the stability margin is defined at multiple values of Δ_{V_i} .

Special Case 3b

There are several coincident vertices located at F_i and several at F_j in which m_c is defined as follows:

$$m_c = \min\{m(i, j)\} = 1; \quad i = \{a, b, \dots\} \quad j = \{s, t, \dots\} \quad (3.152)$$

Pick an $i \in \{a, b, \dots\}$ and $j = \{s, t, \dots\}$. Thus $m(i, j) = 1$ and k_m is determined as in special case 3.

Special Case 3c

This is the same condition as in special case 3b, except that $m_c \geq 2$. For this case, domain splitting is used to divide the domain into subdomains. This is repeated along each of the vertex paths to each coincident critical vertex.

Let the set $\{z\}$ contain z coincident critical vertices at F_z and the set $\{y\}$ contain y critical vertices at F_y . Take the first two elements of the set $\{z\}$, say, a, b . Then $m(a, b) \geq 1$, since both a, b are vertices of the hypercube. Split the domain along the edge between these two vertices with an orthogonal cut. This creates two subdomains D_1 and D_2 , each containing one of the critical vertices a and b . Continue this process $z - 2$ times, creating z subdomains, each having an isolated critical vertex at F_z . Repeat this same process for the critical vertices in $\{y\}$. This creates zy subdomains, each having two critical vertices. Apply the procedures of the preceding special cases to each of these subdomains.

Computing the Real Margin

Figure 3.43 outlines the calculation of the real margin k_m . The algorithm uses the convex hull-based lower bound from Theorem 3.7. The ΔM analysis model is created as defined in Sect. 3.3.1 following the example given in Example 3.4. The state-space model for M is

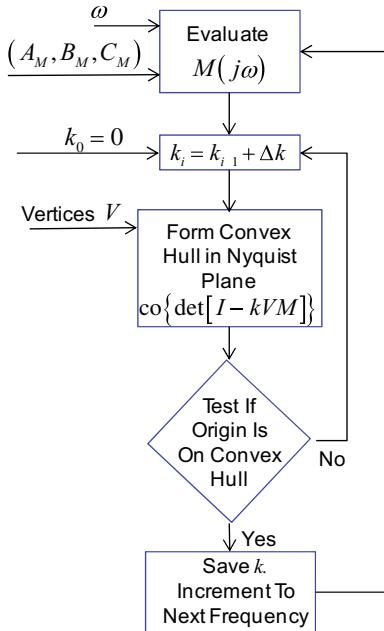
$$A_M = A_0; \quad B_M = [b_1 \dots b_n]; \quad C_M = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (3.153)$$

and the uncertainty matrix Δ is

$$\Delta = \text{diag}[\delta_1 \dots \delta_n] \quad (3.154)$$

A vector of frequencies is selected spanning the range in which the real margin is to be computed. As shown in Fig. 3.41, the matrix M is then evaluated at a given frequency, $M(j\omega) = C_M(sI - A_M)^{-1}B_M$, and the vertices of the hypercube are

Fig. 3.43 Flowchart for computing the real margin k_m



mapped into the Nyquist plane defined in V . Each vertex V_i (2^n of them) maps to a point F_i . A convex hull is then formed using the points F_i , and a zero-exclusion test is made to see if the origin is contained on the convex hull. If not, the value of k is increased, and the mapping/convex hull/zero-exclusion procedure is repeated. We know that for $k = 0$, the $\det[I - kDM] = 1.0$. As k is increased the algorithm stops when either the origin is on the convex hull or some upper limit is reached. This procedure is applied to each frequency in the vector ω and then a plot of k versus frequency is made. The robustness bound is the minimum k versus frequency. The following example demonstrates this analysis.

Example 3.7 Autopilot Sensitivity to Real Parameter Variations Consider the longitudinal dynamics from Example 3.2 with the angle-of-attack controller from Example 3.5. The ΔM analysis model for this problem was formed in Example 3.5. This analysis will determine how sensitive the LQR control law is to accurate values for the aerodynamic stability derivatives used in the design model. We show in Chap. 4 that LQR control laws have infinite margin at the plant-input loop breakpoint. This example will show that these control laws are sensitive to the accuracy of the model parameters.

The state-space triple for M from Example 3.5 is

$$\begin{aligned}
 A_M &= \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 \\ 0 & -1.3046 & 1.0 & -0.2142 & 0 \\ 0 & 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 275,100 & 49,059 & 3753 & -16,861 & -163 \end{bmatrix} \\
 B_M &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.1422 & 0.4628 & 0 & 0 \\ 0 & 0 & -6.9073 & 10.2389 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \\
 C_M &= \begin{bmatrix} 0 & -1.1422 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.4628 & 0 \\ 0 & -6.9073 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10.2389 & 0 \end{bmatrix} \quad (3.155)
 \end{aligned}$$

where the uncertain aerodynamic stability derivatives (Z_α/V , Z_δ/V , M_α , M_δ) are highlighted in red. The uncertainty matrix Δ that models the real parameter uncertainties is a 4×4 diagonal matrix that models the parameter variations using. For $n = 4$ there are $2^4 = 16$ hypercube vertices, modeled as in (3.144).

Figure 3.44 shows a frequency sweep ($\omega = \text{logspace}(-2, 3, 100)$) where the minimum k was determined at each frequency by the $\text{co}\{\det[I - kVM]\}$ intersecting the origin of the Nyquist plane. Also included in the plot is the bound computed using the small gain theorem. The minimum k versus frequency is $k_m = 0.53831$, and it occurs at a frequency of $\omega = 8.7976$ rad/s. The small gain theorem bound is $\min(1/\bar{\sigma}(M)) = 0.27167$.

To determine if the bound $k_m = 0.53831$ is conservative or exact, we must examine the convex hull and the vertices used in forming the convex hull. Figure 3.45 is a plot of the 16 hypercube vertices at $\omega = 8.7976$ rad/s and $k = 0.53831$. The convex hull vertices and uncertainty matrices Δ_{V_i} are

$$\begin{aligned}
 F_{13} &= 1.5867 - 0.0718i \quad \Delta_{V_{13}} = \text{diag}[-1 \ -1 \ 1 \ 1] \\
 F_{15} &= 1.8186 - 0.0700i \quad \Delta_{V_{15}} = \text{diag}[-1 \ -1 \ -1 \ 1] \\
 F_7 &= 2.0505 - 0.0314i \quad \Delta_{V_7} = \text{diag}[1 \ -1 \ -1 \ 1] \\
 F_3 &= 2.0752 - 0.0060i \quad \Delta_{V_3} = \text{diag}[1 \ 1 \ -1 \ 1] \\
 F_4 &= 0.3260 + 0.0787i \quad \Delta_{V_4} = \text{diag}[1 \ 1 \ -1 \ -1] \\
 F_2 &= 0.1062 + 0.0759i \quad \Delta_{V_2} = \text{diag}[1 \ 1 \ 1 \ -1] \\
 F_6 &= 0.0694 + 0.0515i \quad \Delta_{V_6} = \text{diag}[1 \ -1 \ 1 \ -1] \\
 F_{14} &= -0.0000 - 0.0000i \quad \Delta_{V_{14}} = \text{diag}[-1 \ -1 \ 1 \ -1] \quad (3.156)
 \end{aligned}$$

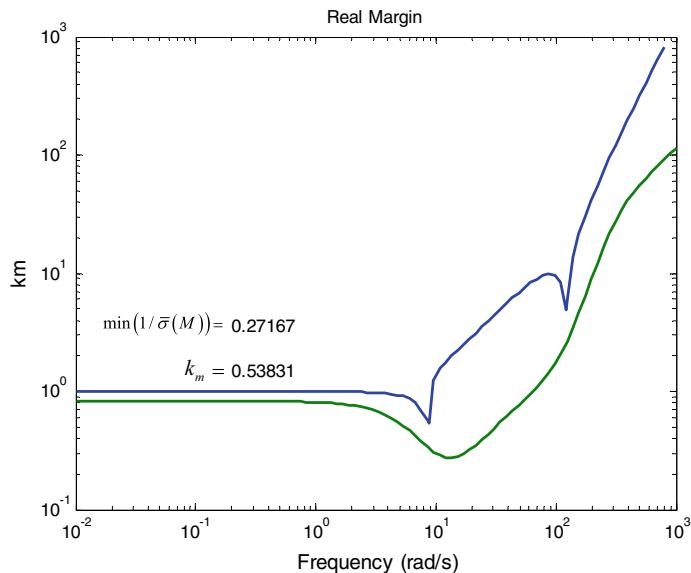


Fig. 3.44 Frequency sweep of the real margin and small gain uncertainty bounds

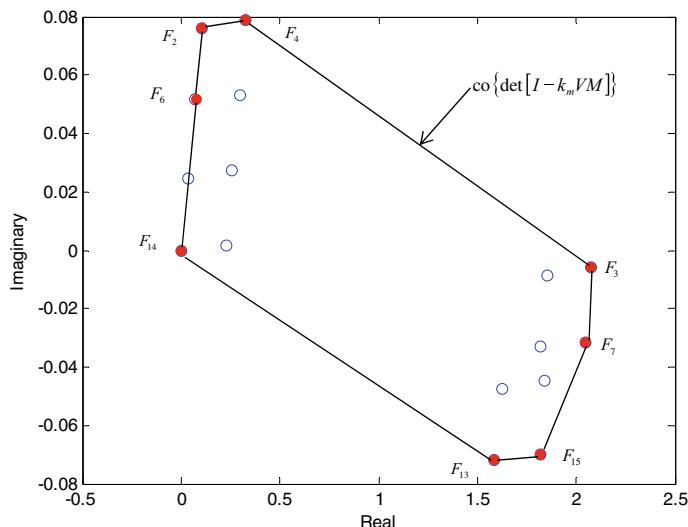


Fig. 3.45 Convex hull for the real margin bound

We see from Fig. 3.45 and that $\Delta_{V_{14}}$ intercepts the origin. To show that the closed-loop system is destabilized using $\Delta = 0.53831\Delta_{V_{14}}$ we insert these uncertainties into the system matrix and compute the eigenvalues of A_{cl} . Doing so yields the following eigenvalues:

$$\lambda(A_{cl}) = \begin{bmatrix} -76.0970 \pm 96.3341j \\ 0.0001 \pm 8.7976j \\ -11.4150 \end{bmatrix} \quad (3.157)$$

which shows two roots just on the $j\omega$ axis.

This analysis shows that this design can tolerate a 53.8% variation in the aerodynamic stability derivatives. We see that this “robust stability margin” is much less than the plant-input margins guaranteed by using LQR controllers (at the plant input). It is well known that classical gain and phase margins, including the vector margin, do not necessarily mean the system is robust to real parameter uncertainties. This state feedback design has a scalar loop gain at the plant input described by $L_u(s) = K_c(sI - \tilde{A})\tilde{B}$, which can also be written explicitly in transfer function form.

$$L_u(s) = \frac{27.4(s_i + 67.8 \pm 90.4j)(s_i + 7.5 \pm 5.1j)}{s(s + 7.6)(s - 6.3)(s_i + 67.8 \pm 90.4j)} = \frac{27.4(s_i + 7.5 \pm 5.1j)}{s(s + 7.6)(s - 6.3)} \quad (3.158)$$

Note that the open-loop system is unstable ($M_\alpha > 0$) and the actuator poles are exactly canceled in $L(s)$ (at the plant input only). The gain margin at the plant input is $[-11, +\infty]$ dB and the phase margin $\pm 60^\circ$. Next, consider a scalar real uncertainty δ_K at the plant input, and compute a root locus, i.e., zeros of $s(s + 7.6)(s - 6.3) + \delta_K 27.4(s_i + 7.5 \pm 5.1j)$. This is plotted in Fig. 3.46 and shows the system is stable for all gain values $\delta_K > 0.28$. This is achieved because of the zero dynamics that exist at this loop breakpoint.

When we analyze the system under real parameter uncertainty, the robustness bounds are determined by the zero dynamics that exist in the $M(s)$ matrix. This is the fundamental challenge in robust control. How do we design a controller that provides “robust” zero dynamics at multiple loop breakpoints simultaneously? We can see this challenge by repeating this root locus analysis for the uncertain parameters. Consider varying just the M_δ stability derivative. The state-space triple for $M(s)$ is

$$A_M = \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 \\ 0 & -1.3046 & 1.0 & -0.2142 & 0 \\ 0 & 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \\ 275,100 & 49,059 & 3753 & -16,861 & -163 \end{bmatrix}$$

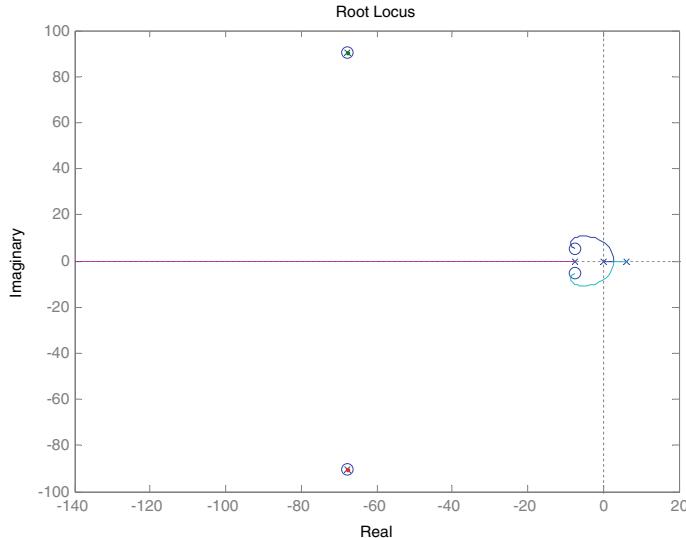


Fig. 3.46 Root locus of LQR $L(s)$ at plant input

$$B_M = \begin{bmatrix} 0 \\ 0 \\ 10.2389 \\ 0 \\ 0 \end{bmatrix}; \quad C_M = [0 \ 0 \ 0 \ -10.2389 \ 0] \quad (3.159)$$

The transfer function for this system is

$$M(s) = \frac{-393449.(s_i + 7.2 \pm 4.7j)}{(s + 14.3)(s_i + 7.2 \pm 10.3j)(s_i + 67.8 \pm 90.4j)} \quad (3.160)$$

A root locus is shown in Fig. 3.47. The roots cross the $j\omega$ axis with a gain of $\delta_{M_\delta} = 0.68$. We see this is larger than the real margin bound $k_m = 0.53831$ from the example because only a single parameter is being varied.

The key point to be understood from this example is that for linear systems the zero dynamics at each loop breakpoint influences the resulting stability robustness. The control architecture as well as the magnitude of the gains in the design influence and change these zero dynamics. Thus, as the bandwidth of the control design changes so does the sensitivity to accurate knowledge of the model parameters.

This fact has led us to combine robust control with adaptive control to improve the system's sensitivity to uncertainties and nonlinearities. The first half of this book is focused on optimal and robust control and how to design the best linear robust controller possible. “Best” in terms of meeting command tracking requirements and being robust to high-frequency unmodeled dynamics and sensor noise. These methods form the baseline control that is then augmented with an adaptive

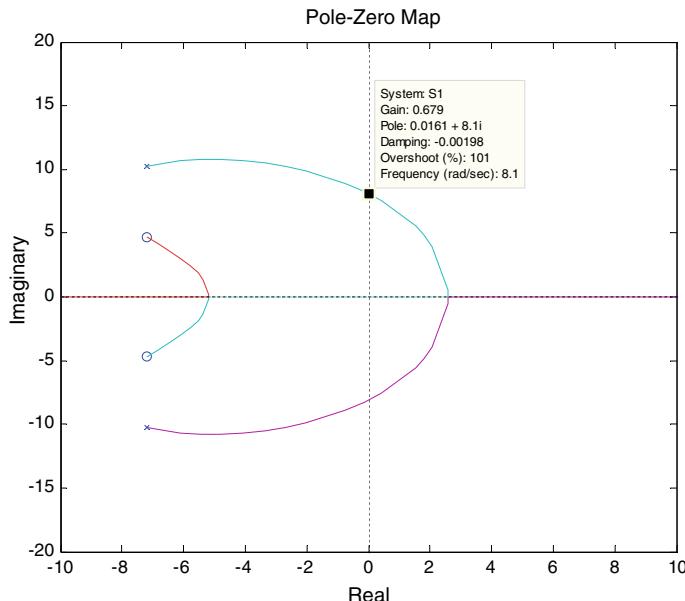


Fig. 3.47 Root locus for $M(s)$ varying only M_δ

increment to further address uncertainties and nonlinearities that the robust control. The second half of the book covers the adaptive control linear and nonlinear systems. Together we have found these methods solve some of the most challenging problems in aerospace control.

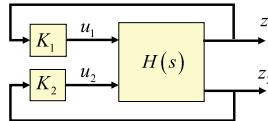
3.5 Conclusions

This chapter presented the theory and practice of using frequency domain methods to analyze robust stability. The ability to design controllers and analyze the stability of multivariable systems has been an enabling technology for the aerospace and other industries. Prior to the development of this theory in 1980s, vehicles were predominantly designed to be stable. The advent of modern control theory, in which the methods in this chapter belong, enables control engineers to basically control any shape of aircraft and other vehicles.

Virtually all of the design methods used today for control system design are model dependent. The accuracy in which we know these models varies, and to measure the data accurately can significantly increase costs. Analyzing the control systems robustness to uncertainties gives the engineer very powerful tools to determine what data is needed to be known accurately and what data is not. Overall this will reduce costs, but more importantly increases the quality of the overall control design, making it perform better and be safer.

3.6 Exercises

Exercise 3.1 Consider the following block diagram



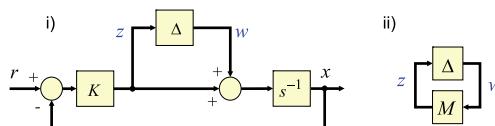
where

$$H(s) = \begin{bmatrix} \frac{3}{s} & \frac{-278}{s(s+6)(s+30)} \\ \frac{0.05}{s} & \frac{-206}{s(s+6)(s+30)} \end{bmatrix}$$

and $K_1 = 5$ and $K_2 = -10$.

- (a) Apply the multivariable Nyquist theorem from Sect. 3.3.2 to this system, examining the return difference matrix $I + KH$, where $K = \text{diag}[K_1 \ K_2]$, and determine stability. Create the multivariable Nyquist plot. This is a plot of the $\det[I + KH]$ and indicates the number of encirclements.
- (b) Plot the singular values of the return difference matrix and stability robustness matrix versus frequency. Compute the singular value gain and phase margins for this system. This is a plot of $\underline{\sigma}[I + L_u]$ and $\underline{\sigma}[I + L_u^{-1}]$ versus frequency. Plot these using a log scale for frequency and magnitude in dB.

Exercise 3.2 Consider the block diagrams shown below. Each block in the diagrams is a scalar.



- (a) Derive a state-space model (A_M, B_M, C_M) for $M(s) = C_M(sI - A_M)^{-1}B_M$ (shown in (ii)), modeling the system shown in (i))
- (b) For $\Delta = 0$, what is the range of gain K that provides closed-loop stability?
- (c) For $K = 1$, sketch the small gain theorem applied to the system in (ii) using your model derived in part (a). This should be a magnitude versus frequency plot.
- (d) What does the sketch in (c) indicate for the system's robustness to uncertainties that are constant, i.e., $\Delta = \text{constant}$?

Exercise 3.3 Consider the longitudinal airframe dynamics and classical control system described in Example 3.1. The model data are for (3.14) and (3.8)

$K_a = -0.0015$	$V = 886.78 \text{ fps}$
$K_q = -0.32$	$Z_\alpha/V = -1.3046$
$a_z = 2.0$	$Z_\delta/V = -0.2142$
$a_q = 6.0$	$M_\alpha = 47.7109$
	$M_\delta = -104.8346$

- (a) Form a closed-loop state-space model and simulate an acceleration step response to show the system is stable and correct.
- (b) Form the loop transfer function matrix at the plant input. Compute stability margins.
- (c) The actuator dynamics were neglected during the controller design. Derive a multiplicative error model for the neglected actuator dynamics, assuming that the actuator dynamics are modeled using the following transfer function:
- (d) $\frac{\delta}{\delta_c} = \frac{1}{\tau s + 1}$
- (e) Form a ΔM robustness analysis model for analyzing these neglected actuator dynamics.
- (f) Determine the largest actuator time constant τ that results in a stable closed-loop system using the small gain theorem as the robustness test.

Exercise 3.4 Consider the longitudinal dynamics $\dot{x} = A_p x + B_p u$, $x = [\alpha \ q]^T$, with

$$[A_p \ B_p] = \begin{bmatrix} -1.2100 & 1.0 \\ 44.2506 & 0 \end{bmatrix} \begin{bmatrix} -0.1987 \\ -97.2313 \end{bmatrix}$$

Build a robust servomechanism model from this chapter:

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & A_p \end{bmatrix}; \quad \tilde{B} = \begin{bmatrix} 0 \\ B_p \end{bmatrix}$$

and close the loop using a state feedback control law $u = -K_c z$, $z = [\int e_\alpha \ \alpha \ q]^T$, with gains

$$K_c = [-2.1598 \ -1.3301 \ -0.1700]$$

Analyze the stability of this system. Plot a Nyquist plot, Bode plot, $\underline{\sigma}[I + L]$ and $\underline{\sigma}[I + L^{-1}]$ versus frequency, and compute singular value stability margins.

Exercise 3.5 Using the classical longitudinal control system from Exercise 3.3, analysis the closed-loop system's robustness to uncertainties in the aerodynamic parameters. Create the ΔM robustness analysis model using as in Example 3.4 considering the uncertainties

$$\left(\frac{Z_\alpha}{V} (1 \pm \delta_1), \frac{Z_\delta}{V} (1 \pm \delta_2), M_\alpha (1 \pm \delta_3), M_\delta (1 \pm \delta_4) \right)$$

Compute the structured singular value (SSV) bound analyzing the sensitivity to these real parameter uncertainties.

References

1. Åström KJ, Murray RM (2008) Feedback systems: an introduction for scientists and engineers. Princeton University Press, Princeton, p 08540
2. Doyle JC (1978) robustness of multiloop linear feedback systems. In: Proceedings of the IEEE conference on decision and control, pp 12–18
3. Wise KA (1992) A comparison of six robustness tests evaluating missile autopilot robustness to uncertain aerodynamics. J Guid Control Dyn 15(4):861–870
4. De Gaston RR, Safonov M (1988) Exact calculation of the multiloop stability margin. IEEE Trans Autom Control 33(2):156–171
5. Morton, BG, McAfoos RM (1985) A Mu-test for robustness analysis of a real-parameter variation problem. In: Proceedings of the American control conference
6. Morton BG (1985) New applications of Mu to real parameter variation problems. In: Proceedings of the 25th IEEE conference on decision and control
7. Doyle JC, Wall JE, Stein G (1982) Performance and robustness analysis for structured uncertainty. In: Proceedings of the 22nd IEEE conference on decision and control, pp 629–636
8. Kuo BC (1971) Automatic control systems, 6th edn. Prentice Hall, New York
9. Zadeh LA, Desoer CA (1963) Linear system theory. McGraw-Hill, New York



Optimal Control and Linear Quadratic Regulators

4

In this chapter we introduce optimal control concepts, linear quadratic regulators and servomechanism controllers for command tracking in the presence of unknown disturbances. We begin by briefly discussing and comparing classical control, modern control, optimal control, and why the optimal control has emerged as a popular and a reliable design method for flight critical control systems in aerospace and other industries. We then proceed by introducing optimal control problems and the resulting Hamilton–Jacobi–Bellman partial differential equation. For linear systems with a quadratic performance index, we develop the linear quadratic regulator. We will cover only infinite-time optimal control problems and explore some very important stability and robustness properties of these systems. Central to the design of optimal control laws is the selection of the penalty matrices in the performance index. We will also discuss asymptotic properties of optimal control solutions with regard to the penalty matrices that in turn will set the stage for detailed design of these controllers in later chapters. Within this section, servomechanism models are introduced and the associated command tracking controllers are derived in the context of the optimal control formalism.

4.1 Introduction

Control systems must provide stability and performance in the presence of model uncertainty and neglected dynamics. This has proven to be a significant challenge, and as our understanding of dynamics and control has improved, practicing engineers and control scientists have been able to develop new vehicle designs that are faster, have greater performance, and perform robustly in very large operational envelopes. These advancements were built upon the foundation created by the classical methods, but were powered by computer aided design tools which

greatly expanded the engineer's ability to solve larger, more complex problems, using advanced techniques.

In general, constructing flight critical or any other industrial control systems using conventional (classical) analytical methods involves iterative one-loop-at-a-time control design and analyzes processes that are costly in time and manpower. These systems are often synthesized by discretizing the vehicle operational envelope at specific points, designing the control system at these points, and guaranteeing robustness to parameter variations by designing large single-loop stability margins and evaluating the design through simulation. These methods worked well on vehicles that are open-loop stable, but as new system configurations emerged that were open-loop unstable in multiple axes, multi-input–multi-output (MIMO) design methods became essential.

In the 1970s and 1980s, the question of robust stability and performance was raised and new control system design and analysis methods emerged, called the “modern control”. These advancements provided the theoretical foundation and mathematics required for optimizing a controller design for MIMO systems, with evaluation of stability and robustness to parameter uncertainties. Using methods for characterizing model uncertainties, controller robustness properties were evaluated, and iterative design tools emerged to achieve robust stability and performance. These modern methods allowed the control system designer to understand and directly address stability and robustness concerns for open-loop unstable MIMO systems. With computer-aided design tools, engineers could readily pose and solve “optimal control” problems for complex systems, and implement control solutions across a large operational envelope using gain scheduling that involves connecting individual controllers via interpolation over the envelope grid points.

Optimal control paradigm is based on minimizing a cost performance index. There are many classes of problems for nonlinear or linear systems, dealing with time-variant or time-invariant dynamics, over a fixed time interval or infinite time, and with different types of performance indexes. Optimal control problems are in general very difficult to solve, except for linear systems with a quadratic performance index. These problems are well understood and produce control laws that have very interesting properties, such as excellent gain and phase margins at the system input breakpoint.

One of the key challenges in using optimal control theory is transforming frequency domain performance and stability requirements from classical control into time domain requirements. A multivariable optimal controller design using a quadratic performance index optimizes the design in the time domain. Satisfying frequency domain requirements such as bandwidth, noise sensitivity, etc., using the optimization performance index, is a challenge. Similarly, quantifying the degree of robustness required to overcome parameter uncertainties is not explicitly posed in the original optimal control problem setup.

The key to using optimal control theory is to develop a method to tune the design parameters to achieve the desired robustness, performance, and stability within the overall system. This is the main focus of the present and the next chapters. This chapter introduces optimal control theory, the linear quadratic regulator,

and the all-important Algebraic matrix Riccati Equation. We will discuss in details some of the excellent properties that optimal controllers produce, which makes them a favorite in many industrial control problems, especially in aerospace. We will also present optimal control design extensions from the regulator framework to command following problems. It is the command following challenge that is most common in the design of aerospace flight critical control systems.

4.2 Optimal Control and the Hamilton–Jacobi–Bellman Equation

The derivation of the Hamilton–Jacobi–Bellman (HJB) equation for optimal control problems will allow us to understand how optimal control regulator problems are posed, and how we can form an optimal control from a performance index minimization problem. Optimal control problems are in general very difficult to solve. There are many books available on the subject. Athans and Falb [1], Kwakernaak and Sivan [2], Anderson and Moore [3] are the three excellent textbooks that deal with necessary and sufficiency conditions, differentiability and continuity assumptions, problem setup, derivations, and solutions for most optimal control problems that can be solved analytically. We will begin by deriving the HJB partial differential equation in a general setting and will then focus on linear systems with quadratic performance indices. We shall only provide a generic framework for derivation of optimal control policies. Readers interested in details are referred to now-classical control textbooks, such as [1–3].

The Hamilton–Jacobi–Bellman Equation

We begin by considering a general dynamic nonlinear system in the form,

$$\dot{x} = f(x, u, t), \quad x(t_0) = x_0 \quad (4.1)$$

where $x \in R^{n_x}$ is the system state and $u \in R^{n_u}$ is the control input. The system starts at time t_0 , with the initial state x_0 . We suppose that $f(x, u, t)$ is continuously differentiable in all arguments. This assumption is sufficient for the initial value problem (4.1) to have the unique solution on a finite time interval [4]. We also assume that T is small enough to reside within the time interval, where the system solutions are defined. We are interested in “optimally” controlling the system dynamics, starting from x_0 and driving the system state to a designated location. The notion of optimality is defined through the integral cost performance index,

$$J = \int_{t_0}^T L(x(\tau), u(\tau), \tau) d\tau + S(x(T)) \quad (4.2)$$

evaluated along the system trajectories $x(t)$ due to applied control input $u(t)$. The instant cost $L(x, u, \tau)$ and the terminal cost $S(x(T))$ are defined as scalar

non-negative functions of their arguments. Essentially, the cost J is our cumulative measure of the overall efforts (controls) and the state energy spent to steer the system from its initial state x_0 to a neighborhood of the terminal manifold $S(x(T)) = 0$.

For example, if the system dynamics are scalar then we can utilize quadratic instant and terminal costs, $L(x, u, \tau) = x^2(\tau) + q u^2(\tau)$ and $S(x(T)) = w x^2(T)$, with positive weights (q, w) . In this case, $L(x, u, \tau)$ can be thought of as the instant kinetic energy of the system, while the terminal cost $S(x(T))$ measures how close we can drive the system state to the origin, in T seconds or less. So, by appropriately choosing the weights (q, w) , we can emphasize the importance of minimizing the kinetic energy spent, while regulating the system state to zero. Later on, we shall address optimal control problems with quadratic cost.

Given the system dynamics (4.1), the control challenge of interest is to find an optimal control policy u^* to minimize the cost index J , over the time interval $[t_0, T]$. When used in (4.1), the optimal control u^* produces the optimal state trajectory x^* over $[t_0, T]$. Clearly, the cost index J in (4.2) depends on the system initial state $x(t_0)$, the selected control policy $u(\cdot) = u(t)|_{t_0 \leq t \leq T}$, and on the initial time t_0 .

$$J = J(x(t_0), u(\cdot), t_0) \quad (4.3)$$

Let J^* denotes the optimal (minimum) cost, when using the optimal control policy u^* .

$$J^*(x_0, t_0) = \min_{u[t_0, T]} J(x_0, u, t_0) = \int_{t_0}^T L(x^*(\tau), u^*(\tau), \tau) d\tau + S(x^*(T)) \quad (4.4)$$

We see that the optimal performance index J^* is a function of the initial state x_0 and the initial time t_0 .

$$J^*(x_0, t_0) = \min_{u[t_0, T]} J(x(t_0), u(\cdot), t_0) = \min_{u[t_0, T]} \left[\int_{t_0}^T L(x, u, \tau) d\tau + S(x(T)) \right] \quad (4.5)$$

Suppose that we start the system at an arbitrary initial condition x and at time t . Then the optimal cost-to-go from x is

$$J^*(x, t) = \min_{u[t, T]} \left[\int_t^T L(x, u, \tau) d\tau + S(x(T)) \right] \quad (4.6)$$

We can break (4.6) into two integrals, from $[t, t_1]$ to $[t_1, T]$,

$$J^*(x, t) = \min_{u[t, T]} \left[\int_t^{t_1} L(x, u, \tau) d\tau + \int_{t_1}^T L(x, u, \tau) d\tau + S(x(T)) \right] \quad (4.7)$$

and then explicitly write the minimization operation over the two intervals.

$$J^*(x, t) = \min_{u[t, t_1]} \min_{u[t_1, T]} \left[\int_t^{t_1} L(x, u, \tau) d\tau + \int_{t_1}^T L(x, u, \tau) d\tau + S(x(T)) \right] \quad (4.8)$$

The main idea here is to divide the integral into time slices, and then at each slice choose the optimal control that minimizes the overall cost J . This argument leads to the principle of optimality, developed by Richard Ernest Bellman in the late 1950s. Here is the original formulation of the principle, as it appears in [5].

Principle of optimality. *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*

The principle of optimality tells us that the optimal cost-to-go from x at time t to a terminal state $x(T)$ can be computed by minimizing the sum of: (a) The cost-to-go from $x = x(t)$ to $x_1 = x(t_1)$; and (b) the optimal cost from x_1 onward. So formally speaking, we can move the second min operation inside the square brackets.

$$J^*(x, t) = \min_{u[t, t_1]} \left[\int_t^{t_1} L(x, u, \tau) d\tau + \underbrace{\min_{u[t_1, T]} \int_{t_1}^T L(x, u, \tau) d\tau + S(x(T))}_{J^*(x_1, t_1)} \right] \quad (4.9)$$

Now, we can see that inside the brackets, the second integral is the optimal cost-to-go from x_1 to $x(T)$.

$$J^*(x, t) = \min_{u[t, t_1]} \left[\int_t^{t_1} L(x, u, \tau) d\tau + J^*(x_1, t_1) \right] \quad (4.10)$$

We define $t_1 = t + \Delta t$ and substitute it into (4.10).

$$J^*(x, t) = \min_{u[t, t+\Delta t]} \left[\int_t^{t+\Delta t} L(x, u, \tau) d\tau + J^*(x(t + \Delta t), t + \Delta t) \right] \quad (4.11)$$

Assuming that all functions are smooth, we can expand the right-hand side of (4.11) in a Taylor series,

$$J^*(x(t), t) = \min_{u[t, t+\Delta t]} \left[L(x, u, \tau) \Delta t + J^*(x, t) + \left(\frac{\partial J^*(x, t)}{\partial x} \right) \Delta x + \underbrace{\frac{\partial J^*(x, t)}{\partial t} \Delta t}_{\text{H.O.T.}} + O(\Delta t^2) \right] \quad (4.12)$$

where $O(\Delta t^2)$ denotes high-order terms (H.O.T.) in the Taylor series expansion. Here we define

$$\frac{\partial J^*}{\partial x}(x, t) = \begin{bmatrix} \frac{\partial J^*}{\partial x_1} & \dots & \frac{\partial J^*}{\partial x_n} \end{bmatrix} \in R^{1 \times n_x} \quad (4.13)$$

as a row vector. We will denote the transpose as $\nabla_x J^*(x, t)$ a column vector. We can cancel $J^*(x, t)$ on each side, since it does not depend on $u[t, t + \Delta t]$, divide both sides by Δt , and get:

$$0 = \min_{u[t, t+\Delta t]} \left[L(x, u, \tau) + \frac{\partial J^*(x, t)}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial J^*(x, t)}{\partial t} + O(\Delta t) \right] \quad (4.14)$$

Letting $\Delta t \rightarrow 0$, gives

$$-\frac{\partial J^*(x, t)}{\partial t} = \min_u \left[L(x(t), u, \tau(t)) + \frac{\partial J^*(x, t)}{\partial x} \underbrace{\dot{x}(t)}_{f(x, u, t)} \right] \quad (4.15)$$

where the system state is defined by the original system dynamics (4.1). We now introduce the Hamiltonian

$$H(x, \nabla_x J^*(x, t), u, t) = L(x, u, \tau) + \frac{\partial J^*(x, t)}{\partial x} f(x, u, t). \quad (4.16)$$

and rewrite (4.15) as

$$-\frac{\partial J^*(x, t)}{\partial t} = \min_u H(x, \nabla_x J^*(x, t), u, t) \quad (4.17)$$

Due to the assumed smoothness of all the functions, it follows that to minimize H with respect to the control u , we can compute the function gradient, and then equate it to zero.

$$\nabla H_u(x, \nabla_x J^*(x, t), u, t) = 0 \quad (4.18)$$

that is, every component of the gradient vector ∇H_u must vanish at the point of optimum. In addition, one needs to check if the derivative of the gradient (a matrix)

$\frac{\partial^2 H}{\partial u^2}$ is positive semidefinite, which would indicate that the point of optimum is the true minima of the Hamiltonian. This inequality is known as the Legendre–Clebsch condition.

Formulation (4.18) allows the functional minimization problem, such as (4.4), to be transformed into a function minimization, which can be solved using ordinary calculus. Let

$$H^*(x, \nabla_x J^*(x, t), t) \equiv \min_u [H(x, \nabla_x J^*(x, t), u, t)] \quad (4.19)$$

If we can solve (4.18) for the optimal control $u = u^*$, and substitute it back into (4.17), we get the Hamilton–Jacobi–Bellman (HJB) partial differential equation (PDE), whose solution is the optimal cost $J^*(x(t), t)$.

$$-\frac{\partial J^*(x, t)}{\partial t} = H^*(x, \nabla_x J^*(x, t), t) \quad (4.20)$$

We need a boundary condition for (4.20) to be well-posed. Setting $t_0 = T$ in the cost index (4.2), yields

$$J^*(x(T), T) = S(x(T)) \quad (4.21)$$

Using Bellman's principle of optimality, we have arrived at sufficient conditions for optimal control solution to exist. The latter is defined by the HJB Eq. (4.20) together with its boundary condition (4.21). The sufficiency of (4.20)–(4.21) for control optimality means that if we can solve the former for $J^*(x, t)$ and calculate $u^*(t)$, then the latter constitutes the optimal control policy for the system (4.1), with respect to the cost index (4.2).

In most optimal control problems, we would be interested in the calculation of the optimal control policy u^* , rather than the optimal cost J^* . Solving (4.20) is still quite difficult, even for low-order problems, in that we still must solve a PDE for the cost function $J^*(x, t)$.

As derived, the optimal policy $u^*(t)$ represents an open-loop control strategy, in the sense that u^* is computed as a function of time t . For practical applications, we would really want a feedback control policy, such as $u^* = u^*(x)$, to enforce robustness and reduce sensitivity of the solution to uncertainties that may exist in the system dynamics. We will see that if the system dynamics are linear, and the performance index penalty function $L(x, u, \tau)$ is quadratic, then the problem is easily solved, and the resulting optimal feedback control policy and the closed-loop system have very useful properties, with formal assurances of stability, performance, and robustness. In the forthcoming chapters, we shall derive and exploit these properties in our use of optimal control to maximize performance and robustness, while minimizing the control effort.

Summary

Dynamics: $\dot{x} = f(x, u, t) \quad x(t_0) = x_0$
 Performance index: $J(x, u, t) = \int_t^T L(x, u, \tau) d\tau + S(x(T))$
 Optimal Cost: $J^*(x, t) = \min_{u[t, T]} [J]$
 Hamiltonian: $H(x, u, t) = L(x, u, t) + \frac{\partial J^*(x, t)}{\partial x} f(x, u)$
 Optimal control: $\nabla H_u(x, u, t) = 0 \Rightarrow \boxed{u^*(t)} \Rightarrow H^*(x, u^*, t)$
 HJB-Equation: $\begin{cases} -\frac{\partial J^*(x, t)}{\partial t} = H^*(x, \nabla_x J^*(x, t), t) \\ J^*(x(T), T) = S(x(T)) \end{cases}$

Example 4.1 In this example we will set up (but not solve) the HJB equation. Consider the system,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2x_1 - 3x_2 + u \end{aligned} \tag{4.22}$$

where $x(0) = [1 \ 2]^T$, with the performance index

$$J = \int_0^1 (x_1^4 + u^2) dt + x_1^2(1) + x_2^2(1) \tag{4.23}$$

For this problem $l(x, u, t) = x_1^4 + u^2$ and $S(x(T)) = x_1^2(1) + x_2^2(1) = x^T(1)x(1)$. The Hamiltonian is

$$\begin{aligned} H(x, u, \nabla_x J^*, t) &= L(x, u, t) + \frac{\partial J^*}{\partial x} f(x, u) \\ &= x_1^4 + u^2 + \frac{\partial J^*}{\partial x_1} \dot{x}_1 + \frac{\partial J^*}{\partial x_2} \dot{x}_2 \\ &= x_1^4 + u^2 + \frac{\partial J^*}{\partial x_1} (x_2) + \frac{\partial J^*}{\partial x_2} (-2x_1 - 3x_2 + u) \end{aligned} \tag{4.24}$$

Now, we minimize the Hamiltonian by differentiating the right-hand side of (4.24) with respect to the control and then equating the resulting derivative to zero. Thus,

$$\left(\nabla H_u = 0 = 2u^* + \frac{\partial J^*}{\partial x_2} \right) \Rightarrow \left(u^* = -\frac{1}{2} \frac{\partial J^*}{\partial x_2} \right) \tag{4.25}$$

Substituting the optimal solution back into (4.24), we get

$$H^*(x, \nabla_x J^*, t) = x_1^4 + \frac{1}{4} \left(\frac{\partial J^*}{\partial x_2} \right)^2 + \frac{\partial J^*}{\partial x_1} x_2 - 2 \frac{\partial J^*}{\partial x_2} x_1 - 3 \frac{\partial J^*}{\partial x_2} x_2 - \frac{1}{2} \left(\frac{\partial J^*}{\partial x_2} \right)^2 \quad (4.26)$$

The HJB equation is then,

$$\begin{aligned} -\frac{\partial J^*}{\partial t} &= H^*(x, \nabla_x J^*(x, t), t) = x_1^4 + \frac{1}{4} \left(\frac{\partial J^*}{\partial x_2} \right)^2 \\ &\quad + \frac{\partial J^*}{\partial x_1} x_2 - \frac{\partial J^*}{\partial x_2} \left(2x_1 - 3x_2 - \frac{1}{2} \right) \end{aligned} \quad (4.27)$$

with the boundary condition $J^*(x, T) = x_1^2(T) + x_2^2(T)$. ■

The basic problem of optimal control design consists of choosing a control policy such that some measure of performance for the corresponding closed-loop system is optimized with respect to the selected performance criterion.

4.2.1 The HJB Equation for Nonlinear Systems Affine in Control

In this section, we consider a special class of systems whose dynamics are linear (affine) in control and have the following form:

$$\dot{x}(t) = f(x(t)) + g(x(t)) u(t) \quad (4.28)$$

where $x \in R^{n_x}$ is the n_x -dimensional state vector, $u \in R^{n_u}$ is the n_u -dimensional control input, $f(x) \in R^{n_x}$ is the n_x -dimensional vector function of the system dynamics, and $g(x) \in R^{n_x} \times R^{n_u}$ is the $(n_x \times n_u)$ -dimensional input matrix. Both functions $f(x)$ and $g(x)$ are assumed to be locally Lipschitz in x . Any piece-wise continuous control input u is admissible. These conditions are sufficient to ensure that for any initial condition

$$x(t_0) = x_0 \quad (4.29)$$

the system dynamics (4.28) have the unique trajectory solution that starts at x_0 . Since the system is autonomous then without a loss of generality the initial time can be set to zero $t_0 = 0$.

The system performance cost criterion (4.2), or equivalently the cost function, is defined here as

$$J(x_0, u(\cdot)) = \int_0^\infty \left(l(x(t)) + u(t)^T R u(t) \right) dt \quad (4.30)$$

The performance cost depends on the system initial state x_0 and on the selected control policy $u(\cdot)$. The scalar function $l(x)$ designates the instantaneous state cost and as such, it is chosen to be positive semidefinite, while R is an $(m \times m)$ symmetric positive strictly definite matrix of control weights.

The optimal control problem consists of choosing the control policy $u^*(\cdot)$ such that the cost function (4.30) attains its minimum on the corresponding closed-loop system trajectory $x^*(t)$ that starts at x_0 . The resulting optimal cost is

$$J^*(x_0) = \min_u \int_0^\infty (l(x(t)) + u(t)^T R u(t)) dt = \int_0^\infty (l(x^*(t)) + u^*(t)^T R u^*(t)) dt \quad (4.31)$$

The control policy $u^*(\cdot)$, if it exists, is said to be optimal with respect to the performance cost (4.30). In this case, $x^*(t)$ denotes the corresponding optimal trajectory. The solution to the optimal control problem can be obtained via the “principle of optimality” discussed in the previous section.

Consider the so-called cost-to-go function

$$V(x(t)) = \min_u \left\{ \int_t^\infty (l(x(\tau)) + u(\tau)^T R u(\tau)) d\tau \right\} \quad (4.32)$$

This is the optimal performance cost associated with the initial condition $x(t)$. Let $\Delta t > 0$ be a constant. Applying Bellman's principle of optimality gives the functional equation.

$$\begin{aligned} V(x(t)) &= \min_u \left\{ \int_t^{t+\Delta t} (l(x(\tau)) + u(\tau)^T R u(\tau)) d\tau + \int_{t+\Delta t}^\infty (l(x(\tau)) + u(\tau)^T R u(\tau)) d\tau \right\} \\ &= \min_u \left\{ \int_t^{t+\Delta t} (l(x(\tau)) + u(\tau)^T R u(\tau)) d\tau + V(x(t + \Delta t)) \right\} \end{aligned} \quad (4.33)$$

Using Taylor series expansion results in

$$V(x(t)) = \min_u \left\{ (l(x(t)) + u(t)^T R u(t)) \Delta t + V(x(t)) + \underbrace{\left(\frac{\partial V(x(t))}{\partial x} \right)^T}_{\nabla V(x(t))} \underbrace{\frac{dx(t)}{dt}}_{(f(x(t)) + g(x(t))u)} \Delta t + O(\Delta t^2) \right\} \quad (4.34)$$

where $O(\Delta t^2)$ denotes the high-order terms associated with the Taylor series expansions and $\nabla V(x(t))$ is the gradient of $V(x)$ (a column vector) with respect

to x , evaluated at $x(t)$. Since $V(x(t))$ is independent of u and it appears on both sides of the Eq. (4.34), this term can be eliminated.

$$0 = \min_u \left\{ \left(l(x(t)) + u(t)^T R u(t) \right) \Delta t + (\nabla V(x(t)))^T (f(x(t)) + g(x(t))u) \Delta t + O(\Delta t)^2 \right\} \quad (4.35)$$

Dividing both sides of (4.35) by Δt , and letting $\Delta t \rightarrow 0$, gives

$$0 = \min_u \left\{ \left(l(x(t)) + u(t)^T R u(t) \right) + (\nabla V(x(t)))^T (f(x(t)) + g(x(t))u) \right\} \quad (4.36)$$

Since (4.36) is valid point-wise for any $x(t)$ and the system (4.28) is autonomous, the time dependence can be dropped.

$$0 = \min_u \left\{ \left(l(x) + u^T R u \right) + (\nabla V(x))^T (f(x) + g(x)u) \right\} \quad (4.37)$$

Equation (4.37) implies that partial derivatives with respect to u of the function inside the min brackets must vanish at the point where the minimum is achieved.

$$2u^T R + (\nabla V(x))^T g(x) = 0 \quad (4.38)$$

Solving (4.38) for u gives the optimal feedback control strategy.

$$u^*(x) = -\frac{1}{2} R^{-1} g^T(x) \nabla V(x) \quad (4.39)$$

This is a necessary condition for a minimum to occur. A sufficient condition requires that the second derivative of (4.37) with respect to control u is strictly positive, which is the case here, since $R = R^T > 0$ is symmetric and positive definite by the definition. Consequently, (4.39) is the minimizing solution for the optimal cost performance index (4.32).

Note that the cost-to-go function $V(x)$ is yet to be determined. Substituting (4.39) into (4.37), results in

$$l(x) + (\nabla V(x))^T f(x) - \frac{1}{4} (\nabla V(x))^T g(x) R^{-1} g^T(x) \nabla V(x) = 0 \quad (4.40)$$

Equation (4.40) is the HJB equation for the selected cost function. Its solution results in the optimal control feedback (4.39). Also from (4.31), it follows that the optimal performance cost can be computed from the cost-to-go performance function.

$$\begin{aligned} J^*(x_0) &= \min_u \int_0^\infty \left(l(x(t)) + u(t)^T R u(t) \right) dt \\ &= \int_0^\infty \left(l(x^*(t)) + u^*(t)^T R u^*(t) \right) dt = V(x^*(0)) = V(x_0) \end{aligned} \quad (4.41)$$

If u^* is the optimal control solution then

$$l(x) + (u^*)^T R u^* + (\nabla V(x))^T (f(x) + g(x) u^*) = 0 \quad (4.42)$$

On the other hand,

$$\dot{V}(x) = (\nabla V(x))^T \dot{x} = (\nabla V(x))^T (f(x) + g(x) u^*) \quad (4.43)$$

Combining (4.42) and (4.43) gives

$$\dot{V}(x) = -l(x) - (u^*)^T R u^* < 0 \quad (4.44)$$

Therefore, if a positive-definite solution of the HJB Eq. (4.40) exists, i.e., if $V(x) > 0$ globally for all $x \in R^n$, then the optimal cost-to-go function becomes a Lyapunov function for the closed-loop system

$$\dot{x} = f(x) - \frac{1}{2} g(x) R^{-1} g^T(x) (\nabla V(x))^T \quad (4.45)$$

These dynamics are obtained by substituting the optimal control (4.39) into the original open-loop system (4.28).

Suppose that the origin is the open-loop system equilibrium, that is let $f(x) = 0$. In this case, we can apply Lyapunov Direct method to conclude that the origin $x^* = 0$ is the globally asymptotically stable equilibrium of the closed-loop system (4.45), i.e., starting from any initial condition x_0 , $\lim_{t \rightarrow \infty} \|x(t)\| = 0$.

As shown in (4.41), once the cost-to-go function $V(x)$ is determined, the total optimal cost associated with the optimal control in (4.39) can be readily computed.

$$J^*(x_0) = V(x_0) \quad (4.46)$$

To summarize, the following steps should be followed when solving the optimal control problem (4.28), (4.30).

1. Find (if possible) a positive-definite solution $V(x)$ of the HJB Eq. (4.40).
2. Compute the optimal control feedback u^* as in (4.39) and the associated optimal cost (4.46).
3. Compose, analyze for stability and simulate the closed-loop system (4.45).

Example 4.2 Optimal Control Design for a First-Order Nonlinear System with Quadratic Cost Consider the scalar nonlinear dynamical system.

$$\dot{x} = \underbrace{a(x)x}_f + u, \quad x(0) = x_0 \quad (4.47)$$

Similar to (4.85), a quadratic cost function can be selected as the system optimality performance cost index.

$$J(x(0), u(\cdot)) = \int_0^\infty (x^2(t) + r u^2(t)) dt \quad (4.48)$$

In (4.48), r is a positive constant control weight. Using (4.39), the optimal control solution is

$$u^* = -\frac{1}{2r} \frac{\partial V(x)}{\partial x} \quad (4.49)$$

The cost-to-go function $V(x)$ must satisfy the HJB Eq. (4.40)

$$x^2 + \frac{\partial V(x)}{\partial x} a(x) x - \frac{1}{4r} \left[\frac{\partial V(x)}{\partial x} \right]^2 = 0 \quad (4.50)$$

or, equivalently

$$\left[\frac{\partial V(x)}{\partial x} \right]^2 - 4r \frac{\partial V(x)}{\partial x} a(x) x - 4r x^2 = 0 \quad (4.51)$$

Using mean-value theorem, the derivative $\frac{\partial V(x)}{\partial x}$ can be written.

$$\frac{\partial V(x)}{\partial x} = \underbrace{\left(\int_0^1 \frac{\partial^2 V(sx)}{(\partial x)^2} ds \right)}_{p(x)} x = p(x)x \quad (4.52)$$

Thus, the two solutions of (4.51) are in the form

$$\frac{\partial V(x)}{\partial x} = p(x)x \quad (4.53)$$

Substituting (4.53) into (4.51) yields the second-order equation.

$$p^2(x) - 4r a(x) p(x) - 4r = 0 \quad (4.54)$$

It has two roots.

$$p(x) = 2r \left(a(x) \pm \sqrt{a^2(x) + \frac{1}{r}} \right) \quad (4.55)$$

The positive solution of (4.55) is selected.

$$p(x) = 2r \left(a(x) + \sqrt{a^2(x) + \frac{1}{r}} \right) > 0, \quad \forall x \quad (4.56)$$

This selection implies that the gradient of $V(x)$ with respect to x is a monotonically increasing function of its argument

$$\frac{\partial V(x)}{\partial x} = 2r \left(a(x) + \sqrt{a^2(x) + \frac{1}{r}} \right) x \quad (4.57)$$

It is also clear that $\frac{\partial V(x)}{\partial x}$ is globally positive definite. Integration yields the cost-to-go function:

$$V(x) = 2r \int_0^x \left\{ \left(a(z) + \sqrt{a^2(z) + \frac{1}{r}} \right) z \right\} dz \quad (4.58)$$

Based on (4.57), it is easy to see that $V(0) = 0$ and that $V(x)$ is positive definite and radially unbounded function of x . It is possible to show that $V(x)$ is the Lyapunov function for the corresponding closed-loop system. In order to prove the latter, we substitute the optimal control solution

$$u^* = -\frac{1}{2r} \frac{\partial V(x)}{\partial x} = -\left(a(x) + \sqrt{a^2(x) + \frac{1}{r}} \right) x \quad (4.59)$$

into the system dynamics (4.47) and derive the closed-loop system.

$$\dot{x} = -\left(\sqrt{a^2(x) + \frac{1}{r}} \right) x \quad (4.60)$$

Using (4.44), the time derivative of $V(x)$ along the system (4.60) trajectories can be calculated.

$$\dot{V}(x) = -l(x) - (u^*)^T R u^* = -x^2 - r (u^*)^2 \leq 0 \quad (4.61)$$

Since $V(x)$ is positive definite, radially unbounded, and its time derivative is strictly negative for all $x \neq 0$, the closed-loop system (4.60) is globally asymptotically stable. Hence, starting from any initial condition $x(0)$, all system trajectories asymptotically converge to the origin. In other words, the optimal control feedback (4.59) globally stabilizes the original system (4.47).

It is of interest to compare the optimal control strategy to the dynamic inversion (DI) controller. The DI control design consists of subtracting out the open-loop system dynamics and assigning the desired closed-loop dynamics instead. Often the

latter is defined as a linear-in-state expression. For (4.47), a DI controller can be selected in the form,

$$u_{\text{DI}} = -(a(x) + \bar{a})x \quad (4.62)$$

In (4.62), \bar{a} is a positive constant. Under the DI feedback, the corresponding closed-loop system dynamics become linear and globally exponentially stable.

$$\dot{x} = -\bar{a}x \quad (4.63)$$

Comparing the optimal closed-loop dynamics (4.60)–(4.63), it is obvious that if one chooses the control weight

$$0 < r \leq \frac{1}{\bar{a}^2} \quad (4.64)$$

then the optimal controller enforces a convergence rate that is the same or better than its DI counterpart. This fact can be proven by applying the Gronwall–Bellman lemma to:

$$\dot{x} = -\left(\sqrt{a^2(x) + \frac{1}{r}}\right)x \leq -\left(\sqrt{\frac{1}{r}}\right)x \leq -\bar{a}x \quad (4.65)$$

Basically, the inequality (4.64) requires the control weight to be small enough in order to allow for a large optimal control feedback, which in turn provides the required rate of exponential convergence of the system trajectories to the origin.

The next example analyzes and compares closed-loop performance characteristics of the optimal and the DI state feedback controllers.

Example 4.3 Optimal Control Solution versus Dynamic Inversion Control for a First-Order Nonlinear System with Quadratic Cost Consider the scalar nonlinear dynamical system.

$$\dot{x} = x^2 + u \quad (4.66)$$

Here $f(x) = x$. Choose $a = r = 1$. Then according to (4.59), the optimal controller can be written as

$$u^* = -\left(x + \sqrt{x^2 + 1}\right)x \quad (4.67)$$

At the same time, a DI feedback (4.62) can be selected to assign the desired closed-loop linear stable behavior, with the unity time constant.

$$u_{\text{DI}} = -(x + 1)x \quad (4.68)$$

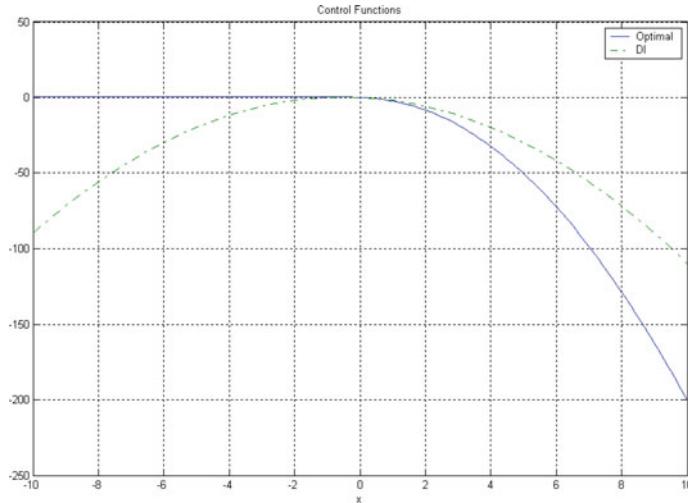


Fig. 4.1 Optimal control policy versus dynamic inversion-based control

Both controllers globally stabilize the system state in (4.66). Figure 4.1 is the key to understanding the advantages of the optimal controller versus the DI strategy.

What is very important to note here is the fact that the optimal control strategy does not cancel the nonlinearity x^2 in the region where it is beneficial from the closed-loop stability point of view. In order to enforce closed-loop stability, the optimal control dominates x^2 function in the $x > 0$ region, while it makes minimum changes to the function in the $x < 0$ region, where the parabola already contributes to the closed-loop system stability.

Starting from a negative initial condition $x(0) = -3$, the closed-loop system performance and the control time histories are shown on Figs. 4.2 and 4.3, correspondingly.

These data show that when compared to the DI feedback, the optimal strategy provides faster stabilization with a lesser control effort.

Changing the initial conditions from negative to positive $x(0) = 3$ results in a similar trend shown in Figs. 4.4 and 4.5.

These data indicate that the optimal controller starts initially with a larger value than the DI feedback, but it quickly reduces its signal, while stabilizing the system faster than the DI control. Once again, the optimal control provided faster convergence with a lesser overall control effort.

In order to assess robustness properties of the two control strategies, we introduce a small constant uncertainty $\varepsilon > 0$ into the system dynamics.

$$\dot{x} = (1 + \varepsilon)x^2 + u \quad (4.69)$$

It can be theoretically proven that for any $\varepsilon > 0$ there exists an initial condition $x(0)$ such that the DI control would destabilize the system dynamics. In other words,

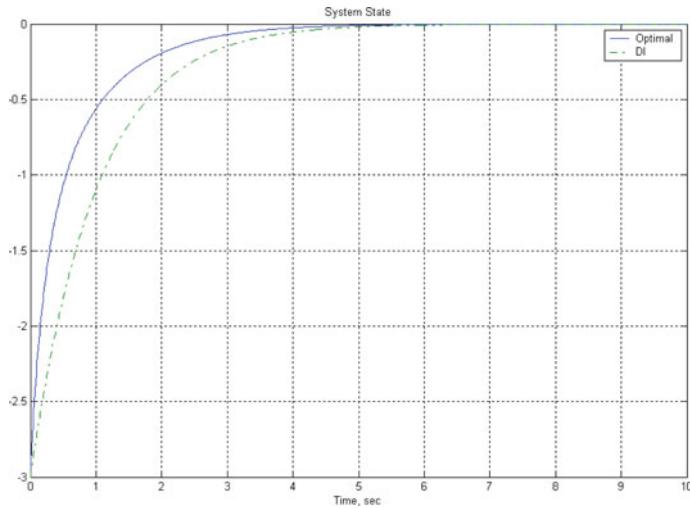


Fig. 4.2 State trajectories under optimal and dynamics inversion controls starting from negative initial condition

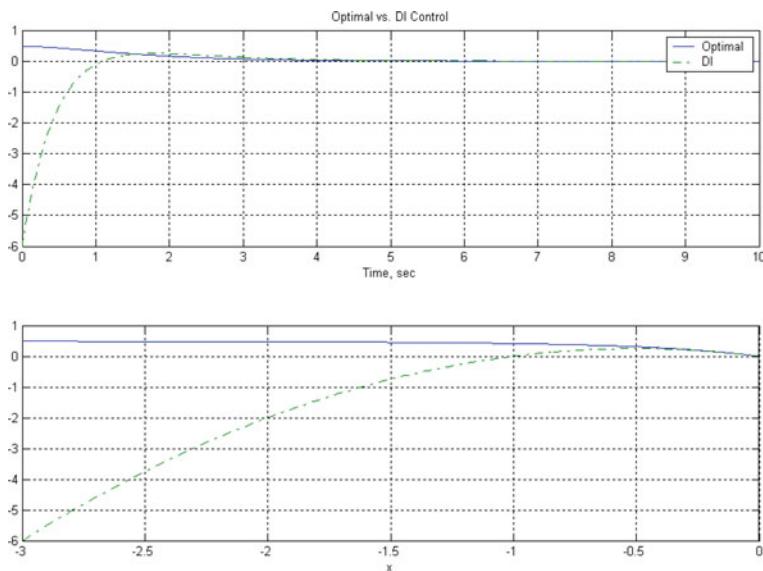


Fig. 4.3 Optimal and dynamics inversion controls starting from negative initial conditions

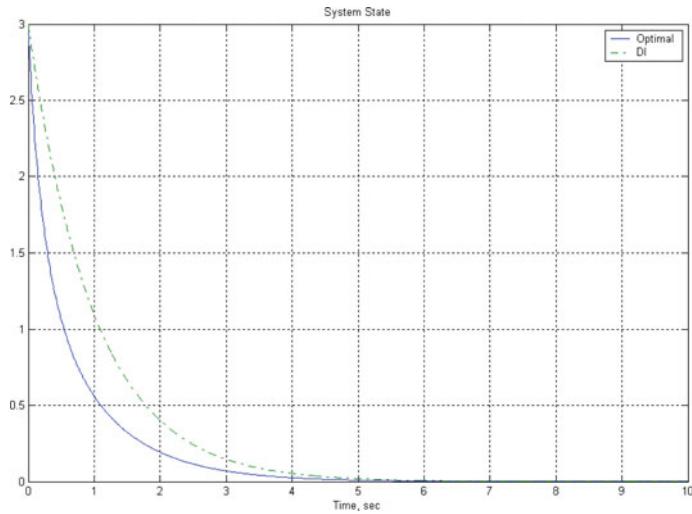


Fig. 4.4 State trajectories under optimal and dynamics inversion controls starting from positive initial condition

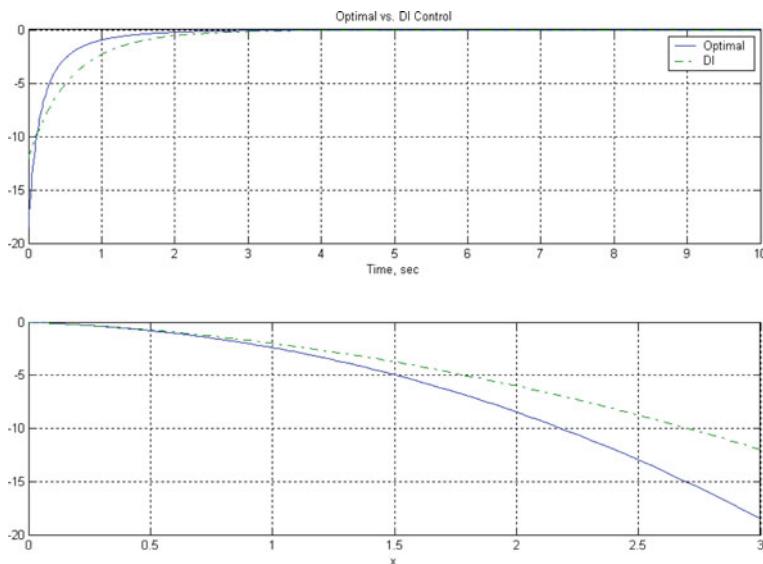


Fig. 4.5 Optimal and dynamics inversion controls starting from positive initial conditions

the DI feedback has no robustness margin due to changes in the system dynamics. On the other hand, the optimal controller (dominates rather than cancels) provides a set of finite robustness bounds with respect to the system uncertainty in (4.69). So, the optimal control feedback is superior to the DI feedback in providing: (a) faster rate of stabilization; (b) minimum control effort; and (c) robustness to system uncertainties. ■

4.3 Linear Quadratic Regulator (LQR)

The linear quadratic regulator (LQR) is one of the most widely used control design methods in industrial applications and particularly in aerospace. Trade studies have been performed comparing properties of various controllers (performance, robustness, control usage) in many different applications. For example, we have found that flight control systems designed using the LQR method have excellent performance, robustness, and minimize the control usage. This method is easily extended (in the next section) to produce command tracking controllers.

Consider the linear non-autonomous system,

$$\dot{x} = A(t)x + B(t)u \quad x(t_0) = x_0 \quad x \in R^{n_x}, u \in R^{n_u} \quad (4.70)$$

with the quadratic performance index,

$$J = \int_{t_0}^T (x^T Q x + u^T R u) d\tau + x^T(T) Q_T x(T) \quad (4.71)$$

where the cost weight matrices (Q, R, Q_T) are symmetric positive semidefinite, positive definite, and positive semidefinite, respectively.

$$Q = Q^T \geq 0, \quad R = R^T > 0, \quad Q_T = Q_T^T \geq 0 \quad (4.72)$$

Following (4.16), the LQR Hamiltonian is

$$H = x^T Q x + u^T R u + \frac{\partial J^*}{\partial x} (A(t)x + B(t)u) \quad (4.73)$$

Taking the gradient of H with respect to u and equating it to zero produces

$$\frac{\partial H}{\partial u} = 2R u + B^T \nabla_x J^*(x, t) = 0 \quad (4.74)$$

where the optimal control is

$$u^* = -\frac{1}{2} R^{-1} B^T \nabla_x J^*(x, t) \quad (4.75)$$

Substituting u^* back into (4.20) yields the HJB equation

$$-\frac{\partial J^*}{\partial t} = x^T Qx - \frac{1}{4} \frac{\partial J^*}{\partial x} B R^{-1} B^T \nabla_x J^* + \frac{\partial J^*}{\partial x} Ax \quad (4.76)$$

which is still quite difficult to solve directly. Fortunately, one can show that the optimal cost J^* is a quadratic time-varying function of the system state [3, Sect. 2.3].

$$J^* = J^*(x(t), t) = x(t)^T P(t) x(t) \quad (4.77)$$

where $P(t) = P^T(t) > 0$. Substituting (4.77) into (4.76), we get

$$\frac{\partial J^*(x, t)}{\partial t} = x^T \dot{P}(t)x \quad \nabla_x J^*(x, t) = 2 P(t)x \quad (4.78)$$

Substituting (4.78) back into (4.76), and factoring out x on both sides, results in,

$$x^T \left[-\dot{P}(t) - P(t)A - A^T P(t) - Q + P(t)B R^{-1} B^T P(t) \right] x = 0 \quad (4.79)$$

with the boundary condition $P(T) = Q_T$. Since this must be satisfied for any state x , the following initial value problem must be true

$$-\dot{P}(t) = P(t)A + A^T P(t) + Q - P(t)B R^{-1} B^T P(t), \quad P(T) = Q_T \quad (4.80)$$

The time-varying matrix ordinary differential equation in (4.80) is called the Riccati equation. Substituting (4.78) into (4.75) yields the optimal control policy

$$u^*(x, t) = -\frac{1}{2} R^{-1} B^T \nabla_x J^* = -\underbrace{R^{-1} B^T P(t)}_{K(t)} x = -K(t)x \quad (4.81)$$

in state feedback form. Note that in (4.80), the Riccati equation is integrated backward in time. Then, the optimal control $u^*(x, t)$ with the feedback gains $K(t)$ are formed using (4.81). For real-time operations, these gains must be precomputed and stored in a lookup table, and the feedback control law would be implemented by looking up the gains in the table. This process is called “gain scheduling”.

Continuing on, we substitute the optimal control (4.81) into the system dynamics (4.70) and obtain the closed-loop system.

$$\begin{aligned} \dot{x} &= A(t)x + B(t)u & x(t_0) &= x_0 \\ u(t) &= -K(t)x \\ \dot{x} &= (A(t) - B(t)K(t))x \end{aligned} \quad (4.82)$$

Summary

Dynamics: $\dot{x} = A(t)x + B(t)u \quad x(t_0) = x_0$

Performance index: $J = \int_{t_0}^T (x^T Q x + u^T R u) d\tau + x^T(T) Q_T x(T)$

Riccati Equation:
$$\begin{cases} -\dot{P}(t) = P(t)A + A^T P(t) + Q - P(t)B R^{-1} B^T P(t) \\ P(T) = Q_T \end{cases}$$

Optimal control: $u^* = -R^{-1} B^T P(t)x = -K(t)x$

Closed-Loop system: $\dot{x} = (A(t) - B(t)K(t))x, \quad x(0) = x_0$

4.3.1 Infinite-Time LQR Problem

In this section, we consider the quadratic performance index on an infinite time interval,

$$J = \int_0^\infty (x^T Q x + u^T R u) d\tau \quad Q = Q^T \geq 0, \quad R = R^T > 0 \quad (4.83)$$

Using (4.39), the linear quadratic regulator (LQR) feedback solution can be calculated for any linear-time-invariant (LTI) system in the form,

$$\dot{x} = \underbrace{Ax}_{f(x)} + \underbrace{Bu}_{g(x)}, \quad x(0) = x_0 \quad (4.84)$$

with the following quadratic cost function

$$J(x_0, u(\cdot)) = \int_0^\infty \left(\underbrace{x^T(t)Qx(t)}_{l(x(t))} + u(t)^T R u(t) \right) dt \quad (4.85)$$

In (4.85), Q is a symmetric positive semidefinite matrix and R is a symmetric strictly positive-definite matrix. These matrices define the significance of state and control contributions in the cost function. To have a well-posed problem, we would require the pair (A, B) to be controllable and the pair $(A, Q^{1/2})$ to be observable. Weaker conditions, such as stabilizable (A, B) and detectable $(A, Q^{1/2})$, are also acceptable. The need for controllability of the system dynamics should be obvious. Clearly, the control cannot stabilize the system and perform as desired if the dynamics are not controllable. Detectability of modes through the performance index guarantees that the unstable modes are penalized, producing a control

that will minimize their contribution to the index. We will see that the numerical choices of the matrices Q and R are very important in achieving performance and robustness in the closed-loop system.

According to (4.40), the corresponding HJB equation becomes

$$x^T Q x + (\nabla V(x))^T A x - \frac{1}{4} (\nabla V(x))^T B R^{-1} B^T \nabla V(x) = 0 \quad (4.86)$$

The cost-to-go is postulated as a quadratic, with a symmetric positive-definite matrix P .

$$V(x) = x^T P x \quad (4.87)$$

In this case, the gradient (column vector) of $V(x)$ with respect to x is

$$\nabla V(x) = 2 P x \quad (4.88)$$

Substituting (4.88) into (4.86) gives

$$x^T Q x + \underbrace{2 x^T P A x}_{x^T (P A + A^T P) x} - x^T P B R^{-1} B^T P x = 0 \quad (4.89)$$

or, equivalently

$$x^T (Q + P A + A^T P - P B R^{-1} B^T P) x = 0 \quad (4.90)$$

Since the algebraic equation in (4.90) must be valid for all vectors x , the matrix P must satisfy the Algebraic Riccati Equation (ARE).

$$P A + A^T P - P B R^{-1} B^T P + Q = 0 \quad (4.91)$$

Define $Q = C C^T$. It turns out that if the matrix pair (A, B) is controllable (stabilizable) and (A, C) is observable (detectable) then the ARE (4.91) has the unique symmetric positive-definite solution P . Using (4.39), the corresponding LQR-optimal linear feedback control policy can be derived.

$$u^* = -\underbrace{R^{-1} B^T P}_{K_x} x = -K_x x \quad (4.92)$$

In (4.92), the feedback gain K_x is often referred to as the steady-state optimal Kalman gain matrix. The total cost associated with the optimal control strategy (4.92) is computed based on (4.46)

$$J^*(x(0)) = x^T(0) P x(0) \quad (4.93)$$

The fundamental property of the LQR control solution is the guaranteed closed-loop global exponential stability. This property can be proven by rewriting the ARE (4.91) in the form of the algebraic Lyapunov equation (ALE).

$$P \underbrace{\left(A - B \underbrace{R^{-1} B^T P}_{K} \right)}_{A_{cl}} + \underbrace{\left(A - B \underbrace{R^{-1} B^T P}_{K} \right)^T}_{A_{cl}} P = - \underbrace{\left(Q + P B R^{-1} B^T P \right)}_{\tilde{Q}} \quad (4.94)$$

So, the LQR-driven closed-loop system matrix A_{cl} satisfies the ALE,

$$P A_{cl} + A_{cl}^T P = -\tilde{Q} \quad (4.95)$$

with the symmetric positive-definite ARE solution P and the symmetric positive semidefinite weight matrix \tilde{Q} . According to the Lyapunov lemma [3], the closed-loop system matrix

$$A_{cl} = A - B \underbrace{R^{-1} B^T P}_{K_x} \quad (4.96)$$

is Hurwitz. Therefore, the origin is the globally exponentially stable equilibrium of the closed-loop system.

In industrial applications with nonlinear process control, the system models are often linearized at the designated operating conditions. Then, LQR-based controllers can be designed at each operating point. The resulting constant feedback gains K_x would be stored in a table and recalled (looked up) in real time for implementation. As we have already mentioned, this is the gain-scheduling control concept.

Substituting the optimal feedback control (4.92) into the open-loop dynamics (4.84) gives the closed-loop system.

$$\dot{x} = \underbrace{(A - BK_x)}_{A_{cl}} x = A_{cl} x \quad (4.97)$$

The LQR formulation guarantees the closed-loop system (4.97) to be exponentially stable (see (4.94) above). This means the eigenvalues of A_{cl} lie in the open left half complex plane, $\text{Re}(\lambda(A_{cl})) < 0$. In this case, the system state is regulated to zero, $x \rightarrow 0$ as $t \rightarrow \infty$, which implies $u \rightarrow 0$, as $t \rightarrow \infty$.

It is often desirable when simulating the dynamics to compute and examine the peak values of the optimal control u and its rate \dot{u} . This gives insights into the demands on the actuation system. If we differentiate u , and assume constant gains K_x , we get

$$\dot{u} = -K_x \dot{x} = -K_x (A - BK_x)x = -K_x A_{cl} x \quad (4.98)$$

We can form a closed-loop simulation model, with outputs x , u , and \dot{u} , as

$$\begin{aligned}\dot{x} &= Ax + Bu \quad u = -K_x x \\ \dot{x} &= (A - BK_x)x = A_{\text{cl}}x \\ y &= \begin{bmatrix} x \\ u \\ \dot{u} \end{bmatrix} = \begin{bmatrix} I_{n_x} \\ -K_x \\ -K_x A_{\text{cl}} \end{bmatrix} x\end{aligned}\tag{4.99}$$

In real-life applications, and especially in flight control applications, it is critical to prevent saturation of the control surface positions and rates. When this happens, nonlinear effects begin to dominate the system response, stability is no longer guaranteed, and the system could depart stable flight. We can see from (4.99) that large gains K_x may cause large control positions and rates. Thus, high gains are undesirable in most industrial control applications. From (4.92) we see that K_x gets large as P gets large. From (4.91) we see that it is the choice of the weights Q and R in the ARE that determines how large the optimal state feedback gains will become.

Summary

Dynamics: $\dot{x} = Ax + Bu$, $x(0) = x_0$

Performance index: $J = \int_0^\infty (x^T Q x + u^T R u) d\tau$

Algebraic Riccati Equation: $PA + A^T P - P B R^{-1} B^T P + Q = 0$

Optimal control: $u = -R^{-1} B^T P x = -K_x x$

Closed-Loop System: $\dot{x} = (A - BK_x)x$, $x(0) = x_0$

Simulation output: $y = \begin{bmatrix} x \\ u \\ \dot{u} \end{bmatrix} = \begin{bmatrix} I_{n_x} \\ -K_x \\ -K_x A_{\text{cl}} \end{bmatrix} x$

Example 4.4 In this second-order system example, we wish to solve for the optimal control and examine the properties of the closed-loop system. Consider the following linear-time-invariant model,

$$\dot{x} = Ax + Bu \quad A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.100}$$

with the performance index

$$J = \int_0^\infty (x_1^2 + ru^2) d\tau \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad R = r \tag{4.101}$$

The eigenvalues of the open-loop system are $\lambda = 0$ and $\lambda = -1$. In the performance index the state penalty matrix Q penalizes the first state of the system. The control penalty r is left as a parameter so we can see how small and large values of r change the closed-loop dynamics. It is always important to check if the design problem is well-posed. Conditions on the plant and on the performance index for a well-posed problem require to check if the unstable modes of the system are controllable, and if the unstable modes are observable through the state penalty matrix. In other words, we need to verify if (A, B) is stabilizable and $(A, Q^{1/2})$ is detectable. First, we compute the controllability matrix.

$$P_c = [B \ AB] = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}_{RK=2} \quad (4.102)$$

Since this matrix has full rank, the system is controllable. So, any unstable modes are controllable. Next, we can factor the state penalty matrix into square roots,

$$Q = (Q^{1/2})^T Q^{1/2} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.103)$$

and then check the observability using the square root of Q .

$$\begin{bmatrix} Q^{1/2} \\ Q^{1/2}A \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.104)$$

Since the observability matrix (4.104) has full rank, all modes of the system are observable through the penalty matrix. Now, we can solve the ARE,

$$PA + A^T P - P B R^{-1} B^T P + Q = 0 \quad (4.105)$$

for P , using A , B , Q , and $R = r$. Let $P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$. Then the ARE is

$$\begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} - \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{1}{r} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = 0 \quad (4.106)$$

Since the Riccati matrix P must be real symmetric and positive definite, from (4.106) we can derive three equations for p_1 , p_2 , and p_3 . These are

$$\begin{aligned} -p_2^2/r + 1 &= 0 \\ p_1 - p_2 - p_2 p_3/r &= 0 \\ 2(p_2 - p_3) - p_3^2/r &= 0 \end{aligned} \quad (4.107)$$

The first equation gives $p_2 = \sqrt{r}$ (both positive and negative values of m must be checked to see which is the solution). Using $p_2 = \sqrt{r}$, p_1 and p_3 are

$$\begin{aligned} p_3 &= r \left(\sqrt{1 + \frac{2}{\sqrt{r}}} - 1 \right) \\ p_1 &= \sqrt{r} \sqrt{1 + \frac{2}{\sqrt{r}}} \end{aligned} \quad (4.108)$$

The constant state feedback gain matrix is

$$K = R^{-1} B^T P = \left[\frac{2}{\sqrt{r}} \sqrt{1 + \frac{2}{\sqrt{r}}} - 1 \right] \quad (4.109)$$

The closed-loop state dynamics are always stable with characteristic equations

$$\phi_{cl}(s) = s^2 + s \sqrt{1 + \frac{2}{\sqrt{r}}} + \frac{1}{\sqrt{r}} \quad (4.110)$$

By varying the control penalty r in (4.110), we can compute a root locus (Fig. 4.6), to show how the numerical choice of R impacts poles of the closed-loop system dynamics.

The root locus data in Fig. 4.6 are the result of changing the control penalty r from 0.001 to 100. For large values of r (small gains), the closed-loop poles are near the

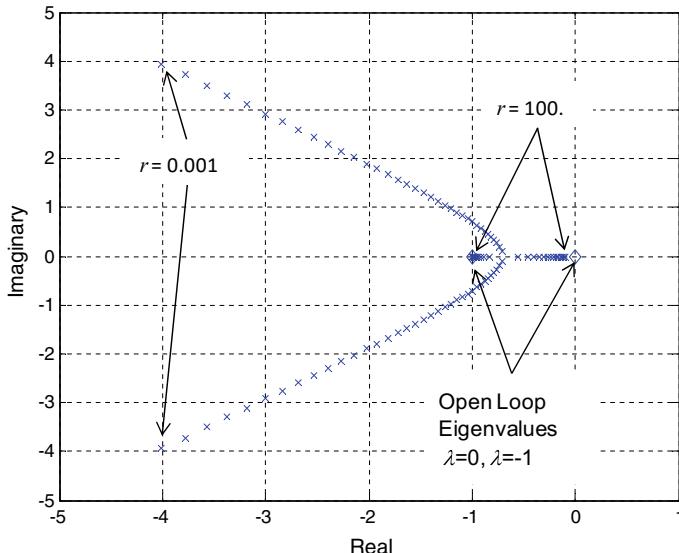


Fig. 4.6 Example 4.4 root locus varying the LQR control penalty parameter

open-loop poles ($r = 100$, $K = [0.1 \ 0.0954]$), producing a slow system response. For small values of r (large gains), the roots follow asymptotes into the left half plane, and the response gets fast ($r = 0.01$, $K = [31.6228 \ 7.0153]$). In general, the values of the optimal feedback gains are proportional to the relative magnitude of Q and R . For a fixed R , large values of Q heavily penalize the state (relative to the control), the resulting optimal feedback gains grow large, and the closed-loop system response gets fast. On the other hand, small values of Q penalize the control more than the state, resulting in smaller control efforts. This also keeps the gains small, producing a slower response. ■

4.3.2 Guaranteed Stability Robustness for State Feedback LQR

In this section, we shall analyze gain and phase stability margins of LQR state feedback controllers,

$$u = -K_x x \quad (4.111)$$

for the LTI dynamics (4.84).

As discussed previously in Sect. 3, gain and phase margins for LTI systems quantify relative stability, or equivalently the sensitivity of the closed-loop stable system with respect to gain and time-delay uncertainties that might be present at the system input break point. Standard requirements for any controller to be accepted in practical applications require a minimum of 6 dB gain margin and 45° phase margin at all control input break points in the system.

Next, we consider an LTI system operating under an LQR state feedback controller and compute the corresponding gain and phase margins that are derived based on the notation of the system loop gain transfer function $L_u(s)$ at the system input break point.

$$L_u(s) = K_x (s I_{n_x} - A)^{-1} B \quad (4.112)$$

As shown in Fig. 4.7, the system loop gain is an $(n_u \times n_u)$ —open-loop transfer function that defines the relationship between the injected signal u_{in} at the plant input and the resultant (returned) output signal u_{out} .

$$u_{\text{out}} = -K_x x = -\underbrace{K_x (s I_{n_x} - A)^{-1} B}_{L_u(s)} u_{\text{in}} = -L_u(s) u_{\text{in}} \quad (4.113)$$

The LQR state feedback gain $K_x \in R^{n_u \times n_x}$ is given by (4.92),

$$K_x = R^{-1} B^T P \quad (4.114)$$

Fig. 4.7 Loop gain computed at the system input break point

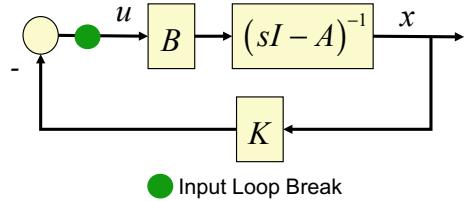
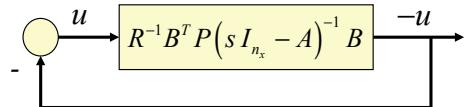


Fig. 4.8 LQR state feedback closed-loop block diagram



where $P \in R^{n_x \times n_x}$ is the unique symmetric strictly positive-definite solution of the ARE (4.91), with the state and control weight matrices $Q \in R^{n_x \times n_x}$ and $R \in R^{n_u \times n_u}$, selected such that $Q = Q^T \geq 0$ and $R = R^T > 0$. The LQR loop gain at the system input is

$$L_u(s) = R^{-1}B^T P(sI_{n_x} - A)^{-1}B \quad (4.115)$$

The corresponding closed-loop state feedback LQR block diagram is shown below in Fig. 4.8.

In robustness analysis, the system return difference matrix is of great importance. It defines flow transformation from the injected signal u_{in} to the difference between the injected and the returned signals.

$$u_{\text{in}} - u_{\text{out}} = u_{\text{in}} + K_x x - = u_{\text{in}} + \underbrace{K_x (sI_{n_x} - A)^{-1} B}_{L_u(s)} u_{\text{in}} = (I_{n_u} + L_u(s))u_{\text{in}} \quad (4.116)$$

The closed-loop transfer function from the command r to the system error $e = r - x$ is called the system sensitivity $S_u(s)$. It is easy to see that the sensitivity transfer function is the matrix inverse of the return difference. In fact, since

$$e = r - y = r - L_u(s)e \quad (4.117)$$

then

$$e = \underbrace{(I_{n_u} + L_u(s))^{-1}}_{S_u(s)} r = S_u(s)r \quad (4.118)$$

When evaluated at the system closed-loop eigenvalues $\lambda \in \mathbb{C}$, the return difference matrix becomes singular.

$$\det(I_{n_u} + L_u(\lambda)) = 0 \quad (4.119)$$

Because of (4.95) and (4.96), the LQR state feedback controller (4.114) makes the closed-loop system globally exponentially stable. Consequently, with the LQR feedback gain K_x from (4.114), the closed-loop eigenvalues of

$$A_{\text{cl}} = A - B K_x \quad (4.120)$$

are forced into the open left half complex plane \mathbb{C}^- .

In what follows, we shall study robustness of the closed-loop system stability (i.e., stability robustness) under gain and time-delay perturbations at the system input. We begin with the derivation of the return difference equality. It will then enable us to prove guaranteed stability robustness properties of the LQR state feedback controllers.

The Return Difference Equality

The immediate goal is to imbed the LQR loop gain (4.115) into the ARE (4.91) and derive a relationship between the optimal loop gain and the system parameters. Toward that end, we rewrite the ARE the form,

$$-P A - A^T P + P B R^{-1} B^T P - Q = 0 \quad (4.121)$$

add and subtract $s P$,

$$P(s I_{n \times n} - A) + (-s I_{n \times n} - A)^T P + P B R^{-1} B^T P - Q = 0 \quad (4.122)$$

right multiply (4.122) by $(s I_{n_x} - A)^{-1}$ and left multiply it by $(-s I_{n_x} - A)^{-1}$.

$$\begin{aligned} & (-s I_{n_x} - A)^{-T} P + P(s I_{n_x} - A)^{-1} \\ & + (-s I_{n_x} - A)^{-T} P B R^{-1} B^T P (s I_{n_x} - A)^{-1} \\ & - (-s I_{n_x} - A)^{-T} Q (s I_{n_x} - A)^{-1} = 0 \end{aligned} \quad (4.123)$$

Define the open-loop system resolvent matrix.

$$\Phi(s) = (s I_{n_x} - A)^{-1} \quad (4.124)$$

Then (4.123) becomes,

$$\Phi(-s)^T P + P \Phi(s) + \Phi(-s)^T P B R^{-1} B^T P \Phi(s) - \Phi(-s)^T Q \Phi(s) = 0 \quad (4.125)$$

and the LQR loop gain (4.115) is

$$L_u(s) = R^{-1} B^T P \Phi(s) B \quad (4.126)$$

Based on (4.126), we can left and right multiply (4.125) by B^T and B , in that order.

$$\begin{aligned} & B^T \Phi(-s)^T P B + B^T P \Phi(s) B \\ & + B^T \Phi(-s)^T P B R^{-1} B^T P \Phi(s) B \\ & - B^T \Phi(-s)^T Q \Phi(s) B = 0 \end{aligned} \quad (4.127)$$

Then we rewrite the above equation in the form,

$$\begin{aligned} & \underbrace{B^T \Phi(-s)^T P B R^{-1} R}_{L_u^T(-s)} + R \underbrace{R^{-1} B^T P \Phi(s) B}_{L_u(s)} \\ & + \underbrace{B^T \Phi(-s)^T P B R^{-1} R}_{L_u^T(-s)} \underbrace{R^{-1} B^T P \Phi(s) B}_{L_u(s)} = B^T \Phi(-s)^T Q \Phi(s) B \end{aligned} \quad (4.128)$$

arriving at

$$L_u^T(-s)R + R L_u(s) + L_u^T(-s)R L_u(s) = B^T \Phi(-s)^T Q \Phi(s) B \quad (4.129)$$

Adding R to both sides,

$$L_u^T(-s)R + R L_u(s) + L_u^T(-s)R L_u(s) + R = R + B^T \Phi(-s)^T Q \Phi(s) B \quad (4.130)$$

results in the *return difference equation* for the LQR state feedback regulators.

$$(I_{n_u} + L_u(-s))^T R (I_{n_u} + L_u(s)) = R + B^T \Phi(-s)^T Q \Phi(s) B \quad (4.131)$$

Setting $s = j \omega$, gives the *return difference inequality*,

$$(I_{n_u} + L_u(-j \omega))^T R (I_{n_u} + L_u(j \omega)) \geq R \quad (4.132)$$

or equivalently

$$(I_{n_u} + L_u^*(j \omega))^T R (I_{n_u} + L_u(j \omega)) \geq R \quad (4.133)$$

where the upper script “*” denotes the complex conjugate transpose operation.

The return difference matrix inequality (4.133) can also be written as

$$(I_{n_u} + R^{-\frac{1}{2}} L_u^*(j \omega) R^{\frac{1}{2}}) (I_{n_u} + R^{\frac{1}{2}} L_u(j \omega) R^{-\frac{1}{2}}) \geq I_{n_u} \quad (4.134)$$

This is the key property of the LQR loop gain transfer function matrix $L_u(s)$. The inequality (4.134) enables stability robustness quantification for the LQR state feedback controllers.

Before going any further, the proven stability properties of the LQR controllers are summarized in the formal statement below.

Theorem 4.2 Given a controllable LTI system (4.84) and a symmetric positive semidefinite matrix $Q \in R^{n_x \times n_x}$, suppose that (A, \sqrt{Q}) is stabilizable. Then for any symmetric positive-definite matrix $R \in R^{n_u \times n_u}$, the ARE has the unique symmetric positive-definite solution $P \in R^{n_x \times n_x}$, the LQR state feedback gain is (4.114), and the closed-loop matrix (4.120) is Hurwitz. In addition, the LQR loop gain (4.115) satisfies the return difference equality (4.131) for all $s \in \mathbb{C}$, and it also satisfies the return difference inequality for $s = j\omega$, with any $\omega \in R$. ■

Consider the open-loop system (4.84) with a scalar input and a scalar output, that is the system dynamics are SISO. Then (4.133) becomes,

$$\left| \underbrace{1 + L_u(j\omega)}_{RD_u(j\omega)} \right| \geq 1 \quad (4.135)$$

and implies that the Nyquist frequency response of the return difference $RD_u(j\omega)$ never enters the unit disk centered at $(-1, j0)$ in the complex plane for all frequencies ω . Therefore, LQR regulators for SISO LTI systems have $[\frac{1}{2}, \infty)$ gain margin and at least 60° phase margin.

For MIMO LTI systems with a diagonal matrix $R = rI_{n_u}$ and a positive constant r , the return difference inequality (4.133) takes the form,

$$(I_{n_u} + L_u^*(j\omega))(I_{n_u} + L_u(j\omega)) \geq I_{n_u} \quad (4.136)$$

and implies that the minimum singular value of the return difference $RD_u(j\omega)$ is no less than unity, uniformly in ω , and thus

$$\underline{\sigma}(RD_u(j\omega)) \geq 1 \quad (4.137)$$

Fig. 4.9 illustrates (4.135) and (4.136) graphically. As discussed in Chap. 3, the inequality (4.137) is a sufficient condition to guarantee MIMO LQR gain and phase margins equal to that of the SISO LQR controllers. In other words, simultaneous variations $[\frac{1}{2}, \infty)$ in gain or $\pm 60^\circ$ in phase, at the system input break point in all control channels, will not destabilize the corresponding closed-loop dynamics.

From (4.137), two other useful inequalities can be derived: one for the system sensitivity transfer function,

$$\overline{\sigma}(S(j\omega)) = \overline{\sigma}(RD_u(j\omega))^{-1} = \frac{1}{\underline{\sigma}(RD_u(j\omega))} \leq 1 \quad (4.138)$$

and the other for the cosensitivity transfer function.

$$\overline{\sigma}(T_u(j\omega)) = \overline{\sigma}(I_{n_u} - S_u(j\omega)) \leq 1 + \overline{\sigma}(S_u(j\omega)) \leq 2 \quad (4.139)$$

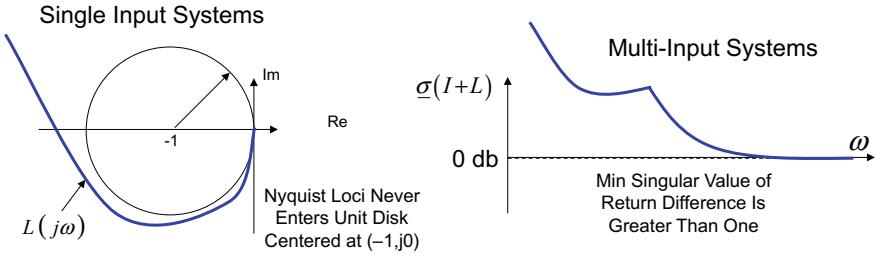
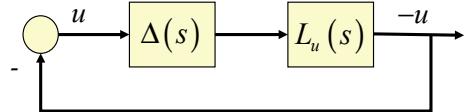


Fig. 4.9 Frequency domain analysis of LQR loop transfer functions

Fig. 4.10 Closed-loop system with a multiplicative perturbation



These two relations can be used to further analyze system robustness to unmodeled dynamics represented by unstructured multiplicative uncertainties [3].

It turns out that for LQR controllers with generic diagonal control weighting matrices, only SISO margins are guaranteed, one per control channel, with the other controls closed and unperturbed [3]. Consider the perturbed system block diagram shown in Fig. 4.10,

where $\Delta(s) \in \mathbb{C}^{n_u \times n_u}$ represents a multiplicative uncertainty perturbation at the system input and $L_u(s) \in \mathbb{C}^{n_u \times n_u}$ is the LQR loop gain from (4.115).

Suppose that the uncertainty transfer function satisfies the following strict matrix inequality.

$$\Delta^*(j\omega)R + R\Delta(j\omega) > R \quad (4.140)$$

Let $0 \leq \varepsilon \leq 1$ be a constant and consider a parameter-dependent uncertainty in the form

$$\Delta_\varepsilon(j\omega) = (1 - \varepsilon)I_{m \times m} + \varepsilon\Delta(j\omega) \quad (4.141)$$

Then

$$\begin{aligned} \Delta_\varepsilon^*(j\omega)R + R\Delta_\varepsilon(j\omega) &= (1 - \varepsilon)R + \varepsilon\Delta^*(j\omega)R + (1 - \varepsilon)R + \varepsilon R\Delta(j\omega) \\ &= 2(1 - \varepsilon)R + \varepsilon \underbrace{(\Delta^*(j\omega)R + R\Delta(j\omega))}_{> R} > (2 - \varepsilon)R > R \end{aligned} \quad (4.142)$$

So, $\Delta_\varepsilon(j\omega)$ satisfies the same strict matrix inequality (4.140), for any $0 \leq \varepsilon \leq 1$. Define

$$\tilde{\Delta}_\varepsilon(j\omega) = R^{\frac{1}{2}}\Delta_\varepsilon(j\omega)R^{-\frac{1}{2}}, \quad \tilde{L}_u(j\omega) = R^{\frac{1}{2}}L_u(j\omega)R^{-\frac{1}{2}} \quad (4.143)$$

Using these new variables, (4.134) and (4.142) can be written as

$$\begin{aligned} \tilde{\Delta}_\varepsilon^*(j\omega) + \tilde{\Delta}_\varepsilon(j\omega) &> I_{n_u} \\ \left(I_{n_u} + \tilde{L}_u^*(j\omega) \right) \left(I_{n_u} + \tilde{L}_u(j\omega) \right) &\geq I_{n_u} \end{aligned} \quad (4.144)$$

We claim that in this case,

$$\det \left[I_{n_u} + \tilde{L}_u(j\omega) \tilde{\Delta}_\varepsilon(j\omega) \right] \neq 0 \quad (4.145)$$

holds true for any $\omega \in R$ and any $0 \leq \varepsilon \leq 1$. To prove that, assume the opposite is true and suppose

$$\det \left[I_{n_u} + \tilde{L}_u(j\omega_0) \tilde{\Delta}_{\varepsilon_0}(j\omega_0) \right] = 0 \quad (4.146)$$

for some $\omega_0 \in R$ and $0 \leq \varepsilon_0 \leq 1$. Then there must exist a vector $w_0 \in R^n$ such that

$$\tilde{L}_u(j\omega_0) \tilde{\Delta}_{\varepsilon_0}(j\omega_0) w_0 = -w_0 \quad (4.147)$$

The return difference inequality in (4.144) implies

$$\tilde{L}_u(j\omega) + \tilde{L}_u^*(j\omega) + \tilde{L}_u^*(j\omega) \tilde{L}_u(j\omega) > 0 \quad (4.148)$$

Set $\omega = \omega_0$. Left multiply by $w^* \tilde{\Delta}_{\varepsilon_0}^*(j\omega_0)$ and right multiply by $\tilde{\Delta}_{\varepsilon_0}(j\omega_0) w_0$.

$$w^* \tilde{\Delta}_{\varepsilon_0}^* \underbrace{\tilde{L}_u \tilde{\Delta}_{\varepsilon_0} w_0}_{-w_0} + \underbrace{w^* \tilde{\Delta}_{\varepsilon_0}^* \tilde{L}_u^* \tilde{\Delta}_{\varepsilon_0} w_0}_{-w_0^*} + \underbrace{w^* \tilde{\Delta}_{\varepsilon_0}^* \tilde{L}_u^* \tilde{L}_u \tilde{\Delta}_{\varepsilon_0} w_0}_{-w_0^*} > 0 \quad (4.149)$$

So,

$$-w^* \tilde{\Delta}_{\varepsilon_0}^* w_0 - w_0^* \tilde{\Delta}_{\varepsilon_0} w_0 + w_0^* w_0 > 0 \quad (4.150)$$

and thus the following inequality must be true,

$$w^* \left(\tilde{\Delta}_{\varepsilon_0}^* + \tilde{\Delta}_{\varepsilon_0} \right) w_0 \leq w_0^* w_0 \quad (4.151)$$

which presents a contradiction to the argument, since (4.144) holds true at the same time. This proves the validity of the claim (4.145).

Furthermore, substituting (4.143) into (4.145), gives the following relation.

$$\begin{aligned} 0 &\neq \det[I_{n_u} + \tilde{L}_u(j\omega)\tilde{\Delta}_\varepsilon(j\omega)] \\ &= \det[I_{n_u} + R^{\frac{1}{2}}L_u(j\omega)R^{-\frac{1}{2}}R^{\frac{1}{2}}\Delta_\varepsilon(j\omega)R^{-\frac{1}{2}}] \\ &= \det[I_{n_u} + R^{\frac{1}{2}}L_u(j\omega)\Delta_\varepsilon(j\omega)R^{-\frac{1}{2}}] = \det[I_{n_u} + L_u(j\omega)\Delta_\varepsilon(j\omega)] \end{aligned} \quad (4.152)$$

Consequently, for all $\omega \in R$ and $0 \leq \varepsilon \leq 1$,

$$\det[I_{n_u} + L_u(j\omega)\Delta_\varepsilon(j\omega)] \neq 0 \quad (4.153)$$

Consider now the perturbed closed-loop system shown in Fig. 4.8. The system eigenvalues satisfy

$$\det[I_{n_u} + L_u(\lambda)\Delta(\lambda)] = 0 \quad (4.154)$$

The nominal system without the uncertainty is globally exponentially stable.

$$\det[I_{n_u} + L_u(\lambda)] = 0, \quad \lambda \in \mathbb{C}^- \quad (4.155)$$

Suppose that at least one of the eigenvalues of the perturbed system is located in the closed right half complex plane \mathbb{C}^+ . Then for some $\omega_0 \in R$ and a constant $0 \leq \varepsilon_0 \leq 1$ the inequality (4.153) is invalid which is a contradiction.

Theorem 4.3 *With the LQR controller, the perturbed system shown in Fig. 4.7 is closed-loop globally exponentially stable for all multiplicative uncertainties that satisfy the assumed inequality (4.140). ■*

Suppose R and $\Delta(s)$ are diagonal. Then (4.140) is satisfied if and only if

$$\Delta_i^*(s) + \Delta_i(s) > 1, \quad \forall 1 \leq i \leq m \quad (4.156)$$

where $\Delta_i(s)$ is i th the diagonal element of $\Delta(s)$.

If $\Delta_i(s)$ is real then (4.156) is satisfied in the interval $(\frac{1}{2}, \infty)$. Consequently, the closed-loop stability of the perturbed system in Fig. 4.10 is preserved for gain variations in any of the control channels within the identified interval. *This is the LQR gain margin.*

If $\Delta_i(s) = e^{-j\varphi}$ with $|\varphi| < \frac{\pi}{3}$ then (4.156) is satisfied and implies that closed-loop stability of the perturbed system remains intact for phase variations within $\pm 60^\circ$ phase variation in any of the control channels. *This is the LQR phase margin.*

Theorem 4.4 *If a diagonal control weight matrix R is selected with strictly positive elements then the corresponding LQR controller with the state feedback gains (4.114) has guaranteed $(\frac{1}{2}, \infty)$ gain margin and $\pm 60^\circ$ phase margin in each control channel. ■*

Note that for LQR controllers with fully populated control weighting matrices, stability margins are not guaranteed and may become impractically small due to cross-coupling in various control channels [3].

4.3.3 LQR Design and Asymptotic Properties

The numerical values in the LQR penalty matrices Q and R determine the eigenstructure of the closed-loop system, $(A - BK_x)V = V\Lambda$. This eigenstructure specifies the system performance and robustness properties as discussed in the previous section. It is very important to properly choose the numerical values for elements in Q and R , and more importantly, it is quintessential to learn how to exploit these matrices to tune the control feedback gains and to achieve the desired performance and robustness in the resulting closed-loop system.

In this section, we shall investigate how the eigenstructure evolves, as the weighting matrices are varied numerically similar to a root locus analysis in classical control. Readers interested in detailed asymptotic analysis may find it in Kwakernaak and Sivan [2].

Consider the following LTI system,

$$\dot{x} = Ax + Bu \quad x \in R^{n_x} \quad u \in R^{n_u} \quad (4.157)$$

with the infinite-time quadratic cost index,

$$J = \int_0^{\infty} (x^T Q x + u^T R u) d\tau \quad (4.158)$$

where $Q = Q^T \geq 0$, $R = R^T > 0$, (A, B) stabilizable, and $(A, Q^{1/2})$ detectable. We assume that there are no transmission zeros on the $j\omega$ axis. Then the ARE for this optimal control problem is

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (4.159)$$

Associated with this ARE is the $2n_x \times 2n_x$ Hamiltonian matrix H is given by,

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \quad (4.160)$$

which can be used to determine the solution to the ARE. The optimal state feedback control is given by

$$u = -R^{-1}B^T Px = -K_x x \quad (4.161)$$

which when substituted into (4.157) yields the closed-loop system:

$$\dot{x} = (A - BK_x)x = A_{\text{cl}}x \quad (4.162)$$

The n_x eigenvalues of the closed-loop system $\lambda(A_{\text{cl}})$ are the stable eigenvalues of the Hamiltonian matrix H . In fact, the Hamiltonian matrix H has $2n_x$ eigenvalues of which n_x have negative real parts (stable) and n_x have positive real parts (unstable, but stable backward in time). Let,

$$\phi_{\text{cl}}(s) = \det[sI_{n_x} - A + BK_x] \quad (4.163)$$

then

$$\det[sI_{2n_x} - H] = \phi_{\text{cl}}(s)\phi_{\text{cl}}(-s) \quad (4.164)$$

The asymptotic properties we desire to explore are those associated with the migration of these eigenvalues as the numerical values in the LQR penalty matrices Q and R are varied. We can examine these eigenvalues (roots of $\phi_{\text{cl}}(s)$) through the polynomial formed by expanding the $\det[sI_{2n_x} - H]$. We begin with some elementary row and column operations on H . First, we multiply the first row of H by $-Q(sI_{n_x} - A)^{-1}$ and add it to the second row. This yields

$$\begin{aligned} \det[sI_{2n_x} - H] &= \det \begin{bmatrix} sI_{n_x} - A & BR^{-1}B^T \\ Q & sI_{n_x} + A^T \end{bmatrix} \\ &= \det \begin{bmatrix} sI_{n_x} - A & BR^{-1}B^T \\ 0 & (sI_{n_x} + A^T) - Q(sI_{n_x} - A)^{-1}BR^{-1}B^T \end{bmatrix} \end{aligned} \quad (4.165)$$

Then,

$$\begin{aligned} \det[sI_{2n_x} - H] &= \det[sI_{n_x} - A] \det \left[(sI_{n_x} + A^T) - Q(sI_{n_x} - A)^{-1}BR^{-1}B^T \right] \\ &= \det[sI_{n_x} - A] \det \left[(sI_{n_x} + A^T) \left\{ I_{n_x} - (sI_{n_x} + A^T)^{-1}Q(sI_{n_x} - A)^{-1}BR^{-1}B^T \right\} \right] \\ &= \det[sI_{n_x} - A] \det[sI_{n_x} + A^T] \det \left[I_{n_x} - (sI_{n_x} + A^T)^{-1}Q(sI_{n_x} - A)^{-1}BR^{-1}B^T \right] \end{aligned} \quad (4.166)$$

We factor the Q and $BR^{-1}B^T$ into products of two square roots: $Q = Q_1^T Q_1$ and $BR^{-1}B^T = R_1 R_1^T$. Next, using the identity $\det[I - AB] = \det[I - BA]$, we get

$$\begin{aligned} & \det \left[I_{n_x} - \underbrace{\left(sI_{n_x} + A^T \right)^{-1} Q_1^T}_{B} \underbrace{Q_1(sI_{n_x} - A)^{-1} R_1 R_1^T}_{A} \right] \\ &= \det \left[I_{n_x} - Q_1(sI_{n_x} - A)^{-1} R_1 R_1^T \left(sI_{n_x} + A^T \right)^{-1} Q_1^T \right] \end{aligned} \quad (4.167)$$

and so,

$$\begin{aligned} & \det[sI_{2n_x} - H] \\ &= \det[sI_{n_x} - A] \det[sI_{n_x} + A^T] \det \left[I_{n_x} - \underbrace{Q_1(sI_{n_x} - A)^{-1} R_1 R_1^T}_{H_1(s)} \left(sI_{n_x} + A^T \right)^{-1} Q_1^T \right] \\ &= \phi_{\text{ol}}(s)(-1)^{n_x} \phi_{\text{ol}}(-s) \det \left[I_{n_x} + H_1(s) H_1^T(-s) \right] \end{aligned} \quad (4.168)$$

where $\phi_{\text{ol}}(s) = \det[sI_{n_x} - A]$ is the open-loop characteristic polynomial. Thus,

$$\phi_{\text{cl}}(s)\phi_{\text{cl}}(-s) = \phi_{\text{ol}}(s)\phi_{\text{ol}}(-s) \det \left[I_{n_x} + H_1(s) H_1^T(-s) \right] \quad (4.169)$$

Let,

$$\det[H_1(s)] = \frac{\psi(s)}{\phi(s)}, \quad \text{where: } H_1(s) = Q_1(sI_{n_x} - A)^{-1} R_1 \quad (4.170)$$

and consider the performance cost index,

$$J = \int_0^\infty \left(x^T Q x + \rho^2 u^T R u \right) d\tau \quad (4.171)$$

with a positive scalar weight $\rho > 0$. We are interested in the behavior as $\rho \rightarrow 0$ and as $\rho \rightarrow \infty$. The zeros of (4.169) are also the zeros of

$$\phi(s)\phi(-s) \det[\rho I_{n_x} + H_1(s) H_1(-s)] \quad (4.172)$$

As $\rho \rightarrow 0$, some of the roots will go to infinity. Those that stay finite, will approach the transmission zeros of the transfer function matrix $H_1(s)$ and their negative values. These finite zeros influence the dynamic response of the optimal regulator. As $\rho \rightarrow \infty$, the roots of $\phi_{\text{cl}}(s)$ are the n_x stable roots of $\phi_{\text{ol}}(s)\phi_{\text{ol}}(-s)$.

That is, if the roots of $\phi_{\text{ol}}(s)$ have positive real part, then the mirror image of them in $\phi_{\text{ol}}(-s)$ will become the stable roots in $\phi_{\text{cl}}(s)$.

We see that shaping the zeros of $H_1(s)$ plays a crucial role in the design of the optimal control. This is done through selection of the LQR penalty matrix weights Q and R . Later in Sect. 4.4, we will use this fact to tune the design of optimal controllers to achieve performance and robustness.

4.4 Command Tracking and Robust Servomechanism Control

Most industrial control problems require the control system to accurately track commands. This requirement distinguishes these problems from regulation in which the state is driven to zero. From classical control theory, we know that in order to track a constant command with zero error, we need to add integral error control action into the controller. For single-input-single-output (SISO) systems, the loop transfer function $L_u(s)$ can be written as,

$$L_u(s) = \frac{K(b_0 s^m + \dots + b_{m-1} s + 1)}{s^p(a_0 s^n + \dots + a_{n-1} s + 1)} \quad (4.173)$$

where the gain K and the polynomial coefficients a_i and b_i are real constants. The *type* of the control system depends upon the order p of the pole of $L(s)$ at $s = 0$. The number of finite zeros, their location, or the location of the poles are not important to specify the system type. The system type p , where $p = 0, 1, 2, \dots$ indicates how many integrators are present in the control system. We know that in order to track a constant command $r(t) = \text{const}$, and to produce zero steady-state error, an integrator is needed, $p \geq 1$, creating (at a minimum) a type 1 system. In order to track a type 1 input, the control system will need two integrators, creating a type 2 system. Thus, to track commands accurately, the class of commanded signals must be known, and the controller must be augmented with enough integrators to produce zero steady-state errors.

When these integrators are added to the control system for command tracking they also provide disturbance rejection within the same class, i.e., a type 1 control system can track constant commands and reject constant disturbances. Similarly, a type 2 system can track ramp inputs and reject ramp disturbances.

Basically, the augmentation of the system with these integrators for command tracking requires embedding into the system a model of the class of signals that the system will track. This is often referred to as the *internal model principle* [6]. For instance, when tracking a constant command and adding a single integrator, we have embedded the command generation internal model $\dot{r} = 0$ into the system.

In the previous sections we have illustrated the use of linear quadratic optimal control theory to design a controller and examined the excellent stability properties provided by that method. The LQR forces the system state to go to zero, forming a type 0 control system. If one wants to track a constant command using such an

LQR controller, the system would have a steady-state offset error, to the command. We know from Eq. (4.173) that in order to track a constant command with zero error, we need to add an integrator, creating a type 1 control system.

A natural extension of the LQR method presented in the previous chapter would be to add an integral control action into the controller to produce zero steady errors, while tracking constant commands. The number of integrators that would need to be added depends upon the commanded signal (whether it is a constant, a ramp, or other type of signal).

This section presents a systematic process for building an augmented state-space model, called the *servomechanism design model* [7]. This state-space description embeds a model of the class of signals to be tracked, such that when optimal control theory is applied, the state regulation provides accurate tracking of the selected class of external commands. This control system is then decomposed into two parts: a servo tracking controller for command following and a state feedback component for stabilization.

Consider the linear-time-invariant (LTI) multi-input–multi-output (MIMO) system dynamics,

$$\begin{aligned}\dot{x} &= Ax + Bu + Ew \\ y &= Cx + Du + Fw\end{aligned}\tag{4.174}$$

where $x \in R^{n_x}$ is the system state, $u \in R^{n_u}$ is the control input, $w \in R^{n_w}$ is the external disturbance, and $y \in R^{n_y}$ is the system regulated output. We assume that (A, B) is controllable, (A, C) is observable, and the system state is accessible.

The control problem of interest is to design a control input u such that the system regulated output y globally asymptotically tracks bounded, smooth and possibly time-variant commands $r(t)$, that are generated by the p -dimensional ordinary differential equation,

$$r^{(p)} + a_1 r^{(p-1)} + \cdots + a_{p-1} \dot{r} + a_p r = 0\tag{4.175}$$

with known scalar coefficients a_i , where $r^{(i)}$ denotes the i th time derivative of r . It is also assumed that the unknown external disturbance $w(t)$ satisfies the same model (4.175).

In what follows, we will embed the command-disturbance generating model (4.175) into the open-loop system dynamics (4.174) and derive the so-called servomechanism model and the corresponding servomechanism controller to track external commands and reject any unknown disturbances, both generated by (4.175).

4.4.1 Servomechanism Control Design Model

For convenience, define $a_0 = 1$ and rewrite (4.175) as

$$\sum_{i=0}^p a_i r^{(p-i)} = 0 \quad (4.176)$$

At this point, no restrictions are placed on the roots of the command-disturbance generating characteristic polynomial (4.176).

$$a(\lambda) = \sum_{i=0}^p a_i \lambda^{(p-i)} = 0 \quad (4.177)$$

As previously mentioned, it is assumed that the unknown time-varying disturbances $w(t)$ satisfy the same dynamics (4.176).

$$\sum_{i=0}^p a_i w^{(p-i)} = 0 \quad (4.178)$$

Let

$$e = y - r \quad (4.179)$$

denote the regulated output tracking error. Then

$$\begin{aligned} \sum_{i=0}^p a_i e^{(p-i)} &= \sum_{i=0}^p a_i y^{(p-i)} - \underbrace{\sum_{i=0}^p a_i r^{(p-i)}}_0 = \sum_{i=0}^p a_i y^{(p-i)} \\ &= C \underbrace{\left(\sum_{i=0}^p a_i x^{(p-i)} \right)}_{\xi} + D \underbrace{\left(\sum_{i=0}^p a_i u^{(p-i)} \right)}_{\mu} + F \underbrace{\left(\sum_{i=0}^p a_i w^{(p-i)} \right)}_0 \end{aligned} \quad (4.180)$$

and so

$$\sum_{i=0}^p a_i e^{(p-i)} = C \xi + D \mu \quad (4.181)$$

where

$$\xi = \sum_{i=0}^p a_i x^{(p-i)}, \quad \mu = \sum_{i=0}^p a_i u^{(p-i)} \quad (4.182)$$

are the aggregated state and control signals, respectively. Differentiating ξ with respect to time gives

$$\begin{aligned}\dot{\xi} &= \sum_{i=0}^p a_i \dot{x}^{(p-i)} = \sum_{i=0}^p a_i \left(A x^{(p-i)} + B u^{(p-i)} + E w^{(p-i)} \right) \\ &= A \underbrace{\left(\sum_{i=0}^p a_i x^{(p-i)} \right)}_{\xi} + B \underbrace{\left(\sum_{i=0}^p a_i u^{(p-i)} \right)}_{\mu} + E \underbrace{\left(\sum_{i=0}^p a_i w^{(p-i)} \right)}_0\end{aligned}\quad (4.183)$$

or equivalently

$$\dot{\xi} = A \xi + B \mu \quad (4.184)$$

Combining (4.181) and (4.184) yields the extended open-loop system.

$$\begin{aligned}\sum_{i=0}^p a_i e^{(p-i)} &= C \xi + D \mu \\ \dot{\xi} &= A \xi + B \mu\end{aligned}\quad (4.185)$$

Let $e_1 = e \in R^{n_u}$. Then the dynamics (4.185) can be written as an $(n_u \times p + n_x)$ -dimensional LTI system.

$$\begin{aligned}\dot{e}_1 &= e_2 \\ \dot{e}_2 &= e_3 \\ &\vdots \\ \dot{e}_{p-1} &= e_p \\ \dot{e}_p &= - \sum_{i=1}^p a_i e_{p-i+1} + C \xi + D \mu \\ \dot{\xi} &= A \xi + B \mu\end{aligned}\quad (4.186)$$

Once again rewriting (4.186) in matrix form results in,

$$\underbrace{\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \vdots \\ \dot{e}_{p-1} \\ \dot{e}_p \\ \dot{\xi} \end{pmatrix}}_{\dot{z}} = \underbrace{\begin{pmatrix} 0_{n_u \times n_u} & I_{n_u} & 0_{n_u \times n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & I_{n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & 0_{n_u \times n_u} & \dots & I_{n_u} & 0_{n_u \times n_x} \\ -a_p I_{n_u} & -a_{p-1} I_{n_u} & -a_{p-2} I_{n_u} & \dots & -a_1 I_{n_u} & C \\ 0_{n_x \times n_u} & 0_{n_x \times n_u} & 0_{n_x \times n_u} & \dots & 0_{n_x \times n_u} & A \end{pmatrix}}_{\tilde{A}} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{p-1} \\ e_p \\ \xi \end{pmatrix} \quad z$$

$$\begin{aligned}
& + \underbrace{\begin{pmatrix} 0_{n_u \times n_u} \\ 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ D \\ B \end{pmatrix}}_{\tilde{B}} \mu \\
e_1 &= \underbrace{\left(I_{n_u} \ 0_{n_u \times n_u} \ 0_{n_u \times n_u} \ \dots \ 0_{n_u \times n_u} \ 0_{n_u \times n_x} \right)}_{\tilde{C}} z
\end{aligned} \tag{4.187}$$

which in turn represents the servomechanism *design model*.

$$\begin{aligned}
\dot{z} &= \tilde{A} z + \tilde{B} \mu \\
e_1 &= \tilde{C} z
\end{aligned} \tag{4.188}$$

Note that the problem of asymptotic command tracking for (4.174) is reduced to stabilization of the first state component e_1 in (4.188). The main advantage of using the command-disturbance generating model (4.188) is the absence of external disturbances in the system dynamics.

Example 4.5 Constant Command Tracking Consider a scalar constant command r . According to (4.175), this gives $\dot{r} = 0$ ($p = 1$), with $a_1 = 0$. The command error is $e = y_c - r$ where $y_c = C_c x + D_c u$. The servomechanism design model using (4.188) is given by

$$\begin{aligned}
\dot{z} &= \tilde{A} z + \tilde{B} \mu \quad z = \begin{bmatrix} e \\ \dot{x} \end{bmatrix}, \mu = \dot{u} \\
\tilde{A} &= \begin{bmatrix} 0 & C_c \\ 0 & A \end{bmatrix}, \tilde{B} = \begin{bmatrix} D_c \\ B \end{bmatrix}, \tilde{C} = [1 \ 0_{1 \times n_x}]
\end{aligned} \tag{4.189}$$

Example 4.6 Sinusoidal Command Tracking Consider a scalar sinusoidal command $r(t) = \sin(\omega t)$. This gives $\ddot{r} = -\omega^2 r$, ($p = 2$), with $a_1 = 0$, $a_2 = -\omega_0^2$, (see Eq. (4.175)). The command error is $e = y_c - r$ where $y_c = C_c x + D_c u$. The state-space system using (4.188) is given by:

$$\begin{aligned}
\dot{z} &= \tilde{A} z + \tilde{B} \mu \quad z = \begin{bmatrix} e \\ \xi \end{bmatrix}, \quad \xi = \ddot{x} - \omega^2 x, \quad \mu = \ddot{u} - \omega^2 u, \\
\tilde{A} &= \begin{bmatrix} 0 & 1 & 0 \\ -\omega^2 & 0 & C_c \\ 0 & 0 & A \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ D_c \\ B \end{bmatrix}, \quad \tilde{C} = [1 \ 0 \ 0_{1 \times n_x}]
\end{aligned} \tag{4.190}$$

Example 4.7 Constant Command Tracking in a Scalar System Our knowledge from classical control tells us that a type 1 controller is needed to track a constant command. Using a scalar system, this example will build a state-space model and illustrate how to design an integral control for tracking constant commands. Consider the following scalar system,

$$\begin{aligned}\dot{x} &= -2x + u + w \\ y &= x\end{aligned}\tag{4.191}$$

where x is the state, u is the control, and w is a non-measurable constant disturbance. Hence $A = [-2]$, $B = [1]$, $E = [1]$, $C = [1]$, and $D = [0]$. The goal is for the output y (same as the state x) to track a constant command r , with zero steady-state error. The constant scalar command is modeled using (4.175) as

$$\dot{r} = 0, \quad p = 1, \quad a_1 = 0\tag{4.192}$$

The robust servo design model (4.188) is

$$\tilde{A} = \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix}; \quad \tilde{B} = \begin{bmatrix} D \\ B \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad \tilde{C} = [1 \ 0]\tag{4.193}$$

in which we see that (\tilde{A}, \tilde{B}) form a controllable pair. The feedback control law is $\mu = -Kz$. For this example it is desired that the closed-loop dynamics have a characteristic polynomial of $\phi_{cl}(s) = (s+2)^2 + 4 = s^2 + 4s + 8$ (pole placement problem). The feedback control is

$$\mu = -[K_1 \ K_x] \begin{bmatrix} e \\ \dot{x} \end{bmatrix}\tag{4.194}$$

The closed-loop system is $\dot{z} = (A - BK)z$ with characteristic polynomial $\phi_{cl}(s) = \det[sI - \tilde{A} + \tilde{B}K_z]$. Substitute for (\tilde{A}, \tilde{B}) keeping the gains as parameters, expand the determinant, and equate to the desired closed-loop characteristic polynomial

$$\det[sI - \tilde{A} + \tilde{B}K_z] = s^2 + (2 + K_x)s + K_1 = s^2 + 4s + 8\tag{4.195}$$

Equating coefficients of s yields two equations in the two unknown gains that can be solved for, the gains $[K_1 \ K_x] = [8 \ 2]$. The control $u = \int \mu$ and is

$$u = -K \int z dt = -[8 \ 2] \begin{bmatrix} \int e dt \\ x \end{bmatrix} = -8 \int e dt - 2x + \text{constant of integration}\tag{4.196}$$

In the implementation, the constant of integration is ignored. Figure 4.11 illustrates the system, (controller, plant, and disturbance).

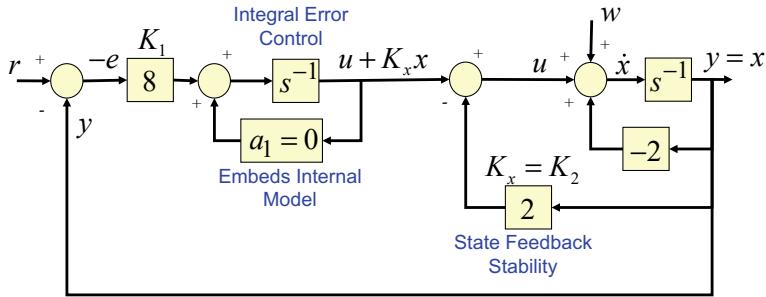


Fig. 4.11 Example 4.7 block diagram of the control and system dynamics

4.4.2 Servomechanism Model Controllability

Before proceeding any further, we need to study controllability of the servomechanism model (4.188). Toward that end, we shall use the Hautus controllability test.

$$[(A, B) \text{ is controllable}] \Leftrightarrow [\text{rank}(s I_{n_x} - A \ B) = n, \ \forall s = \lambda(A)] \quad (4.197)$$

For the extended system (4.188), the Hautus test requires

$$\begin{aligned} \text{rank} & \left[\begin{array}{cccccc} s I_{n_u} & -I_{n_u} & 0_{n_u \times n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & s I_{n_u} & -I_{n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & 0_{n_u \times n_u} & \dots & -I_{n_u} & 0_{n_u \times n_x} \\ a_p I_{n_u} & a_{p-1} I_{n_u} & a_{p-2} I_{n_u} & \dots & (s + a_1) I_{n_u} & -C \\ 0_{n_x \times n_u} & 0_{n_x \times n_u} & 0_{n_x \times n_u} & \dots & 0_{n_x \times n_u} & s I_{n_x} - A \end{array} \right] \begin{array}{c} 0_{n_u \times n_u} \\ 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ D \\ B \end{array} \\ & = n_u \times p + n_x, \quad \forall s = \lambda(\tilde{A}) \end{aligned} \quad (4.198)$$

It is known that elementary row operations applied to $\tilde{H}(\lambda) = (\lambda I_{ext} - \tilde{A} \ \tilde{B})$ do not change the matrix rank. So we can multiply the first n_u -rows of $\tilde{H}(\lambda)$ by $\frac{a_p}{s}$ and subtract the result from the second-to-last n_u -rows.

$$\begin{aligned} \text{rank } \tilde{H}(\lambda) &= \text{rank} \left(\begin{array}{cccccc} s I_{n_u} & -I_{n_u} & 0_{n_u \times n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & s I_{n_u} & -I_{n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & 0_{n_u \times n_u} & \dots & -I_{n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & (a_{p-1} + \frac{a_p}{s}) I_{n_u} & a_{p-2} I_{n_u} & \dots & (s + a_1) I_{n_u} & -C \\ 0_{n_x \times n_u} & 0_{n_x \times n_u} & 0_{n_x \times n_u} & \dots & 0_{n_x \times n_u} & s I_{n_x} - A \end{array} \right) \begin{array}{c} 0_{n_u \times n_u} \\ 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ D \\ B \end{array} \end{aligned} \quad (4.199)$$

We perform a similar operation, multiplying the second set of n_u -rows by $\left(\frac{a_{p-1} + \frac{a_{p-1}}{s}}{s}\right)$ and subtracting the result from the second-to-last set of m -rows.

$$\text{rank } \tilde{H}(\lambda) =$$

$$\text{rank} \begin{pmatrix} s I_{n_u} & -I_{n_u} & 0_{n_u \times n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & s I_{n_u} & -I_{n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & 0_{n_u \times n_u} & \dots & -I_{n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & \left(a_{p-2} - \frac{a_{p-1} + \frac{a_{p-1}}{s}}{s}\right) I_{n_u} & \dots & (s+a_1) I_{n_u} & -C \\ 0_{n_x \times n_u} & 0_{n_x \times n_u} & 0_{n_x \times n_u} & \dots & 0_{n_x \times n_u} & s I_{n_x} - A \end{pmatrix} \begin{pmatrix} 0_{n_u \times n_u} \\ 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ D \\ B \end{pmatrix} \quad (4.200)$$

Continuing using similar row operations results in,

$$\text{rank } \tilde{H}(\lambda) = \text{rank} \begin{pmatrix} s I_{n_u} & -I_{n_u} & 0_{n_u \times n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & s I_{n_u} & -I_{n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & 0_{n_u \times n_u} & \dots & -I_{n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & 0_{n_u \times n_u} & \dots & \frac{a(s)}{s^{p-1}} I_{n_u} & -C \\ 0_{n_x \times n_u} & 0_{n_x \times n_u} & 0_{n_x \times n_u} & \dots & 0_{n_x \times n_u} & s I_{n_x} - A \end{pmatrix} \begin{pmatrix} 0_{n_u \times n_u} \\ 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ D \\ B \end{pmatrix} \quad (4.201)$$

where $a(\lambda)$ is the command-disturbance generating characteristic polynomial (4.177). Clearly, the first $m \times (p-1)$ rows of the matrix on the right-hand side are linearly independent for any s , and consequently

$$\text{rank } \tilde{H}(\lambda) = n_u \times (p-1) + \text{rank} \begin{pmatrix} \frac{a(s)}{s^{p-1}} I_{n_u \times n_u} & -C & D \\ 0_{n_x \times n_u} & s I_{n_x} - A & B \end{pmatrix} \quad (4.202)$$

Therefore,

$$\left[\text{rank } \tilde{H}(\lambda) = n_u \times p + n_x \right] \Leftrightarrow \left[\text{rank} \begin{pmatrix} \frac{a(s)}{s^{p-1}} I_{n_u} & -C & D \\ 0_{n_x \times n_u} & s I_{n_x} - A & B \end{pmatrix} = n_u + n_x \right] \quad (4.203)$$

If $s = \lambda(a)$ then for the Hautus test to be true, it is sufficient to require

$$\underbrace{\text{rank} \begin{pmatrix} -C & D \\ s I_{n_x} - A & B \end{pmatrix}}_{H(s)} = n_u + n_x, \quad \forall s = \lambda(a) \quad (4.204)$$

Converse is also true. On the other hand, if $s \neq \lambda(a)$ then (4.203) holds due to the assumed controllability of the original system matrix pair (A, B) .

Recall that a complex number s is called the system transmission zero if $\text{rank } H(s) < n_u + n_x$. Thus, the servomechanism model (4.188) is controllable if and only if the following two conditions are met:

1. (A, B) is controllable.
2. The system transmission zeros are not the eigenvalues of the command-disturbance generating polynomial $a(s)$.

These two conditions are sufficient and necessary for controllability of the servomechanism model (4.188).

4.4.3 Servomechanism Control Design

Assuming that the servomechanism model controllability conditions hold, we can use a control design method, such as pole placement or LQR, to find a stabilizing controller in state feedback form,

$$\begin{aligned} \mu &= -K_z z = -\underbrace{(K_p \ K_{p-1} \ K_{p-2} \ \dots \ K_1 \ K_x)}_{K_z} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{p-1} \\ e_p \\ \xi \end{pmatrix} \\ &= -\sum_{i=1}^p K_{p+1-i} e_i - K_x \xi \end{aligned} \quad (4.205)$$

where $K_i \in R^{n_u \times n_u}$ are tracking error feedback gain matrices and $K_x \in R^{n_u \times n_x}$ is the state feedback gain on ξ . These gains are designed using the servomechanism design model from (4.188) where the control law (4.205) is

$$\mu = \overset{(p)}{\mu} - \sum_{i=1}^p a_i \overset{(p-i)}{\mu} = -\sum_{i=1}^p K_i \overset{(p-i)}{e} - K_x \left[\overset{(p)}{x} - \sum_{i=1}^p a_i \overset{(p-i)}{x} \right] \quad (4.206)$$

Integrating (4.206) p -times gives the control solution u for the original system model in (4.174) as:

$$u = -K_x x + \sum_{i=1}^p s^{-i} (a_i(u + K_x x) - K_i e) \quad (4.207)$$

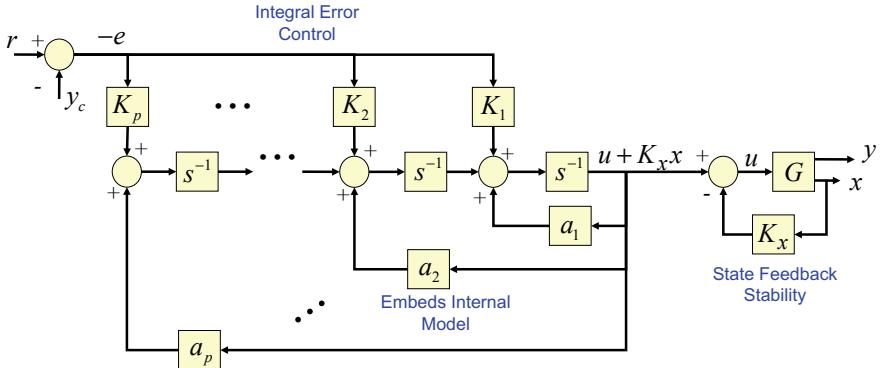


Fig. 4.12 Robust servomechanism block diagram

Figure 4.12 is a block diagram illustrating the system of (4.174) (represented as G) connected to the robust servomechanism state feedback control law.

The state feedback term $(-K_x x)$ enforces closed-loop stability of the plant. The p integrators and their gain matrices provide integral error control, and the feedback loops internal to the controller with coefficients a_i embed the internal model of the signal being tracked. When the servomechanism gains are designed using an LQR the closed-loop system using state feedback is guaranteed to be globally exponentially stable and it will force the system regulated output y_c to track the command signal $r(t)$ with zero steady-state error.

The gain blocks in Fig. 4.12 are matrices of dimension $n_u \times n_u$. This prevents one from collapsing the controller into transfer functions to gain insight into the servomechanism dynamics. In most applications the command signal r to be tracked is a scalar, with the system dynamics (4.174) having a scalar control, $n_u = 1$. In such systems the controller in Fig. 4.12 can be reduced into a scalar transfer function. Using the notation from (4.182) and (4.186), the controller (4.205) can be written as

$$u^{(p)} + \sum_{i=1}^p a_i u^{(p-i)} = - \sum_{i=1}^p K_i e^{(p-i)} - K_x \left(x^{(p)} + \sum_{i=1}^p a_i x^{(p-i)} \right) \quad (4.208)$$

Taking the Laplace transform of both sides,

$$\left(s^p + \sum_{i=1}^p a_i s^{p-i} \right) u = - \left(\sum_{i=1}^p K_i s^{p-i} \right) e - K_x \left(s^p + \sum_{i=1}^p a_i s^{p-i} \right) x \quad (4.209)$$

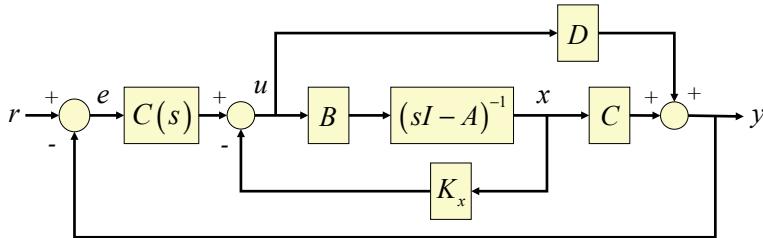


Fig. 4.13 Error-state feedback servocontroller block diagram

gives the command tracking control solution for the original system (4.174), in the form of an implementable strictly proper dynamic feedback.

$$u = - \left(\frac{\sum_{i=1}^p K_i s^{p-i}}{s^p + \sum_{i=1}^p a_i s^{p-i}} \right) e - K_x x \quad (4.210)$$

Consider strictly proper transfer function matrix.

$$C(s) = \frac{\sum_{i=1}^p K_i s^{p-i}}{s^p + \sum_{i=1}^p a_i s^{p-i}} = \frac{K_1 s^{p-1} + K_2 s^{p-2} + \dots + K_{p-1} s + K_p}{s^p + a_1 s^{p-1} + a_2 s^{p-2} + \dots + a_{p-1} s + a_p} \quad (4.211)$$

Then the servocontroller (4.210) can be written in the error-state feedback form.

$$u = -C(s) e - K_x x = C(s) (r - y) - K_x x \quad (4.212)$$

Figure 4.13 shows the controller block diagram.

The servocontrol solution (4.211), (4.212) embeds the external command model (4.176) into its dynamic compensator $C(s)$ (4.211). We can also view this architecture as an inner–outer-loop dynamics feedback controller. The inner loop is represented by state feedback gain K_x . This loop stabilizes the system dynamics. The outer loop is formed with the dynamic compensator $C(s)$ that embeds an internal model of the command signal and drives the system tracking error e to zero.

It is interesting to note that if all feedback gains in (4.205) are computed using the LQR design method then the inner loop is always stabilizing, even when the outer-loop compensator is set to zero.

Example 4.8 First-Order Servomechanism Compensators For $p = 1$, the characteristic polynomial (4.175) $a(s) = s + a_1$ is of first order, and consequently, the servomechanism command tracking controller (4.210) becomes

$$u = -\left(\frac{K_1}{s + a_1}\right)e - K_x x$$

In this case, the system regulated output y will track any command r and reject any disturbance w that satisfy the same differential equation but with possibly different and unknown coefficients.

$$\dot{w} = -a_1 w, \quad \dot{r} = -a_1 r$$

If $a_1 < 0$ then both the disturbance and the command grow exponentially without bound. The required control effort also becomes unbounded since the controller tracking error transfer function has a pole in the right half plane. From a practical point of view, an unbounded controller is unacceptable. This immediately puts a realistic restriction on the class of allowable commands and disturbances: $a_1 \geq 0$.

It is interesting to consider the case when $a_1 = 0$. This results in a Proportional + Integral (PI) feedback controller in its classical form.

$$u = K_1\left(\frac{r - y}{s}\right) - K_x x$$

Such a controller will track constant commands, reject constant disturbances, and keep all signals in the closed-loop system uniformly bounded. These properties are consistent with the classical PI control design concept. ■

It is evident that for the control solution (4.210) to become practical, it is necessary to impose the following restriction on the class of allowable commands and disturbances: $\operatorname{Re}\lambda(a(s)) \leq 0$. In other words, we preclude unstable eigenvalues in the command and disturbance generation dynamics. However, eigenvalues on the $j\omega$ -axis are allowed and, in that case, special care must be undertaken. Command generation models of higher than unity order can generate unbounded signals. Because of that they are often implemented with position and rate limiters to ensure bounded commands and thus bounded controls during the system operation.

Next we derive the closed-loop dynamics, with the servocontroller (4.210) and the original open-loop system (4.174).

$$\dot{x} = Ax + Bu + Ew$$

$$y = Cx + Du + Fw$$

$$u = - \left(\frac{\sum_{i=1}^p K_i s^{p-i}}{s^p + \sum_{i=1}^p a_i s^{p-i}} \right) \underbrace{(y - r)}_e - K_x x \quad (4.213)$$

$\underbrace{C(s)}$

We need to represent the dynamic servocontroller (4.210) in the state-space form. Similar to (4.205) we define the controller state-space model,

$$\begin{aligned} \dot{v}_1 &= v_2 \\ \dot{v}_2 &= v_3 \\ &\vdots \\ \dot{v}_{p-1} &= v_p \\ \dot{v}_p &= - \sum_{i=1}^p a_{p-i+1} v_i + \underbrace{(y - r)}_e \\ u &= - \sum_{i=1}^p K_{p-i+1} v_i - K_x x \end{aligned} \quad (4.214)$$

and rewrite the controller dynamics in matrix form.

$$\begin{aligned} u &= - \sum_{i=1}^p K_{p-i+1} v_i - K_x x \\ &= - \underbrace{\begin{pmatrix} K_p & K_{p-1} & K_{p-2} & \dots & K_1 & K_x \end{pmatrix}}_{K_z} \underbrace{\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{p-1} \\ v_p \\ x \end{pmatrix}}_{\tilde{x}} = -K_z \tilde{x} \end{aligned} \quad (4.215)$$

It is interesting to compare (4.215)–(4.205). Even though the two formulations look similar, they have completely different meaning. In the servocontrol definition (4.215), $\tilde{x} \in R^{n_x+p}$ denotes the extended system state vector and v_p is a filtered signal of the tracking error $e = y - r$. For $p > 1$ and $1 \leq i \leq p - 1$, v_i is the integral of v_{i+1} . Consequently for command generation models that have higher than unity order, the filtered tracking error v_p is processed by a bank of $(p - 1)$ integrators.

Based on (4.215), the extended open-loop system dynamics, the servocontroller and the regulated output can be written in matrix form.

$$\begin{aligned}
 \begin{pmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \vdots \\ \dot{v}_{p-1} \\ \dot{v}_p \\ \dot{x} \end{pmatrix} &= \underbrace{\begin{pmatrix} 0_{n_u \times n_u} & I_{n_u} & 0_{n_u \times n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & I_{n_u \times n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & 0_{n_u \times n_u} & \dots & I_{n_u} & 0_{n_u \times n_x} \\ -a_p I_{n_u} & -a_{p-1} I_{n_u} & -a_{p-2} I_{n_u} & \dots & -a_1 I_{n_u} & C \\ 0_{n_x \times n_u} & 0_{n_x \times n_u} & 0_{n_x \times n_u} & \dots & 0_{n_x \times n_u} & A \end{pmatrix}}_{\tilde{A}} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{p-1} \\ v_p \\ x \end{pmatrix} \\
 &+ \underbrace{\begin{pmatrix} 0_{n_u \times n_u} \\ 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ D \\ B \end{pmatrix}}_{\tilde{B}} u + \underbrace{\begin{pmatrix} 0_{n_u \times n_u} \\ 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ F \\ E \end{pmatrix}}_{\tilde{E}} w + \underbrace{\begin{pmatrix} 0_{n_u \times n_u} \\ 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ -I_{n_u} \\ 0_{n_u \times n_u} \end{pmatrix}}_{\tilde{B}_{cmd}} r \\
 u &= -K_z \tilde{x}, \quad y = \underbrace{\begin{pmatrix} 0_{n_u \times n_u} & 0_{n_u \times n_u} & 0_{n_u \times n_u} & \dots & 0_{n_u \times n_u} & C_{n_u \times n_x} \end{pmatrix}}_{\tilde{C}} \tilde{x} \\
 &\quad + D u + F w
 \end{aligned} \tag{4.216}$$

We will often refer to the controller in (4.216) as the robust servolinear quadratic regulator, or in short RSLQR. Substituting control feedback and regulated output expressions into the extended open-loop system gives the closed-loop dynamics.

$$\begin{aligned}
 \dot{\tilde{x}} &= \underbrace{\left(\tilde{A} - \tilde{B} K_z \right)}_{\tilde{A}_{cl}} \tilde{x} + \tilde{B}_{cmd} r + \tilde{E} w \\
 y &= \underbrace{\left(\tilde{C} - D K_z \right)}_{\tilde{C}_{cl}} \tilde{x} + F w
 \end{aligned} \tag{4.217}$$

This system is driven by an external command $r(t)$ and a disturbance $w(t)$. Both of these signals are generated by the same model (4.175).

In Chap. 2 we introduced state models for a common plant (2.58) and a controller (2.59). These models were then coupled to form a closed-loop simulation model and loop gain frequency domain analysis models. We want to implement the RSLQR control from (4.216) using the controller given by

$$\dot{x}_c = A_c x_c + B_{c1} y + B_{c2} r$$

$$u = C_c x_c + D_{c1} y + D_{c2} r \quad (4.218)$$

The control in (4.216) is a state feedback control ($y = x$). Substituting into (4.218) we have:

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} x + B_{c2} r \\ u &= C_c x_c + D_{c1} x + D_{c2} r\end{aligned}\quad (4.219)$$

with

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0_{n_u \times n_u} & I_{n_u} & \cdots & 0_{n_u \times n_u} \\ \vdots & \ddots & \cdots & 0_{n_u \times n_u} \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & \cdots & I_{n_u} \\ a_p I_{n_u} - D_c K_p & \cdots & \cdots a_1 I_{n_u} - D_c K_1 \\ [-K_p \cdots -K_2 -K_1] \end{bmatrix} & \begin{bmatrix} 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ C_c - D_c K_x \\ [-K_x] \end{bmatrix} & \begin{bmatrix} 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ -I_{n_u} \\ [0] \end{bmatrix} \end{bmatrix} \quad (4.220)$$

Example 4.9 The Robust Servo Controller For Example 4.4 In Example 4.7 the robust servo control u was given as

$$u = -8 \int \text{edt} - 2x \quad (4.221)$$

The state-space model for the controller using (4.220) is

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & [1] & [-1] \\ [-8] & [-2] & [0] \end{bmatrix} \quad (4.222)$$

In most industrial applications the commanded signal $r(t)$ is assumed to be a constant. For example, in flight control such a command could represent the stick force coming from a pilot, or the guidance command coming from the outer-loop steering algorithms. Even though these command signals are not actually constant, designing and implementing a type 1 control system has proven very effective in most applications, and the RSLQR will provide zero steady-state error command tracking.

To achieve good transient response characteristics, tuning of the LQR PI matrices Q and R is required. Understanding how these matrices effect the control gains, and how the control gains influence the closed-loop system response, is key to achieving a good design.

It is important in the design of a realistic control system to be mindful of the “size” of the feedback gains in K_z . In aerospace applications, gains that are too large amplify sensor noise, drive the actuators with high rates, and cause issues and challenges with flexible body dynamics, called structural mode interaction. The feedback gains K_c depend upon the numerical values in Q and R . As $\|Q\|_2$ becomes large, the gains get large, as $\|R\|_2$ is made small, the gains get large, thus $\|K_z\|_2 \sim \|Q\|_2/\|R\|_2$.

Summary

$$\dot{x} = Ax + Bu + Ew$$

Dynamics:

$$y = Cx + Du$$

$$\text{Command model: } \overset{(p)}{r} = \sum_{i=1}^p a_i \overset{(p-i)}{r}; \text{ Disturbance model: } \overset{(p)}{w} = \sum_{i=1}^p a_i \overset{(p-i)}{w}$$

$$\text{State model: } \xi = \overset{(p)}{\dot{x}} - \sum_{i=1}^p a_i \overset{(p-i)}{x}; \text{ Control model: } \mu = \overset{(p)}{\dot{u}} - \sum_{i=1}^p a_i \overset{(p-i)}{u}$$

$$\text{Augmented state vector: } z = \begin{bmatrix} e & \dot{e} & \dots & \overset{(p-1)}{e} & \xi \end{bmatrix}$$

$$\text{Performance index: } J = \int_0^\infty (z^T Q z + \mu^T R \mu) d\tau \quad Q = Q^T \geq 0, R = R^T > 0$$

$$\text{Control design model: } \dot{z} = \tilde{A}z + \tilde{B}\mu; (\tilde{A}, \tilde{B}) \text{ controllable. } (\tilde{A}, Q^{1/2}) \text{ detectable.}$$

$$\tilde{A} = \begin{bmatrix} 0_{n_u \times n_u} & I_{n_u} & 0_{n_u \times n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & I_{n_u} & \dots & 0_{n_u \times n_u} & 0_{n_u \times n_x} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & 0_{n_u \times n_u} & \dots & I_{n_u} & 0_{n_u \times n_x} \\ -a_p I_{n_u} & -a_{p-1} I_{n_u} & -a_{p-2} I_{n_u} & \dots & -a_1 I_{n_u} & C \\ 0_{n_x \times n_u} & 0_{n_x \times n_u} & 0_{n_x \times n_u} & \dots & 0_{n_x \times n_u} & A \end{bmatrix};$$

$$\tilde{B} = \begin{bmatrix} 0_{n_u \times n_u} \\ 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ D \\ B \end{bmatrix}$$

$$\text{ARE: } P\tilde{A} + \tilde{A}^T P + Q - P\tilde{B}R^{-1}\tilde{B}^T P = 0 \quad \mu = -R^{-1}\tilde{B}^T P z = -K_z z$$

$$\text{Control: } u = -K_x x + \sum_{i=1}^p s^{-i} \left(a_i \left(\overset{(p-i)}{u} + K_x \overset{(p-i)}{x} \right) - K_i \overset{(p-i)}{e} \right)$$

Controller: $\dot{x}_c = A_c x_c + B_{c1}x + B_{c2}r$
 $u = C_c x_c + D_{c1}x + D_{c2}r$

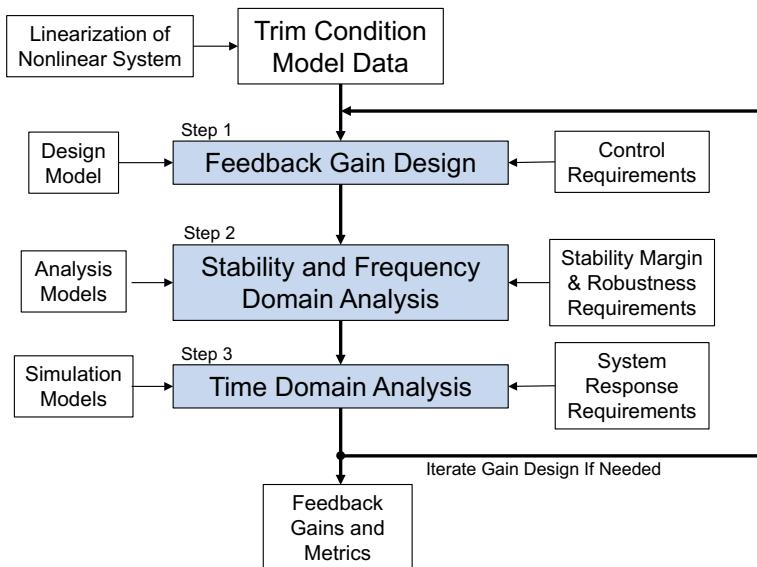
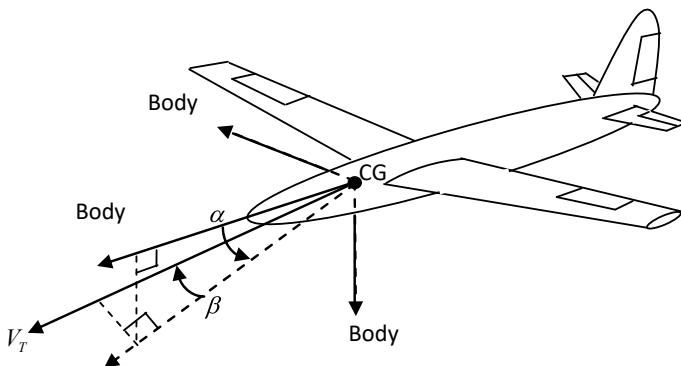
$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} 0_{n_u \times n_u} & I_{n_u} & \cdots & 0_{n_u \times n_u} \\ \vdots & \ddots & \cdots & 0_{n_u \times n_u} \\ 0_{n_u \times n_u} & 0_{n_u \times n_u} & \cdots & I_{n_u} \\ a_p I_{n_u} - D_c K_p & \cdots & \cdots & a_1 I_{n_u} - D_c K_1 \\ -K_p & \cdots & -K_2 & -K_1 \end{bmatrix} \begin{bmatrix} 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ C_c - D_c K_x \\ [-K_x] \end{bmatrix} \begin{bmatrix} 0_{n_u \times n_u} \\ \vdots \\ 0_{n_u \times n_u} \\ -I_{n_u} \\ [0] \end{bmatrix}$$

The following example will illustrate how to choose parameters within Q and how to select a design that performs well, has a reasonable bandwidth, and does not result in high actuator rates. The processes for selecting the LQR penalty weights forms LQR design charts that show important time domain and frequency domain metrics plotted versus loop gain-crossover frequency. Viewing this information, while using the design charts, allows the control system engineer to select the desired bandwidth of the design and to perform the necessary trade studies required to meet the desired closed-loop system performance design goals. This process also prevents large feedback gains from being selected, which can introduce challenges later in the simulation and analysis of the control system.

Example 4.10 LQR Q-Matrix Parameter Selection Using Design Charts In this example we introduce design charts to aid in selecting numerical values in the LQR weighting matrix Q . These charts plot important system design metrics versus loop gain-crossover frequency (LGCF) to provide insight and visualize how the design metrics vary as gains are increased in size to get a faster response. Figure 4.14 illustrates our model-based flight control design process discussed in Chap. 2. The figure shows a three-step design process for a single-equilibrium design point. The process begins with the linearization at an equilibrium design point. This populates the feedback control design model in Step 1 with trim condition model data. The feedback gain design model is the lowest-order model required to adequately represent the dynamics and variables to be controlled. Keeping this model low order reduces the number of feedback variables needed for implementation and reduces control system complexity. In Step 1 the feedback gains are designed and then analyzed in the frequency domain in Step 2 and in the time domain in Step 3.

Consider the pitch plane dynamics of an unpiloted aircraft (Fig. 4.15), given as

$$\begin{aligned} \dot{\alpha} &= \frac{Z_\alpha}{V} \alpha + \frac{Z_\delta}{V} \delta + q \\ \dot{q} &= M_\alpha \alpha + M_\delta \delta + M_q q \end{aligned} \quad (4.223)$$

**Fig. 4.14** Model-based control system design**Fig. 4.15** Unpiloted aircraft

It is desired to design an acceleration command $r = A_{zc}$ flight control system. We will assume that the command is constant and will design an RSLQR controller with integral feedback. We will design a constant gain matrix K_z for a single flight condition and will assume gain scheduling will be used to interpolate the gains between conditions (other design points). Normal acceleration A_z (ft/s^2) dynamics are described by

$$A_z = -V \dot{\gamma} = Z_\alpha \alpha + Z_\delta \delta \quad (4.224)$$

In Step 1 of Fig. 4.14 the RSLQR gain design model is defined. For this example, the design model is the same as in Example 4.5, (4.189). With $r = \text{constant}$, $\dot{r} = 0$, $p = 1$, we add a single integrator into the design model to form a Type-1 controller. The resulting RSLQR design model $\dot{z} = \tilde{A}z + \tilde{B}\mu$ state vector and the control input are

$$\begin{aligned} z &= [e \dot{x}^T]^T = [e_{A_z} \dot{\alpha} \dot{q}]^T \in R^3 \\ \mu &= \dot{\delta}_e \in R \end{aligned} \quad (4.225)$$

where $e_{A_z} = A_z - A_{z_c}$. Using (4.223) and (4.224) gives the RSLQR gain design model as

$$\begin{bmatrix} \dot{e}_{A_z} \\ \ddot{\alpha} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & Z_\alpha & 0 \\ 0 & Z_\alpha/V & 1 \\ 0 & M_\alpha & M_q \end{bmatrix} \begin{bmatrix} e_{A_z} \\ \dot{\alpha} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} Z_\delta \\ Z_\delta/V \\ M_\delta \end{bmatrix} \dot{\delta}_{e_c} \quad (4.226)$$

This will produce a (Proportional + Integral) state feedback control law for the aircraft.

$$u = -K_1 \int (A_z - A_{z_c}) dt - K_2 \alpha - K_3 q \quad (4.227)$$

The objective in the design is to provide a fast response and track constant acceleration commands with zero steady-state error, while avoiding using large gains. The RSLQR performance index (4.85) for this problem is

$$J = \int_0^\infty (z^T Q z + \mu^2 R) d\tau \quad (4.228)$$

Since the control is a scalar we set $R = 1$ [since R can be factored out of the integral in (4.228)]. The Q matrix is formed to penalize only the error state e in (4.225). This gives

$$z^T Q z = [e_{A_z} \dot{\alpha} \dot{q}] \begin{bmatrix} q_{11} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_{A_z} \\ \dot{\alpha} \\ \dot{q} \end{bmatrix} = q_{11} e_{A_z}^2, \quad (4.229)$$

Substituting (4.229) into (4.228) gives the performance index as

$$J = \int_0^\infty (q_{11} e^2 + \mu^2) d\tau \quad (4.230)$$

If we check observability of the pair $(\tilde{A}, Q^{1/2})$ we find the system to be observable through this choice of Q .

In Step 2 (Fig. 4.14), the frequency domain analysis model is formed. This model should include anything known about the system that produces significant phase lag (see discussion in Chap. 2). In this example, we will incorporate a second-order model of the actuator. We will use this model for both frequency domain analysis as well as time domain simulation (Step 3). The actuator model is given as

$$\ddot{\delta}_e = -2\zeta_a \omega_a \dot{\delta}_e + \omega_a^2 (\delta_c - \delta_e) \quad (4.231)$$

At a flight condition of Mach 0.3, 5000 ft altitude, and a trim angle of attack α of 5° , the open-loop plant model data (stability and control derivatives), speed, and actuator parameters are:

$$\begin{aligned} Z_\alpha/V &= -1.05273(1/\text{s}) \\ Z_\delta/V &= -0.0343(1/\text{s}) \\ M_\alpha &= -2.3294(1/\text{s}^2) \\ M_q &= -1.03341(1/\text{s}^2) \\ M_\delta &= -1.1684(1/\text{s}^2) \\ V &= 329.127(\text{ft/s}) \\ \omega_a &= 2\pi^* 13.(\text{rad/s}) \\ \zeta_a &= 0.6 \end{aligned} \quad (4.232)$$

Substituting the data into (4.226) yields

$$\tilde{A} = \begin{bmatrix} 0 & -346.48 & 0 \\ 0 & -1.0527 & 1 \\ 0 & -2.3294 & -1.0334 \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} -11.289 \\ -0.0343 \\ -1.1684 \end{bmatrix} \quad (4.233)$$

Before starting a gain design, it is important to check model controllability (see Chap. 2 Sect. 2.5.2). We evaluate (4.198) of the pair and find that the system is indeed controllable. The open-loop eigenvalues of \tilde{A} are $\lambda_i = 0, -1.04 \pm 1.53j$. To proceed, we will define the plant and controller models to be used for frequency domain and time domain analysis. The plant model is

$$\begin{aligned} \dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u \end{aligned} \quad (4.234)$$

where $x = [\alpha \ q \ \delta_e \ \dot{\delta}_e]^T$, $u = \delta_{e_c}$, and $y = [A_z \ \alpha \ q \ \delta_e \ \dot{\delta}_e]^T$. The state-space matrices are

$$\begin{bmatrix} A_p & B_p \\ C_p & D_p \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} Z_\alpha/V & 1 & Z_\delta/V & 0 \\ M_\alpha & M_q & M_\delta & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_a^2 & -2\zeta_a\omega_a \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_a^2 \end{bmatrix} \\ \begin{bmatrix} Z_\alpha & 0 & Z_\delta & 0 \\ I_4 & & & \end{bmatrix} & \begin{bmatrix} 0_{5 \times 1} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} -1.053 & 1 & -0.0343 & 0 \\ -2.329 & -1.033 & -1.168 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -6671.9 & -98.02 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6671.9 \end{bmatrix} \\ \begin{bmatrix} -346.48 & 0 & -11.289 & 0 \\ I_4 & & & \end{bmatrix} & \begin{bmatrix} 0_{5 \times 1} \end{bmatrix} \end{bmatrix} \quad (4.235)$$

The controller model implementing (4.227) is

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} y_{cmd} \\ u &= C_c x_c + D_{c1} y + D_{c2} y_{cmd} \end{aligned} \quad (4.236)$$

where $x_c = \int e_{A_z}$, $y = [A_z \ \alpha \ q \ \delta_e \ \dot{\delta}_e]^T$, and $y_{cmd} = A_{z_c}$. The state-space matrices are

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} & [-1] \\ [-K_1] & \begin{bmatrix} 0 & -K_2 & -K_3 & 0 & 0 \end{bmatrix} & [0] \end{bmatrix} \quad (4.237)$$

The LQR design charts are formed by sweeping q_{11} values from a small to large, solving for the feedback gains for each value of q_{11} , and examining the open-loop and closed-loop system properties. Figure 4.16 illustrates the process. The computation steps are:

1. Set $R = I_{n_u}$ and the value of q_{11} in Q from (4.229).
2. Solve the ARE using $(\tilde{A}, \tilde{B}, Q, R)$ and compute the feedback gain matrix K_z in (4.205).
3. Form the closed-loop system in (4.217). Compute the eigenvalues and store for plotting (root locus type plot)
4. Simulate the closed-loop system to a step command and extract time domain performance metrics. These are rise time, settling time, percent command overshoot, percent command undershoot, max control and control rate. Store the metrics and the time responses for plotting.

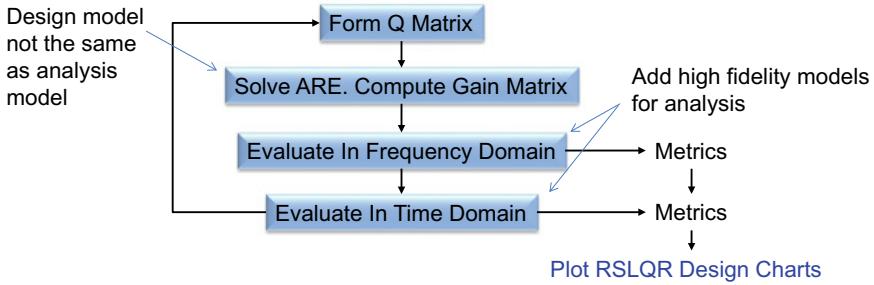


Fig. 4.16 Computational procedure for creating RSLQR design charts

5. Evaluate the design in the frequency domain and compute frequency domain metrics. At the plant input this includes the LGCF, $\|S_u\|_\infty$, and $\|T_u\|_\infty$, and plotting $\bar{\sigma}(L_u)$ and $\underline{\sigma}(I_{n_u} + L_u)$ versus frequency. At the plant output this includes computing $\|S_y\|_\infty$ and $\|T_y\|_\infty$ for the commanded variable. Since this example is a SIMO system we will also compute Nyquist and Bode plots. Store the frequency response metrics and curves for plotting.
6. Loop back to (1) and increase q_{11} until the numerical range is complete.
7. Plot the time domain and frequency domain metrics versus LGCF.
8. Plot the time responses and frequency response plots overlaying them so as to see how they evolve with increasing gain.

For this command tracking system, it is desired to track the acceleration command with zero error and minimize the rise time and settling time, all in response to the command, without driving the control surface actuators with large gains. Large gains will cause large actuator deflections and rates, which are not desirable. This creates a trade study, in which the bandwidth must be limited in order not to exceed actuator limitations. Also, large gains amplify sensor noise, reduce stability margins, and make the system sensitive to unmodeled high-frequency dynamics (like flexible body modes).

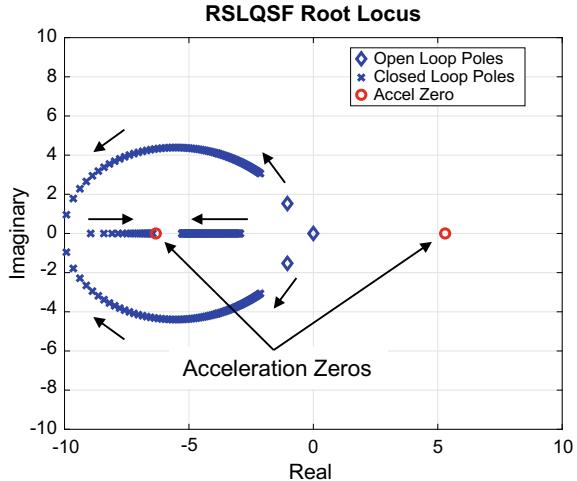
For this flight condition, the range of the LQR penalty q_{11} is selected to be $q_{11} = [10^{-2}, 10^{0.5}]$, using 100 design points. A small value of q_{11} will produce a slow response, $q_{11} = 10^{-2}$. A large value will produce a faster response, but since there are unmodeled dynamics (actuator) in the plant model large gains can cause the system to lose stability. When solving a new problem some experimentation with this range may be required.

For this linear system, the time response will depend upon the location of the closed-loop poles in the s-plane. Looping through the above calculations in Fig. 4.16, the eigenvalues of the closed-loop system matrix $(\tilde{A} - \tilde{B}K_z)$ are plotted to form a root locus. The data are shown in Fig. 4.17.

Also plotted are the open-loop poles (diamonds) and the zeros of the A_z/δ_e transfer function, which includes a non-minimum phase zero in the right half plane

Fig. 4.17 RSLQR

short-period dynamics root locus. Actuator poles at $-49.0 \pm 65.3j$ not shown



(RHP). The open-loop short-period dynamics are stable at this flight condition (diamonds), with the open-loop poles located in the left half plane (LHP). The two finite zeros of the acceleration transfer function A_z/δ_e are -6.33 and 5.29 . We see in the figure two closed-loop poles migrating toward the LHP zero at -6.33 . As discussed earlier on asymptotic properties of regulators and the root square locus, (4.172), Fig. 4.17 shows the RHP zero is mirrored into the LHP, and two of the closed-loop poles, one from the integrator and the other from the short-period, are approaching this region on the negative real axis. The remaining short-period pole moves out asymptotically to infinity along the negative real axis. The speed of response of this system will be limited by the two poles migrating toward the LHP zero at -6.33 . One pole will get canceled by the LHP zero, with the other pole defining the speed of response. As the LQR penalty q_{11} gets larger, this pole approaches the mirrored zero and will not move past it. As discussed in Chap. 2, the pole closest to the origin will dominate the response. As the gains get larger with increasing q_{11} the response will be limited by this poles' location in the LHP.

In response to a constant step command, the time domain response metrics of interest here are:

- 63% rise time, 95% settling time
- Percent overshoot, percent undershoot (because the system is non-minimum phase)
- max control (elevon) deflection, max control (elevon) rate.

In addition, we need to look at the frequency domain performance and robustness metrics, such as:

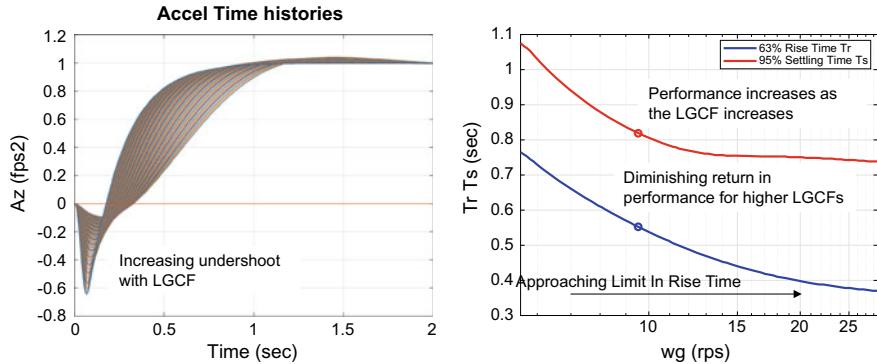


Fig. 4.18 Acceleration time histories and rise time (blue) and settling time (red) versus loop gain-crossover frequency ω_g

- $\sigma_S = 1/\|S_u\|_\infty$ and $\sigma_T = 1/\|T_u\|_\infty$ (used to compute singular value stability margins)
- $\|S_y\|_\infty$ and $\|T_y\|_\infty$ (indicates peaking in S and T for the acceleration signal).

These metrics, plotted versus ω_c , are used to determine how the increasing bandwidth of the system affects the system characteristics, indicating a desired value for q_{11} . The plots form the RSLQR design charts.

As with most control system design procedures, there is not a single answer to determining a set of gains that are acceptable. It is for the designer to make a reasonable selection. Once a suitable design is chosen, the associated gain matrix K_z is then stored in a table to create a gain-scheduled control for real-time implementation.

Figure 4.18 shows the acceleration time histories and the rise time and settling time plotted against LGCF ω_g .

As ω_g increases, the system responds more quickly to the step command. As seen from the figure, there is a diminishing in terms of speed of response as the LGCF increases. This is also evident from the root locus in Fig. 4.17. As the dominant poles approach the zero locations at -6.33 and -5.29 , the change in the pole location (the further it moves into the LHP) diminishes with the increasing gains. The poles headed toward infinity along the asymptotes continue to move, but their contribution to the response ($e^{\lambda t}$) dies quickly as the eigenvalues get large and negative. This result indicates that large gains do not make the system respond more quickly. There is a limit defined by the RHP zero that limits performance.

Figure 4.19 shows the percent overshoot, percent undershoot, max elevon (tail actuated control surface) deflection per g commanded, and max elevon rate per g commanded, versus the LGCF ω_g .

At lower values of ω_g , the response slightly overshoots the command, causing an overshoot. Command overshoot in flight control systems needs to be minimized in order to maintain limits and placards on the aircraft. As the integrator

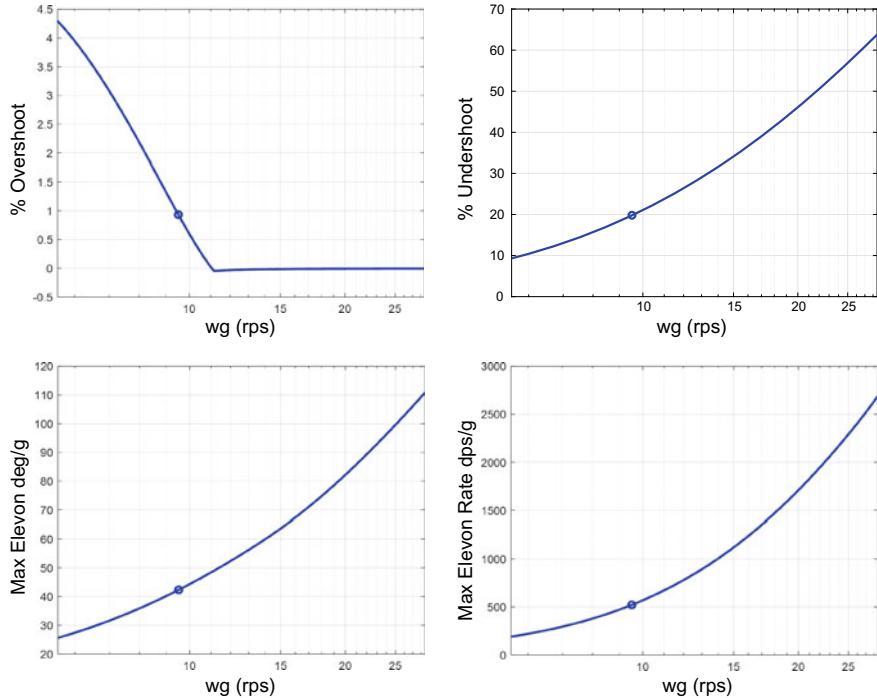


Fig. 4.19 Percent overshoot, undershoot, max elevon deflection, and rate versus loop gain-crossover frequency ω_c

gain increases (as q_{11} increases), above 11 rps ω_g the response has no command overshoot. This metric by itself indicates a desire for larger gains. The percent undershoot, characteristic of non-minimum phase responses, continues to increase with increasing ω_g . This response characteristic is undesirable and also needs to be minimized. Unfortunately, it increases with increasing ω_g . This metric indicates a desire for lower gains. Both the max deflection and max rate increase with increasing ω_g . It is critical in flight control systems not to have excessive deflections and rates in response to changes in the command. Electric actuators typically used in unpiloted aircraft systems draw current proportional to the peak rate (at these normal operating conditions). High rates then cause significant power draw. Also, if the surface becomes rate saturated, this nonlinear effect can significantly degrade stability. As shown in the figure, the deflection and rate increase almost exponentially with increasing ω_g . These metrics also indicate a need for lower gains. As seen in this figure, some of the metrics tend toward increasing the gains and some tend toward decreasing the gains.

Figure 4.20 shows two frequency response metrics, $\sigma_S = 1/\|S_u\|_\infty$ and $\sigma_T = 1/\|T_u\|_\infty$, plotted versus LGCF. In Chap. 3, Eq. 3.72, we showed how to compute singular value-based gain and phase margins using these quantities. The singular

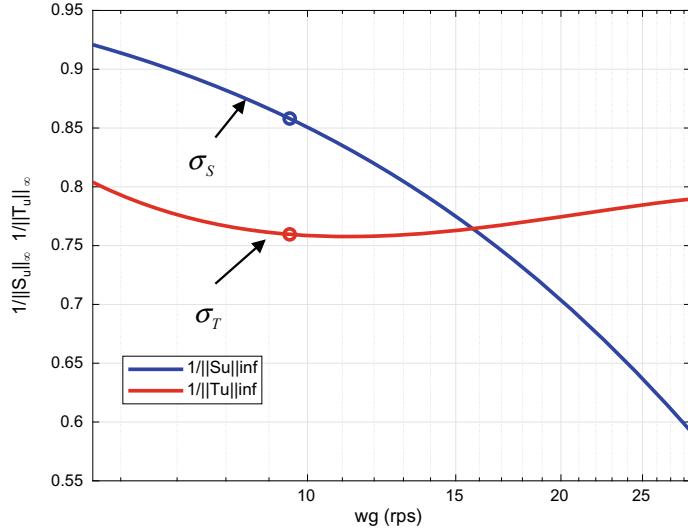


Fig. 4.20 Singular value frequency domain metrics versus loop gain-crossover frequency ω_g

value gain margin is $GM = [1 - \sigma_T, 1/1 - \sigma_S]$, which shows the larger these quantities the larger the GM (similar with phase margin). $\sigma_S = \min_{\omega} \underline{\sigma}(I + L_u)$ is the minimum of the minimum singular value of the return difference dynamics. We see that as the LGCF is increased the return difference matrix is becoming more singular. Since the frequency domain analysis model contains an actuator model, as the LGCF is increased the phase lag from the actuator destabilizes the system. This causes σ_S to decrease with increasing LGCF. This metric tends to favor lower gains. The metric $\sigma_T = 1/\|T_u\|_\infty$ is a measure of the damping in the dominant poles of the closed-loop system. We would like to maximize σ_T . The figure shows that this metric tends to favor larger gains.

In balancing the positive and negative trends indicated by these metrics, a design condition $q_{11} = 0.081113$ was selected. This is the value of q_{11} where the percent undershoot is 20%. This design point was selected to keep the undershoot no greater than 20%. For this design condition, the states A_z , q , δ_e , and $\dot{\delta}_e$ are plotted versus time in Fig. 4.21.

Note that there is minimal to the unit command. For this approach flight condition, the response is quick, without the use of large gains.

The gain matrix K_z is

$$K_z = [0.2848 \ -37.114 \ -9.1957] \quad (4.238)$$

The controller implementing this design is

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & [1 \ 0 \ 0 \ 0 \ 0] & [-1] \\ [-0.2848] & [0 \ 37.114 \ 9.1957 \ 0 \ 0] & [0] \end{bmatrix}. \quad (4.239)$$

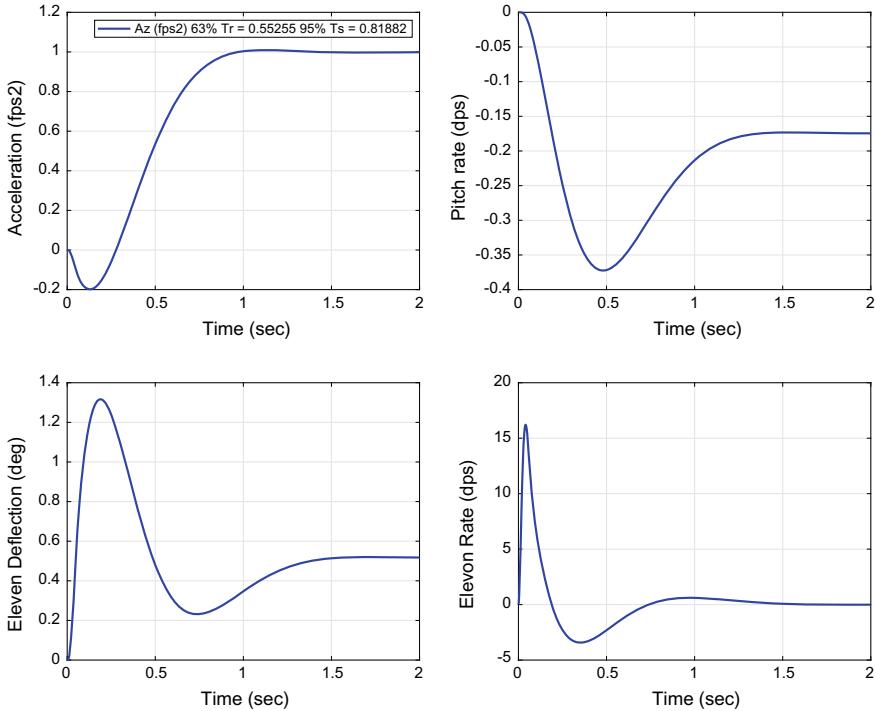


Fig. 4.21 States of the system responding to a unit acceleration step command

The basic problem of optimal control design consists of choosing a control policy such that some measure of performance for the corresponding closed-loop system is optimized with respect to the selected performance criterion.

4.5 Conclusions

In this chapter, we briefly discussed optimal control theory and the linear quadratic regulator and showed how to use this theory for command tracking servomechanism design problems. Many control systems today are designed using this method due to its optimality, frequency domain guarantees, and the ease of the design. For command tracking control systems a certain number of integrators are needed to provide zero steady-state tracking error. In this chapter we also discussed how to formulate this problem within a state-space framework, and how to use optimal control to design the command tracking control system. We also discussed the excellent frequency domain properties of LQR controllers, and for our robust servomechanism controllers we have these same excellent properties.

One of the key take-a-ways from the chapter should be the systematic process for development of design charts for selecting numerical weights in optimal control

problems. It is very easy to use too large of numerical weightings in the LQR performance index, and these large weights would lead to high gains and a high loop gain-crossover frequency. It is critical to be able to determine the bandwidth that is needed in the design to meet performance requirements and not to drive the actuation system implementing the control too hard.

4.6 Exercises

Exercise 4.1 Consider

$$\begin{aligned}\dot{x}_1 &= x_2 & t_0 &= 0 & x(0) &= \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\ \dot{x}_2 &= -2x_1 - 3x_2 + u & T &= 1 \\ J &= \int_0^T (x_1^4 + u^2) d\tau + x_1^2(T) + x_2^2(T)\end{aligned}$$

Set up (but not solve) the HJB equation with the corresponding boundary conditions.

Exercise 4.2 Given,

$$\begin{aligned}\dot{x} &= -x + u \\ J &= \int_0^1 (x^2 + u^2) d\tau + x^2(1)\end{aligned}$$

use the ARE to find the optimal feedback control $u^*(t) = -K x(t)$. Draw the closed-loop system block diagram.

Exercise 4.3 For

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_1 + u \\ J &= \int_0^\infty (x_1^2 + u^2) d\tau\end{aligned}$$

find the LQR-optimal control policy u to minimize the cost J .

Exercise 4.4 Consider the longitudinal aircraft dynamics given in Chap. 1, Exercise 1.2. This linear model represents the aircraft incremental dynamics, with respect to a trim condition. Design an infinite-time LQR to regulate the state vector to zero. Simulate the design with an initial state vector $x(0) = [10 \text{ ft/s} \ 0.1 \text{ rad} \ 0.1 \text{ rad/s} \ 0 \text{ rad}]$.

Exercise 4.5 Consider a second-order system modeled by the input–output equation $\ddot{y} = u$. A feedback controller $u = -k_1 y - k_2 \dot{y}$ is to be designed such that the performance index

$$J = \int_0^\infty (4y^2 + u^2) d\tau$$

is minimized.

- (a) Find k_1 and k_2 .
- (b) What are the closed-loop eigenvalues?

Exercise 4.6 Consider the following system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u, \quad y = \begin{bmatrix} 0 & \gamma^2 \end{bmatrix}x$$

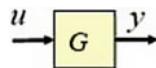
For $\gamma = 2$, an LQR state feedback controller $u = -k_1 x_1 - k_2 x_2$ is to be designed such that the performance index

$$J = \int_0^\infty (y^2 + u^2) d\tau$$

is minimized.

- (a) What are the open-loop eigenvalues?
- (b) Compute optimal gains k_1 and k_2 .
- (c) What are the closed-loop eigenvalues?
- (d) Consider $\lim \gamma \rightarrow 0$, what will the closed-loop eigenvalues be?
- (e) Consider $\lim \gamma \rightarrow \infty$, what will the closed-loop eigenvalues be?

Exercise 4.7 Consider the system with input u and output y shown in the block diagram:



where $G(s) = \frac{s-2}{s-1}$.

- (a) Derive a state-space model for this system.

- (b) Using the state-space model from a), design a linear quadratic regulator (LQR) optimal controller using state feedback that minimizes the cost index $J = \int_0^\infty (y^2(\tau) - 2x(\tau)u(\tau))d\tau$ where y is the output shown in the block diagram. List the (A, B, Q, R) matrices, show the Riccati equation, its solution, and the state feedback gain matrix K_x that results.
- (c) Form the Hamiltonian matrix for this LQR problem. What are the eigenvalues of the Hamiltonian matrix.
- (d) It is desired to form a closed-loop state-space model using the plant and common controller form so that the system state x , output y and the control u from the block diagram are available for plotting.

$$\text{Plant: } \begin{aligned} \dot{x}_p &= A_p x_p + B_p u \\ y_p &= C_p x_p + D_p u \end{aligned} \quad \text{Common Controller: } \begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y_p + B_{c2} r \\ u &= C_c x_c + D_{c1} y_p + D_{c2} r. \end{aligned}$$

- (i) Define the matrices (A_p, B_p, C_p, D_p) for the above plant state-space model with input u and output y_p such that the plant connects to the state feedback controller and has the variables defined for plotting.
- (ii) Define the matrices $(A_c, B_{c1}, B_{c2}, C_c, D_{c1}, D_{c2})$ for the above common controller state-space model with input y_p and output u implementing the LQR state feedback control such that the plant connects to the state feedback controller.

Exercise 4.8 Consider the system $G(s)$ with input u and output y from Exercise 4.7.

- (a) Derive a state-space model for this system.
- (b) It is desired for the output y to track a constant command r with zero error. Derive the robust servomatrices (\tilde{A}, \tilde{B}) .
- (c) Using (\tilde{A}, \tilde{B}) from b) and a cost index $J = \int_0^\infty ((\int e)^2 + u^2)d\tau$, design a state feedback optimal control. List your Riccati matrix and the gain matrix.
- (d) What are the closed-loop eigenvalues for this system?
- (e) It is desired to form a closed-loop state-space model using the plant and common controller form so that the system state x and output y from the block diagram are available for plotting.

$$\text{Plant: } \begin{aligned} \dot{x}_p &= A_p x_p + B_p u \\ y_p &= C_p x_p + D_p u \end{aligned} \quad \text{Common Controller: } \begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y_p + B_{c2} r \\ u &= C_c x_c + D_{c1} y_p + D_{c2} r. \end{aligned}$$

- (i) Define the matrices (A_p, B_p, C_p, D_p) for the above plant state-space model with input u and output y_p such that the plant connects to the state feedback controller and has the variables defined for plotting.
- (ii) Define the matrices $(A_c, B_{c1}, B_{c2}, C_c, D_{c1}, D_{c2})$ for the above common controller state-space model with input y_p and output u implementing the LQR state feedback control such that the plant connects to the state feedback controller.
- (f) Sketch the output y from a constant command $r = 1$.

- (g) Give an expression (do not numerically evaluate it) to compute the steady-state value for the state of the plant in response to the constant command $r = 1$.

Exercise 4.9 Consider the linear system $\dot{x} = Ax + Bu$ with LQR performance index

$$J = \frac{1}{2} \int_0^{\infty} (q x^T Q x + \rho u^T R u) d\tau$$

where $Q = Q^T \geq 0$, $R = R^T > 0$, and $(A, Q^{1/2})$ observable. Use the state feedback control $u = -Kx$ with $K = R^{-1}B^T P$, where P is the ARE solution matrix. What happens to the eigenvalues of the closed-loop system as:

- (a) $q \rightarrow 0$.
- (b) $\rho \rightarrow 0$.

Exercise 4.10 Consider the following scalar linear quadratic command tracking problem

$$\begin{aligned} \dot{x} &= u \\ J &= \frac{1}{2} \int_0^T [(x - r)^2 + \rho u^2] d\tau + \frac{1}{2} q_T (x(T) - r(T))^2 \end{aligned}$$

with $q_T > 0$, $\rho > 0$, T fixed, and a known reference (command) input $r(t)$.

- (a) What is the HJB equation for this problem? (Eliminate u). Include boundary conditions.
- (b) Find a solution for J^* in the form:

$$J^*(x, t) = \frac{1}{2} P(t)x^2 + g(t)x + w(t)$$

Find differential equations for P , g , and w , such that the HJB equation is satisfied. Include boundary conditions. Derive but do not solve the related equations.

References

1. Athans M, Falb PL (2006) Optimal control: an introduction to the theory and its applications. Dover Books on Engineering
2. Kwakernaak H, Sivan R (1972) Linear optimal control systems. Wiley, New York
3. Anderson BDO, Moore JB (1990) Optimal control, linear quadratic methods. Dover, Mineola, New York

4. Khalil H (1996) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, NJ, p 07458
5. Bellman RE (2003) Dynamic programming. Dover, Mineola, New York
6. Francis BA, Wonham WM (1976) The internal model principle of control theory. *Automatica* 12(5):457–465
7. Davidson EJ, Goldenberg A (1975) Robust control of the servomechanism problem. *Automatica* 11:461–471



State Feedback H_∞ Optimal Control

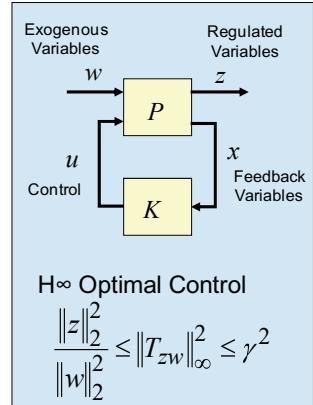
5

This chapter presents full information state feedback H_∞ optimal control. This control synthesis method uses state-space methods to achieve stability, performance, and robustness and allows for the direct loop shaping in the frequency domain. The chapter begins with an introduction of various norms used in control system design and analysis, followed by methods of specifying stability and performance specifications in the frequency domain. This logically leads into loop shaping using frequency-dependent weights. The state feedback control law is then synthesized using an algebraic Riccati equation approach called γ -iteration. This method is applied to a UAV design example. This control synthesis method is an excellent approach that teaches design engineers important properties in both the time domain and frequency domain, and more importantly how to achieve these properties in a closed-loop design.

5.1 Introduction

Since control theory became an engineering discipline, mathematicians and engineers have searched for control system design methods that would simultaneously satisfy stability, performance, and robustness requirements in a single design step. In the 1980s, this problem was posed for multi-input–multi-output (MIMO) systems and the design method called H_∞ optimal control emerged. This method allows the engineer to design, using state-space models, a controller that satisfies important frequency domain requirements, often referred to as loop shaping. These requirements include shaping the sensitivity function $S(s)$, complementary sensitivity $T(s)$, the control activity $U(s)$, the loop gain $L(s)$ and its associated crossover frequency ω_c (rad/s).

Fig. 5.1 H_∞ optimal control block diagram



In flight control systems for manned and unmanned aircraft configurations, robust performance and stability requirements necessitate the use of optimally designed flight control systems to achieve stability, command tracking performance, to minimize control effort, and to be robust to inaccuracies in the model description. Robust performance requirements are generally driven by high maneuver rates needed for agile flight. Robust stability requirements are often related to large flight envelopes and uncertainties in the plant dynamics created by uncertain aerodynamics, actuation, and flexible body dynamics.

H_∞ optimal control allows the control system engineer to address these challenges in the design of the flight control system. The topology of a general H_∞ controller design problem is shown in Fig. 5.1. A state-space model for the plant is

$$\begin{aligned}\dot{x} &= Ax + Bu + Ew \\ z &= Cx + D_1u + D_2w\end{aligned}\tag{5.1}$$

where $x \in R^{n_x}$ is the state, $u \in R^{n_u}$ the control, $w \in R^{n_w}$ the exogenous disturbance, and $z \in R^{n_z}$ a collection of variables to be regulated. The design goal is to minimize the regulated variables z in response to the exogenous input w while providing internal stability. This is equivalent to minimizing the infinity norm of the transfer function matrix. If all the states are available for feedback, then the resulting H_∞ problem is referred to as a full information feedback problem. The solution results in a feedback compensator whose feedback gains are calculated by solving a single Algebraic Riccati Equation.

The state-space solution of linear H_∞ optimal control problems can be found in Doyle et al. [1]. This same problem of reducing the H_∞ norm of a closed-loop system has been viewed as a two-person zero-sum differential game in Basar and Bernhard [2], where the solution is related to certain Algebraic Riccati Equations. This approach, for nonlinear systems, has been pursued in Basar

and Bernhard [2] and in Helton [3]. For nonlinear systems, the Riccati equation is replaced with a particular Hamilton–Jacobi equation known as Isaacs equations [4, p. 67, Eq. (4.2.1)]. This type of optimal control is referred to as nonlinear H_∞ and/or \mathcal{L}_2 -gain optimal control. A design example can be found in Wise and Sedwick [5].

We begin with a review of common norms for signal and systems, proceed to show how to engineer both stability and performance specifications in the frequency domain, and then demonstrate how to achieve loop shaping using frequency-dependent weights. The loop shaping ideas presented here are very similar to using lead-lag filters, low-pass filters, notch filters, etc., from classical control theory. If the reader is not familiar with frequency domain analysis then this chapter should be reviewed prior to working through this chapter. Once an understanding of how the state-space design model is engineered, the full information state feedback controller is derived. A flight control design example using an unmanned aircraft pitch autopilot is presented to show how to implement the concepts of this chapter.

5.2 Norms for Signals and Systems

In control system design we are usually concerned with the “size” of certain signals within the system. These signals may be commands, errors, states, outputs, or internal variables within the dynamics. In Chap. 1 we discussed norms applied to vectors and matrices. In this section we extend these ideas to linear systems. Consider piece-wise continuous scalar signals $u(t)$ which map $(-\infty, \infty)$ to $(-\infty, \infty)$. Define the 1-norm, 2-norm, and ∞ -norm as

$$\begin{aligned}\|u\|_1 &= \int_{-\infty}^{\infty} |u(t)| dt \\ \|u\|_2 &= \left(\int_{-\infty}^{\infty} |u(t)|^2 dt \right)^{\frac{1}{2}} \\ \|u\|_{\infty} &= \sup_t |u(t)|\end{aligned}\tag{5.2}$$

Suppose u is a current through a 1Ω resistor. Then the power is equal to u^2 , and the total energy is the integral of u^2 , which is the norm $\|u\|_2^2$.

Power Signals

The average power of a signal is the average of its instantaneous power. The average power of u is

$$\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T u^2(t) dt \quad (5.3)$$

If this limit exists then the signal is called a power signal, and we denote the limit as $\text{pow}(u)$, given as

$$\text{pow}(u) = \left(\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T u^2(t) dt \right)^{1/2} \quad (5.4)$$

The $\text{pow}(u) = \left(\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T u^2(t) dt \right)^{1/2}$ is not a norm. It does not satisfy the axiom that $\|u\| = 0 \rightarrow u(t) = 0$ for all $t \in (-\infty, \infty)$. Nonzero signals can have zero average power.

Example 5.1 If $\|u\|_2 < \infty$ then $\text{pow}(u) = 0$. Consider $\|u\|_2 < \infty$, then

$$\frac{1}{2T} \int_{-T}^T u^2(t) dt \leq \underbrace{\frac{1}{2T} \|u\|_2^2}_{\rightarrow 0 \text{ as } T \rightarrow \infty} \quad (5.5)$$

Since $\|u\|_2^2 < \infty$, taking the limit $T \rightarrow \infty$, yields $\text{pow}(u) = 0$.

Example 5.2 If $\|u\|_\infty < \infty$ then $\text{pow}(u) \leq \|u\|_\infty$. Using (5.5), we get

$$\frac{1}{2T} \int_{-T}^T u^2(t) dt \leq \frac{1}{2T} \int_{-T}^T \|u\|_\infty^2 dt = \|u\|_\infty^2 \frac{1}{2T} \int_{-T}^T dt = \|u\|_\infty^2 \quad (5.6)$$

Norms for Systems

Consider the norms for stable scalar transfer functions in which

$$\frac{1}{2T} \int_{-T}^T u^2(t) dt \leq \frac{1}{2T} \int_{-T}^T \|u\|_\infty^2 dt = \|u\|_\infty^2 \frac{1}{2T} \int_{-T}^T dt = \|u\|_\infty^2$$

The convolution response of the system is

$$y = G * u \rightarrow y(t) = \int_{-\infty}^{\infty} g(t - \tau)u(\tau)d\tau \quad (5.7)$$

Typical terms for the transfer function are:

- G stable \rightarrow that G is analytic in the closed RHP ($\text{Re } s \geq 0$).
- G proper $\rightarrow G(j\omega)$ is finite (order of the denominator \geq order of numerator).
- G strictly proper $\rightarrow G(j\infty) = 0$ (order of denominator $>$ order of numerator).
- $G(j\infty) = 0$ biproper $\rightarrow G$ and G^{-1} are both proper.

From Parseval's theorem, for a stable G , we have

$$\|G\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega \right)^{1/2} = \left(\int_{-\infty}^{\infty} g^2(t) dt \right)^{1/2} = \|g\|_2 \quad (5.8)$$

For a stable G , $\|G\|_2$ is finite if and only if G is strictly proper with no poles on $j\omega$ axis. For a strictly proper G with no poles on $j\omega$ axis the $\|G\|_2^2$ can be expressed as,

$$\begin{aligned} \|G\|_2^2 &= \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega \\ &= \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} G(-s)G(s)ds = \frac{1}{2\pi j} \oint G(-s)G(s)ds \end{aligned} \quad (5.9)$$

which can be evaluated using residues of the transfer function. $\|G\|_\infty$ is the peak of the Bode plot of G . $\|G\|_\infty$ is finite if and only if G is proper with no poles on $j\omega$ axis. Also, there is a submultiplicative property of the ∞ -norm: $\|GH\|_\infty \leq \|G\|_\infty \|H\|_\infty$, which allows us to bound the combined system via norms on its elements. The above norms for signals and systems allow us to form and understand the amplification or attenuation in the responses of systems and signals of interest.

If we know how big the input signal u is, how big will the output y be? Table 5.1 illustrates this norm relationship for a stable strictly proper G .

Example 5.3 The (1,1) table entry above is formed as follows:

$$y(t) = \int_{-\infty}^{\infty} g(t - \tau)u(\tau)d\tau = \int_{-\infty}^{\infty} g(t - \tau)\delta(\tau)d\tau = \int_{-\infty}^{\infty} g(t)d\tau = g(t)$$

Table 5.1 Output signal norms for stable transfer functions with specific input signals

	$u(t) = \delta(t)$	$u(t) = \sin(\omega t)$
$\ y\ _2$	$\ G\ _2$	∞
$\ y\ _\infty$	$\ G\ _\infty$	$ G(j\omega) $
Pow(y)	0	$\frac{1}{\sqrt{2}} G(j\omega) $

Table 5.2 Output signal norms for stable transfer functions with specific input signals

	$\ u\ _2$	$\ u\ _\infty$	Pow(u)
$\ y\ _2$	$\ G\ _\infty$	∞	∞
$\ y\ _\infty$	$\ G\ _2$	$\ G\ _1$	∞
Pow(y)	0	$\leq \ G\ _\infty$	$\ G\ _\infty$

$$\|y\|_2 = \|G\|_2 = \|g\|_2 \quad (5.10)$$

Suppose u is not fixed as in the above table, but can be any signal with 2-norm ≤ 1 . The result is often called the system gain and is equal to the $\|G\|_\infty$. Table 5.2 illustrates the response for finite 2-norm, ∞ -norm, and pow signals.

Example 5.4 The (1,1) table entry above is formed as follows. For $\|u\|_2 \leq 1$ we want $\|y\|_2$:

$$\|y\|_2^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)|^2 |u(j\omega)|^2 d\omega \leq \|G\|_\infty^2 \frac{1}{2\pi} \int_{-\infty}^{\infty} |u(j\omega)|^2 d\omega = \|G\|_\infty^2 \|u\|_2^2 \quad (5.11)$$

Computing Norms for Systems

For single-input-single-output linear time-invariant systems,

$$\begin{aligned} \dot{x} &= Ax + bu \\ y &= cx \end{aligned} \quad (5.12)$$

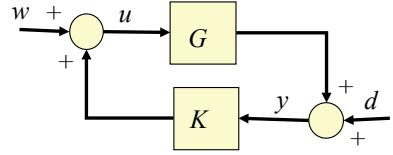
the transfer function is $G(s) = c(sI - A)^{-1}b$. If the system matrix A is stable, the matrix exponential

$$e^{At} = I + tA + \frac{t^2}{2!}A^2 + \dots \quad (5.13)$$

converges uniformly in time. Let

$$P = \int_0^\infty e^{A\tau} bb^T e^{A^T\tau} d\tau \quad (5.14)$$

Fig. 5.2 Block diagram of a linear closed-loop system



then

$$AP + PA^T + bb^T = 0 \quad (5.15)$$

and the 2-norm of G is given by

$$\|G\|_2 = \left(cPc^T \right)^{1/2} \quad (5.16)$$

Proof

$$G(t) = ce^{At}b$$

$$\|G\|_2^2 = \int_0^\infty ce^{A\tau} bb^T e^{A^T \tau} c^T d\tau = c \int_0^\infty e^{A\tau} bb^T e^{A^T \tau} d\tau c^T = cPc^T \quad (5.17)$$

Well Posedness and Stability

Consider the system interconnection shown in the block diagram of Fig. 5.2.

The loop equations for the system shown in Fig. 4.2 are

$$\begin{aligned} w &= u - Ky \\ d &= y - Gu \end{aligned} \quad (5.18)$$

Arranging inputs and outputs into a vector yields

$$\begin{bmatrix} w \\ d \end{bmatrix} = \begin{bmatrix} I & -K \\ -G & I \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} \quad (5.19)$$

Solving for the outputs yields:

$$\begin{bmatrix} u \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}}_H^{-1} \begin{bmatrix} w \\ d \end{bmatrix} \quad (5.20)$$

Suppose G and K are proper, and let H denote the closed-loop transfer function matrix. In this case, the feedback system is well posed if and only if

$$\det(I - G(\infty)K(\infty)) \neq 0$$

The system is internally stable if and only if H is stable.

Proof

$$\det \begin{bmatrix} I & -K \\ -G & I \end{bmatrix} = \det \begin{bmatrix} I & 0 \\ -G & I - GK \end{bmatrix} \det \begin{bmatrix} I & -K \\ 0 & I \end{bmatrix} = \det[I - GK] \quad (5.21)$$

So, it is easy to see that the closed-loop transfer function H will be proper if and only if $\det(I - G(\infty)K(\infty)) \neq 0$.

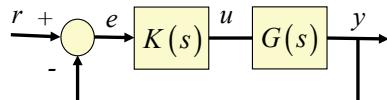
5.3 Stability and Performance Specifications in the Frequency Domain

Classical control system design methods using transfer functions were focused around achieving certain frequency response characteristics, viewed from either Bode, Nyquist, or Nichols charts. For single-input-single-output systems, these classical methods easily incorporated command tracking performance, stability margin, and plant roll-off design features that achieved the desired system response characteristics. When control system design using state-space methods was introduced to address multi-input-multi-output control design challenges, one of the complaints raised about the design methods was the lack of focus, or attention, toward achieving frequency domain properties. In this section we will discuss how to achieve frequency response design goals within a state-space format. Readers not familiar with frequency response analysis should review Chap. 5 before proceeding into designing controllers using H_∞ optimal control.

Consider the control system shown in Fig. 5.3. For this system the loop gain at the plant input $L_u(s) = K(s)G(s)$ is a square matrix that has dimension equal to the number of inputs for the system, with units equal to those variables in the control vector u . Figure 5.4 illustrates frequency domain requirements for $L_u(s)$.

In order to track commands at low frequency the loop gain must have sufficient magnitude. In order to be robust to high-frequency noise and unmodeled high-frequency dynamics, the loop gain must roll off and be sufficiently small. In the frequency band between these conflicting requirements is where the loop gain

Fig. 5.3 Feedback control system



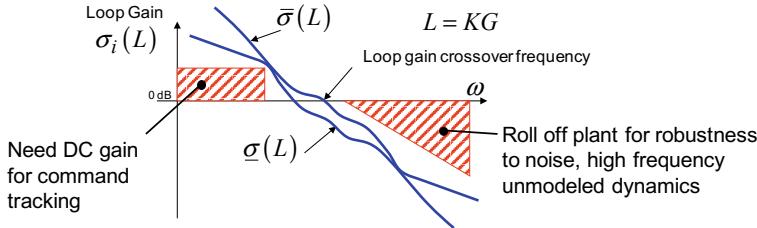


Fig. 5.4 Singular value frequency response requirements for a loop transfer function

crosses 0 dB and defines the loop gain-crossover frequency, ω_c . As illustrated in the figure, the singular values of $L_u(s)$ are $\sigma_i(L_u)$, with $\underline{\sigma}(L_u)$ and $\bar{\sigma}(L_u)$ denoting the minimum and maximum, respectively. We refer to the frequency at which $\bar{\sigma}(L_u)$ crosses 0 dB as the loop gain-crossover frequency ω_c .

For SISO systems, $L_u(s)$ is a scalar with singular value $\underline{\sigma}(L_u) = \bar{\sigma}(L_u) = |L_u|$. Large gain at low frequencies would be obtained by using a large proportional gain and/or integral control (type 1 control). Roll-off at high frequencies would be obtained by using low-pass or elliptical filters, depending upon the amount of roll-off needed.

Stability margins would be computed from $L_u(s)$ to indicate the robustness of the design. As discussed in Chap. 3, for MIMO systems the singular value margins are computed from the sensitivity $S_u(s) = (I_{n_u} + L_u(s))^{-1}$ and complementary sensitivity $T_u(s) = (I_{n_u} + L_u^{-1}(s))^{-1}$ computed at the plant-input loop breakpoint.

Command tracking performance can also be viewed by examining the sensitivity function $S_y(s)$, given by

$$e(s) = (I + L_y(s))^{-1} r(s) = S_y(s) r(s) \quad (5.22)$$

Figure 5.5 illustrates singular value frequency response requirements for the sensitivity function $S_y(s)$.

At low frequency, where $L_y(s)$ needs to be large, $S_y(s)$ needs to be small. At high frequencies where $L_y(s)$ needs to be small, $S_y(s)$ is near unity. From

Fig. 5.5 Singular value frequency response requirements for the sensitivity function

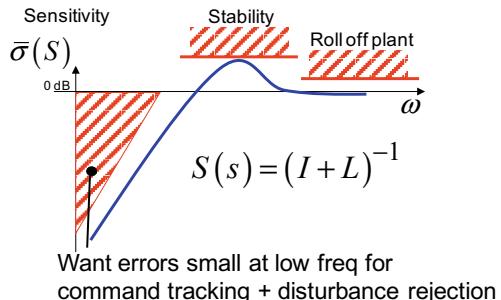
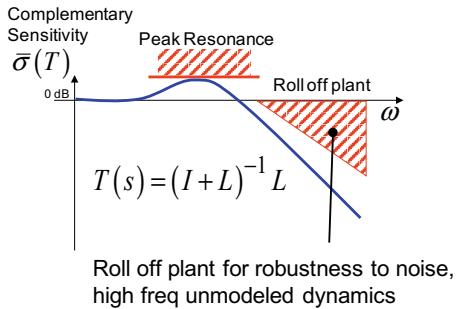


Fig. 5.6 Singular value frequency response requirements for the complementary sensitivity function



Chap. 3 we know that stability margins are determined from the near singularity of the return difference $(I + L_y = S_y^{-1})$ as measured by its minimum of $\underline{\sigma}(I + L_y)$ versus frequency, which equates to the peak of the sensitivity, or $\|S\|_\infty$. We know from the Bode integral log theorem that as we push the sensitivity lower in magnitude to achieve faster response, the peak pops creating a system that is more sensitive and less stable.

The complementary sensitivity $T_y(s)$ is defined as:

$$y(s) = (I + L_y)^{-1} L_y r(s) = T_y(s) r(s) \quad (5.23)$$

Figure 5.6 illustrates the complementary sensitivity function $T_y(s)$ singular value frequency response requirements.

This is also known as the closed-loop transfer function. At low frequencies where the loop gain is large, $T_y(s)$ is near unity. At high frequencies where the loop gain must roll off to be robust to high-frequency noise and unmodeled dynamics, $T_y(s)$ is small. At the peak of $T_y(s)$, $\|T_y\|_\infty$, the closed-loop system has a resonance in which frequencies at the peak are amplified by the system. If we were to approximate the system with an equivalent second-order system, the peak indicates low damping. This indicates that the dominant poles of the system are close to the $j\omega$ axis.

The control activity is amount of control used in responding to commands and rejecting disturbances. In general, it is desirable to minimize control usage at all frequencies, making sure that the actuators responding to the control signals are not position or rate saturated. In the frequency domain, constant weight on penalizing the control activity is usually used.

5.4 Loop Shaping Using Frequency-Dependent Weights

Figures 5.4, 5.5, and 5.6 show how to shape the loop gain, sensitivity, and complementary sensitivity to achieve command tracking performance, robustness to high-frequency noise and unmodeled dynamics, as well as acceptable stability margins. These concepts are central to designing H_∞ optimal controllers. The design

procedure minimizes the ∞ -norm of a system response matrix. This response matrix contains a frequency weighted sensitivity, complementary sensitivity, and control activity. The procedure for building a state-space design model is straightforward.

The weighting filters used to shape the loops in the H_∞ optimal control design should be selected to be minimum order. Each state in the weighting filters adds a state to the controller. In gain-scheduled flight control applications, high-order controllers can introduce transients in the response as the scheduling variables change. Thus, low-order controllers are typically desirable.

Consider the block diagram shown in Fig. 5.7 for the plant model in (5.1). The scalar variable z_1 is a weighted error variable to be regulated. The idea is to weight the error response to a command, $e = S_y r$, with a weighting filter W_S that is the inverse of the desired shape for S_y , so that when the $\|W_S S_y\|_\infty$ is minimized, it will shape S_y . Figure 4.8 illustrates this design process.

The second regulated variable in Fig. 5.7 is z_2 which is the weighted complementary sensitivity. Figure 5.9 illustrates the design process for shaping the complementary sensitivity.

The third regulated variable in Fig. 5.7 is z_3 which is a weighted control activity. This variable is multiplied here by a constant to penalize control activity at all frequencies. If it was needed to penalize some frequencies more than others, the process for selecting the weighting filter would be similar to that shown in Figs. 5.8 and 5.9.

The numerical choice of weighting filters will change with each application. Designing these filters is often difficult, with the degree of difficulty being comparable to selecting lead-lag filters in classical control design for improving gain or

Fig. 5.7 Block diagram showing weighted sensitivity, control activity, and complementary sensitivity

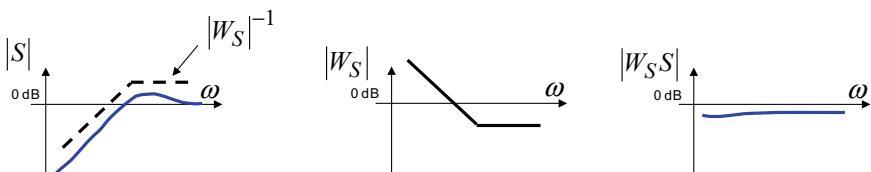
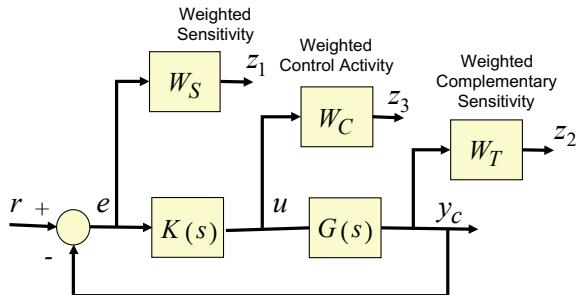


Fig. 5.8 Process for shaping the sensitivity function through a shaping filter

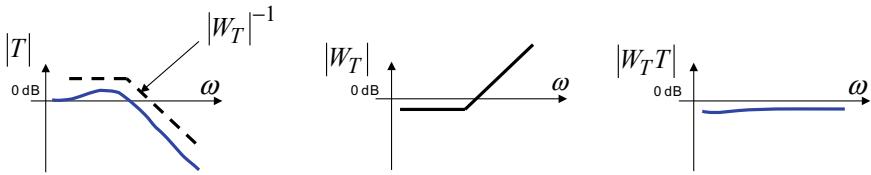


Fig. 5.9 Process for shaping the complementary sensitivity function through a shaping filter

phase margins. Figure 5.10 illustrates a typical sensitivity frequency response with low-frequency command tracking and stability margin requirements, and a typical first-order weighting filter for achieving the shape. The low-frequency behavior of W_S demonstrates an integrator property with a slope of 20 dB per decade. The gain K in W_S is chosen to produce a magnitude of W_S of -3 dB at the desired loop gain-crossover frequency ω_c . The zero in W_S is chosen at the desired ω_c . The -3 dB magnitude will limit the peak of S_y ($\|S_y\|_\infty$) thus producing adequate stability margins. These design rules and model for W_S can be used to shape S and keep the order of the weighting filter to a low number.

Figure 5.11 illustrates a typical complementary sensitivity frequency response which constrains the peak (peak resonance) and adds roll-off for robustness to uncertain and unmodeled high-frequency dynamics. The first-order weighting filter used here is similar in shape to a lead-lag filter. The low-frequency behavior of W_T demonstrates a flat profile versus frequency which will constrain the peak resonance. The zero in W_T can be chosen smaller than the desired ω_c , with the gain K chosen so that the $|W_T|$ is 0 dB at ω_c . Some iteration of this may be needed to converge and obtain a desirable constraint for the peak resonance. The pole in W_T is chosen somewhat arbitrarily and should be chosen high enough in frequency to provide the minimum attenuation needed in the high frequency range. One should not make it too high in frequency so as to keep the digital implementation of the control algorithm (via a computer) reasonable.

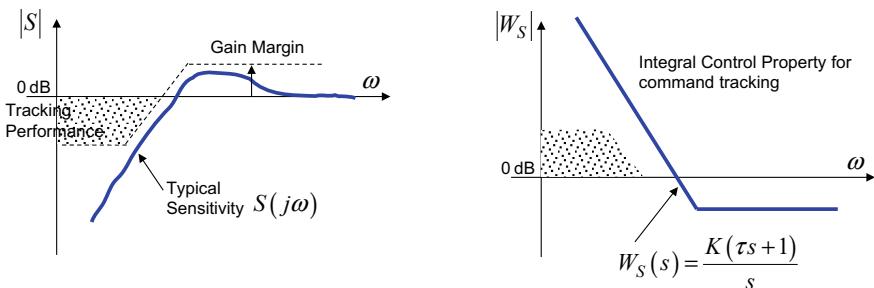


Fig. 5.10 First-order sensitivity weighting filter design

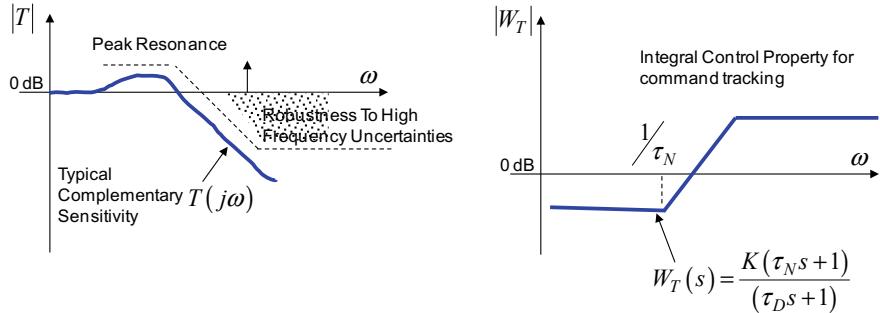


Fig. 5.11 First-order complementary sensitivity weighting filter design

5.5 State Feedback H_∞ Optimal Control

In this section, the state feedback control law is synthesized using an Algebraic Riccati Equation approach called γ -iteration. Consider the following linear time-invariant model

$$\begin{aligned}\dot{x} &= Ax + Bu + Ew \\ z &= Cx + D_1u + D_2w\end{aligned}\tag{5.24}$$

and cost function

$$J(u, w) = \frac{1}{2} \int_{t_0}^T (z^T z - \gamma^2 w^T w) d\tau\tag{5.25}$$

where $\gamma \geq 0$ and with t_0, x_0 given and $T, x(T)$ free. Our goal is to find the optimal control (minimizing control) u^* and maximizing disturbance w^* such that

$$J(u^*, w) \leq J(u^*, w^*) \leq J(u, w^*)\tag{5.26}$$

Examine the response z in (5.24) from the exogenous variable w

$$\|z\|_{\text{RMS}}^2 \leq \|T_{zw}\|_\infty^2 \|w\|_{\text{RMS}}^2\tag{5.27}$$

where T_{zw} is the closed-loop transfer function model from w to z . Choose a positive γ such that $\gamma \geq \|T_{zw}\|_\infty$. Then, from (5.27),

$$\|z\|_{\text{RMS}} - \gamma^2 \|w\|_{\text{RMS}} \leq 0\tag{5.28}$$

Using this γ , substitute (5.24) into (5.25), to obtain

$$\begin{aligned} J(u, w) &= \frac{1}{2} \int_{t_0}^T (z^T z - \gamma^2 w^T w) d\tau \\ &= \frac{1}{2} \int_{t_0}^T ([Cx + D_1 u + D_2 w]^T [Cx + D_1 u + D_2 w] - \gamma^2 w^T w) d\tau \\ &= \frac{1}{2} \int_{t_0}^T \left(\begin{array}{l} x^T C^T C x + 2x^T [C^T D_1 \ C^T D_2] \begin{bmatrix} u \\ w \end{bmatrix} \\ + \begin{bmatrix} u \\ w \end{bmatrix}^T \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} \end{array} \right) d\tau \end{aligned} \quad (5.29)$$

Let

$$S = [C^T D_1 \ C^T D_2], \ R = \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix}, \ \tilde{u} = \begin{bmatrix} u \\ w \end{bmatrix} \quad (5.30)$$

Then (5.29) becomes,

$$J(u, w) = \frac{1}{2} \int_{t_0}^T (x^T C^T C x + 2x^T S \tilde{u} + \tilde{u}^T R \tilde{u}) d\tau \quad (5.31)$$

which is an LQR problem that has cross terms between the state x and extended control \tilde{u} . Next, rewrite the plant model in (5.24) using the extended control as

$$\dot{x} = Ax + \tilde{B}\tilde{u} \quad (5.32)$$

where $\tilde{B} = [B \ E]$. We can write the Hamiltonian for this LQR problem as

$$H = 1/2(x^T C^T C x + 2x^T S \tilde{u} + \tilde{u}^T R \tilde{u}) + p^T(Ax + \tilde{B}\tilde{u}) \quad (5.33)$$

The necessary condition for the optimal control \tilde{u}^* is

$$\nabla H_{\tilde{u}} = 0 = R\tilde{u} + S^T x + \tilde{B}^T p \quad (5.34)$$

Solving for the optimal \tilde{u}^* gives

$$\tilde{u}^* = -R^{-1}(S^T x + \tilde{B}^T p) \quad (5.35)$$

The differential equation for the costate is

$$\dot{p} = -\nabla H_x = -C^T Cx - A^T p - S\tilde{u} \quad (5.36)$$

with $p(T) = 0$. Substituting (5.35) into (5.32), and combining with (5.36), we can write the Hamiltonian system as

$$\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} A & -\tilde{B}R^{-1}\tilde{B}^T \\ -C^T C + SR^{-1}S & -A^T + SR\tilde{B}^T \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} \quad (5.37)$$

The next step is to manipulate this first-order differential equation to eliminate the costate p and create a Riccati equation whose solution will give (5.35). The solution to (5.37) is derived from the state transition matrix. Assume the state transition matrix for (5.37) is

$$\Phi(T, t) = \begin{bmatrix} \phi_{xx}(T, t) & \phi_{xp}(T, t) \\ \phi_{px}(T, t) & \phi_{pp}(T, t) \end{bmatrix} \quad (5.38)$$

Then, $p(T) = \phi_{px}x + \phi_{pp}p$. Solving for p , yields

$$p = \underbrace{\phi_{pp}^{-1}\phi_{px}}_P x = Px \quad (5.39)$$

Differentiating results in

$$\dot{p} = \dot{P}x + P\dot{x} \quad (5.40)$$

From (5.37), we have

$$\dot{P}x + P\dot{x} = \left(-C^T C + SR^{-1}S\right)x + \left(-A^T + SR\tilde{B}^T\right)p \quad (5.41)$$

Substituting for \dot{x} using (5.32) and replacing p using (5.39) and factoring out x on the right yields the Riccati equation

$$-\dot{P} = PA + A^T P + C^T C - \left[P\tilde{B} + S\right]R^{-1}\left[\tilde{B}^T P + S^T\right] \quad (5.42)$$

whose solution P is used to form the state feedback control law as

$$\tilde{u} = -R^{-1}\left(B^T P + S^T\right)x \quad (5.43)$$

For the infinite time problem, (5.42) becomes an Algebraic Riccati Equation (ARE). The ARE is used in most applications. The following theorem summarizes the assumptions needed for the problem to be well posed.

Theorem 5.1 (Doyle et al. [1]) Consider the linear time-invariant system described in (5.24) where $x \in R^{n_x}$, $u \in R^{n_u}$, $w \in R^{n_w}$, and $z \in R^{n_z}$. Assume

1. (A, B, C, D_1) has no zeros on the $j\omega$ -axis
2. (A, B) stabilizable
3. D_1 is injective $\left((D_1^T D_1)^{-1}\right)$ exists

Then the following statements are equivalent:

- (i) There exists a state feedback control $\tilde{u} = -K_\infty x$ such that the closed-loop system is internally stable and $\|T_{zw}\|_\infty < \gamma$.
- (ii) $D_2^T D_2 < \gamma^2 I$ and there exists a $P \geq 0$ that solves the following ARE:

$$PA + A^T P + C^T C - \begin{bmatrix} B^T P + D_1^T C \\ E^T P + D_2^T C \end{bmatrix}^T \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} B^T P + D_1^T C \\ E^T P + D_2^T C \end{bmatrix} = 0 \quad (5.44)$$

and the optimal control u is

$$u = [I_{n_u} \ 0] \tilde{u} = -[I_{n_u} \ 0] R^{-1} (B^T P + S^T) x = -K_\infty x \quad (5.45)$$

5.6 Controller Design Using γ -Iteration

In this section, we build a control design model that embeds the sensitivity, complementary sensitivity, and control activity weighting filters from Sect. 4.4 into a state-space model, and then solves for the state feedback gain matrix (5.45) using a method called γ -iteration. The design model needs to be of the form of (5.24). Define the regulated variables in vector z to comprise sensitivity, complementary sensitivity, and control activity variables.

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Sensitivity – to track commands.
Complementary Sensitivity – to roll-off plant, limit bandwidth.
Control Activity – minimize control usage.

From Sect. 4.4, the weighting filter W_S should be designed to be the inverse of the desired loop shape for $S(s)$, the weighting filter W_T should be designed to be the inverse of the desired loop shape for $T(s)$, and the control activity penalty to

penalize control activity in the desired frequency range. To build the H_∞ -controller state-space design model the plant and weighting filters all need to be represented in a state-space format. The plant dynamics are modeled as

$$\begin{aligned}\dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u\end{aligned}\quad (5.46)$$

where y is the variable to be commanded. The sensitivity $S(s)$ weighting filter W_S is modeled as

$$\begin{aligned}\dot{x}_s &= A_S x_s + B_S (y - r) \\ z_1 &= C_S x_s + D_S (y - r)\end{aligned}\quad (5.47)$$

where r is the command. The variable z_1 is the weighted sensitivity. The complementary sensitivity $T(s)$ weighting filter W_T is modeled as:

$$\begin{aligned}\dot{x}_T &= A_T x_T + B_T y \\ z_2 &= C_T x_T + D_T y\end{aligned}\quad (5.48)$$

where the variable z_2 is the weighted complementary sensitivity. The control activity model must be selected in such a way to satisfy the requirement that D_1 matrix from (5.24) is injective. How to select this variable is demonstrated in Example 4.5.

The γ -iteration algorithm used here is summarized in the following five steps:

Algorithm 5.1 H_∞ Control— γ -Iteration Method

1. Pick a starting γ larger than what is anticipated as the optimal γ . This will start the binary search used to converge to the optimal value.
2. Form the LQR matrices using γ from (5.31).
3. Solve the Algebraic Riccati Equation (5.44) for the matrix P .
4. Check that $P > 0$ and that $\text{Re}(\lambda(A_{CL})) < 0$. If these tests pass, reduce γ , and go back to step 2. If the test fail, increase γ , and go back to step 2. A minimum step size needs to be established and used to determine when γ has converged to γ_{\min} .
5. Once the bisection search has converged to a γ_{\min} , form the feedback control using (5.45).

When using the above process care must be exercised as γ approaches γ_{\min} . It is typical that the R matrix in (5.44) becomes ill-conditioned as γ approaches γ_{\min} . The ARE solvers in most commercial tools are sensitive to this, and the resulting $P > 0$ matrix actually does not solve the ARE. This is easily tested by forming (5.44) and computing the norm on the result. The result should be a zero matrix,

with 2-norm less than 10^{-5} . We have found that once the algorithm has converged to γ_{\min} , it is prudent to increase from the γ_{\min} value slightly to reduce the feedback gain magnitudes and improve the accuracy of the solution to the ARE. We will demonstrate this in the upcoming example.

Summary

$$\text{Dynamics: } \dot{x} = Ax + Bu + Ew$$

$$z = Cx + D_1u + D_2w$$

$$\text{Performance index: } J = \frac{1}{2} \int_0^\infty (x^T C^T C x + 2x^T S \tilde{u} + \tilde{u}^T R \tilde{u}) d\tau$$

$$S = [C^T D_1 \quad C^T D_2], \quad R = \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix}, \quad \tilde{u} = \begin{bmatrix} u \\ w \end{bmatrix}$$

Controller Design

1. Pick a starting γ
2. Form LQR matrices
3. Solve

$$PA + A^T P + C^T C$$

$$-\begin{bmatrix} B^T P + D_1^T C \\ E^T P + D_2^T C \end{bmatrix}^T \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} B^T P + D_1^T C \\ E^T P + D_2^T C \end{bmatrix} = 0$$

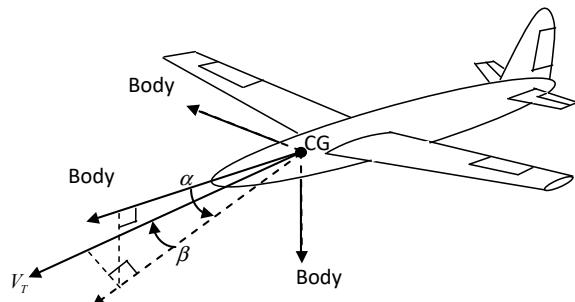
4. Check $P > 0$. Compute feedback gains: $K_\infty = [I_{n_u} \quad 0]R^{-1}(B^T P + S^T)$

Check eigenvalues of closed loop system are stable: $\text{Re}(\lambda(A - BK_\infty)) < 0$

5. Decrease γ until γ_{\min}

Example 5.5 H_∞ Flight Control Design Consider the design of the longitudinal flight control system for the unpiloted aircraft shown in Fig. 5.12.

Fig. 5.12 Unpiloted aircraft



The pitch-plane dynamics are given as

$$\begin{aligned}\dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta + q \\ \dot{q} &= M_\alpha\alpha + M_\delta\delta + M_q q\end{aligned}\quad (5.49)$$

It is desired to design an acceleration command $r = A_{zc}$ flight control system. We will assume that the command is constant, and will design a H_∞ controller using full state feedback. The feedback control law will consist of a constant gain matrix K_c at a single flight condition and will assume gain scheduling will be used to interpolate the gains between conditions (other design points). Normal acceleration A_z (ft/s²) is given by

$$A_z = -V\dot{\gamma} = Z_\alpha\alpha + Z_\delta\delta \quad (5.50)$$

We can introduce A_z directly as a state variable by replacing the angle-of-attack α state. Differentiate Eq. (5.50) to form the differential equation for A_z , and then substitute for $\dot{\alpha}$ from Eq. (5.49). This produces

$$\begin{aligned}\dot{A}_z &= Z_\alpha A_z + VZ_\alpha q + VZ_\delta \dot{\delta}_e \\ \dot{q} &= \frac{M_\alpha}{VZ_\alpha} A_z + M_q q + \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) \delta_e\end{aligned}\quad (5.51)$$

Next, introduce a second-order actuator model for the elevator. This is given as

$$\ddot{\delta}_e = -2\zeta_a\omega_a \dot{\delta}_e + \omega_a^2(\delta_c - \delta_e) \quad (5.52)$$

where δ is the angular position and δ_c command. Combining Eqs. (5.51) and (5.52) forms our plant model written in state-space form as

$$\begin{bmatrix} \dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix} = \begin{bmatrix} Z_\alpha & VZ_\alpha & 0 & VZ_\delta \\ M_\alpha/VZ_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha}\right) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_a^2 & -2\zeta_a\omega_a \end{bmatrix} \begin{bmatrix} A_z \\ q \\ \delta_e \\ \dot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_a^2 \end{bmatrix} \delta_c \quad (5.53)$$

Assume each of the state variables is available for feedback. Equation (5.53) represents the aircraft's dynamics for the plant model expressed in (5.46). This model needs to be combined with the sensitivity weighting filter, complementary sensitivity weighting filter, and the control activity penalty. To satisfy Theorem 4.1 requirements for D_1 to be injective, the control activity is penalized by weighting the variable $\ddot{\delta}_e$ in (5.53) as the control activity. This defines the third regulated variable z_3 expressed as

$$z_3 = W_C \ddot{\delta}_e \quad (5.54)$$

where $\ddot{\delta}$ is formed as

$$\ddot{\delta}_e = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}}_{C_{\ddot{\delta}}} \dot{x} = C_{\ddot{\delta}}(A_p x + B_p u) \quad (5.55)$$

which gives z_3 as

$$z_3 = W_C C_{\ddot{\delta}}(A_p x + B_p u) \quad (5.56)$$

To form the H_∞ controller design model, we combine the plant model (5.46) with the sensitivity weighting filter (5.47) and complementary sensitivity weighting filter (5.48) as

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{x}_S \\ \dot{x}_T \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} &= \begin{bmatrix} A_p & 0 & 0 \\ -B_S C_p & A_S & 0 \\ B_T C_p & 0 & A_T \\ -D_S C_p & C_S & 0 \\ D_T C_p & 0 & C_T \\ W_C C_c A_p & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ x_S \\ x_T \\ x \\ x_S \\ x_T \end{bmatrix} + \begin{bmatrix} B_p \\ -B_S D_p \\ B_T D_p \\ -D_S D_p \\ D_T D_p \\ W_C C_c B_p \end{bmatrix} \delta_c + \begin{bmatrix} 0 \\ B_S \\ 0 \\ D_S \\ 0 \\ 0 \end{bmatrix} r \end{aligned} \quad (5.57)$$

which is of the form of (5.24). For this flight condition the plant model data are

$$\begin{aligned} Z_\alpha &= -1.05273 \text{ (1/s)}; \\ Z_\delta &= -0.0343 \text{ (1/s)}; \\ M_\alpha &= -2.3294 \text{ (1/s}^2\text{)}; \\ M_q &= -1.03341 \text{ (1/s}^2\text{)}; \\ M_\delta &= -1.1684 \text{ (1/s}^2\text{)}; \\ V &= 329.127 \text{ (ft/s)}; \\ \omega_a &= 2\pi * 13. \text{ (rad/s)}; \\ \zeta_a &= 0.6; \end{aligned} \quad (5.58)$$

The sensitivity and complementary weighting filter designs are created by first defining (selecting) the desired loop gain-crossover frequency ω_c . For this flight condition we will set $\omega_c = 2$ (Hz). The sensitivity weighting filter coefficients from Fig. 5.10 are formed by choosing ω_c and the gain K . W_S from Fig. 5.10 is

$$W_S(s) = \frac{K(\tau s + 1)}{s} \quad (5.59)$$

The time constant is computed as

$$\tau = 1/2\pi\omega_c \text{ (\omega_c in Hz)} \quad (5.60)$$

The gain K is chosen to be

$$K = 0.5/\tau \quad (5.61)$$

The state-space model for (5.59) is

$$(A_S, B_S, C_S, D_S) = (0, 1, 6.2832, 0.5) \quad (5.62)$$

The blue curve in Fig. 5.13 shows the frequency response for (5.59). The integral action at low frequency will weight $S_y(s)$ to provide the desired command tracking. The complementary sensitivity weighting filter from Fig. 5.11 is

$$W_T(s) = \frac{K(\tau_N s + 1)}{(\tau_D s + 1)} \quad (5.63)$$

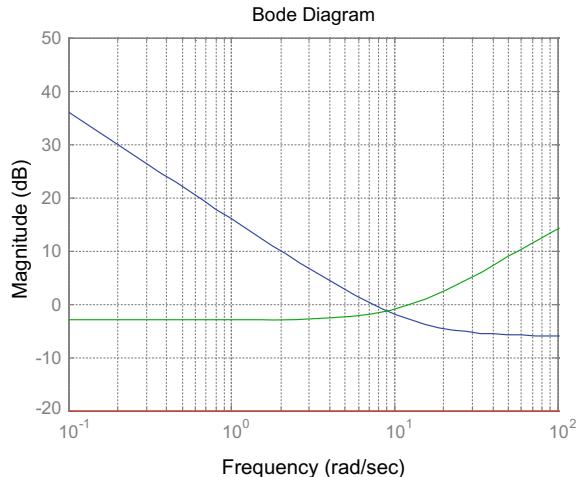
where

$$\begin{aligned} \tau_N &= 1/2\pi\omega_c \ (\omega_c \text{ in Hz}) \\ \tau_D &= 0.005 \ (s) \\ K &= 0.707 \end{aligned} \quad (5.64)$$

The green curve in Fig. 4.13 shows the frequency response for (5.63).
The state-space model for (5.63) is

$$(A_T, B_T, C_T, D_T) = (-200, 1, -2109.1, 11.252) \quad (5.65)$$

Fig. 5.13 Sensitivity, and complementary sensitivity, and control activity weighting filter frequency responses



This model was formed by using the *tf2ss* command in MATLAB®. The weight that penalizes the control activity, W_C , from (5.56) is

$$W_C = 0.1 \quad (5.66)$$

This penalty is plotted in Fig. 5.13 as the red curve.

Now that the desired loop shapes have been engineered, the γ -iteration process is used to form the state feedback control. The binary search algorithm was started with $\gamma_{\max} = 20$ and $\gamma_{\min} = 1$. The algorithm converged with the final γ of 2.544781744 (it is important to include at least 9 decimal places so that results can be reproduced by others). For this minimum γ , the six feedback gains are

$$\begin{aligned} K_\infty = & [210764319.5 - 14676259029.5 \ 3206078184.0 \\ & 29261950.9 \ 730137814.4 - 44949185.4] \end{aligned} \quad (5.67)$$

which are too large to be considered for real-world implementation. The Riccati matrix P was substituted back into the ARE and the 2-norm of the sum computed. The 2-norm = 301,572,105,986.4265, which shows that the ill-conditioning has destroyed the accuracy. By increasing γ slightly to $\gamma = 2.549781744$ the solution becomes accurate with 2-norm = 1.8849×10^{-5} with the feedback gains

$$K_\infty = [0.927881 \ -64.7534 \ 13.1623 \ 0.11795 \ 3.21046 \ -0.151197] \quad (5.68)$$

The H_∞ state feedback controller can be implemented in the following state-space format:

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \quad (5.69)$$

with

$$\begin{aligned} \begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} &= \begin{bmatrix} \begin{bmatrix} A_S & 0 \\ 0 & A_T \end{bmatrix} & \begin{bmatrix} B_S \\ B_T \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ -K_\infty(5:6) & [-K_\infty(1:4)] \end{bmatrix} & \begin{bmatrix} -B_S \\ 0 \\ [0] \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & -200 & 0 \\ -3.2105 & 0.1512 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \\ [0] \end{bmatrix} \end{bmatrix} \\ &= \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & -200 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \\ [0] \end{bmatrix} \\ \begin{bmatrix} -0.9279 & 64.7534 & -13.1623 & -0.1179 \end{bmatrix} & \end{bmatrix} \end{aligned} \quad (5.70)$$

where $x_c = [x_T \ x_S]^T$, $y = [A_z \ q \ \delta_e \ \dot{\delta}_e]^T$, $r = A_{zc}$, and $u = \delta_c$.

To evaluate the design a step simulation of the closed-loop system was performed. The states A_z , q , δ_e , and $\dot{\delta}_e$ are plotted versus time in Fig. 5.14. The controller is a second-order system (one state for W_S and one state for W_T). Note that there is no overshoot to the unit command. Using this approach flight condition, the response is quick without the use of large gains. This simulation can be compared with the Robust Servomechanism design Example 4.10 from Chap. 4. We see the H_∞ control design as responses with a slight overshoot in the acceleration response with an increase in the non-minimum phase undershoot at the initiation of the step command.

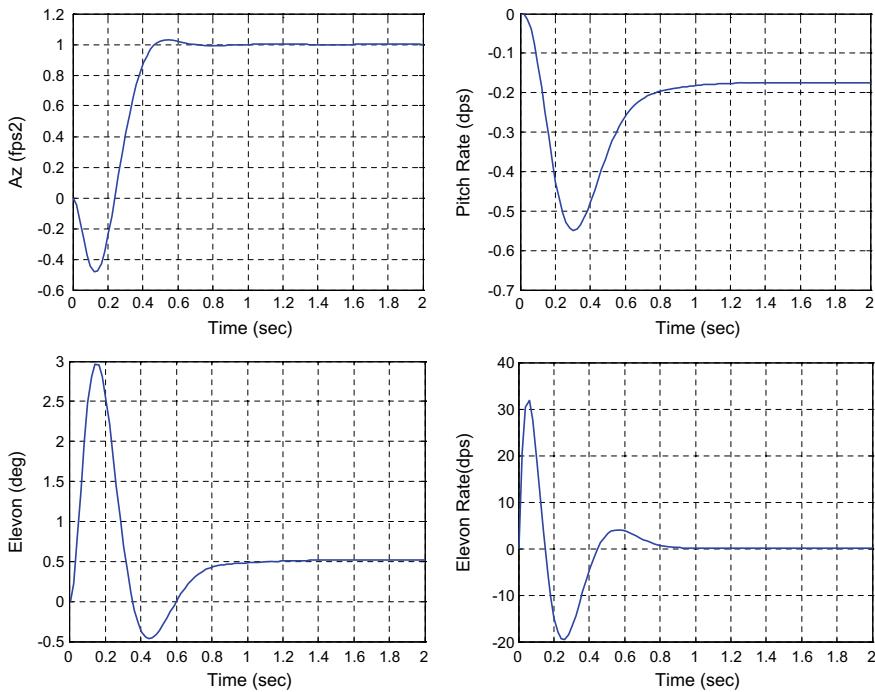


Fig. 5.14 States of the system responding to a unit acceleration step command

5.7 Conclusions

One of the most important development in the 1980s was H_∞ optimal control, and the understanding it gave engineers in performing trades between time domain requirements and frequency domain requirements. We presented the full information state feedback H_∞ optimal control, but output feedback versions also exist. We refer the student wishing to explore output feedback to see [2, 6].

5.8 Exercises

Exercise 5.1 Consider the design of a longitudinal (pitch-plane) autopilot. Using H_∞ state feedback, design a pitch autopilot commanding angle of attack α . Use the following dynamics model as the nominal plant model:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{Z_\alpha}{V} & 1 \\ M_\alpha & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} \frac{Z_{\delta_e}}{V} \\ M_{\delta_e} \end{bmatrix} \delta_e$$

and use data for $\frac{Z_\alpha}{V} = -1.21$; $\frac{Z_\delta}{V} = -0.1987$; $M_\alpha = 44.2506$; $M_\delta = -97.2313$. Add second-order actuator dynamics for the elevator. Design the autopilot to track

a constant angle-of-attack commands. Use the γ -iteration approach outlined in Sect. 4.5.

Exercise 5.2 Consider the longitudinal dynamics of a transport aircraft as given in Chap. 1, Exercise 1.2. Design a H_∞ state feedback controller to track a constant speed command and a constant angle-of-attack command. Use the γ -iteration approach outlined in Sect. 4.5.

Exercise 5.3 Consider the lateral-directional dynamics of a transport aircraft as given in Chap. 1, Exercise 1.4. Design a H_∞ state feedback controller to track a constant stability axis roll rate p_s command (see Eq. (1.22)) and regulate sideslip angle β . Assume $\alpha_0 = 6^\circ$.

References

1. Doyle JC, Glover K, Khargonekar PP, Francis BA (1989) State space solutions to standard H_2 and H_∞ control problems. *IEEE Trans Autom Control* 34(8):831–846
2. Basar T, Bernhard P (1990) H_∞ -optimal control and related minimax design problems, a dynamic game approach. Birkhäuser, Berlin
3. Ball JA, Helton JW (1989) Control for nonlinear plants: connection with differential games. In: Proceedings of the 28th conference on decision and control (Tampa, FL). Inst. of Electrical and Electronics Engineers, Piscataway, NJ, pp 956–962
4. Isaacs R (1965) Differential games. Wiley, New York, p 67
5. Wise KA, Sedwick JL (1996) Nonlinear H_∞ optimal control for agile missiles. *J Guid Control Dyn* 19(1):157–165
6. Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, New Jersey



Output Feedback Control and State Observers

6

Output feedback control design methods are needed when the states are not directly available for use as feedback. For linear time invariant (LTI) multi-input–multi-output (MIMO) dynamical systems there are many design methods to choose from. They range from classical control-based root locus and pole placement techniques to optimal control-based full state observers. The system's time domain performance and frequency domain stability and robustness properties will depend upon the eigenvalues/poles that are achievable through feedback. In Chap. 4 we discussed state feedback control using optimal control and the linear quadratic regulator (LQR), and extended the LQR approach to the robust servomechanism LQR (RSLQR) for command tracking. The state feedback LQR and RSLQR designs yield excellent time domain performance and frequency domain robustness properties. In this chapter we discuss output feedback control methods that build from and retain key properties of the state feedback LQR and RSLQR design using output feedback.

We begin by discussing projective control theory. Projective control is a method of retaining the most dominant eigenstructure of a full state feedback design using only output feedback, where the optimal state feedback regulator (LQR or RSLQR) serves as the reference solution (desired eigenstructure). Projective controls can be implemented using static (constant gain) and/or dynamic output feedback. Static projective controls are output feedback controls that retain an invariant subspace of the reference dynamics in the resulting closed-loop system. For dynamic output feedback, a low-order compensator can be built that can retain additional eigenvalues and associated eigenvectors from the state feedback eigenstructure. The uniqueness of this method is that the designer has the option of choosing the order of the dynamic compensator (with some restrictions), rather than having a full-order compensator.

The concept of a state estimator, or equivalently a state observer, is one of the fundamentals in dynamics and control using output feedback. In this chapter we first discuss full state observers for LTI MIMO dynamical systems and develop observer-based dynamic output feedback controllers. Their design is based on the Separation Principle. It allows to independently design a state observer and use its state estimate in a control feedback architecture, as if the estimate is the true state of the system. This important property yields a design that retains the eigenvalues of the state feedback design for the system model. We analyze relative stability of observer-based output feedback control architectures and show that in general using state observers for control can degrade stability. We present a constructive method to recover optimal state feedback stability margins for LTI systems, operating under dynamic output feedback. The method is called the observer-based loop transfer recovery (OBLTR). The main feature of OBLTR is its inherent ability to systematically design dynamic output feedback controllers and at the same time, recover MIMO gain and phase stability margins of optimal state feedback linear quadratic regulators.

Throughout the chapter we illustrate practicality and significance of OBLTR and projective control design methods via several examples that are motivated by aerospace applications. In these examples the time domain and frequency domain properties of the system are discussed in detail.

6.1 Output Feedback Using Projective Controls

Projective control is an output feedback design method used to obtain a partial eigenstructure of a state feedback controller ($u = -K_x x, x \in R^{n_x}, u \in R^{n_u}$) using static (constant gain) and/or dynamic output feedback. The design retains the dominant performance and robustness properties of the state feedback design. For static output feedback ($u = -K_y y, y \in R^{n_y}$), n_y eigenvalues and associated eigenvectors of a state feedback design can be retained. For dynamic output feedback, a low-order compensator can be built that can retain additional eigenvalues and associated eigenvectors from the state feedback eigenstructure. The order of the compensator can be increased until the entire state feedback eigenstructure is retained.

Linear quadratic regulator (LQR) designs generally give good performance characteristics and stability margins with the availability of the states required for implementation. In many practical designs not all the states are available for feedback. For these problems' dynamic compensators (state observers, Kalman filters, state predictors) are required for implementation.

There have been numerous studies on constructing dynamic regulators. General procedures in the time domain are based upon observer theory and pole placement. An observer is a dynamic system whose output variables are the estimates of the states of the original system. Pole placement is used to make the observer

dynamics faster than the dynamics being estimated. Typically, using these theories does not provide the designer with a low-order dynamic compensator which yields satisfactory results.

An alternative approach is to use projective controls [1–3]. The projective control methodology combines optimal control (state feedback design) with eigenstructure assignment. The uniqueness of this method is that the designer has the option of choosing the order of the dynamic compensator (with some restrictions), rather than having a full-order compensator. The design goal using projective controls is to retain the dominant dynamics as if the states were available for feedback.

Consider the problem of designing output feedback regulators for a linear time-invariant system described by

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}\tag{6.1}$$

with $C = [I_{n_y} \ 0]$, $x \in R^{n_x}$, $u \in R^{n_u}$, and $y \in R^{n_y}$, and the triple (A, B, C) controllable and observable. Let the resulting LQR state feedback control be characterized by

$$u = -R^{-1}B^T Px = -K_x x\tag{6.2}$$

where the symmetric positive-definite matrix P satisfies

$$A^T P + PA - PBR^{-1}B^T P + Q = 0\tag{6.3}$$

with $Q \geq 0$, $R > 0$, and the pair $(A, Q^{1/2})$ observable. Suppose that using the state feedback control law, Eq. (6.2), the above system yields satisfactory closed-loop reference dynamics, described by

$$\dot{x} = \underbrace{(A - BK_x)}_F x = Fx\tag{6.4}$$

If there are n_y outputs available for feedback as described in Eq. (6.1) using the control law

$$u = -K_y y\tag{6.5}$$

then n_y eigenvalues (Λ_{n_y}) and their associated eigenvectors (X_{n_y}) of the state feedback design can be retained using the static output feedback gains given by

$$K_y = K_x X_{n_y} (C X_{n_y})^{-1}\tag{6.6}$$

where the eigenvectors X_{n_y} and eigenvalues Λ_{n_y} satisfy the eigen equation for the state feedback system as

$$FX_{n_y} = X_{n_y}\Lambda_{n_y} \quad (6.7)$$

Using the output feedback control law from Eq. (6.5), the closed-loop system is

$$\dot{x} = \underbrace{(A - BK_y C)}_{A_{cl}} x = A_{cl}x \quad (6.8)$$

with eigenstructure

$$\begin{aligned} A_{cl}X_{n_y} &= (A - BK_y C)X_{n_y} = \left(A - BK_x X_{n_y}(CX_{n_y})^{-1}C \right)X_{n_y} \\ &= (A - BK_x)X_{n_y} = X_{n_y}\Lambda_{n_y} \end{aligned} \quad (6.9)$$

which captures the n_y eigenvalues (Λ_{n_y}) and their associated eigenvectors (X_{n_y}) of the state feedback design. The remaining ($n_x - n_y$) eigenvalues in Eq. (6.8) may not result in closed-loop stability nor satisfactory performance. In this case additional eigenstructure from A_{cl} can be retained using dynamic compensation.

Consider the closed-loop system given by Eq. (6.4). Introduce partitions in the matrices A and F as follows:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}; \quad F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \quad (6.10)$$

where A_{11} and $F_{11} \in R^{n_y \times n_y}$. Denote X and Λ as the eigenvector matrix and spectrum of F ($FX = X\Lambda$), respectively, from Eq. (6.4). From this eigenstructure select the desired dominant dynamics by selecting and placing their eigenvalues in Λ_{n_y} (n_y -eigenvalues) and associated eigenvectors in X_{n_y} ($FX_{n_y} = X_{n_y}\Lambda_{n_y}$). The eigenstructure of the closed-loop output feedback system, using Eq. (6.5), has a spectrum

$$\Lambda(A_{cl}) = \Lambda_{n_y} \cup \Lambda(A_r)$$

where the residual dynamics satisfy

$$A_r = VAY \quad (6.11)$$

with $V \in C^{(n_x - n_y) \times n_x}$ and $Y \in C^{n_x \times (n_x - n_y)}$ satisfying the conditions $CY = 0$, $VX_{n_y} = 0$, and $VY = I$. If the static feedback projective control, Eq. (6.5) does not produce an acceptable result a p^{th} -order dynamic observer of the form

$$\begin{aligned}\dot{z} &= H_d z + D_d y \\ u &= -N_d z - K_d y\end{aligned}\quad (6.12)$$

can be used to extend the eigenstructure. Introduce an extended system by combining the observer (6.12) with the reference dynamics (6.10), described by

$$\dot{x}_e = F_e x_e; \quad F_e = \begin{bmatrix} H_d & D_d & 0 \\ 0 & F_{11} & F_{12} \\ 0 & F_{21} & F_{22} \end{bmatrix} \quad (6.13)$$

Using the observer states as additional outputs for feedback an $(n_y + p)$ -dimensional invariant subspace can be retained. Denote (X_p, Λ_p) as the additional p -dimensional invariant subspace from the reference dynamics. The eigenstructure from the extended system can be decomposed as

$$\begin{bmatrix} H_d & D_d & 0 \\ 0 & F_{11} & F_{12} \\ 0 & F_{21} & F_{22} \end{bmatrix} \begin{bmatrix} W_p & W_{n_y} \\ X_{p1} & X_{n_y1} \\ X_{p2} & X_{n_y2} \end{bmatrix} = \begin{bmatrix} W_p & W_{n_y} \\ X_{p1} & X_{n_y1} \\ X_{p2} & X_{n_y2} \end{bmatrix} \begin{bmatrix} \Lambda_p & 0 \\ 0 & \Lambda_{n_y} \end{bmatrix} \quad (6.14)$$

where W_p and W_{n_y} depend upon the observer matrices H_d and D_d . Define the following matrices

$$\begin{aligned}N_0 &= X_{n_y2} X_{n_y1}^{-1} \\ B_0 &= X_{p2} - N_0 X_{p1} \\ A_r &= A_{22} - N_0 A_{12}\end{aligned}\quad (6.15)$$

The dynamic compensator matrices (H_d, D_d, N_d, K_d) can all be parameterized by a free gain matrix P_0 . Select P_0 to stabilize the residual dynamics, given by

$$A_{re} = A_r + B_0 P_0 A_{12} \quad (6.16)$$

which is an output feedback stabilization problem. The dynamic compensator gain matrices are then given by

$$\begin{aligned}H_d &= \Lambda_p + P_0(A_{12} - B_1 K_{x_2})B_0 \\ D_d &= P_0(A_{11} - B_1 K_{x_1} + (A_{12} - B_1 K_{x_2})N_0) - H_d P_0 \\ N_d &= K_{x_2} B_0 \\ K_d &= K_{x_1} + K_{x_2}(N_0 - B_0 P_0)\end{aligned}\quad (6.17)$$

where

$$K_x = [K_{x_1} \ K_{x_2}]; \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$$

Example 6.1 Static Output Feedback Design Using Projective Control This example applies projective control theory to an air vehicle flight control design problem. The state feedback design (reference eigenstructure) to be retained with the projective control is the robust servo linear quadratic regulator (RSLQR) design from Example 4.10 from Chap. 4. There are three basic steps to applying the projective control approach.

1. Step 1: Design the reference eigenstructure (using a state feedback approach).
2. Step 2: Design a static projective controller using output feedback. Evaluate the adequacy of the design. If not adequate, proceed to step 3.
3. Step 3: Design a low-order dynamic compensator recovering more of the entire eigenstructure of the state feedback design, iterate by adding to the eigenstructure, as required.

Step 3 is necessary only if the compensator designed in step 2 is not satisfactory. Time domain and frequency domain analyses are performed after each design step.

Consider the design of the longitudinal flight control system for an unpiloted aircraft as shown in Fig. 6.1. The pitch plane dynamics are given as

$$\begin{aligned}\dot{\alpha} &= \frac{Z_\alpha}{V}\alpha + \frac{Z_\delta}{V}\delta + q \\ \dot{q} &= M_\alpha\alpha + M_\delta\delta + M_qq\end{aligned}\quad (6.18)$$

It is desired to design an acceleration command $r = A_{zc}$ flight control system. We will assume that the command is constant, and will design a RSLQR controller with integral control. We will design a constant gain matrix K_z for a single flight condition, and will assume gain scheduling will be used to interpolate the gains

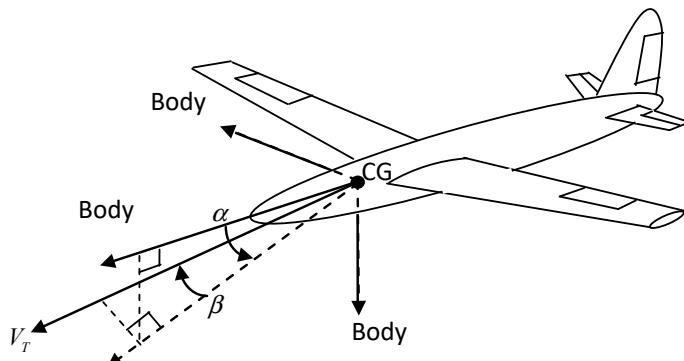


Fig. 6.1 Unpiloted aircraft

between conditions (other design points). Normal acceleration A_z (ft/s²) is given by

$$A_z = -V\dot{\gamma} = Z_\alpha\alpha + Z_\delta\delta \quad (6.19)$$

We can introduce A_z directly as a state variable by replacing the angle-of-attack α state. Differentiate Eq. (6.19) to form the differential equation for \dot{A}_z , and then substitute for $\dot{\alpha}$ from Eq. (6.18). This produces

$$\begin{aligned}\dot{A}_z &= \frac{Z_\alpha}{V} A_z + Z_\alpha q + Z_\delta \dot{\delta}_e \\ \dot{q} &= \frac{M_\alpha}{Z_\alpha} A_z + M_q q + \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha} \right) \delta_e\end{aligned} \quad (6.20)$$

Next, introduce a second-order actuator model for the elevator. This is given as

$$\ddot{\delta}_e = -2\zeta_a \omega_a \dot{\delta}_e + \omega_a^2 (\delta_c - \delta_e) \quad (6.21)$$

Combining (6.20) and (6.21), forms the plant model written in matrix form as

$$\begin{bmatrix} \dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix} = \begin{bmatrix} Z_\alpha/V & Z_\alpha & 0 & Z_\delta \\ M_\alpha/Z_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha} \right) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_a^2 & -2\zeta_a \omega_a \end{bmatrix} \begin{bmatrix} A_z \\ q \\ \delta_e \\ \dot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_a^2 \end{bmatrix} \delta_c \quad (6.22)$$

Since $r = \text{constant}$, $\dot{r} = 0$, $p = 1$, and we need to add one integrator to form a type-1 controller. The state vector (Eq. 3.17) for the robust servomechanism design model is

$$z = [e \ \dot{x}^T]^T \quad (6.23)$$

with the design model $\dot{z} = \tilde{A}z + \tilde{B}\mu$ given as

$$\begin{bmatrix} \dot{e} \\ \dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & Z_\alpha/V & Z_\alpha & 0 & Z_\delta \\ 0 & M_\alpha/Z_\alpha & M_q & \left(M_\delta - \frac{M_\alpha Z_\delta}{Z_\alpha} \right) & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_a^2 & -2\zeta_a \omega_a \end{bmatrix} \begin{bmatrix} e \\ \dot{A}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \omega_a^2 \end{bmatrix} \dot{\delta}_c \quad (6.24)$$

At a flight condition of Mach 0.3, 5000 ft altitude, and a trim angle-of-attack α of 5 degrees, the plant model is

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -1.053 & -346.5 & 0 & -11.29 \\ 0 & 0.007 & -1.033 & -1.093 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -6672. & -98.02 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 6672. \end{bmatrix} \quad (6.25)$$

The objective in the design of the gain matrix is to track the acceleration command with zero error without using large gains. The design begins by setting $R = 1$ and selecting a Q matrix that penalizes the error state e in Eq. (6.24). Thus, the performance index in Eq. is

$$J = \int_0^\infty (z^T Q z + \mu^2) d\tau \quad (6.26)$$

We start by inserting the parameter q_{11} in the (1,1) element,

$$z^T Q z = z^T \begin{bmatrix} q_{11} & & & & \\ 0 & 0 & & & \\ & 0 & & & \\ 0 & 0 & & & \\ & & 0 & & \end{bmatrix} \begin{bmatrix} e \\ \dot{q} \\ \dot{A}_z \\ \dot{\delta}_e \\ \ddot{\delta}_e \end{bmatrix}, \quad (6.27)$$

and set the other matrix elements to zero. This will penalize the error in tracking the command. Substituting Eq. (6.27) into Eq. (6.26) gives the performance index as

$$J = \int_0^\infty (q_{11} e_{A_z}^2 + \mu^2) d\tau, \quad (6.28)$$

LQR design charts were used in Chap. 4 to select the LQR penalty parameter q_{11} in (6.28). We will use the same numerical value in this example to design the state feedback control, and then will compare the projective control to the state feedback design.

Step 1: Design Reference Eigenstructure From Chap. 4 Example 4.10 we designed a RSLQR using a 3-state gain design model. The control law was

$$u = -K_1 \int e_{A_z} - K_2 \alpha - K_3 q. \quad (6.29)$$

Here, using the gain design model of (6.24), we will have a control law given as

$$u = -K_1 \int e_{A_z} - K_2 A_z - K_3 q - K_4 \delta_e - K_5 \dot{\delta}_e, \quad (6.30)$$

with, $q_{11} = 8.1113\text{e-}02$. This produces the RSLQR state feedback gain matrix

$$K_x = [0.2848 \ 0.10872 \ -9.6075 \ 1.3622 \ 1.3161\text{e-}03] \quad (6.31)$$

For this design, the states A_z , q , δ_e , and $\dot{\delta}_e$ are plotted versus time in Fig. 6.2. Note that there is no overshoot to the unit command. For this approach flight condition, the response is quick without the use of large gains. It is desired to keep this same behavior, to the degree possible, in the output feedback projective control design.

Step 2: Design a Static Projective Controller For the static projective control output feedback design, the $\int e$, A_z , and q states are the desired feedbacks. The integral error state is part of the controller, so it is available. The acceleration and rate feedbacks come directly from the inertial measurement unit, and are available. The actuator states in the model would require additional sensors within the actuator, and are assumed not to be available for feedback. In the static projective controller design these states will be projected out. The closed-loop matrix $F = (\tilde{A} - \tilde{B}K_x)$ from (6.4) is

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -1.053 & -346.5 & 0 & -11.29 \\ 0 & 0.007 & -1.033 & -1.093 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -1900.2 & -725.4 & 64100. & -15760. & -106.8 \end{bmatrix} \quad (6.32)$$

with eigenstructure

$$\Lambda = \text{diag}[-3.789 \ -3.551 \ \pm 3.979j, -48.973 \ \pm 65.438j] \quad (6.33)$$

$$X = \begin{bmatrix} 2.5259\text{e-}01 & -1.2001\text{e-}01 - 1.3449\text{e-}01j & -1.2001\text{e-}01 + 1.3449\text{e-}01j \\ -9.5702\text{e-}01 & 9.6128\text{e-}01 & 9.6128\text{e-}01 \\ -1.2031\text{e-}02 & 3.4427\text{e-}04 - 9.9986\text{e-}03j & 3.4427\text{e-}04 + 9.9986\text{e-}03j \\ -3.6233\text{e-}02 & -2.9707\text{e-}02 - 2.4293\text{e-}02j & -2.9707\text{e-}02 + 2.4293\text{e-}02j \\ 1.3728\text{e-}01 & 2.0215\text{e-}01 - 3.1947\text{e-}02j & 2.0215\text{e-}01 + 3.1947\text{e-}02j \\ & 4.8294\text{e-}04 - 1.6202\text{e-}03j & 4.8294\text{e-}04 + 1.6202\text{e-}03j \\ & 8.2257\text{e-}02 + 1.1096\text{e-}01j & 8.2257\text{e-}02 - 1.1096\text{e-}01j \\ & 8.2257\text{e-}02 + 1.1096\text{e-}01j & 8.2257\text{e-}02 - 1.1096\text{e-}01j \\ & -7.2698\text{e-}03 - 9.6999\text{e-}03j & -7.2698\text{e-}03 + 9.6999\text{e-}03j \\ & 9.9034\text{e-}01 & 9.9034\text{e-}01 \end{bmatrix} \quad (6.34)$$

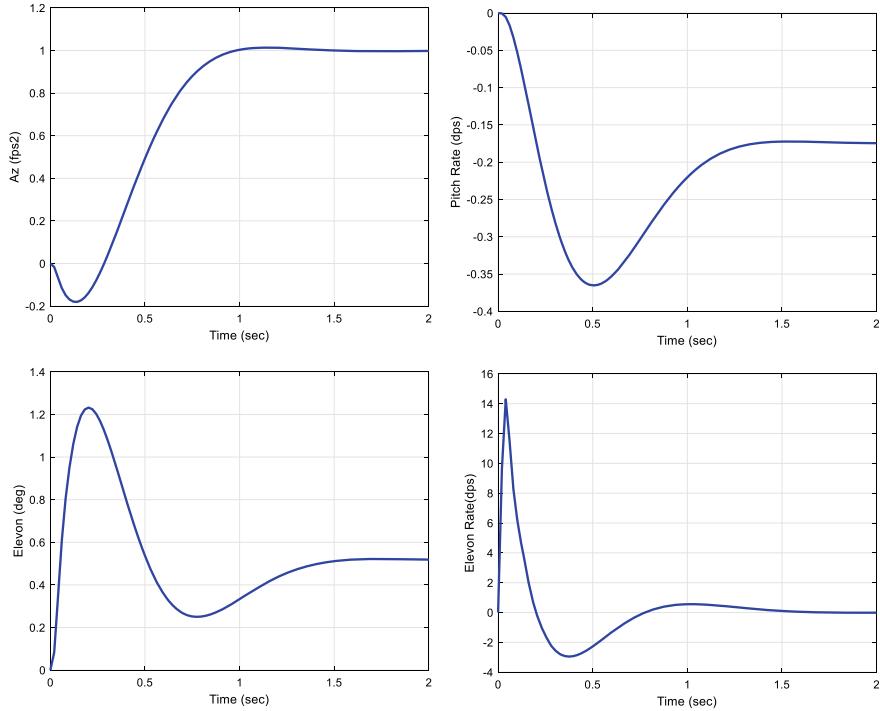


Fig. 6.2 States of the system responding to a unit acceleration step command

The dominant eigenvalues (short-period plus integrator) $\lambda_i = \{-3.789, -3.551 \pm 3.979j\}$ are associated with the first three eigenvectors in (6.34). These are the dynamics that are to be retained in the static output feedback design. From (6.6) the output matrix is

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (6.35)$$

and the eigenvectors are

$$X_{n_y} = \begin{bmatrix} 2.5259e-01 & -1.2001e-01 - 1.3449e-01j \\ -9.5702e-01 & 9.6128e-01 \\ -1.2031e-02 & 3.4427e-04 - 9.9986e-03j \\ -3.6233e-02 & -2.9707e-02 - 2.4293e-02j \\ 1.3728e-01 & 2.0215e-01 - 3.1947e-02j \end{bmatrix} \begin{bmatrix} -1.2001e-01 + 1.3449e-01j \\ 9.6128e-01 \\ 3.4427e-04 + 9.9986e-03j \\ -2.9707e-02 + 2.4293e-02j \\ 2.0215e-01 + 3.1947e-02j \end{bmatrix} \quad (6.36)$$

The static output feedback gain matrix is computed as

$$K_y = K_x X_{n_y} (C X_{n_y})^{-1} = [1.1097e-01 \ 4.3178e-02 \ -3.9555e+00] \quad (6.37)$$

Fig. 6.3 Static output feedback acceleration command control

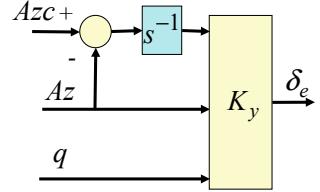


Figure 6.3 shows a block diagram of the output feedback control. The closed-loop system using the static output feedback is $A_{cl} = A - BK_yC$. The eigenstructure for this system is

$$\Lambda = \text{diag}[-3.789 - 3.551 \pm 3.979j, -44.607 \pm 24.718j] \quad (6.38)$$

$$X = \begin{bmatrix} 2.5259e-01 & \begin{bmatrix} -1.2001e-01 - 1.3449e-01j \\ 9.6128e-01 \\ 3.4427e-04 - 9.9986e-03j \\ -2.9707e-02 - 2.4293e-02j \\ 2.0215e-01 - 3.1947e-02j \end{bmatrix} \\ -9.5702e-01 & \begin{bmatrix} -1.2001e-01 + 1.3449e-01j \\ 9.6128e-01 \\ 3.4427e-04 + 9.9986e-03j \\ -2.9707e-02 + 2.4293e-02j \\ 2.0215e-01 + 3.1947e-02j \end{bmatrix} \\ -1.2031e-02 & \begin{bmatrix} -2.2767e-03 - 3.6260e-03j \\ 1.9118e-01 + 1.0547e-01j \\ -2.3298e-04 - 3.8094e-04j \\ -1.6734e-02 - 9.2730e-03j \\ 9.7567e-01 \end{bmatrix} \\ -3.6233e-02 & \begin{bmatrix} -2.2767e-03 + 3.6260e-03j \\ 1.9118e-01 - 1.0547e-01j \\ -2.3298e-04 + 3.8094e-04j \\ -1.6734e-02 + 9.2730e-03j \\ 9.7567e-01 \end{bmatrix} \\ 1.3728e-01 & \begin{bmatrix} -2.2767e-03 \\ 1.9118e-01 \\ -2.3298e-04 \\ -1.6734e-02 \\ 9.7567e-01 \end{bmatrix} \end{bmatrix} \quad (6.39)$$

Note that the eigenvalues $\Lambda_i = \text{diag}\{-3.789, -3.551 \pm 3.979j\}$ and their associated eigenvectors from the state feedback design are retained in the closed-loop output feedback design system. The remaining two eigenvalues which forms the residuals dynamics are $\Lambda_i = \text{diag}[-44.607 \pm 24.718j]$.

The static output feedback controller shown in Fig. 6.3 implementing the RSLQR design can be implemented in the following state-space format

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r \end{aligned} \quad (6.40)$$

with

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & [C_c] & [-1] \\ [-K_y(1)] & [-K_y(2 : 3)] & [0] \end{bmatrix} \\ = \begin{bmatrix} [0] & [1 \ 0] & [-1] \\ [-1.1097e-01] & [-4.3178e-02 \ 3.96] & [0] \end{bmatrix} \quad (6.41)$$

where $x_c = \int e$, $y = [A_z \ q]^T$, $r = A_{zc}$, and $u = \delta_c$.

For this static output feedback design method to be effective, care must be taken to keep the bandwidth reasonable and not destabilize the residual dynamics. As the bandwidth of the state feedback design is increased to have the system

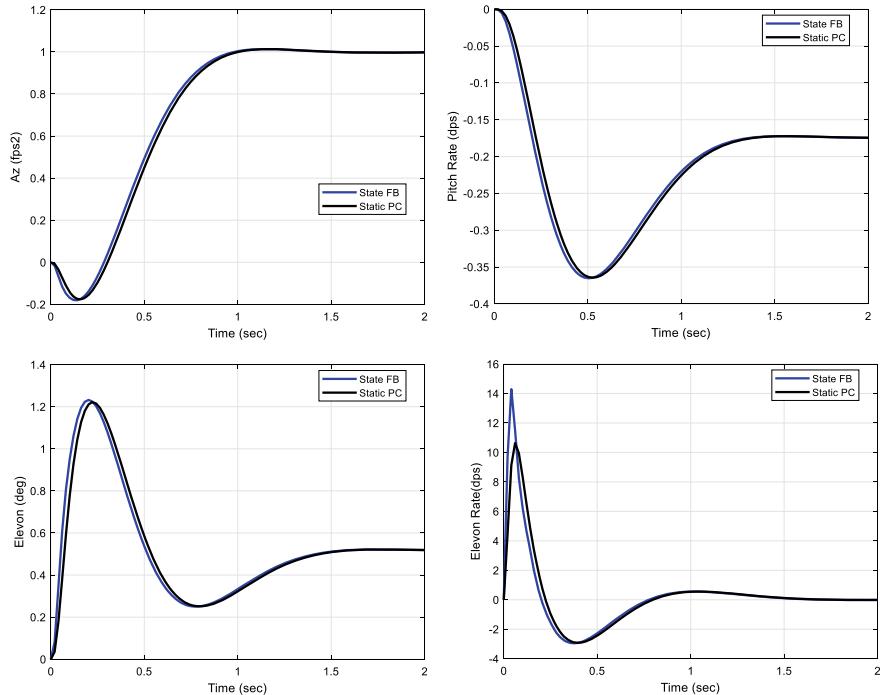


Fig. 6.4 Comparison of state feedback and output feedback time histories

respond faster (an increase in the loop gain-crossover frequency, Fig. 3.5) the larger gains destabilizes the residuals dynamics. The next step in the design process is to compare this design with the state feedback design to determine changes in performance and stability robustness, if any, and to determine if they are acceptable. Figure 6.4 compares the state feedback and output feedback time histories. The static projective control response slightly lags the state feedback design, and is acceptable from a time domain perspective. By capturing the dominant eigenvalues of the state feedback design in the output feedback design, the time response is very close. Next is to compare the design in the frequency domain. Figure 6.5 shows plots of the magnitude and phase of L_u , the magnitude of $1 + L_u$, and the magnitude of $1/T_u$, with the loop gain formed at the actuator command input. We see from the plots of $1 + L_u$ that the output feedback design has a decrease in the stability robustness. This is also seen in the Nyquist plot, Fig. 6.6, which shows the gain margin and phase margin of the design. The output feedback design has a gain margin of 5.12 dB and phase margin of 30.3 deg. This would be unacceptable for flight, so the design would need to be improved by either (1) decreasing the bandwidth until acceptable stability margins were obtained, or (2) designing a low-order dynamic compensator to recover the performance and margins of the

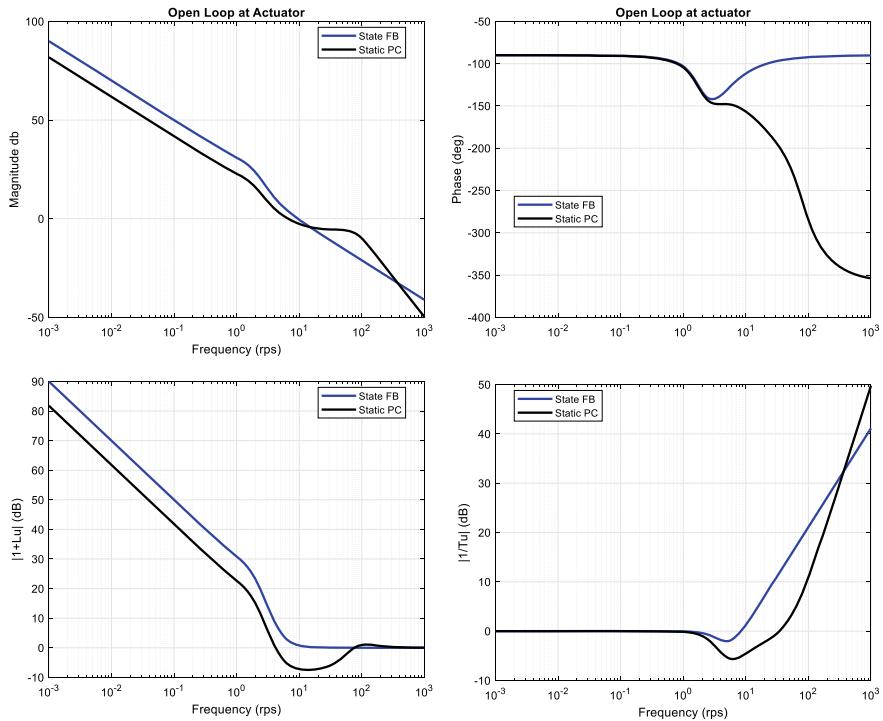


Fig. 6.5 Comparison of state feedback and output feedback frequency domain loop shapes

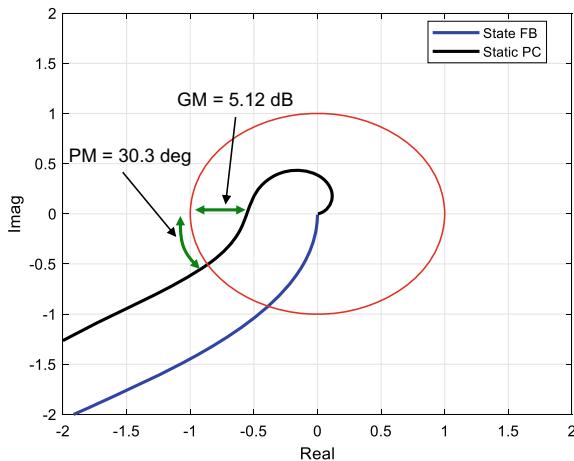


Fig. 6.6 Comparison of state feedback and output feedback

state feedback design. We will select the second option and proceed to design a low-order dynamic compensator.

Step 3: Design a Low-order Dynamic Compensator For this example, a second-order compensator will retain the entire state feedback eigenstructure. To begin the design we need to partition the matrices A and F as in (6.10), and the eigenvectors as in (6.14)

$$\begin{aligned} A_{11} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1.053 & -346.5 \\ 0 & 0.007 & -1.033 \end{bmatrix} & A_{12} &= \begin{bmatrix} 0 & 0 \\ 0 & -11.29 \\ -1.093 & 0 \end{bmatrix} \\ A_{21} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & A_{22} &= \begin{bmatrix} 0 & 1 \\ -6672. & -98.02 \end{bmatrix} \end{aligned} \quad (6.42)$$

$$\begin{aligned} F_{11} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1.053 & -346.5 \\ 0 & 0.007 & -1.033 \end{bmatrix} & F_{12} &= \begin{bmatrix} 0 & 0 \\ 0 & -11.29 \\ -1.093 & 0 \end{bmatrix} \\ F_{21} &= \begin{bmatrix} 0 & 0 & 0 \\ -1900.2 & -725.40 & 64100. \end{bmatrix} & F_{22} &= \begin{bmatrix} 0 & 1 \\ -15760. & -106.80 \end{bmatrix} \end{aligned} \quad (6.43)$$

$$\begin{aligned} X_{n_{y1}} &= \begin{bmatrix} -1.2001e-01 & -1.3449e-01i & -1.2001e-01 + 1.3449e-01i & 2.5259e-01 \\ 9.6128e-01 & & 9.6128e-01 & -9.5702e-01 \\ 3.4427e-04 & -9.9986e-03i & 3.4427e-04 + 9.9986e-03i & -1.2031e-02 \end{bmatrix} \\ X_{n_{y2}} &= \begin{bmatrix} -2.9707e-02 & -2.4293e-02i & -2.9707e-02 + 2.4293e-02i & -3.6233e-02 \\ 2.0215e-01 & -3.1947e-02i & 2.0215e-01 + 3.1947e-02i & 1.3728e-01 \end{bmatrix} \\ X_{p1} &= \begin{bmatrix} 4.8294e-04 & -1.6202e-03 \\ 8.2257e-02 & 1.1096e-01 \\ 5.0819e-05 & -1.6723e-04 \end{bmatrix} \\ X_{p2} &= \begin{bmatrix} -7.2698e-03 & -9.6999e-03 \\ 9.9034e-01 & 0 \end{bmatrix} \end{aligned} \quad (6.44)$$

The compensator design (6.12) requires selecting a gain matrix P_0 such that the residual dynamics A_r in (6.16) are stable. The matrices needed to form A_r are N_0 and B_0

$$N_0 = X_{n_{y2}} X_{n_{y1}}^{-1} = \begin{bmatrix} -1.2887e-01 & -4.8484e-02 & 4.1630e+00 \\ 1.2948e+00 & 3.7703e-01 & -1.4220e+01 \end{bmatrix} \quad (6.45)$$

$$B_0 = X_{p2} - N_0 X_{p1} = \begin{bmatrix} -3.4310e-03 & -3.8330e-03 \\ 9.5942e-01 & -4.2114e-02 \end{bmatrix} \quad (6.46)$$

$$A_r = A_{22} - N_0 A_{12} = \begin{bmatrix} 4.5481e+00 & 4.5267e-01 \\ -6.6874e+03 & -9.3761e+01 \end{bmatrix} \quad (6.47)$$

Using the dynamic compensator in (6.12) with matrices defined in (6.17), the compensator is designed by choosing the free parameter matrix P_0 such that the residual dynamics in (6.16) are stable. For this example

$$\begin{aligned} A_{re} = A_r + B_0 P_0 A_{12} &= \begin{bmatrix} 4.5481e+00 & 4.5267e-01 \\ -6.6874e+03 & -9.3761e+01 \end{bmatrix} \\ &+ \begin{bmatrix} -3.4310e-03 & -3.8330e-03 \\ 9.5942e-01 & -4.2114e-02 \end{bmatrix} P_0 \begin{bmatrix} 0 & 0 \\ 0 & -11.29 \\ -1.093 & 0 \end{bmatrix} \quad (6.48) \end{aligned}$$

By multiplying out the matrices in (6.48) one can determine which elements of P_0 need to be chosen. This matrix is designed using a tuning process in which the elements are increased in magnitude until a suitable design is obtained (trial and error). After some tuning the following matrix was obtained

$$P_0 = \begin{bmatrix} 0 & 2 & -500 \\ 0 & 2 & -2000000 \end{bmatrix} \quad (6.49)$$

The zero elements in the first column were found not to matter. They were made zero to reduce the control usage. Substituting this P_0 into (6.48) yields

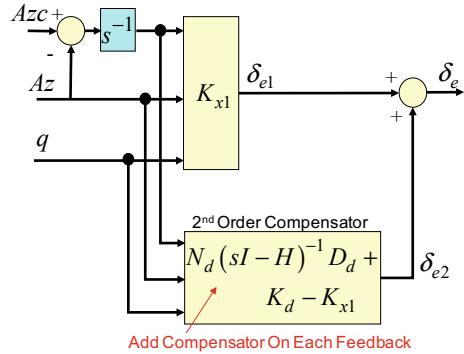
$$A_{re} = \begin{bmatrix} -8.3484e+02 & 6.1667e-01 \\ -1.5365e+04 & -1.1447e+02 \end{bmatrix}; \quad \lambda_i(A_{re}) = [-821.44 \ -127.88] \quad (6.50)$$

Figure 6.7 illustrates a block diagram for the controller. Substituting P_0 into (6.17) yields

$$\begin{aligned} H_d &= \begin{bmatrix} -7.2533e+01 & 6.4233e+01 \\ -8.3671e+02 & -8.8556e+02 \end{bmatrix}; \\ D_d &= \begin{bmatrix} -9.9626e+01 & -2.3864e+01 & 1.2813e+07 \\ -2.8186e+04 & -8.5044e+03 & -1.7641e+08 \end{bmatrix} \\ N_d &= [-3.4109e-03 \ -5.2767e-03]; \\ K_d &= [1.1097e-01 \ 6.0553e-02 \ -1.0610e+03] \quad (6.51) \end{aligned}$$

The dynamic output feedback controller shown in Fig. 6.7 capturing the eigenstructure of the RSLQR design can be implemented in the following state-space format

Fig. 6.7 Low-order dynamic projective controller block diagram



$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r\end{aligned}\quad (6.52)$$

with

$$\begin{aligned}&\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} \\&= \begin{bmatrix} \begin{bmatrix} 0 & 0_{1 \times 2} \\ D_d(:, 1) & H_d \end{bmatrix} & \begin{bmatrix} C_c \\ D_d(:, 2 : 3) \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} -K_d(1) & -N_d \end{bmatrix} & \begin{bmatrix} -K_d(2 : 3) \end{bmatrix} & [0] \end{bmatrix} \\&= \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ -9.9626e+01 & -9.9626e+01 & 6.4233e+01 \\ -2.8186e+04 & -8.3671e+02 & -8.8556e+02 \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ -2.3864e+01 & 1.2813e+07 \\ -8.5044e+03 & -1.7641e+08 \\ -6.0553e-02 & 1.0610e+03 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \\ 0 \\ [0] \end{bmatrix} \\ \begin{bmatrix} -1.1097e-01 & 3.4109e-03 & 5.2767e-03 \end{bmatrix} & \end{bmatrix}\end{aligned}\quad (6.53)$$

where $x_c = [\int e \ x_{c2} \ x_{c3}]^T$, $y = [A_z \ q]^T$, $r = A_{zc}$, and $u = \delta_c$.

A step-input simulation of the closed-loop system using the dynamic controller shows results that equal the state feedback design. Figure 6.8 compares the designs in the frequency domain showing plots of the magnitude and phase of L_u , the magnitude of $1 + L_u$, and the magnitude of $1/T_u$, with the loop gain formed at the actuator command input. Figure 6.9 shows the Nyquist plot (L is a scalar), which shows the gain margin and phase margin of the design. The second-order compensator does an excellent job of recovering the state feedback design eigenstructure and properties using a low-order compensator with output feedback.

Summary

In this section we discussed the design of output feedback controllers that retain the closed-loop eigenstructure of an LQR or RSLQR state feedback reference

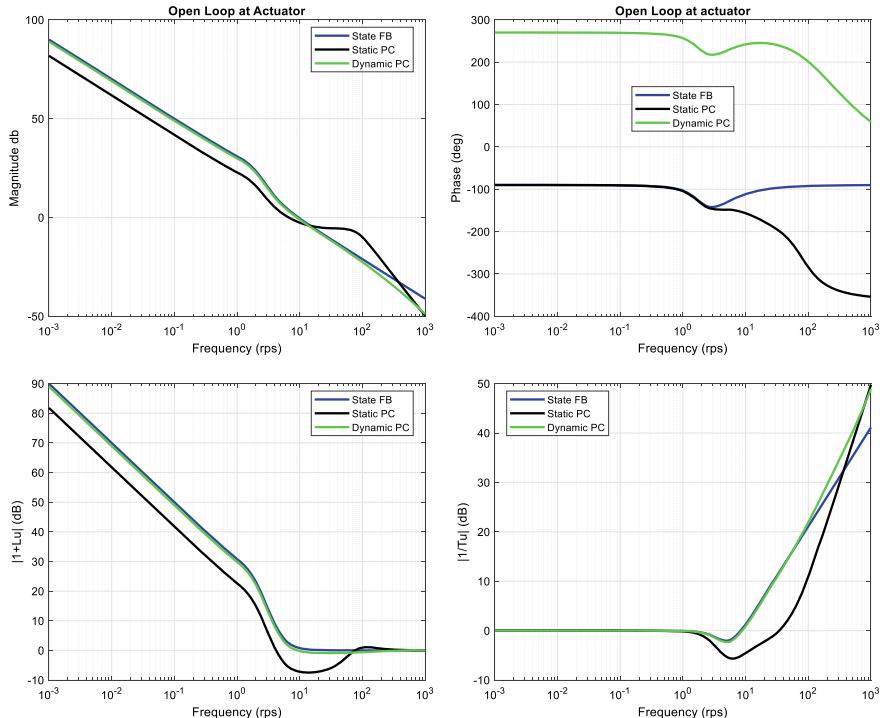


Fig. 6.8 Frequency response comparison between the state feedback, static, and dynamic output feedback projective controllers

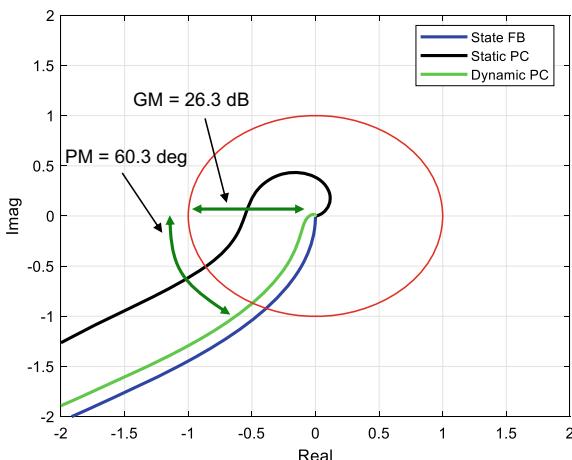


Fig. 6.9 Nyquist plot comparison between the state feedback, static, and dynamic output feedback projective controllers

design. For constant gain or static output feedback, n_y eigenvalues and associated eigenvectors could be retained. For dynamic output feedback, a low-order compensator was constructed to retain additional eigenvalues and associated eigenvectors whose number equals to the order of the compensator. When we examined the time domain and frequency domain properties of projective controllers, we see that by retaining the dominant eigenvalues of the state feedback reference design the closed-loop system using output feedback retains similar time domain performance. However, the frequency domain analysis of these controllers showed that stability margins can be degraded, with an improvement in margins when using a dynamic compensator.

In systems where there is frequency separation between the system dynamics and the systems' subsystems (actuators, sensors, sampling rate), projective controllers work very well. When a system lacks frequency separation, the phase lag from the subsystems degrades the stability robustness. We saw that by adding a dynamic compensator we could improve the stability robustness as compared to static output feedback.

In the remainder of this chapter we discuss control design using full state observers. The tuning of a control design in observer-based flight control laws can be a challenge. Performance requirements, stability margins at both plant-input and output loop break points, sensor noise amplification, and loop attenuation for structural mode interaction concerns all must be met simultaneously. Classical tuning rules for observer design in Bryson and Ho² and Kwakernak and Sivan³ provide insight, but lack direct applicability to today's challenging unstable aircraft configurations. Historically, these observer-based architectures have been shown to suffer from poor stability margins at certain loop break points, and require a loop transfer recovery (LTR) procedure to improve their margins. In the next section we discuss a system squaring-up procedure that introduces left half plane transmission zeros into the observer design process so that when LTR is applied, the system recovers the excellent stability robustness of the state feedback reference design. This important controller design capability allows the designer to exploit the filtering characteristics of the observer while minimizing the phase lag impact on stability margins.

6.2 Full-Order State Observers for Linear Time-Invariant Systems

We begin with the development of a full-order state observer/estimator for the linear time-invariant (LTI) plant,

$$\dot{x} = Ax + Bu, \quad y = Cx + Du \quad (6.54)$$

where $x \in R^{n_x}$ denotes the n_x -dimensional state, $u \in R^{n_u}$ is the n_u -dimensional vector of controls, and $y \in R^{n_y}$ is the measured output. The system output measurements represent signals that would be recorded online (i.e., during the system

operation) by physical devices, called the sensors. The practical assumptions on the dimensions of the system state, control, and measured output are

$$n_u < n_y < n_x \quad (6.55)$$

As seen from (6.55), there are less controls than output measurements, and less output measurements than states in the LTI system (6.54). The observer design task is to reconstruct the system state x using only the output measurements y and the known control inputs u .

The original concept of a state observer is due to David G. Luenberger [4–6], focusing on state estimation and reconstruction for continuous-time deterministic LTI systems (6.54). Design extensions also exist for time varying, nonlinear, discrete, and stochastic dynamics.

Introduction to state observers for LTI systems can be readily found in many textbooks on dynamics and controls, such as [7, 8], to name a few. We begin with basic definitions.

Definition 6.1 The n_x -dimensional system

$$\dot{\hat{x}} = F \hat{x} + G y + H u \quad (6.56)$$

is a full-order state observer for the n_x -dimensional LTI dynamics (6.54), if starting from the same initial conditions,

$$\hat{x}(0) = x(0) \quad (6.57)$$

and for any control input $u(t)$, the observer and the system trajectories are the same.

$$\hat{x}(t) = x(t), \quad \forall t \geq 0 \quad (6.58)$$

Necessary and sufficient conditions for matrices (F, G, H) to form the full-order state observer are formulated by D. G. Luenberger for time-varying linear systems [6]. Methods to construct reduced-order observers, whose dimension is less than that of the system state x , can be found in [7, 8].

In what follows, only full-order state observers for LTI systems will be considered.

Theorem 6.1 *The system (6.56) is a full-order state observer for (6.54) if and only if,*

$$F = A - L C, \quad G = L, \quad H = B - L D \quad (6.59)$$

where $L \in R^{n_x \times n_y}$ is an arbitrary gain matrix. ■

Proof of Theorem 6.1 Can be accomplished by direct substitution of (6.59) into (6.56) and comparing the resulting dynamics to that of the LTI system (6.54) [4–6]. It gives the full-order state observer dynamics,

$$\dot{\hat{x}} = A \hat{x} + B u + L(y - \hat{y}) \quad (6.60)$$

where

$$\hat{y} = C \hat{x} + D u \quad (6.61)$$

is the observer predicted output.

The matrix $L \in R^{n_x \times n_y}$ in (6.60) is called the observer gain. If $L = 0_{n_x \times n_y}$ then (6.60) is an open-loop state observer. Otherwise, (6.60) represents a closed-loop full-order observer dynamics.

Note that open-loop and closed-loop observers satisfy conditions (6.59). However, for open-loop unstable systems, open-loop observer trajectories will diverge from the system trajectories, if the two systems are started from different initial conditions, no matter how small those are.

From theoretical and practical points of view, observer sensitivity to initial conditions is highly undesirable. It can be prevented within the closed-loop observer framework by enforcing global exponential stability of the observer dynamics.

Rewriting (6.60),

$$\dot{\hat{x}} = (A - L C) \hat{x} + (B - L D) u + L y \quad (6.62)$$

shows that the observer closed-loop matrix

$$A_o = A - L C \quad (6.63)$$

defines stability of the observer dynamics (6.62).

Theorem 6.2 *The observer estimation error,*

$$e_x(t) = \hat{x}(t) - x(t) \quad (6.64)$$

can be made globally exponentially stable if and only if the matrix pair (A, C) is observable or detectable. ■

Proof of Theorem 6.2 Subtracting the system dynamics (6.54) from that of the observer (6.60), gives the estimation error dynamics,

$$\dot{e}_x = (A - L C) e_x \quad (6.65)$$

whose global exponential stability is equivalent to the observability/detectability of (A, C) . ■

Since the eigenvalues of $(A - L C)$ and $(A^T - C^T L^T)$ are the same then the poles of the observer closed-loop matrix (6.63) can be arbitrarily placed in the open left half complex plane \mathbb{C}^- by appropriately choosing the observer gain L . The existence of a gain is predicated by the controllability (stabilizability) of (A^T, C^T) which is equivalent to the observability (detectability) of (A, C) . This phenomenon is called the Duality Principle.

6.2.1 The Separation Principle

Suppose that a gain matrix $K_x \in R^{n_u \times n_x}$ is found such that the linear state feedback controller (6.54) renders stable (Hurwitz) closed-loop system matrix,

$$A_{cl} = A - B K_x \quad (6.66)$$

with all of its eigenvalues located in the open left half complex plane \mathbb{C}^- . In our case, the system state components are not accessible for controller realization. Because of that, a state feedback control policy is not realizable.

In order to synthesize a feasible control policy using the system measurements, we shall use the state of the observer (6.60) to form the observer-based linear feedback control signal.

$$u = -K_x \hat{x} \quad (6.67)$$

The combined system with the observer dynamics in the loop are

$$\begin{aligned} \dot{x} &= Ax + Bu, \quad y = Cx + Du \\ \dot{\hat{x}} &= A\hat{x} + Bu + L(y - \hat{y}), \quad \hat{y} = C\hat{x} + Du \end{aligned} \quad (6.68)$$

Substituting (6.67) into (6.68), the (system + observer) closed-loop dynamics become,

$$\begin{aligned} \dot{x} &= Ax - BK_x \hat{x} \\ \dot{\hat{x}} &= (A - BK_x - LC)\hat{x} + LCx \end{aligned} \quad (6.69)$$

or equivalently

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & -BK_x \\ LC & A - BK_x - LC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \quad (6.70)$$

Consider the state similarity transformation.

$$\begin{bmatrix} x \\ e_x \end{bmatrix} = \begin{bmatrix} I_{n_x} & 0_{n_x \times n_x} \\ -I_{n_x} & I_{n_x} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \quad (6.71)$$

Because of (6.65), it is easy to see that in the new coordinates, the closed-loop system dynamics (6.70) become upper block-diagonal.

$$\begin{bmatrix} \dot{x} \\ \dot{e}_x \end{bmatrix} = \begin{bmatrix} A - B K_x & -B K_x \\ 0 & A - L C \end{bmatrix} \begin{bmatrix} x \\ e_x \end{bmatrix} \quad (6.72)$$

Consequently, the eigenvalues of (6.72) consist of the controller induced closed-loop eigenvalues $\lambda(A - B K_x)$ and the eigenvalues $\lambda(A - L C)$ of the closed-loop full-order state observer dynamics. So the controller and the observer gains, K and L , can be designed independent of each other to enforce global exponential closed-loop stability of the combined system (6.72). These arguments substantiate the Separation Principle [7, 8] in the design of observer-based control systems.

Theorem 6.3 Suppose that the LTI dynamics (6.54) are controllable (stabilizable) and observable (detectable). Then an observer-based linear output feedback dynamic controller in the form

$$\begin{aligned} \dot{\hat{x}} &= (A - B K_x - L(C - D K_x)) \hat{x} + L y \\ u &= -K_x \hat{x} \end{aligned} \quad (6.73)$$

can be found such that the corresponding $(2n_x)$ -dimensional closed-loop system (6.70) is globally exponentially stable. If (A, B) is controllable and (A, C) is observable, the combined system closed-loop eigenvalues can be assigned independently by appropriately choosing the controller and the observer gains.

$$\begin{aligned} \det \left[\lambda I_{2n_x} - \begin{bmatrix} A & -B K_x \\ L C & A - B K_x - L C \end{bmatrix} \right] &= \det \left[\lambda I_{2n_x} - \begin{bmatrix} A - B K_x & -B K_x \\ 0 & A - L C \end{bmatrix} \right] \\ &= \det[\lambda I_{n_x} - A + B K_x] \det[\lambda I_{n_x} - A + L C] \end{aligned} \quad (6.74)$$

The Separation Principle (6.74) decouples controller and observer designs in the sense that the total system eigenvalues can be assigned by solving the state feedback control and the observer pole placement problems separately, while calculating suitable gains K_x and L , independent of each other.

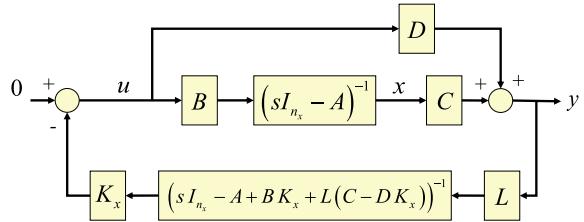
$$\text{eig} \begin{bmatrix} A & -B K_x \\ L C & A - B K_x - L C \end{bmatrix} = \text{eig}[A - B K_x] \cup \text{eig}[A - L C] \quad (6.75)$$

The observer-based control system block diagram is given in Fig. 6.10.

The block diagram clearly shows extra dynamics due to the state observer in the loop. With the output feedback controller (6.73), the loop gain transfer function matrix computed at the input break point is

$$L_u \text{OF}(s) = K_x (s I_{n \times n} - A + B K_x + L(C - D K_x))^{-1} L \left(C(s I_{n \times n} - A)^{-1} B + D \right) \quad (6.76)$$

Fig. 6.10 The observer-based output feedback control block diagram



which is quite different from the loop gain at the same input break point and due to the state feedback control law

$$L_{u \text{ SF}}(s) = K_x (C(s I_{n \times n} - A)^{-1} B + D) \quad (6.77)$$

The closed-loop system state dynamics can be written via (6.72),

$$\dot{x} = (A - B K_x)x - B K_x e_x \quad (6.78)$$

with the observer state estimation error $e_x(t)$ decaying exponentially fast, as specified in (6.65). Selecting the observer gains L to make the estimation error decay faster than the system dynamics is essential to an overall satisfactory closed-loop performance of the system (6.70), whose dynamics are equivalent to that of (6.72). A good practical rule of thumb is to place the observer eigenvalues $\text{eig}(A - L C)$ such that their natural frequencies are three to four times larger than the eigenvalues $\text{eig}(A - B K_x)$ achieved under the state feedback gain K_x .

It turns out that making the observer dynamics too fast is not a good practice. Stability margins might significantly degrade and the overall design would become impractical and sensitive to model imperfections such as gain and time delay uncertainties. We shall analyze these effects for a generic class of LQR optimal servomechanism systems whose design is detailed in the next section.

6.2.2 Observer-Based Optimal Servomechanism Design

Consider the controllable and observable n_p -dimensional LTI dynamics,

$$\begin{aligned} \dot{x}_p &= A_p x_p + B_p u \\ y_{\text{meas}} &= C_p \text{meas} x_p + D_p \text{meas} u, \quad y_{\text{reg}} = C_p \text{reg} x_p + D_p \text{reg} u \end{aligned} \quad (6.79)$$

with n_u -controls $u \in R^{n_u}$, using the output measurements $y_{\text{meas}} \in R^{n_{\text{meas}}}$, and with the regulated outputs $y_{\text{reg}} \in R^{n_{\text{reg}}}$. Suppose that the system state is not accessible. Only measured regulated outputs are available for control. We shall also assume that there are as many regulated outputs as the number of the system control inputs. Finally, we assume

$$(n_{\text{meas}} + n_{\text{reg}}) \leq n_y, \quad n_{\text{reg}} = n_u \quad (6.80)$$

which also implies that the total number of the output measurements (sensors) exceeds the number of the system control inputs. This property will be exploited in Sect. 6.5, where a constructive method is developed to asymptotically recover optimal stability margins of the state feedback linear quadratic regulator (LQR).

For clarity of exposition, we discuss only full-order state observers. Extensions to using reduced-order observers are possible but will not be addressed here.

Without a loss of generality, the control-feedforward matrix $D_p \text{meas}$ in the output measurement definition (6.79) is set to zero. This is possible due to the fact that the observability of an LTI system remains valid even when some of the system output measurements are replaced with their integrals. This leads to the LTI dynamics without the control-feedforward terms in the measurements but not necessarily in the regulated outputs.

$$\begin{aligned}\dot{x}_p &= A_p x_p + B_p u \\ y_{\text{meas}} &= C_p \text{meas} x_p, \quad y_{\text{reg}} = C_p \text{reg} x_p + D_p \text{reg} u\end{aligned}\tag{6.81}$$

The problem of interest is to synthesize a control policy $u(\cdot)$ based on available output signals y_{meas} and y_{reg} such that the regulated output y_{reg} tracks bounded external commands $y_{\text{cmd}}(t) \in R^{n_u}$ with adequately small errors. To achieve the control goal, the servomechanism design [9] from Chap. 4 can be applied. The main idea is to imbed a tracking error integrator into a control solution and then design a Type I Proportional + Integral (PI) controller, which shall track constant commands and reject constant disturbances, globally asymptotically and with zero steady-state errors. When a sufficiently slow changing external command is presented, the PI controller will be able to follow it with bounded errors as desired.

We begin with the definition of the integrated tracking error,

$$\dot{e}_{yI} = y_{\text{reg}} - y_{\text{cmd}}\tag{6.82}$$

merge its dynamics with that of the open-loop LTI system (6.81),

$$\begin{aligned}\underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} &= \underbrace{\begin{pmatrix} 0_{n_u \times n_u} & C_p \text{reg} \\ 0_{n_p \times n_u} & A_p \end{pmatrix}}_{\tilde{A}} \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} D_p \text{reg} \\ B_p \end{pmatrix}}_{\tilde{B}} u + \underbrace{\begin{pmatrix} -I_{n_u} \\ 0_{n_p \times n_u} \end{pmatrix}}_{B_{\text{cmd}}} y_{\text{cmd}} \\ y_{\text{meas}} &= \underbrace{\begin{pmatrix} I_{n_u} & 0_{n_u \times n_p} \\ 0_{n_u \times n_u} & C_p \text{meas} \end{pmatrix}}_{C_{\text{meas}}} \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x, \quad y_{\text{reg}} = \underbrace{\begin{pmatrix} 0_{n_u \times n_u} & C_p \text{reg} \end{pmatrix}}_{C_{\text{reg}}} \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x + \underbrace{D_p \text{reg}}_{D_{\text{reg}}} u\end{aligned}\tag{6.83}$$

and form the extended $n_{\text{ext}} = (n_p + n_u)$ -dimensional open-loop plant.

$$\begin{aligned}\dot{x} &= \tilde{A} x + \tilde{B} u + B_{\text{cmd}} y_{\text{cmd}} \\ y_{\text{meas}} &= C_{\text{meas}} x, \quad y_{\text{reg}} = C_{\text{reg}} x + D_{\text{reg}} u\end{aligned}\tag{6.84}$$

Controllability of the extended dynamics (6.84) is preserved as long as,

$$\det \begin{bmatrix} A_p & B_p \\ C_{p \text{ reg}} & D_{p \text{ reg}} \end{bmatrix} \neq 0 \quad (6.85)$$

that is the original controllable system (6.81) has no transmission zeros at the origin, (from the control input to the regulated output).

For a constant external command y_{cmd} , a state feedback servomechanism control policy could be found to stabilize the extended state vector $x \in R^{n_{\text{ext}}}$. Because of the integrated tracking error dynamics (6.82), in equilibrium the regulated output y_{reg} is equal to the desired commanded value y_{cmd} , component wise. A state feedback servo-controller,

$$u = -K_{\text{ext}} x \quad (6.86)$$

can be found to stabilize the extended system dynamics (6.84) and force all closed-loop system trajectories to converge to the extended system equilibrium. In this case, the system regulated output y_{reg} will track constant external commands y_{cmd} and reject constant disturbances globally, asymptotically and with zero tracking errors. We derived state feedback servo-controllers in Chap. 4.

Next, we present derivations for optimal dynamic output feedback servo-controllers. These controllers can be constructed in the following three steps:

1. RSLQR: Use the RSLQR method from Chap. 4 with the system matrices (\tilde{A}, \tilde{B}) and suitably chosen state-control weights (Q, R) to calculate optimal state feedback gain K_{ext} .
2. Observer: Calculate the ARE-based observer gain L_v via the dual LQR design for (\tilde{A}^T, C^T) with weights for the process and the measurement noise in the system [7].
3. Form the optimal observer-based output feedback controller as in (6.73).

For the RSLQR state feedback design in Step 1, we select state and control matrices, $Q = Q^T \geq 0$ and $R = R^T > 0$, such that (\tilde{A}, \sqrt{Q}) is observable, calculate the unique positive-definite solution $P = P^T > 0$ of the Algebraic Riccati Equation (ARE),

$$P \tilde{A} + \tilde{A}^T P - P \tilde{B} R^{-1} \tilde{B}^T P + Q = 0 \quad (6.87)$$

compute the RSLQR optimal state feedback gain,

$$K_{\text{ext}} = R^{-1} B^T P \quad (6.88)$$

and define the corresponding RSLQR optimal state feedback control policy,

$$u = -K_{\text{ext}} x = -[K_I \ K_x] \begin{bmatrix} e_y I \\ x_p \end{bmatrix} = K_I s^{-1} (y_{\text{cmd}} - y_{\text{reg}}) - K_x x_p \quad (6.89)$$

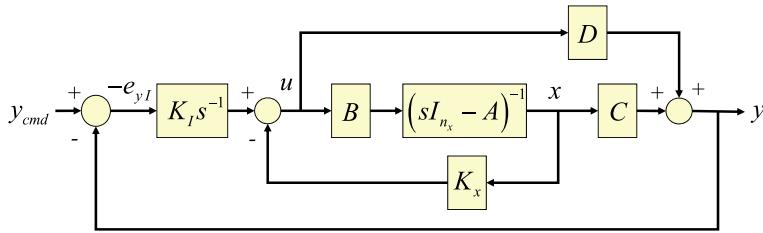


Fig. 6.11 The LQR optimal state feedback control block diagram

with the RSLQR gain K_{ext} partitioned into an integral and a proportional component, $K_I \in R^{n_u \times n_u}$ and $K_x \in R^{n_u \times n_p}$. Such a partition allows to explicitly rewrite the LQR solution (6.89) in terms of the original open-loop LTI system parameters. The RSLQR state feedback control block diagram is shown in Fig. 6.11.

In a stochastic setting, Step 2 gives the steady-state Kalman filter gain and, in this case, the full-order state observer becomes the Kalman–Bucy filter [7]. In the present deterministic case, the process and the measurement weight matrices for the observer design become the tuning parameters to achieve and recover the desired LQR state feedback gain and phase margins. The optimal state observer design is dual to that of the LQR controller [7, 8]. The design begins with a parameter-dependent filter ARE in the form,

$$P_v \tilde{A}^T + \tilde{A} P_v + Q_v - P_v C_{\text{meas}}^T R_v^{-1} C_{\text{meas}} P_v = 0 \quad (6.90)$$

where $Q_v \in R^{n_{\text{ext}} \times n_{\text{ext}}}$ and $R_v \in R^{n_u \times n_u}$ are symmetric positive-definite parameter-dependent matrices, and $v > 0$ is a constant. Specific forms for the weights will be introduced and discussed in the next section. Suffice to say, the unique symmetric positive-definite solution $P_v \in R^{n \times n}$ of the ARE (6.90) leads to the steady-state Kalman filter gain definition,

$$L_v = P_v C_{\text{meas}}^T R_v^{-1} \quad (6.91)$$

and subsequently to the formulation of the optimal Luenberger full-order state observer dynamics.

$$\dot{\hat{x}} = A \hat{x} + B u + B_{\text{cmd}} y_{\text{cmd}} + L_v (y_{\text{meas}} - \hat{y}_{\text{meas}}), \quad \hat{y}_{\text{meas}} = C_{\text{meas}} \hat{x} \quad (6.92)$$

Finally, the optimal observer-based output feedback controller is constructed via the Separation Principle, that is the control signal is formed using the LQR gains (6.88) while feeding back the observer state \hat{x} , instead of the system true state x , which is not accessible for formulating control policies.

$$u = -K_{\text{ext}} \hat{x} \quad (6.93)$$

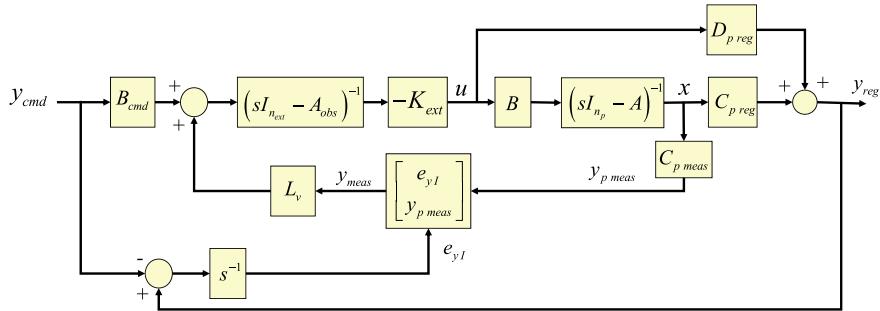


Fig. 6.12 The closed-loop system block diagram with an observer-based servomechanism

Substituting (6.93) into (6.92), we arrive at the closed-loop observer dynamics,

$$\dot{\hat{x}} = \underbrace{(A - B K_{ext} - L_v C_{meas})}_{A_{obs}} \hat{x} + B_{cmd} y_{cmd} + L_v y_{meas} \quad (6.94)$$

that are driven by the external command y_{cmd} and the output measurements y_{meas} . Together, Eqs. (6.93) and (6.94) form a dynamic observer-based optimal servomechanism controller in the form that is similar to (6.73). Figure 6.12 shows the corresponding closed-loop system block diagram.

It is imperative to verify stability of the observer matrix A_{obs} in (6.94). The design guarantees closed-loop stability of the (Observer + Controller) system but not necessarily provides a stable control solution (6.94). For all practical reasons, only stable controllers shall be utilized and that requires verification of A_{obs} stability, as an integral part of the overall observer-based servomechanism design process.

In the next two sections, we derive a numerically efficient methodology to construct observer-based dynamic servo-controllers that are able to recover RSLQR state feedback margins at the system input break point. We shall call this method the observer-based loop transfer recovery, or in short OBLTR. The OBLTR theoretical foundation is based on certain asymptotic properties of the Algebraic Riccati Equation.

6.2.3 Asymptotic Properties of the Algebraic Riccati Equation

Consider a parameter-dependent n_x -dimensional Algebraic Riccati Equation (ARE) in the form,

$$P_v A + A^T P_v - P_v B R_v^{-1} B^T P_v + Q_v = 0 \quad (6.95)$$

where the system matrix pairs (A, B) and (A, C) are controllable and observable, with $A \in R^{n_x \times n_x}$, $B \in R^{n_x \times n_u}$, $C \in R^{n_u \times n_x}$, and $n_u \leq n_x$. The subscript “ v ” in (6.95) denotes a constant scalar positive parameter. It enters into the definition of the ARE weight matrices,

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) C^T C, \quad R_v = \left(\frac{v}{v+1} \right) R_0 \quad (6.96)$$

where $Q_0 \in R^{n_x \times n_x}$ and $R_0 \in R^{n_u \times n_u}$ are both symmetric and strictly positive definite. This formulation originated in [10].

For any fixed constant $v > 0$, (6.95) has the unique symmetric positive-definite solution $P_v > 0$. Such an ARE arises in the optimal linear quadratic regulator (LQR) control problems for linear time-invariant dynamics,

$$\dot{x} = A x + B u, \quad y = C x \quad (6.97)$$

with a quadratic minimization criterion in the form,

$$\begin{aligned} J_v &= \int_0^\infty \left(x^T Q_v x + u^T R_v u \right) dt \\ &= \int_0^\infty \left(x^T \left(Q_0 + \left(\frac{v+1}{v} \right) C^T C \right) x + \left(\frac{v}{v+1} \right) u^T R_0 u \right) dt \\ &= \int_0^\infty \left(x^T Q_0 x + \left(\frac{v+1}{v} \right) y^T y + \left(\frac{v}{v+1} \right) u^T R_0 u \right) dt \end{aligned} \quad (6.98)$$

From (6.98), we can immediately see that small values of v in the system cost (6.98) lead to larger weights on the system output y and to smaller weights on the system input u . So, the resulting optimal control policy gives faster output stabilization at the expense of a larger control effort. In that context, v parameterizes LQR weights and allows to study a tradeoff between closed-loop system output performance and an optimal control effort required to achieve it. We shall study properties of ARE solutions as $v \rightarrow 0$. In order to formally do that, we need to introduce basic concepts in asymptotic analysis.

Introduction to Asymptotic Orders

Let R^n represent the Euclidean n -dimensional space, R^+ be the set of all positive real numbers, and let $R^{n \times m}$ denote the space of all n -by- m matrices, with integers n and m . For any $x \in R^n$, we write $\|x\|$ for a Euclidean vector norm of x , and $\|A\|$ to be the corresponding induced matrix norm, for $A \in R^{n \times m}$.

We shall use the Bachmann–Landau asymptotic-order notation, denoted by the “Big O” and the “Small o” symbols [11, 12].

Definition 6.3 (*The “Big O” Symbol*) Given any two parameter-dependent functions (maps) $f(x; \varepsilon)$ and $g(x; \varepsilon)$, from a domain $X \subset R^n$ to another domain $Y \subset R^m$, with a scalar parameter $\varepsilon \in E \subset R$ from an interval E , we say that

$$f(x; \varepsilon) = O(g(x; \varepsilon)) \quad (6.99)$$

if for each $x \in X$, there exists a positive scalar $k(x)$ such that,

$$\|f(x; \varepsilon)\| \leq k(x)\|g(x; \varepsilon)\| \quad (6.100)$$

for all $\varepsilon \in E$. The asymptotic order (6.99) is said to be uniformly valid in the domain X if $k(x)$ is a finite constant independent of x .

Choose $\varepsilon_0 \in E$ and suppose that the two limits exist.

$$\lim_{\varepsilon \rightarrow \varepsilon_0} \|f(x; \varepsilon)\| = \|f(x; \varepsilon_0)\|, \quad \lim_{\varepsilon \rightarrow \varepsilon_0} \|g(x; \varepsilon)\| = \|g(x; \varepsilon_0)\| \quad (6.101)$$

Then we can define,

$$f(x; \varepsilon) = O(g(x; \varepsilon)), \quad \text{as } \varepsilon \rightarrow \varepsilon_0 \quad (6.102)$$

if for each $x \in X$, there exists a positive scalar $k(x)$ and a neighborhood/interval $N(x, \varepsilon_0)$ of $\varepsilon = \varepsilon_0$, such that (6.100) holds for all $\varepsilon \in N(x, \varepsilon_0)$. In addition, if the set $N(x, \varepsilon_0) = N(\varepsilon_0)$ is independent of x then (6.102) is said to be uniformly valid in x .

Without a loss of generality, we proceed to study asymptotics with respect to the origin and assume that $\varepsilon_0 = 0$. In this case, the asymptotic-order relation (6.102) defines the convergence rate of $\|f(x; \varepsilon)\|$ to $\|f(x; 0)\|$, as $\varepsilon \rightarrow 0$, while holding x fixed. Specifically, for every fixed $x \in X$, $\|f(x; \varepsilon)\|$ converges to its limit $\|f(x; 0)\|$ no slower than $\|g(x; \varepsilon)\|$ converges to $\|g(x; 0)\|$, as $\varepsilon \rightarrow 0$. This convergence may hold uniformly in X , yet it could completely fail outside of X .

The asymptotic-order relations (6.99) and (6.102) define the Big O symbol. In order to shorten notation, we often would drop both arguments and write $f = O(g)$ instead.

We shall often use $O(1)$ to define uniformly bounded functions. However, note that $f(x; \varepsilon) = O(1)$ and $g(x; \varepsilon) = O(1)$ does not imply $f(x; \varepsilon) = g(x; \varepsilon)$. These asymptotic equalities should be understood in the “left-to-right” sense only, meaning that both functions, $f(x; \varepsilon)$ and $g(x; \varepsilon)$, are uniformly bounded for all $x \in X$, as $\varepsilon \rightarrow 0$.

For example, $f(x; \varepsilon) = x + \varepsilon = O(1)$ and $g(x; \varepsilon) = \sin(x)\varepsilon^2 = O(1)$, for all $0 < x < 1$ and $0 < \varepsilon < 1$. However, these are two very different functions, $f(x; \varepsilon) \neq g(x; \varepsilon)$, and their only common feature is that both functions are uniformly bounded on the noted intervals. In addition, $g(x; \varepsilon) = \sin(x)\varepsilon^2 = O(\varepsilon^2) = O(\varepsilon) = O(1)$. That is an asymptotic property of $g(x; \varepsilon)$. It says that the function decays as fast as $O(\varepsilon^2)$, uniformly on the bounded interval $x \in (0, 1)$, as $\varepsilon \rightarrow 0$.

Consider another example: $f(x; \varepsilon) = \frac{\varepsilon}{x}$ for $[0 < x < 1, 0 < \varepsilon < 1]$. In this case, $f(x; \varepsilon) = O(\varepsilon) = O(1)$ for any fixed value of x but this order is not uniform, since for any fixed ε the function tends to infinity as $x \rightarrow 0$.

Next, we introduce the so-called “Small o” symbol. In addition to specifying an asymptotic bound, the “Small o” allows to quantify the rate of convergence of a parameter-dependent function with respect to another parameter-dependent function.

Definition 6.4 (*The “Small o” Symbol*) For a given domain $X \subset R^n$, the statement

$$f(x; \varepsilon) = o(g(x; \varepsilon)), \quad \text{as } \varepsilon \rightarrow 0 \quad (6.103)$$

means that for each $x \in X$ and any given $\delta > 0$, there exists an ε -interval $N(x, \delta) = \{\varepsilon : 0 < \varepsilon \leq \varepsilon_1(x, \delta)\}$ such that

$$\|f(x; \varepsilon)\| \leq \delta \|g(x; \varepsilon)\| \quad (6.104)$$

for all $\varepsilon \in N(x, \delta)$. We say that (6.103) is uniformly valid in X if $\varepsilon_1(x, \delta) = \varepsilon_1(\delta)$ is independent of x .

Assuming that $\|g(x; \varepsilon)\| \neq 0$, we can interpret the “Small o” symbol via a limiting relation, point-wise in x .

$$\lim_{\varepsilon \rightarrow 0} \frac{\|f(x; \varepsilon)\|}{\|g(x; \varepsilon)\|} = 0 \quad (6.105)$$

Because of that, the notation $f \ll g$ is often used to indicate (6.103). Also we would frequently write $f = o(g)$ to shorten notation (6.103).

Consider two scalar parameter-dependent functions, $f(\varepsilon)$ and $g(\varepsilon)$, that are independent of the domain argument x and have finite limits at $\varepsilon = 0$. Also suppose that $g(\varepsilon) \neq 0$ in a neighborhood of 0. Then $f = O(g)$ implies that the functions ratio is bounded, that is $\frac{f}{g} = O(1)$, while $f = o(g)$ means $\frac{f}{g} = o(1)$, or equivalently $\lim_{\varepsilon \rightarrow 0} \frac{f(\varepsilon)}{g(\varepsilon)} = 0$. In other words, in this case the functions ration is bounded and tends to zero, as $\varepsilon \rightarrow 0$.

Using “Big O” and “Small o” definitions, we can perform various asymptotic operations.

Lemma 6.1 *For any parameter-dependent function $f(\varepsilon)$ that has a finite limit at zero, the following asymptotics take place as $\varepsilon \rightarrow 0$.*

$$\begin{aligned} O(f) &= f(O(1)) \\ O(O(f)) &= O(f) \\ o(o(f)) &= o(f) \end{aligned} \quad (6.106)$$

If $f(\varepsilon) = O(g(\varepsilon))$ for any function $g(\varepsilon)$ whose limit at zero is finite, then

$$\begin{aligned} O(o(f)) &= o(O(f)) = o(g) \\ O(f)O(g) &= O(fg) \\ O(f)o(g) &= o(f)o(g) = o(fg) \end{aligned} \quad (6.107)$$

Proof of lemma is straightforward and left to the reader as an exercise.

The following example illustrates the idea of interpreting Taylor series expansions as asymptotic-order relations. For a smooth scalar function with finite derivatives at the origin $f^{(k)}(0)$, we can write its Taylor series approximation.

$$f(\varepsilon) = \underbrace{\sum_{k=0}^n \frac{f^{(k)}(0)}{k!} \varepsilon^k}_{g(\varepsilon)} + O(\varepsilon^{n+1}) = g(\varepsilon) + O(\varepsilon^{n+1}) \quad (6.108)$$

This expression also defines an asymptotic order for the function approximation error.

$$E(\varepsilon) = f(\varepsilon) - g(\varepsilon) = O(\varepsilon^{n+1}) \quad (6.109)$$

So as $\varepsilon \rightarrow 0$, the approximation error converges to zero at the rate of $O(\varepsilon^{n+1})$.

In asymptotic analysis, the “Big O” symbol is more informative than the “Small o” in the sense that it gives precision to the behavior of the function in question. If for example $f(\varepsilon) = 1 + \varepsilon + O(\varepsilon^2)$ then $(f(\varepsilon) - 1 - \varepsilon)$ decays to zero like $O(\varepsilon^2)$, and specifies the rate of asymptotic behavior. On the other hand, the same function can be written as $f(\varepsilon) = 1 + \varepsilon + o(\varepsilon)$. It tells us that $(f(\varepsilon) - 1 - \varepsilon)$ tends to zero faster than ε but does not define how fast. So “Big O” defines rates of convergence, while “Small o” indicates convergence to zero without rate specification.

We note that, $o(1)$ denotes all functions that tend to zero as $\varepsilon \rightarrow 0$. At the same time, $O(1)$ represents a set of all bounded functions. Clearly, $O(1)$ is a larger set than $o(1)$. In fact, we can show that for $0 \leq \varepsilon \ll 1$, $o(1) = O(1)$ but not vice versa. It is important to understand that this is not an equality relation. It merely states the fact that a parameter-dependent function that tends to zero is bounded for sufficiently small parameter values.

Asymptotic Expansions

The O - and o -symbols can be easily extended to parameter-dependent matrices $A(x; \varepsilon) \in R^{n \times n}$, using vector-induced matrix norms. For example, given a matrix $A_\varepsilon(x) = A(x; \varepsilon) \in R^{n \times n}$, the matrix asymptotic expansion

$$A_\varepsilon(x) = A_0(x) + A_1(x)\varepsilon + O(\varepsilon^2), \quad \text{as } \varepsilon \rightarrow 0 \quad (6.110)$$

means that for every x from a domain $X \subset R^n$,

$$\lim_{\varepsilon \rightarrow 0} \|A_\varepsilon(x) - A_0(x) - A_1(x)\varepsilon\| = \lim_{\varepsilon \rightarrow 0} \|O(\varepsilon^2)\| = 0 \quad (6.111)$$

and the convergence rate in (6.111) is no slower than ε^2 , for every fixed x . We immediately note that there is a difference between the asymptotic expansion (6.110) and, for example, the Taylor series expansion of a state-parameter-dependent matrix $A_\varepsilon(x)$. In this case, a Taylor series expansion may not even exist, since in (6.110) a differentiability of $A_\varepsilon(x)$ with respect to ε is not assumed.

In (6.110), the parameter-dependent matrix $A_\varepsilon(x)$ is expanded using polynomials in ε , as the asymptotic expansion basis functions. Can other than polynomial basis functions be used to perform asymptotic expansions? For a given $A_\varepsilon(x)$, how do we define suitable basis functions to perform asymptotic expansions? The following definition [11] lays out the framework for asymptotic expansions of a given function.

Definition 6.4 A sequence of functions $\{\varphi_n(\varepsilon)\}_{n=1}^\infty$ is called an asymptotic sequence if

$$\varphi_{n+1}(\varepsilon) = o(\varphi_n(\varepsilon)), \quad \varepsilon \rightarrow 0 \quad (6.112)$$

for each $n = 1, 2, \dots$

As expected, polynomials $\{\varepsilon^{n-1}\}_{n=1}^\infty$ satisfy (6.112) and thus represent an asymptotic sequence. So does the following finite sequence,

$$\varphi_1(\varepsilon) = \log \varepsilon, \quad \varphi_2(\varepsilon) = 1, \quad \varphi_3(\varepsilon) = \varepsilon \log \varepsilon, \quad \varphi_4(\varepsilon) = \varepsilon \quad (6.113)$$

where the first component becomes unbounded as $\varepsilon \rightarrow 0$. Thus the definition (6.112) does not preclude having one or more infinite terms in an asymptotic sequence.

Definition 6.5 Consider a parameter-dependent function $f(x; \varepsilon)$ defined on a domain X of x and for some neighborhood of $\varepsilon = 0$. Let $\{\varphi_n(\varepsilon)\}$ be a given asymptotic sequence. If there exists a sequence of functions $\{f_n(x)\}$ on X such that,

$$f(x; \varepsilon) = \sum_{n=1}^N \varphi_n(\varepsilon) f_n(x) + o(\varphi_N(\varepsilon)) \quad (6.114)$$

then (6.114) is called the asymptotic expansion of $f(x; \varepsilon)$ to N terms as $\varepsilon \rightarrow 0$ with respect to the sequence $\{\varphi_n(\varepsilon)\}$.

For an explicitly defined function $f(x; \varepsilon)$ and a given asymptotic sequence $\{\varphi_n(\varepsilon)\}$, the sequence $\{f_n(x)\}$ can be found uniquely by repeated application of definition (6.114).

$$\begin{aligned}
 f(x; \varepsilon) &= \varphi_1(\varepsilon)f_1(x) + o(\varphi_1(\varepsilon)) \Rightarrow f_1(x) = \lim_{\varepsilon \rightarrow 0} \frac{f(x; \varepsilon)}{\varphi_1(\varepsilon)} \\
 f(x; \varepsilon) &= \varphi_1(\varepsilon)f_1(x) + \varphi_2(\varepsilon)f_2(x) \\
 &\quad + o(\varphi_2(\varepsilon)) \Rightarrow f_2(x) = \lim_{\varepsilon \rightarrow 0} \frac{f(x; \varepsilon) - \varphi_1(\varepsilon)f_1(x)}{\varphi_2(\varepsilon)} \\
 &\dots \\
 f(x; \varepsilon) &= \sum_{n=1}^N \varphi_n(\varepsilon)f_n(x) + o(\varphi_N(\varepsilon)) \\
 \Rightarrow f_n(x) &= \lim_{\varepsilon \rightarrow 0} \frac{f(x; \varepsilon) - \sum_{n=1}^{N-1} \varphi_n(\varepsilon)f_n(x)}{\varphi_N(\varepsilon)}
 \end{aligned} \tag{6.115}$$

Once again, we would like to emphasize that in contrast to Taylor series expansions, asymptotic expansions do not require differentiability of the function under consideration.

Often it is of interest to asymptotically expand an unknown function that might be defined via an algebraic relation such as the ARE in (6.95). In that case, a suitable asymptotic sequence $\{\varphi_n(\varepsilon)\}$ may not be known beforehand but it can be determined using definitions (6.112), (6.114) and the original relation that the function is a solution of.

Asymptotic Properties of the Algebraic Riccati Equation

We now return to ARE (6.95) and study asymptotic properties of its unique symmetric positive-definite solution with respect to parameter-dependent weights from (6.96). Using these weights, we get

$$P_v A + A^T P_v - \left(1 + \frac{1}{v}\right) P_v B R_0^{-1} B^T P_v + Q_0 + \left(1 + \frac{1}{v}\right) C^T C = 0 \tag{6.116}$$

or, equivalently

$$\begin{aligned}
 P_v A + A^T P_v - P_v B R_0^{-1} B^T P_v + Q_0 + C^T C + \frac{1}{v} [C^T C - P_v B R_0^{-1} B^T P_v] \\
 = 0
 \end{aligned} \tag{6.117}$$

We are interested in analyzing asymptotic properties of the ARE unique symmetric positive-definite solution P_v , as $v \rightarrow 0$. In what follows, we will be using v as the asymptotic parameter instead of the “classical notation” ε .

Rewriting (6.117) as,

$$v \left(P_v A + A^T P_v + Q_0 \right) + (v+1) \left(C^T C - P_v B R_0^{-1} B^T P_v \right) = 0 \quad (6.118)$$

and setting $v = 0$, gives the limiting relation for the corresponding ARE solution.

$$P_0 B R_0^{-1} B^T P_0 = C^T C \quad (6.119)$$

To better understand ARE solution dependence on v , we consider the following asymptotic expansion,

$$P_v = P_0 + \varphi_1(v) P_1 + o(\varphi_1(v)), \quad v \rightarrow 0 \quad (6.120)$$

where a scalar function $\varphi_1(v)$ and a positive-definite symmetric matrix P_1 need to be determined such that the remainder $(P_v - P_0 - \varphi_1(v) P_1)$ is of order $o(\varphi_1(v))$.

$$\lim_{v \rightarrow 0} \frac{\|P_v - P_0 - \varphi_1(v) P_1\|}{\varphi_1(v)} = 0 \quad (6.121)$$

This is the first step in building an asymptotic series expansion for P_v . We shall follow definitions (6.112) and (6.120) to do so. Substituting (6.120) into (6.118), yields

$$\begin{aligned} 0 &= v \left((P_0 + \varphi_1 P_1 + o(\varphi_1)) A + A^T (P_0 + \varphi_1 P_1 + o(\varphi_1)) + Q_0 \right) \\ &\quad + (v+1) \left(C^T C - \underbrace{(P_0 + \varphi_1 P_1 + o(\varphi_1)) B R_0^{-1} B^T (P_0 + \varphi_1 P_1 + o(\varphi_1))}_Z \right) \end{aligned} \quad (6.122)$$

Using “Small o” definition (6.103), we can simplify Z term in (6.122).

$$\begin{aligned} Z &= (P_0 + \varphi_1 P_1 + o(\varphi_1)) B R_0^{-1} B^T (P_0 + \varphi_1 P_1 + o(\varphi_1)) \\ &= P_0 B R_0^{-1} B^T P_0 + \varphi_1 P_0 B R_0^{-1} B^T P_1 + \underbrace{o(\varphi_1) P_0 B R_0^{-1} B^T}_{o(\varphi_1)} \\ &\quad + \varphi_1 P_1 B R_0^{-1} B^T P_0 + \underbrace{\varphi_1 \left(\varphi_1 P_1 B R_0^{-1} B^T P_1 + o(\varphi_1) P_1 B R_0^{-1} B^T \right)}_{o(\varphi_1)} \end{aligned}$$

$$\begin{aligned}
& + \underbrace{o(\varphi_1) B R_0^{-1} B^T (P_0 + \varphi_1 P_1 + o(\varphi_1))}_{o(\varphi_1)} \\
& = P_0 B R_0^{-1} B^T P_0 + \varphi_1 \left(P_1 B R_0^{-1} B^T P_0 + P_0 B R_0^{-1} B^T P_1 \right) + o(\varphi_1) \quad (6.123)
\end{aligned}$$

Then (6.122) becomes

$$\begin{aligned}
0 &= v \left((P_0 + \varphi_1 P_1 + o(\varphi_1)) A + A^T (P_0 + \varphi_1 P_1 + o(\varphi_1)) + Q_0 \right) \\
&+ (v+1) \left(\underbrace{C^T C - P_0 B R_0^{-1} B^T P_0}_0 \right) \\
&- (v+1) \left(\varphi_1 \left(P_1 B R_0^{-1} B^T P_0 + P_0 B R_0^{-1} B^T P_1 \right) + o(\varphi_1) \right) \\
&= v \left((P_0 + \varphi_1 P_1 + o(\varphi_1)) A + A^T (P_0 + \varphi_1 P_1 + o(\varphi_1)) + Q_0 \right) \\
&- (v+1) \left(\varphi_1 \left(P_1 B R_0^{-1} B^T P_0 + P_0 B R_0^{-1} B^T P_1 \right) + o(\varphi_1) \right) \quad (6.124)
\end{aligned}$$

We can rewrite (6.124) as,

$$\begin{aligned}
&\frac{v}{v+1} \left((P_0 + \varphi_1 P_1 + o(\varphi_1)) A + A^T (P_0 + \varphi_1 P_1 + o(\varphi_1)) + Q_0 \right) \\
&= \varphi_1 \left(P_1 B R_0^{-1} B^T P_0 + P_0 B R_0^{-1} B^T P_1 \right) + o(\varphi_1) \quad (6.125)
\end{aligned}$$

Since for $v > 0$, $\left(\frac{v}{v+1}\right)o(\varphi_1) = o(\varphi_1)$ then

$$\begin{aligned}
&\left(\frac{v}{v+1}\right) \left(P_0 A + A^T P_0 + Q_0 \right) + \left(\frac{v}{v+1}\right) \varphi_1 \left(P_1 A + A^T P_1 \right) \\
&- \varphi_1 \left(P_1 B R_0^{-1} B^T P_0 + P_0 B R_0^{-1} B^T P_1 \right) = o(\varphi_1) \quad (6.126)
\end{aligned}$$

This asymptotic relation needs to be consistent and hold as $v \rightarrow 0$. Suppose we select $\varphi_1(v) = o\left(\frac{v}{v+1}\right)$ then (6.126) reduces to

$$\underbrace{\left(\frac{v}{v+1}\right) \left(P_0 A + A^T P_0 + Q_0 \right)}_{O(v)} = o(\varphi_1) = o\left(o\left(\frac{v}{v+1}\right)\right) = o\left(\frac{v}{v+1}\right) = o(v) \quad (6.127)$$

which is a contradiction since $O(v) \neq o(v)$.

On the other hand, if we choose

$$\varphi_1(v) = O\left(\frac{v}{v+1}\right) = O(v), \quad v \rightarrow 0 \quad (6.128)$$

then (6.126) has the consistent asymptotic order.

$$\begin{aligned} & \underbrace{\left(\frac{v}{v+1}\right)\left(P_0A + A^TP_0 + Q_0\right)}_{O(v)} + \underbrace{O\left(\left(\frac{v}{v+1}\right)^2\right)\left(P_1A + A^TP_1\right)}_{O(v^2)=O(v)} \\ & - \underbrace{O\left(\frac{v}{v+1}\right)\left(P_1B R_0^{-1}B^T P_0 + P_0B R_0^{-1}B^T P_1\right)}_{O(v)} = o(\varphi_1) = o(O(v)) = O(v) \end{aligned} \quad (6.129)$$

We note that (6.128) does not define $\varphi_1(v)$ uniquely. Instead, we have found the function order of magnitude in asymptotic sense as $v \rightarrow 0$, such that (6.126) becomes a consistent asymptotic relation.

Based on (6.128), we select $\varphi_1(v) = v$ and consider the following asymptotic expansion for the ARE solution,

$$P_v = P_0 + P_1 v + O(v^2), \quad \text{as } v \rightarrow 0 \quad (6.130)$$

with the big O symbol $O(v^2)$ denoting a v -dependent $(n \times n)$ -matrix, whose induced norm tends to zero no slower than v^2 , as $v \rightarrow 0$.

$$\lim_{v \rightarrow 0} \|P_v - P_0 - P_1 v\| = \lim_{v \rightarrow 0} \|O(v^2)\| = 0 \quad (6.131)$$

For matrices satisfying (6.130), we can also define

$$P_0 = \lim_{v \rightarrow 0} P_v$$

which means $\lim_{v \rightarrow 0} \|P_v - P_0\| = 0$. As previously noted, limits of parameter-dependent matrices are understood in terms of their induced norms.

Before proceeding any further, we need to introduce a square root of a matrix according to [13].

Definition 6.6 An $(n \times n)$ matrix $S = P^{\frac{1}{2}} = \sqrt{P}$ is called a square root of a symmetric positive-definite $(n \times n)$ matrix P , if $P = S^T S$.

It is not so difficult to see that matrix square roots are by no means unique. However, we can define the unique square root by taking S to be symmetric.

We are now fully equipped to state and prove several interesting asymptotic properties related to the unique symmetric positive-definite solution P_v of the

parameter-dependent ARE (6.117). Later on, these important asymptotics will allow us to perform stability robustness analysis for observer-based control systems, as well as for systems operating under adaptive control (Chaps. 13–15).

Theorem 6.4 Consider parameter-dependent ARE (6.95) with any two controllable and observable matrix pairs, (A, B) and (A, C) , and with two symmetric positive-definite matrices Q_v and R_v from (6.96), where $A \in R^{n \times n}$, $B \in R^{m \times n}$, $C \in R^{p \times n}$, $Q_v \in R^{n \times n}$, $R_v \in R^{m \times m}$, and n, p, m are positive integers. Then for any positive scalar parameter v , ARE (6.95) has the unique symmetric positive-definite solution P_v .

If $p = m$, $\det(C B) \neq 0$, and the transfer function $G(s) = C(s I_{n \times n} - A)^{-1} B$ is minimum phase then the ARE solution P_v can be represented by the asymptotic expansion (6.130), while the following statements hold true:

1. P_0 and P_1 are symmetric.
2. P_0 is the unique symmetric strictly positive-definite solution of the algebraic Lyapunov equation:

$$P_0 \left(A - B R_0^{-1} B^T P_1 \right) + \left(A - B R_0^{-1} B^T P_1 \right)^T P_0 + Q_0 = 0 \quad (6.132)$$

3. There exists a unitary matrix $W \in R^{m \times m}$ such that:

$$P_0 B = C^T W^T \sqrt{R_0} \quad (6.133)$$

4. The unitary matrix W in (6.133) can be chosen as:

$$W = (U V)^T \quad (6.134)$$

where U and V are two unitary matrices defined by the singular value decomposition,

$$C B R_0^{-\frac{1}{2}} = U \Lambda V \quad (6.135)$$

and Λ represents the diagonal matrix of the corresponding singular values.

5. P_v is invertible for any $v \geq 0$, and for any unit vector $x \in R^n$.

$$\lim_{v \rightarrow 0} x^T P_v x \geq \lambda_{\min}(P_0) > 0 \quad (6.136)$$

where $\lambda_{\min}(P_0)$ denotes the minimum eigenvalue of P_0 .

6. The following asymptotic relation holds:

$$P_v B = C^T W^T \sqrt{R_0} + O(v), \quad \text{as } v \rightarrow 0 \quad (6.137)$$

Before proving the theorem, an immediate remark is in order. Relations (6.132) and (6.133) imply that the transfer function

$$\begin{aligned} G_0(s) &= B^T P_0 \left(s I_{n \times n} - A + B R_0^{-1} B^T P_1 \right)^{-1} B \\ &= \sqrt{R_0} W C \left(s I_{n \times n} - A + B R_0^{-1} B^T P_1 \right)^{-1} B \end{aligned}$$

becomes strictly positive real (SPR) [8] via feedback $u = -R_0^{-1} B^T P_1 x$, when the latter is applied to the linear dynamics (6.97). At the same time, the asymptotic expansions (6.130) and (6.137) mean that the transfer function,

$$G_v(s) = B^T P_v \left(s I_{n \times n} - A + B R_v^{-1} B^T P_v \right)^{-1} B \quad (6.138)$$

which is SPR by the design, approaches the transfer function,

$$G_y(s) = \sqrt{R_0} W C \left(s I_{n \times n} - A + B R_0^{-1} B^T P_0 \right)^{-1} B \quad (6.139)$$

that is

$$G_v(s) = G_y(s) + O(v), \quad \text{as } v \rightarrow 0 \quad (6.140)$$

point-wise in s .

Proof of Theorem 6.4 Existence and uniqueness of P_v is the well-known fact. We proceed by showing that matrices P_0 and P_1 in (6.130) are symmetric. Using (6.130) gives

$$P_0 = \lim_{v \rightarrow 0} P_v = \lim_{v \rightarrow 0} P_v^T = P_0^T \quad (6.141)$$

Consequently,

$$P_1 = \lim_{v \rightarrow 0} \frac{1}{v} (P_v - P_0) = \lim_{v \rightarrow 0} \frac{1}{v} (P_v^T - P_0^T) = P_1^T \quad (6.142)$$

Next, we substitute (6.130) into (6.117).

$$\begin{aligned} & v \left[(P_0 + P_1 v + O(v^2)) A + A^T (P_0 + P_1 v + O(v^2)) \right] \\ & - v \left[(P_0 + P_1 v + O(v^2)) B R_0^{-1} B^T (P_0 + P_1 v + O(v^2)) + Q_0 + C^T C \right] \\ & + C^T C - (P_0 + P_1 v + O(v^2)) B R_0^{-1} B^T (P_0 + P_1 v + O(v^2)) = 0 \quad (6.143) \end{aligned}$$

Collecting the zero-order terms in v , gives

$$C^T C - P_0 B R_0^{-1} B^T P_0 = 0 \quad (6.144)$$

So the matrix solution $(P_0 B)$ of (6.144) may be expressed as in (6.133), whose validity can be verified by direct substitution into (6.144).

Collecting first-order terms in v from (6.143), gives

$$\underbrace{P_0 \left(A - B R_0^{-1} B^T P_1 \right)}_{\tilde{A}} + \left(A - B R_0^{-1} B^T P_1 \right)^T P_0 - P_0 B R_0^{-1} B^T P_0 + \underbrace{\left(Q_0 + C^T C \right)}_{\tilde{Q}} = 0 \quad (6.145)$$

or, equivalently

$$P_0 \tilde{A} + \tilde{A}^T P_0 - P_0 B R_0^{-1} B^T P_0 + \tilde{Q} = 0 \quad (6.146)$$

Since a feedback connection, such as $u = -R_0^{-1} B^T P_1 x$, does not change controllability of (A, B) then (\tilde{A}, B) is also controllable. Moreover, since $\tilde{Q} = \tilde{Q}^T > 0$, then the ARE (6.146) has the unique symmetric positive-definite solution $P_0 = P_0^T > 0$. Finally, using (6.144) in (6.145), gives (6.132) and thus proves the second claim of the theorem.

Choosing the unitary matrix W as in (6.134), while using (6.135), results in

$$\begin{aligned} B^T P_0 B &= B^T C^T W^T \sqrt{R_0} = \sqrt{R_0} \left(R^{-\frac{1}{2}} B^T C^T \right) W^T \sqrt{R_0} \\ &= \sqrt{R_0} V^T \Lambda \underbrace{U^T U}_{I_m \times m} V \sqrt{R_0} = \sqrt{R_0} \left(V^T \Lambda V \right) \sqrt{R_0} > 0 \quad (6.147) \end{aligned}$$

Note that this particular choice of W supports the established positive-definiteness property of P_0 .

For any unit vector $x \in R^n$

$$\lim_{v \rightarrow 0} x^T P_v x = \lim_{v \rightarrow 0} x^T [P_0 + O(v)]x = x^T P_0 x \geq \lambda_{\min}(P_0) > 0 \quad (6.148)$$

We know that the ARE solution P_v is invertible for any fixed $v > 0$. Also, from (6.148) it follows that for a sufficiently small $v \geq 0$, the eigenvalues of P_v are bounded away from zero. Therefore, P_v is invertible globally and for any $v \geq 0$. Finally, we note that (6.137) is a direct consequence of (6.130) and (6.133). The proof of the theorem is complete. ■

Let us now make the following substitutions into the ARE (6.95),

$$A := A^T, \quad B := C^T \quad (6.149)$$

The resulting equation becomes,

$$P_v A^T + A P_v - P_v C^T R_v^{-1} C P_v + Q_v = 0 \quad (6.150)$$

where according to (6.96),

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) B B^T, \quad R_v = \frac{v}{v+1} R_0 \quad (6.151)$$

The reader may have noticed that such an ARE arises in the design of Kalman filters and Luenberger observers. Substituting (6.151) into (6.150), gives

$$P_v A^T + A P_v - \left(1 + \frac{1}{v} \right) P_v C^T R_0^{-1} C P_v + Q_0 + \left(1 + \frac{1}{v} \right) B B^T = 0 \quad (6.152)$$

or, equivalently

$$\begin{aligned} P_v A^T + A P_v - P_v C^T R_0^{-1} C P_v + Q_0 + B B^T + \frac{1}{v} [B B^T - P_v C^T R_0^{-1} C P_v] \\ = 0 \end{aligned} \quad (6.153)$$

For the parameter-dependent ARE in (6.153), all statements from Theorem 6.2 can be easily reformulated. These claims are summarized (without proofs) below.

Corollary 6.1 *Suppose that all assumptions from Theorem 6.4 hold. Then the unique positive-definite solution P_v of the ARE (6.150), with the weight matrices Q_v and R_v from (6.151), can be represented by the asymptotic expansion (6.130). Moreover, the following statements hold*

1. P_0 and P_1 are symmetric.
2. P_0 is the unique symmetric strictly positive-definite solution of the following algebraic Lyapunov equation:

$$P_0 \left(A - C^T R_0^{-1} C P_1 \right)^T + \left(A - C^T R_0^{-1} C P_1 \right) P_0 + Q_0 = 0 \quad (6.154)$$

3. There exists a unitary matrix $W \in R^{m \times m}$ such that:

$$P_0 C^T = B W^T \sqrt{R_0} \quad (6.155)$$

4. The unitary matrix W in (6.133) can be chosen as:

$$W = (U V)^T \quad (6.156)$$

where U and V are two unitary matrices, defined by the singular value decomposition,

$$B^T C^T R_0^{-\frac{1}{2}} = U \Lambda V \quad (6.157)$$

and Λ represents the diagonal matrix of the corresponding singular values.

5. P_v is invertible for any $v \geq 0$, and

$$\lim_{v \rightarrow 0} x^T P_v x \geq \lambda_{\min}(P_0) > 0 \quad (6.158)$$

where $\lambda_{\min}(P_0)$ denotes the minimum eigenvalue of P_0 .

6. The following asymptotic relation holds:

$$P_v C^T = B W^T \sqrt{R_0} + O(v), \quad \text{as } v \rightarrow 0 \quad (6.159)$$

Later in the book, we shall use the above statements to design observer-based controllers and model reference adaptive control (MRAC) systems with smooth transient dynamics. At the moment, let us make the following remark: Since P_v is invertible for any $v \geq 0$, one can define the matrix inverse,

$$\tilde{P}_v = P_v^{-1} \quad (6.160)$$

and analyze its property using an asymptotic expansion in the form

$$\tilde{P}_v = \tilde{P}_0 + O(v), \quad \text{as } v \rightarrow 0 \quad (6.161)$$

Substituting (6.161) into $\tilde{P}_v P_v = I_{n \times n}$, gives

$$I_{n \times n} = \tilde{P}_v P_v = (\tilde{P}_0 + O(v)) (P_0 + O(v)) = \tilde{P}_0 P_0 + O(v), \quad \text{as } v \rightarrow 0 \quad (6.162)$$

Consequently,

$$I_{n \times n} = \lim_{v \rightarrow 0} \tilde{P}_v P_v = \tilde{P}_0 P_0 \quad (6.163)$$

and therefore

$$\left\{ \left[\tilde{P}_0 = P_0^{-1} \right] \Rightarrow \left[P_v^{-1} = P_0^{-1} + O(v) \right] \right\}, \quad \text{as } v \rightarrow 0 \quad (6.164)$$

Using (6.164) and (6.159), yields

$$\begin{aligned} C^T &= \tilde{P}_v \left(B W^T \sqrt{R_0} + O(v) \right) = \tilde{P}_v B W^T \sqrt{R_0} + \left(P_0^{-1} + O(v) \right) O(v) \\ &= \tilde{P}_v B W^T \sqrt{R_0} + O(v) \end{aligned} \quad (6.165)$$

and as a result, we obtain the asymptotic relation,

$$\tilde{P}_v B = C^T R_0^{-\frac{1}{2}} W + O(v) \quad (6.166)$$

which we shall employ in the design of robust and adaptive output feedback controllers later on in the book. This concludes asymptotic analysis of parameter-dependent ARE solutions.

6.2.4 The Squaring-Up Method

The asymptotics (6.166) will be used extensively to develop a constructive numerically efficient algorithm for the design of a full-order Luenberger state observer with guaranteed margins at the system input.

In the previous section, we have shown that under the following three conditions,

- The number of outputs and inputs in the system are the same: $n_y = n_u$.
- The system relative degree is one: $\det(C B) \neq 0$.
- The transfer function $G(s) = C(s I_{n \times n} - A)^{-1} B$ is minimum phase.

the ARE solution P_v and its inverse P_v^{-1} exist, and both matrices are symmetric strictly positive definite, uniformly in $v \geq 0$. We have also shown that the asymptotic relation (6.166) takes place, as $v \rightarrow 0$. However, the three assumed conditions are quite restrictive and most practical systems may not satisfy these assumptions.

Consider a generic class of linear time-invariant (LTI) systems whose input-output dynamics are not square, that is the number of the system inputs may or may not be the same as the number of its outputs. For these systems, existence of the asymptotics (6.166) is not guaranteed. In order to overcome the three restrictive

assumptions in Theorem 6.4, we are going to modify the system input–output dynamics to make it square, minimum phase and relative degree 1.

Practical methods to “square-up” LTI MIMO systems can be found in [14]. The squaring-up is accomplished by adding pseudo (i.e., fictitious) inputs or outputs. This procedure allows to enforce the three needed assumptions, and eventually it leads to the desired asymptotic relation (6.166).

We shall deal with “tall” LTI MIMO systems that have more outputs than inputs. In industrial applications, this is a reasonable supposition since system outputs represent sensors, and their number and locations can be chosen by the system designer [15, 16]. The squaring-up problem for a tall LTI MIMO system, with n_u inputs and ($n_y > n_u$) outputs,

$$\begin{array}{c} n_u\text{-Inputs} \\ \Downarrow \\ n_y\text{-Outputs} \Rightarrow \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in R^{(n_x+n_y) \times (n_x+n_u)} \end{array}$$

can be stated as follows [14]: “Given the state matrix $A \in R^{n_x \times n_x}$, the input matrix $B \in R^{n_x \times n_u}$, and the output matrices $C \in R^{n_y \times n_x}$, $D \in R^{p \times m}$, with ($n_x > n_u$, $n_y > n_u$), determine pseudo-input matrices $B_2 \in R^{n_x \times (n_y - n_u)}$ and $D_2 \in R^{n_y \times (n_y - n_u)}$, such that the resulting square system with n_y inputs and n_y outputs,

$$\begin{array}{c} n_y\text{-Inputs} \\ \Downarrow \\ n_y\text{-Outputs} \Rightarrow \begin{pmatrix} A & \bar{B} = (B, B_2) \\ C & \bar{D} = (D, D_2) \end{pmatrix} \in R^{(n_x+n_y) \times (n_x+n_y)} \end{array}$$

has its transmission zeros in the open left half complex plane, \mathbb{C}^- ” and $\text{rank}(C \bar{B}) = n_y$.

In [14], several constructive algorithms for solving the squaring-up problem are given. Systems that we shall encounter in this chapter will have no feedforward connections, that is $D = 0_{n_y \times n_u}$. In this case, the squaring-up problem is reduced to finding a pseudo-input matrix $B_2 \in R^{n_x \times (n_y - n_u)}$, such that the square system,

$$\begin{array}{c} n_y\text{-Inputs} \\ \Downarrow \\ n_y\text{-Outputs} \Rightarrow \begin{pmatrix} A & \bar{B} = (B, B_2) \\ C & 0_{p \times p} \end{pmatrix} \in R^{(n_x+n_y) \times (n_x+n_y)} \end{array}$$

has its transmission zeros in the open left half complex plane \mathbb{C}^- and $\text{rank}(C \bar{B}) = n_y$.

Squaring-up problems have multiple solutions. A set of sufficient conditions for a solution to exist is given below.

1. (A, C) is observable.
2. The system is tall $\text{rank}(C) = n_y > n_u = \text{rank}(B) = n_u$ and $\text{rank}(C B) = n_u$.
3. There are no finite transmission zeros in the closed right-half complex plane.

The first two conditions come from [14], while the third one is almost always satisfied for tall systems [7, 13].

If a squaring-up solution is found, the resulting system transfer function becomes square and has unity vector relative degree. Note however that individual transfer function components, such as the one from the control input to the regulated output, may have unstable zeros and arbitrary relative degree.

Next, we present the squaring-up design methodology for tall LTI MIMO systems,

$$\begin{aligned}\dot{x} &= A x + B u \\ y_{\text{reg}} &= C_{\text{reg}} x_{\text{reg}} + D_{\text{reg}} u \\ y_{\text{sens}} &= C_{\text{sens}} x_p + D_{\text{sens}} u\end{aligned}\quad (6.167)$$

where $x \in R^{n_x}$ denotes the n_x -dimensional state, $u \in R^{n_u}$ is the n_u -dimensional vector of controls, $y_{\text{reg}} \in R^{n_{y_{\text{reg}}}}$ represents the regulated output and $y_{\text{sens}} \in R^{n_{y_{\text{sens}}}}$ denotes other sensed measurements in the system. Define combined output matrices,

$$C_{\text{meas}} = \begin{pmatrix} C_{\text{reg}} \\ C_{\text{sens}} \end{pmatrix} \in R^{n_{y_{\text{meas}}} \times n_x}, \quad D_{p \text{ meas}} = \begin{pmatrix} D_{\text{reg}} \\ D_{\text{sens}} \end{pmatrix} \in R^{n_{y_{\text{meas}}} \times n_x} \quad (6.168)$$

where

$$n_{y_{\text{meas}}} = n_{y_{\text{reg}}} + n_{y_{\text{sens}}} \quad (6.169)$$

is the total number of the measured outputs in the system. The open-loop dynamics ($u = 0$) are allowed to be unstable and have arbitrary relative degree from the control input to the regulated output. However, the system finite transmission zeros (if they exist) are required to be stable.

Definition 6.3 A complex number $s_0 \in \mathbb{C}$ is the finite transmission zero of (6.54) if the Rosenbrock system matrix,

$$R(s) = \begin{pmatrix} A - s I_{n_x} & B \\ C_{\text{meas}} & D_{\text{meas}} \end{pmatrix} \quad (6.170)$$

drops rank at $s = s_0$.

$$\text{rank}(R(s_0)) = \text{rank} \begin{pmatrix} A - s_0 I_{n_x} & B \\ C_{\text{meas}} & D_{\text{meas}} \end{pmatrix} < n_x + \min(n_{y_{\text{meas}}}, n_u) \quad (6.171)$$

The system is called minimum phase if all finite transmission zeros are located in the open left half complex plane \mathbb{C}^- .

Derivation of the squaring-up method will be shown for the LTI MIMO dynamics without control-feedforward connections.

$$\begin{aligned}\dot{x} &= Ax + Bu, \quad y = Cx \\ x &\in R^{n_x}, \quad u \in R^{n_u}, \quad y \in R^{n_y}\end{aligned}\tag{6.172}$$

If the number of the system output measurements n_y is greater than the number of the control inputs n_u , the system is called “tall”. On the other hand, if $n_y < n_u$, the system is “wide”. Finally, if $n_y = n_u$ the system is “square”. Originally developed for a class of wide systems, the squaring-up method is easily restated for tall systems, via a duality argument and with the corresponding set of sufficient conditions.

For clarity, we focus on LTI MIMO systems that are controllable, observable, tall, and minimum phase.

Assumption 6.1

- (A, B) is controllable and $\text{rank}(B) = n_u \leq n_x$.
- (C, A) is observable and $\text{rank}(C) = n_y > n_u$.
- The tall system is minimum phase.

Controllability condition in Assumption 6.1 eliminates existence of input decoupling zeros in the system. Together with assumed observability, the system transmission zeros are exactly the same values in the complex plane where the system Rosenbrock matrix drops rank. The minimum phase assumption is not restrictive since most tall systems have no finite transmission zeros.

Squaring-up a tall system necessitates finding $(n_y - n_u)$ columns $B_2 \in R^{n_x \times (n_y - n_u)}$ such that all $(n_x - n_y)$ finite transmission zeros of the newly formed $(n_y \times n_y)$ -square system, with the added $(n_y - n_u)$ “fictitious” inputs, are placed at the prescribed locations in \mathbb{C}^- . Columns in B_2 can be thought of as extra input-directions for process disturbances entering the system dynamics.

After squaring-up, the modified system with the newly added inputs becomes $(n_y \times n_y)$ square and minimum phase which in turn means that $(n_x + n_y) \times (n_x + n_y)$ square Rosenbrock system matrix can drop rank only in the left half complex plane.

$$\boxed{\text{rank}\left(\begin{array}{cc} A - s_0 I_{n_x} & \bar{B} \\ C & 0_{n_y \times n_y} \end{array}\right) < n_x + n_y} \Rightarrow s_0 \in \mathbb{C}^-\tag{6.173}$$

Remark 6.1 For servomechanism design, we start with the original system and add the integrated tracking error,

$$\dot{e}_{yI} = y_{\text{reg}} - y_{\text{cmd}} \quad (6.174)$$

as an additional state component, arriving at the tall extended open-loop system,

$$\underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0_{n_u \times n_u} & C_p \text{ reg} \\ 0_{n_u \times n_u} & A_p \end{pmatrix}}_A \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} D_p \text{ reg} \\ B_p \end{pmatrix}}_B u + \underbrace{\begin{pmatrix} -I_{n_u} \\ 0_{n_x \times n_u} \end{pmatrix}}_{B_{\text{cmd}}} y_{\text{cmd}} \quad (6.175)$$

with $n_{\text{ext}} = n_x + n_u$ states, m controls and $n_y = n_u + n_{y \text{ sens}} > n_u$ output measurements.

$$y_{\text{meas}} = \underbrace{\begin{pmatrix} I_{n_u} & 0_{n_u \times n_x} \\ 0_{n_{y \text{ sens}} \times n_u} & C_p \text{ sens} \end{pmatrix}}_C \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x = C x \in R^{\overbrace{(n_u + n_{y \text{ sens}})}^{n_y} \times 1} \quad (6.176)$$

As seen from (6.176), the regulated output signal, when considered as a measurement, is replaced by the integrated tracking error e_{yI} . With such a redefinition of the output measurements, it can be shown that the system observability is preserved. Also, with the addition of (6.174), the original system dynamics with a possibly nonzero control-feedforward term is transformed into the extended dynamics, whose corresponding D -matrix is zero. ■

Next, we present a theoretical proof and derive a numerical algorithm to efficiently solve the squaring-up problem. Adding extra columns B_2 to the system input matrix B , yields $(n_x + n_y) \times (n_x + n_y)$ square Rosenbrock system matrix,

$$R(s) = \begin{pmatrix} A - s I_{n_x} & \bar{B} = (B \ B_2) \\ C & 0_{n_y \times n_y} \end{pmatrix} \quad (6.177)$$

where $B_2 \in R^{n_x \times (n_y - n_u)}$ needs to be found such that $R(s)$ drops rank at the prescribed transmission zeros in \mathbb{C}^- . With zero D -matrix, there are a total of $(n_x - n_y)$ finite transmission zeros in the $(n_y \times n_y)$ square system.

Consider singular value decomposition (SVD),

$$C = U_{n_y \times n_y} \left(\underbrace{\text{diag}(s_1, \dots, s_{n_y})}_{n_y \times n_y} 0_{n_y \times (n_x - n_y)} \right) \underbrace{V^T}_{n_x \times n_x} \quad (6.178)$$

with a diagonal matrix of singular values $S_{n_y \times n_y} = \text{diag}(s_1, \dots, s_{n_y})$ and introduce state similarity transformation.

$$z = V^T x \quad (6.179)$$

In the new coordinates,

$$\begin{aligned}\dot{z} &= \underbrace{\left(V^T A V \right)}_{A_z} z + \underbrace{\left(V^T B \right)}_{B_z} u, \quad y = \underbrace{\left(C V \right)}_{C_z} z \\ C_z &= C V = \left(\begin{array}{c|c} U S_{n_y \times n_y} & 0_{n_y \times (n_x - n_y)} \\ \hline C_{z1} & \end{array} \right) \end{aligned} \quad (6.180)$$

and the Rosenbrock system matrix (6.177) becomes,

$$R_z(s) = \left(\begin{array}{cc} \boxed{A_z - s I_n}_{n_x \times n_x} & \begin{array}{c} (B_{z11})_{n_y \times n_u} \\ (B_{z21})_{(n_x - n_y) \times n_u} \end{array} \\ \boxed{C_{z1}}_{n_y \times n_y} 0_{p \times (n_x - n_y)} & 0_{n_y \times n_u} \end{array} \right) \quad (6.181)$$

where

$$B_z = \left(\begin{array}{c} (B_{z11})_{n_y \times n_u} \\ (B_{z21})_{(n_x - n_y) \times n_u} \end{array} \right) \quad (6.182)$$

Clearly,

$$\begin{aligned} n_u &= \text{rank}(C B) = \text{rank} \left(\left(\boxed{C_{z1}}_{n_y \times n_y} 0_{n_y \times (n_x - n_y)} \right) \left(\begin{array}{c} B_{z11} \\ B_{z21} \end{array} \right) \right) \\ &= \text{rank}(C_{z1} B_{z11}) = \text{rank}(B_{z11}) \end{aligned} \quad (6.183)$$

and therefore, B_{z11} has full rank n_u . Consequently, there exists $(B_{z12})_{n_y \times (n_y - n_u)}$ such that the resultant square matrix $B_{z1} = \left(\begin{array}{cc} (B_{z11})_{n_y \times n_u} & (B_{z12})_{n_y \times (n_y - n_u)} \end{array} \right)$ is not singular. To find such a matrix, it is sufficient to consider SVD,

$$(B_{z11})_{n_y \times n_u} = \underbrace{\left(U_{n_y \times n_u} \ U_{n_y \times (n_y - n_u)} \right)}_{n_y \times n_y} \underbrace{\left(\begin{array}{c} \text{diag}(s_1, \dots, s_{n_u}) \\ \hline 0_{(n_y - n_u) \times n_u} \end{array} \right)}_{n_u \times n_u} \underbrace{V^T}_{n_u \times n_u} \quad (6.184)$$

and choose

$$(B_{z12})_{n_y \times (n_y - n_u)} = U_{n_y \times (n_y - n_u)} \quad (6.185)$$

to be complimentary orthogonal to B_{z11} .

Next, we define

$$B_z 2 = \left((B_{z21})_{(n_x - n_y) \times n_u} \ (B_{z22})_{(n_x - n_y) \times (n_y - n_u)} \right) \quad (6.186)$$

form the extended matrix,

$$B_z = \begin{pmatrix} \boxed{B_{z1}}_{n_y \times n_y} \\ \boxed{B_{z2}}_{(n_x - n_y) \times n_y} \end{pmatrix} \in R^{n_x \times n_y} \quad (6.187)$$

and consider the corresponding square Rosenbrock system matrix in the new coordinates.

$$\bar{R}_z(s) = \begin{pmatrix} \boxed{A_z - s I_{n_x}}_{n_x \times n_x} & \boxed{B_{z1}}_{n_y \times n_y} \\ \boxed{C_{z1}}_{n_y \times n_y} & 0_{n_y \times (n_x - n_y)} \end{pmatrix} \quad (6.188)$$

The transmission zero placement problem is reduced to finding $(n_x - n_y) \times (n_y - n_u)$ matrix B_{z22} such that $\bar{R}_z(s)$ drops rank at the designated $(n_x - n_y)$ locations in the left half complex plane \mathbb{C}^- . We can explicitly calculate the rank of the Rosenbrock matrix (6.188).

$$\begin{aligned} & \text{rank}(\bar{R}_z(s)) \\ &= \text{rank} \left\{ \begin{pmatrix} I_{n_y} & 0 \\ -B_{z2} B_{z1}^{-1} & I_{n_x - p} \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{z11} - s I_{n_y} & A_{z12} & B_{z1} \\ A_{z21} & A_{z22} - s I_{n_x - p} & B_{z2} \\ C_{z1} & 0_{n_y \times (n_x - n_y)} & 0_{n_y \times n_u} \end{pmatrix} \right\} \\ &= \text{rank} \left(\begin{array}{ccc|c} A_{z11} - s I_{n_y} & A_{z12} & \boxed{B_{z1}}_{n_y \times n_y} & \\ * & A_{z22} - B_{z2} B_{z1}^{-1} A_{z12} - s I_{n_x - n_y} & 0_{(n_x - n_y) \times n_y} & \\ C_{z1} & 0_{n_y \times (n_x - n_y)} & 0_{n_y \times n_y} & (n+p) \times (n+p) \end{array} \right) \quad (6.189) \end{aligned}$$

It easy to see that $\bar{R}_z(s)$ loses rank at the eigenvalues of

$$A_{z22cl} = A_{z22} - B_{z2} \underbrace{\left(B_{z1}^{-1} A_{z12} \right)}_{\tilde{A}_{z12}} = A_{z22} - B_{z2} \tilde{A}_{z12} \quad (6.190)$$

Multiplying the first row in (6.189) by non-singular B_{z1}^{-1} does not change the matrix rank.

$$\text{rank}(\bar{R}_z(s))$$

$$= \text{rank} \begin{pmatrix} B_z^{-1} (A_z 11 - s I_{n_y}) & \tilde{A}_z 12 & I_{n_y} \\ * & \boxed{A_z 22 - B_z 2 \tilde{A}_z 12 - s I_{n_x - n_y}} & 0_{(n_x - n_y) \times n_y} \\ C_z 1 & 0_{n_y \times (n_x - n_y)} & 0_{n_y \times n_y} \end{pmatrix} \quad (6.191)$$

Since the original pair (A, C) is observable then $(A_z 22, \tilde{A}_z 12)$ is also observable. Consequently, $B_z 2$ can be found to assign the target transmission zeros as the desired eigenvalues of A_{z22cl} in (6.190). However, not all elements of $B_z 2$ can be assigned arbitrarily. By definition (6.188), only $B_z 22$ is free to be selected. Consider matrix partitioning,

$$\tilde{A}_z 12 = \begin{pmatrix} \tilde{A}_z 121 \\ \tilde{A}_z 122 \end{pmatrix} \quad (6.192)$$

and rewrite (6.190).

$$A_{z22cl} = \left(A_z 22 - B_z 21 \tilde{A}_z 121 \right) - B_z 22 \tilde{A}_z 122 \quad (6.193)$$

It is easy to see that if the matrix pair $\left(\left(A_z 22 - B_z 21 \tilde{A}_z 121 \right), \tilde{A}_z 122 \right)$ is observable then $B_z 22 \in R^{(n_x - n_y) \times (n_y - n_u)}$ can be found to assign the $(n_x - n_y)$ desired transmission zeros to become the eigenvalues of A_{z22cl} . We can use pole placement method to do so. Alternatively, LQR design can be employed to find $B_z 22$ such that A_{z22cl} becomes Hurwitz. In that case, desired transmission zeros do not have to be explicitly specified.

The observability property of $\left(\left(A_z 22 - B_z 21 \tilde{A}_z 121 \right), \tilde{A}_z 122 \right)$ is discussed next. Rewrite (6.191).

$$\begin{aligned} \text{rank} (\bar{R}_z(s)) &= \text{rank} \begin{pmatrix} B_z^{-1} (A_z 11 - s I_{n_y}) & \tilde{A}_z 12 & I_{n_y} \\ * & \boxed{A_z 22 - B_z 2 \tilde{A}_z 12 - s I_{n_x - n_y}} & 0_{(n_x - n_y) \times n_y} \\ C_z 1 & 0_{n_y \times (n_x - n_y)} & 0_{n_y \times n_y} \end{pmatrix} \\ &= \text{rank} \begin{pmatrix} * & \boxed{\begin{matrix} \tilde{A}_z 121 \\ \tilde{A}_z 122 \end{matrix}} & \boxed{\begin{matrix} I_{n_u} & 0_{n_u \times (n_y - n_u)} \\ 0_{(n_y - n_u) \times n_u} & I_{n_y - n_u} \end{matrix}} \\ * & \boxed{\left(A_z 22 - B_z 21 \tilde{A}_z 121 \right) - B_z 22 \tilde{A}_z 122 - s I_{n_x - n_y}} & 0_{(n_x - n_y) \times n_y} \\ C_z 1 & 0_{n_y \times (n_x - n_y)} & 0_{n_y \times n_y} \end{pmatrix} \end{aligned}$$

$$= \text{rank} \begin{pmatrix} * & \boxed{\begin{matrix} 0_{n_u \times (n_x - n_y)} \\ \tilde{A}_{z122} \end{matrix}} & \boxed{\begin{matrix} I_{n_u} & 0_{n_u \times (n_y - n_u)} \\ 0_{(n_y - n_u) \times n_u} & I_{n_y - n_u} \end{matrix}} \\ * & \boxed{\begin{matrix} (A_{z22} - B_{z21}\tilde{A}_{z121}) - B_{z22}\tilde{A}_{z122} - sI_{n_x - n_y} \\ 0_{n_y \times (n_x - n_y)} \end{matrix}} & \begin{matrix} 0_{(n_x - n_y) \times n_y} \\ 0_{n_y \times n_y} \end{matrix} \\ C_{z1} & & \end{pmatrix} \quad (6.194)$$

The last expression in (6.194) results from right-multiplying the third column in the second expression by \tilde{A}_{z121} and subtracting it from the second column. It is rank preserving. If $((A_{z22} - B_{z21}\tilde{A}_{z121}), \tilde{A}_{z122})$ is not observable then there exists s_0 such that Rosenbrock matrix $\bar{R}_z(s_0)$ is rank deficient. That means s_0 is a finite transmission zero in the original tall system, which is a contradiction to the argument. So $((A_{z22} - B_{z21}\tilde{A}_{z121}), \tilde{A}_{z122})$ must be observable and the squaring-up procedure is justified.

A concise step-by-step summary of the squaring-up method is given in Table 6.1.

Table 6.1 The squaring-up method for tall LTI MIMO systems

Tall LTI MIMO system: $\begin{pmatrix} A & B \\ C & D \end{pmatrix} \in R^{(n_x+n_y) \times (n_x+n_u)}$, $n_y > n_u$

Check Assumption 6.1

Perform SVD: (6.178)

Apply state transformation: (6.179)

Compute B_{z12} using SVD (6.184) and (6.185)

Compute \tilde{A}_{z12} via (6.190) and (6.192)

Select $(n - p)$ target transmission zeros in \mathbb{C}^-

Compute B_{z22} to assign target transmission zeros to become eigenvalues of A_{z22cl} (6.193)

Form extended matrix B_z (6.187)

Compute squared-up matrix $\bar{B} = V B_z = \begin{pmatrix} B & B_2 \end{pmatrix} \in R^{n \times p}$ in original coordinates

Squared-up LTI MIMO system: $\begin{pmatrix} \bar{B} \\ A \begin{pmatrix} B & B_2 \end{pmatrix} \\ C \end{pmatrix} \in R^{(n_x+n_y) \times (n_x+n_y)}$

Example 6.2 Squaring-Up Design for Aircraft Short-Period Dynamics In this example, we shall use Table 6.1 to square-up aircraft short-period dynamics,

$$\underbrace{\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix}}_{\dot{x}_p} = \underbrace{\begin{pmatrix} \frac{Z_\alpha}{V_0} & 1 + \frac{Z_q}{V_0} \\ M_\alpha & M_q \end{pmatrix}}_{A_p} \underbrace{\begin{pmatrix} \alpha \\ q \end{pmatrix}}_{x_p} + \underbrace{\begin{pmatrix} \frac{Z_{\delta_e}}{V_0} \\ M_{\delta_e} \end{pmatrix}}_{B_p} \underbrace{\delta_e}_u$$

augmented with the integrated acceleration tracking error.

$$\dot{e}_{yI} = \underbrace{A_z}_{y_{\text{reg}}} - \underbrace{A_{z\text{cmd}}}_{y_{\text{cmd}}}$$

The corresponding extended dynamics are in the standard LTI form,

$$\boxed{\begin{aligned} \dot{e}_{yI} &= C_{p\text{reg}} x_p + D_{p\text{reg}} u - y_{\text{cmd}} \\ \dot{x}_p &= A_p x_p + B_p u \end{aligned}}$$

$$\Leftrightarrow \boxed{\begin{aligned} \underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} &= \underbrace{\begin{pmatrix} 0 & C_{p\text{reg}} \\ 0_{2 \times 1} & A_p \end{pmatrix}}_A \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} D_{p\text{reg}} \\ B_p \end{pmatrix}}_B u + \underbrace{\begin{pmatrix} -1 \\ 0_{2 \times 1} \end{pmatrix}}_{B_{\text{cmd}}} y_{\text{cmd}} \\ \Leftrightarrow \boxed{\dot{x} = A x + B u + B_{\text{cmd}} y_{\text{cmd}}} \end{aligned}}$$

where $x = (e_{yI} \ \alpha \ q)^T$ is the system state vector and $B_{\text{cmd}} = (-1 \ 0 \ 0)^T$ is the B -matrix due to external command $y_{\text{cmd}} = A_{z\text{cmd}}$. Output measurements for control design consist of the integrated tracking error e_{yI} and the aircraft body axis pitch rate q .

$$y_{\text{meas}} = \begin{pmatrix} e_{yI} \\ q \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{C_{\text{meas}}} \underbrace{\begin{pmatrix} e_{yI} \\ \alpha \\ q \end{pmatrix}}_x = C_{\text{meas}} x$$

So, the extended open-loop dynamics are tall, with one input and two outputs. It is straightforward to verify that the system is controllable, observable and has no unstable transmission zeros. Note that the original open-loop system has an unstable zero in the transfer function between the elevator input and the acceleration output.

We will perform the squaring-up process analytically. That allows to exploit and expose key features of the method. Our task at hand is to calculate a column

vector $B_2 \in R^{3 \times 1}$ such that the squared-up system has relative degree one and its transmission zero is in the open left-hand complex plane.

$$\text{Find } B_2 \in R^{3 \times 1} : \boxed{\overline{B} = (B, B_2)}$$

$$\Rightarrow \boxed{\det(C \overline{B}) \neq 0 \wedge \left(\det \begin{pmatrix} A - s I_3 & \overline{B} \\ C & 0 \end{pmatrix} = 0 \Rightarrow \boxed{s \in \mathbb{C}^-} \right)}$$

Specifically, we need to calculate $B_2 = (b_1 \ b_2 \ b_3)^T$ such that the system Rosenbrock matrix,

$$R_z(\lambda) = \begin{pmatrix} A - s I & \overline{B} \\ C_{\text{meas}} & 0 \end{pmatrix} = \begin{pmatrix} -s & Z_\alpha & 0 & Z_{\delta_e} & b_1 \\ 0 & \frac{Z_\alpha}{V_0} - s & 1 + \frac{Z_q}{V_0} & \frac{Z_{\delta_e}}{V_0} & b_2 \\ 0 & M_\alpha & M_q - s & M_{\delta_e} & b_3 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

drops rank only at the prescribed transmission zero $s_0 \in \mathbb{C}^-$. In this case, only a single transmission zero can be placed. Therefore, $s_0 < 0$ is a real number, not complex.

Exchanging 2nd and 3rd rows and columns in $S(\lambda)$ does not change the matrix rank.

$$\overline{R}_z(\lambda) = T_{2 \leftrightarrow 3} R_z(\lambda) T_{2 \leftrightarrow 3}^{-1} = \begin{pmatrix} s & 0 & Z_\alpha & Z_{\delta_e} & b_1 \\ 0 & M_q - s & M_\alpha & M_{\delta_e} & b_3 \\ 0 & 1 + \frac{Z_q}{V_0} & \frac{Z_\alpha}{V_0} - s & \frac{Z_{\delta_e}}{V_0} & b_2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

We chose $b_3 = 0$. Then

$$\overline{R}_z(s) = \begin{pmatrix} s & 0 & Z_\alpha & Z_{\delta_e} & b_1 \\ 0 & M_q - s & M_\alpha & M_{\delta_e} & 0 \\ 0 & 1 + \frac{Z_q}{V_0} & \frac{Z_\alpha}{V_0} - s & \frac{Z_{\delta_e}}{V_0} & b_2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$A_{z12} = \begin{pmatrix} Z_\alpha \\ M_\alpha \end{pmatrix}, \quad A_{z22} = \frac{Z_\alpha}{V_0}, \quad B_{z1} = \begin{pmatrix} Z_{\delta_e} & b_1 \\ M_{\delta_e} & 0 \end{pmatrix}, \quad B_{z2} = \begin{pmatrix} \frac{Z_{\delta_e}}{V_0} & b_2 \end{pmatrix}$$

and clearly, B_{z1} is non-singular. These matrices are defined based on (6.189).

Given the desired transmission zero $s_0 < 0$, we need to select b_1 and b_2 such that $s_0 = A_{z22} - B_{z2} B_{z1}^{-1} A_{z12}$.

$$\begin{aligned} s_0 &= A_{z22} - B_{z2} B_{z1}^{-1} A_{z12} = \frac{Z_\alpha}{V_0} - \left(\frac{Z_{\delta_e}}{V_0} b_2 \right) \begin{pmatrix} Z_{\delta_e} & b_1 \\ M_{\delta_e} & 0 \end{pmatrix}^{-1} \begin{pmatrix} Z_\alpha \\ M_\alpha \end{pmatrix} \\ &= \frac{Z_\alpha}{V_0} + \frac{1}{M_{\delta_e} b_1} \begin{pmatrix} Z_{\delta_e} & b_2 \end{pmatrix} \begin{pmatrix} 0 & -b_1 \\ -M_{\delta_e} & Z_{\delta_e} \end{pmatrix} \begin{pmatrix} Z_\alpha \\ M_\alpha \end{pmatrix} \\ &= \frac{Z_\alpha}{V_0} + \frac{1}{M_{\delta_e} b_1} \left(-M_{\delta_e} b_2 - \frac{Z_{\delta_e}}{V_0} b_1 + Z_{\delta_e} b_2 \right) \begin{pmatrix} Z_\alpha \\ M_\alpha \end{pmatrix} \\ &= \left(\frac{b_2}{b_1} - \frac{1}{V_0} \right) \left(\frac{M_\alpha Z_{\delta_e}}{M_{\delta_e}} - Z_\alpha \right) \end{aligned}$$

Consequently, b_1 and b_2 can be selected using a positive scalar $k > 0$,

$$\frac{b_2}{b_1} = \frac{1}{V_0} - k \operatorname{sgn} \left(\frac{M_\alpha Z_{\delta_e}}{M_{\delta_e}} - Z_\alpha \right), \quad \forall k > 0$$

and in that case, k is defined to enforce the desired transmission zero.

$$\boxed{s_0 = -k \left| \frac{M_\alpha Z_{\delta_e}}{M_{\delta_e}} - Z_\alpha \right| < 0} \Rightarrow \boxed{k = \frac{s_0}{\left| \frac{M_\alpha Z_{\delta_e}}{M_{\delta_e}} - Z_\alpha \right|}}$$

We can explore an alternative way to solving this particular squaring-up problem. Instead of zeroing out b_3 , let us compute a generic formula for achieving the desired transmission zero.

$$\begin{aligned} s_0 &= A_{z22} - B_{z2} B_{z1}^{-1} A_{z12} = \frac{Z_\alpha}{V_0} - \left(\frac{Z_{\delta_e}}{V_0} b_2 \right) \begin{pmatrix} Z_{\delta_e} & b_1 \\ M_{\delta_e} & b_3 \end{pmatrix}^{-1} \begin{pmatrix} Z_\alpha \\ M_\alpha \end{pmatrix} \\ &= \frac{Z_\alpha}{V_0} + \frac{1}{(Z_{\delta_e} b_3 - M_{\delta_e} b_1)} \begin{pmatrix} Z_{\delta_e} & b_2 \end{pmatrix} \begin{pmatrix} b_3 & -b_1 \\ -M_{\delta_e} & Z_{\delta_e} \end{pmatrix} \begin{pmatrix} Z_\alpha \\ M_\alpha \end{pmatrix} \\ &= \frac{Z_\alpha}{V_0} + \frac{1}{(Z_{\delta_e} b_3 - M_{\delta_e} b_1)} \left(\left(\frac{Z_{\delta_e}}{V_0} b_3 - b_2 M_{\delta_e} \right) Z_\alpha + \left(-\frac{Z_{\delta_e}}{V_0} b_1 + b_2 Z_{\delta_e} \right) M_\alpha \right) \\ &= \frac{Z_\alpha}{V_0} + \frac{1}{(Z_{\delta_e} b_3 - M_{\delta_e} b_1)} \left(\frac{Z_{\delta_e}}{V_0} (b_3 - b_1) + b_2 (Z_{\delta_e} M_\alpha - M_{\delta_e} Z_\alpha) \right) \end{aligned}$$

So,

$$s_0 = \frac{Z_\alpha}{V_0} + \frac{1}{(Z_{\delta_e} b_3 - M_{\delta_e} b_1)} \left(\frac{Z_{\delta_e}}{V_0} (b_3 - b_1) + b_2 (Z_{\delta_e} M_\alpha - M_{\delta_e} Z_\alpha) \right)$$

and we can select b_1 and b_3 .

$$\boxed{b_1 = Z_\alpha, \quad b_3 = M_\alpha} \Rightarrow \boxed{s_0 = \frac{Z_\alpha}{V_0} + \left[b_2 + \frac{Z_{\delta_e}(M_\alpha - Z_\alpha)}{V_0(Z_{\delta_e}M_\alpha - M_{\delta_e}Z_\alpha)} \right]}$$

The resulting transmission zero formula is parameterized with respect to b_2 .

In some control design applications, it might be beneficial to place the target transmission zero at the short-period stable zero location $\left(\frac{Z_\alpha}{V_0}\right)$. That can be accomplished by a proper selection of b_2 .

$$\boxed{b_2 = -\frac{Z_{\delta_e}(M_\alpha - Z_\alpha)}{V_0(Z_{\delta_e}M_\alpha - M_{\delta_e}Z_\alpha)}} \Rightarrow \boxed{s_0 = \frac{Z_\alpha}{V_0} < 0}$$

The resulting squaring-up solution is

$$b_1 = Z_\alpha, \quad b_2 = -\frac{Z_{\delta_e}(M_\alpha - Z_\alpha)}{V_0(Z_{\delta_e}M_\alpha - M_{\delta_e}Z_\alpha)}, \quad b_3 = M_\alpha$$

In this example, we were able to directly solve the squaring-up problem and correlate its solution to the aircraft short-period stability and control derivatives. For generic MIMO LTI tall systems, the squaring-up design should follow the steps from Table 6.1. ■

In the next sections, we shall exploit the squaring-up technique to enforce the asymptotic relation (6.166), which will subsequently facilitate the design of robust observer-based output feedback controllers with quantifiable performance and stability robustness guarantees.

6.3 Observer-Based Control with Loop Transfer Recovery

Suppose that the extended open-loop system (6.84) is driven by the observer-based output feedback control (6.92) and (6.93). By definition, the extended system is tall, controllable, and observable.

$$\begin{aligned} \underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} &= \underbrace{\begin{pmatrix} 0_{n_u \times n_u} & C_p \text{reg} \\ 0_{n_x \times n_u} & A_p \end{pmatrix}}_{\tilde{A}} \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} D_p \text{reg} \\ B_p \end{pmatrix}}_{\tilde{B}} u + \underbrace{\begin{pmatrix} -I_{n_u} \\ 0_{n_x \times n_u} \end{pmatrix}}_{B_{\text{cmd}}} y_{\text{cmd}} \\ y_{\text{meas}} &= \underbrace{\begin{pmatrix} I_{n_u} & 0_{n_u \times n_x} \\ 0_{n_u \times n_u} & C_p \text{meas} \end{pmatrix}}_{C_{\text{meas}}} \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x \end{aligned} \tag{6.195}$$

Assuming that there are no finite unstable transmission zeros in the tall system

$$y_{\text{meas}} = C_{\text{meas}} \left(s I_{n_{\text{ext}}} - \tilde{A} \right)^{-1} \tilde{B} u \quad (6.196)$$

the squaring-up method [14] from the previous section can be applied to assign the system transmission zeros at the desired locations in \mathbb{C}^- . The squaring-up is accomplished by adding pseudo-control inputs $B_2 \in R^{n_{\text{ext}} \times (n_y - n_u)}$.

$$\boxed{\overline{B} = (\tilde{B}, B_2) \Rightarrow \det(C_{\text{meas}} \overline{B}) \neq 0 \wedge \left(\det \begin{pmatrix} s I_{n_{\text{ext}}} - \tilde{A} & \overline{B} \\ C_{\text{meas}} & 0 \end{pmatrix} = 0 \Rightarrow \boxed{s \in \mathbb{C}^-} \right)} \quad (6.197)$$

In this case, the observer gain

$$L_v = P_v C_{\text{meas}}^T R_v^{-1} \quad (6.198)$$

depends on the unique symmetric positive-definite solution P_v of the ARE,

$$P_v \tilde{A}^T + \tilde{A} P_v + Q_v - P_v C_{\text{meas}}^T R_v^{-1} C_{\text{meas}} P_v = 0 \quad (6.199)$$

with a positive constant parameter v , the weight matrices,

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) \overline{B} \overline{B}^T, \quad R_v = \frac{v}{v+1} R_0 \quad (6.200)$$

and the squared-up matrix \overline{B} from (6.197). At the same time, the RSLQR state feedback controller gain K_{ext} is given by (6.87) and (6.88).

With the controller and the observer gains (K_{ext}, L_v) designed using the corresponding ARE-s, the closed-loop system dynamics become globally exponentially stable. This fact is due to the Separation Principle discussed in Sect. 6.3.

Equations (6.92), (6.93), (6.198)–(6.200) represent the observer-based loop transfer recovery (OBLTR) control system. Its name signifies certain loop transfer recovery capabilities that will be derived and analyzed in detail.

Of interest now is the closed-loop system relative stability. Specifically, we are going to analyze MIMO gain and phase margins via the system loop gain computed at the plant-input loop break point as shown in Fig. 6.13. We start with the state observer dynamics (6.94),

$$\dot{\hat{x}} = \underbrace{\left(\tilde{A} - \tilde{B} K_{\text{ext}} - L_v C_{\text{meas}} \right)}_{A_{\text{obs}}} \hat{x} + B_{\text{cmd}} y_{\text{cmd}} + L_v y_{\text{meas}} \quad (6.201)$$

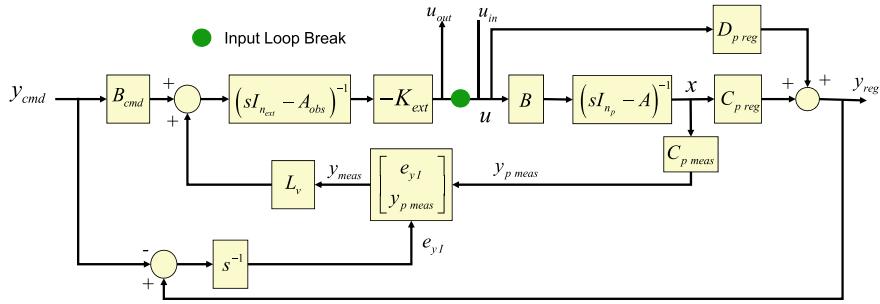


Fig. 6.13 Input loop gain calculation block diagram

From Fig. 6.13, the loop gain transfer function from the injected signal u_{in} to the resultant controller output u_{out} , can be derived as

$$\begin{aligned} u_{out} &= -K_{ext} \hat{x} = -K_{ext} (s I_{n_{ext}} - A_{obs})^{-1} L_v C_{meas} x \\ &= \underbrace{-K_{ext} (s I_{n_{ext}} - \tilde{A} + \tilde{B} K_{ext} + L_v C_{meas})^{-1} L_v C_{meas} (s I_{n_{ext}} - \tilde{A})^{-1} u_{in}}_{\text{Loop Gain @ Input Break Point: } L_u(s)} \end{aligned} \quad (6.202)$$

So, the loop gain L_u at the plant-input loop break point is

$$L_u(s) = K_{ext} (s I_{n_{ext}} - \tilde{A} + \tilde{B} K_{ext} + L_v C_{meas})^{-1} L_v C_{meas} (s I_{n_{ext}} - \tilde{A})^{-1} \tilde{B} \quad (6.203)$$

In what follows, we shall prove loop transfer recovery (LTR) properties of the OBLTR dynamic compensator. Specifically we show that as $v \rightarrow 0$ the system loop gain (6.203) asymptotically approaches that of the LQR state feedback loop gain,

$$L_{u\ lqr}(s) = R^{-1} \tilde{B}^T P (s I_{n_{ext}} - \tilde{A})^{-1} \tilde{B} = K_{ext} (s I_{n_{ext}} - \tilde{A})^{-1} \tilde{B} \quad (6.204)$$

point-wise in s , with the LQR optimal gain matrix K_{ext} from (6.87) and (6.88). Note that the square open-loop extended dynamics

$$G(s) = C_{meas} (s I_{n_{ext}} - A)^{-1} \bar{B} \quad (6.205)$$

from the system inputs (including the pseudo-input) to the measured output are minimum phase by construction via the squaring-up method of Sect. 6.2.4. This

particular property is required to ensure existence and validity of the asymptotic expansions discussed in Sect. 6.2.3.

Based on (6.198), the observer gain asymptotics are,

$$L_v = P_v C_{\text{meas}}^T R_v^{-1} = \left(1 + \frac{1}{v}\right) \underbrace{P_v C_{\text{meas}}^T R_0^{-1}}_{\overline{B} W^T R_0^{-\frac{1}{2}} + O(v)} = \left(1 + \frac{1}{v}\right) \left(\overline{B} W^T R_0^{-\frac{1}{2}} + O(v)\right) \quad (6.206)$$

where the unitary matrix

$$W = V^T U^T \quad (6.207)$$

is defined through the singular value decomposition (SVD) of

$$R_0^{-\frac{1}{2}} C B = V^T \Lambda U^T \quad (6.208)$$

Using (6.206), further gives

$$\lim_{v \rightarrow 0} (v L_v) = \lim_{v \rightarrow 0} \left((v + 1) \overline{B} W^T R_0^{-\frac{1}{2}} + O(v) \right) = \overline{B} W^T R_0^{-\frac{1}{2}} \quad (6.209)$$

It is easy to see that the identities

$$(I + X Y) X = X (I + Y X) \Leftrightarrow X(I + Y X)^{-1} = (I + X Y)^{-1} X \quad (6.210)$$

are valid for any matrices X and Y of the corresponding dimensions and assuming that matrix inverses exist.

Rewrite the loop gain (6.203),

$$\begin{aligned} L_u(s) &= K_{\text{ext}} \left(\underbrace{\left(s I_{n_{\text{ext}}} - A + B K_{\text{ext}} \right)}_{\Psi^{-1}(s)} + L_v C_{\text{meas}} \right)^{-1} L_v C_{\text{meas}} \underbrace{\left(s I_{n_{\text{ext}}} - A \right)^{-1} B}_{\Phi(s)} \\ &= K_{\text{ext}} (\Psi^{-1}(s) + L_v C_{\text{meas}})^{-1} L_v C_{\text{meas}} \Phi(s) B \\ &= K_{\text{ext}} (I_{n_{\text{ext}}} + \Psi(s) L_v C_{\text{meas}})^{-1} \Psi(s) L_v C_{\text{meas}} \Phi(s) B \end{aligned} \quad (6.211)$$

and use (6.210) to assert

$$\left(I_{n_{\text{ext}}} + \underbrace{\Psi L_v}_{X} \underbrace{C_{\text{meas}}}_{Y} \right)^{-1} \underbrace{\Psi L_v}_{X} = \underbrace{\Psi L_v}_{X} \left(I_{n_y} + \underbrace{C_{\text{meas}} \Psi L_v}_{Y} \right)^{-1} \quad (6.212)$$

Then (6.211) becomes

$$\begin{aligned} L_u(s) &= K_{\text{ext}} \Psi(s) L_v (I_{n_y} + C_{\text{meas}} \Psi(s) L_v)^{-1} C_{\text{meas}} \Phi(s) B \\ &= K_{\text{ext}} \Psi(s) (v I_{n_y} + C_{\text{meas}} \Psi(s) (v L_v))^{-1} C_{\text{meas}} \Phi(s) B \end{aligned} \quad (6.213)$$

Because of (6.209), the loop gain limit as $v \rightarrow 0$ can be computed point-wise in s .

$$\begin{aligned} L_u(s) &\xrightarrow[v \rightarrow 0]{} K_{\text{ext}} \Psi(s) \bar{B} W^T R_0^{-\frac{1}{2}} \left(C_{\text{meas}} \Psi(s) \bar{B} W^T R_0^{-\frac{1}{2}} \right)^{-1} C_{\text{meas}} \Phi(s) B \\ &= K_{\text{ext}} \Psi(s) \bar{B} (C_{\text{meas}} \Psi(s) \bar{B})^{-1} C_{\text{meas}} \Phi(s) B \end{aligned} \quad (6.214)$$

By the definitions from (6.211),

$$\left. \begin{aligned} \Psi^{-1}(s) &= (s I_{n_{\text{ext}}} - A + B K_{\text{ext}}) \\ \Phi(s) &= (s I_{n_{\text{ext}}} - A)^{-1} \end{aligned} \right\} \Rightarrow \Psi(s) = \Phi(s) (I_{n_{\text{ext}}} + B K_{\text{ext}} \Phi(s))^{-1} \quad (6.215)$$

and consequently

$$\begin{aligned} L_u(s) &\xrightarrow[v \rightarrow 0]{} K_{\text{ext}} \Phi (I_{n_{\text{ext}}} + B K_{\text{ext}} \Phi)^{-1} \bar{B} \\ &\quad \times \left(C_{\text{meas}} \Phi (I_{n_{\text{ext}}} + \tilde{B} K_{\text{ext}} \Phi)^{-1} \bar{B} \right)^{-1} C_{\text{meas}} \Phi \tilde{B} \end{aligned} \quad (6.216)$$

Introduce,

$$\bar{K} = \begin{pmatrix} K_{\text{ext}} \\ 0_{(n_y - n_u) \times n_{\text{ext}}} \end{pmatrix} \quad (6.217)$$

and immediately note that

$$\bar{B} \bar{K} = (\tilde{B} B_2) \begin{pmatrix} K_{\text{ext}} \\ 0_{(n_y - n_u) \times n_{\text{ext}}} \end{pmatrix} = \tilde{B} K_{\text{ext}} \quad (6.218)$$

So (6.216) becomes

$$L_u(s) \xrightarrow[v \rightarrow 0]{} K_{\text{ext}} \Phi (I_{n_{\text{ext}}} + \bar{B} \bar{K} \Phi)^{-1} \bar{B} (C_{\text{meas}} \Phi (I_{n_{\text{ext}}} + \bar{B} \bar{K} \Phi)^{-1} \bar{B})^{-1} C_{\text{meas}} \Phi \tilde{B} \quad (6.219)$$

Using (6.210) once again, further gives

$$\left(I_{n_{\text{ext}}} + \underbrace{\overline{B}}_X \underbrace{\overline{K} \Phi(s)}_Y \right)^{-1} \underbrace{\overline{B}}_X = \underbrace{\overline{B}}_X \left(I_{n_u} + \underbrace{\overline{K} \Phi(s)}_Y \underbrace{\overline{B}}_X \right)^{-1} \quad (6.220)$$

and in this case, the limiting relation (6.216) takes the form

$$\begin{aligned} L_u(s) &\xrightarrow[v \rightarrow 0]{} K_{\text{ext}} \Phi \overline{B} (I_{n_u} + \overline{K} \Phi \overline{B})^{-1} \left(C_{\text{meas}} \Phi \overline{B} (I_{n_u} + \overline{K} \Phi \overline{B})^{-1} \right)^{-1} C_{\text{meas}} \Phi \tilde{B} \\ &= K_{\text{ext}} \Phi(s) \overline{B} (C_{\text{meas}} \Phi(s) \overline{B})^{-1} C_{\text{meas}} \Phi(s) \tilde{B} \\ &= K_{\text{ext}} \Phi(s) \overline{B} (C_{\text{meas}} \Phi(s) \overline{B})^{-1} (C_{\text{meas}} \Phi(s) \overline{B}) \begin{pmatrix} I_{n_u} \\ 0_{(n_y - n_u) \times n_u} \end{pmatrix} \\ &= K_{\text{ext}} \Phi(s) \overline{B} \begin{pmatrix} I_{n_u} \\ 0_{(n_y - n_u) \times n_u} \end{pmatrix} = \boxed{K_{\text{ext}} \Phi(s) \tilde{B}} \end{aligned} \quad (6.221)$$

Existence of the left-hand side limiting expression in (6.221) is predicated on the non-singularity of the open-loop transfer function matrix $(C_{\text{meas}} \Phi(s) \overline{B})^{-1}$. From the margins analysis point of view, it is sufficient to require that this matrix is non-singular in the closed right-half complex plane $\overline{\mathbb{C}}^+$. The squaring-up method ensures just that. The limiting expression in (6.221) gives the desired asymptotics,

$$L_u(s) \xrightarrow[v \rightarrow 0]{} K_{\text{ext}} \left(s I_{n_{\text{ext}}} - \tilde{A} \right)^{-1} \tilde{B} = L_{u \text{lqr}}(s) \quad (6.222)$$

point-wise in $s \in \mathbb{C}^+$, where $L_{u \text{lqr}}(s)$ is the input loop gain of the LQR PI optimal state feedback controller whose gain is given in (6.88).

The asymptotics (6.222) represents the loop transfer recovery (LTR) property of the OBLTR controller (6.198), (6.199), and (6.201). This is summarized in the theorem below.

Theorem 6.5 Consider the tall controllable observable LTI MIMO system (6.84) and suppose that there are no finite unstable transmission zeros in the tall open-loop dynamics (6.196). Also consider the full-order state observer (6.201), whose optimal gain is defined by the parameter-dependent ARE (6.199). Let the ARE weights be selected as in (6.200), by means of the squaring-up procedure (6.197). Then

- (a) The system (6.84) with the observer-based dynamic output feedback control $u = -K_{\text{ext}} \hat{x}$ is globally exponentially stable.
- (b) The regulated output tracks constant commands and rejects constant disturbances with zero steady-state errors.
- (c) The LTR relation (6.222) takes place, that is the system input loop gain $L_u(s)$ (6.203) asymptotically approaches the optimal LQR state feedback loop gain $L_{u \text{lqr}}(s)$, as $v \rightarrow 0$, point-wise in $s \in \mathbb{C}^+$. ■

Table 6.2 Observer-based control with loop transfer recovery (OBLTR)

Open-loop extended system with output measurements	$\dot{x} = \tilde{A}x + \tilde{B}u + B_{\text{cmd}}y_{\text{cmd}}$, (6.195) $y_{\text{meas}} = C_{\text{meas}}x$
Squaring-up	$\bar{B} = \begin{pmatrix} \tilde{B} & B_2 \end{pmatrix} \Rightarrow$ $\det(C_{\text{meas}}\bar{B}) \neq 0$ $\text{zeros}\left[C_{\text{meas}}\left(sI_{n_{\text{ext}}} - \tilde{A}\right)^{-1}\bar{B}\right] \in \mathbb{C}^-$
ARE weights	$Q_v = Q_0 + \left(\frac{v+1}{v}\right)\bar{B}\bar{B}^T, \quad R_v = \frac{v}{v+1}R_0$
Observer ARE	$P_v\tilde{A}^T + \tilde{A}P_v + Q_v - P_vC_{\text{meas}}^TR_v^{-1}C_{\text{meas}}P_v = 0$
Observer gain	$L_v = P_vC^TR_v^{-1}$
Full-order state observer	$\dot{\hat{x}} = \tilde{A}\hat{x} + \tilde{B}u + L_v(y - \hat{y}) + B_{\text{cmd}}y_{\text{cmd}}, \quad \hat{y} = C\hat{x}$
Control feedback	$u = -K_{\text{ext}}\hat{x}$

The LTR property (6.222) provides theoretical foundations to design OBLTR servomechanism controllers to guarantee stable closed-loop tracking performance and also recover the LQR optimal state feedback MIMO gain and phase margins.

A synopsis of the OBLTR design is given in Table 6.2.

In essence, achieving closed-loop stability and tracking performance with guaranteed margins while using measured outputs only are the main features of the presented OBLTR design.

6.3.1 OBLTR Design Process and Examples

Aerospace control designs have both system-level time domain performance requirements and frequency domain robustness requirements that must be met for safe, reliable system operation. Time domain performance is typically measured by how fast the system responds to a command, and how accurately the command is achieved. Also, how fast do the actuators move/respond to the command. Are rate limits in actuation achieved? Linear and nonlinear simulations are used to evaluate the design and gather time domain metrics. Frequency domain robustness is typically measured through stability margins and loop attenuation at high frequencies. Several transfer functions (as discussed in Chap. 3) are examined to evaluate the adequacy of the design in the frequency domain and to form frequency domain metrics.

Many of today's vehicle design problems have flight condition dependent models that are open-loop unstable, and when using acceleration feedback, are also nonminimum phase. The challenge of controlling a system with both right half plane (RHP) poles and zeros in its transfer function is one of the most challenging control design problems. For open-loop unstable systems, feedback gains that are large in magnitude are usually needed to move RHP poles into the open left half

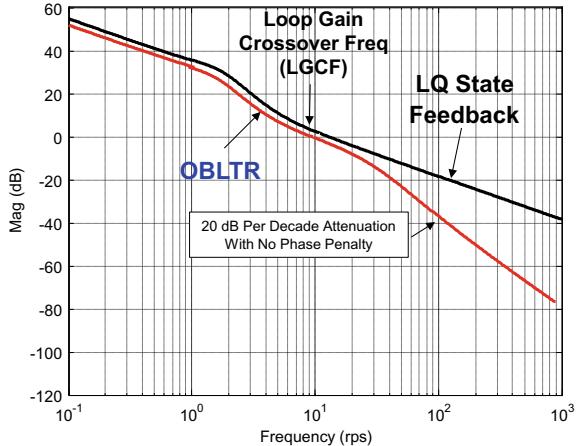
plane (LHP) in order to stabilize the closed-loop system and achieve the desired performance. But large gains are very difficult to implement in practice. Any control system design must be balanced by examining both time domain performance and frequency domain robustness. The design must meet performance requirements, actuation system limitations, stability margins requirements, sensor noise attenuation requirements, and ensure loop gain attenuation at high frequencies for structural mode interaction (SMI) concerns. System resonances must be examined and prevented. Historically, low pass and notch filtering are added to the control system to attenuate noise and prevent SMI. These (any) filters can add significant phase lag, and when combined with a design that has large feedback gains, can create challenges in trying to meet stability margin (gain and phase margin) requirements due to the phase lag. Several iterations in the gain design process would usually be needed as filters and other subsystems are added to balance the system performance and stability robustness.

OBLTR control theory has some unique features that can help balance the challenge between achieving time domain performance and frequency domain robustness. When the observer design is tuned properly using LTR, OBLTR can provide an additional 20 dB per decade of filtering, with no to minimal phase penalty (less than adding a low pass filter). OBLTR can be used on systems that have either full state measurements or just output measurements for feedback. When the states are available as measurements, ($n_y = n_x$), the observer filters the states providing the additional 20 dB per decade of filtering. Thus, the control design engineer can exploit this property to meet the system requirements (noise and SMI attenuation) without adding as much additional filtering. When a reduced number of output signals are available for feedback, ($n_y < n_x$), the observer is required to implement the RSLQR state feedback gain solution. In that case, the same process for tuning the observer with LTR is followed, and the OBLTR control law still provides the desired 20 dB per decade of filtering roll-off with minimal phase penalty.

An observer design that is tuned properly using LTR will recover the RSLQR loop gain up to and including where the loop gain crosses 0 dB. Then the observer rolls off providing the additional attenuation. Figure 6.14 illustrates this property in the Bode magnitude plot. There will be some phase lag due to the observer. The amount of phase lag will depend upon the frequency separation between the RSLQR eigenvalues $\lambda(\tilde{A} - \tilde{B}K_{\text{ext}})$ and the observer eigenvalues $\lambda(\tilde{A} - L_v C_{\text{meas}})$.

The mathematics discussed in the previous section on asymptotics and the squaring-up procedure may be “intimidating” for the control practitioner. Our intent here is to emphasize the importance of using control system design methods that have a solid theoretical foundation, and this text and references cited within provide just that. The discussion and examples in this section will demonstrate the OBLTR control design process that can be used for a baseline control algorithm. They will show how system time domain performance requirements are achieved in the tuning of the RSLQR, and how the observer parameters are selected and

Fig. 6.14 Bode magnitude plot comparing LQR and OBLTR feedback



tuned to achieve a robust dynamic output control solution. One can then envision how this process can be automated in design scripts and used to reduce touch labor in the overall control system design for a system that is engineered to function in a wide operational envelope.

In later chapters, we extend model reference adaptive control (MRAC) algorithms and connect them to the OBLTR control law. This link is one of the main advancements in control theory captured within this text, connecting robust and adaptive control designs. As we show in Chaps. 13 and 14, in OBLTR MRAC controlled systems, the observer serves as a closed-loop reference model within the adaptive algorithm and reduces transients during parameter adaptation.

OBLTR Design Process

Equation (6.94) with Fig. 6.12 shows the OBLTR servomechanism control law. The implementation is

$$\begin{aligned}\dot{\hat{x}} &= A_{\text{obs}}\hat{x} + B_{\text{cmd}}y_{\text{cmd}} + L_v y_{\text{meas}} \\ u &= -K_{\text{ext}}\hat{x}\end{aligned}\quad (6.223)$$

Table 6.3 outlines a design process for forming the necessary models and designing the OBLTR control law gains. The first two steps form the models, one for design, and the other for analysis. This is followed by the RSLQR design, then the observer development. The table lists the parameters to be defined in each step to complete the design.

Step 1: Form Linear Model for Gain Design and Analysis The OBLTR gain design model uses the extended state RSLQR design model given as

$$\underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{\hat{x}}_p \end{pmatrix}}_{\dot{\hat{x}}} = \underbrace{\begin{pmatrix} 0_{n_u \times n_u} & C_p \text{ reg} \\ 0_{n_x \times n_u} & A_p \end{pmatrix}}_{\tilde{A}} \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} D_p \text{ reg} \\ B_p \end{pmatrix}}_{\tilde{B}} u + \underbrace{\begin{pmatrix} -I_{n_u} \\ 0_{n_x \times n_u} \end{pmatrix}}_{B_{\text{cmd}}} y_{\text{cmd}}$$

Table 6.3 Design steps for OBLTR servomechanism design

Design steps	Parameters to be defined	Product
1. Form the servomechanism linear model (\tilde{A}, \tilde{B}) for gain design, define regulated variable and measurements	$x_p, y_p \text{ meas}, y_{\text{reg}}$	Form the low-order state-space plant model for RSLQR gain design, form the measurement vector $y_p \text{ meas}$, and the command variables y_{reg} to track command y_{cmd}
2. Form the linear plant model for time domain simulation and frequency domain analysis (A, B, C, D)	x, y	This model includes all vehicle subsystem models, any time delays in the system, and any filtering on measurement and command signals. Everything and anything that contributes phase lag or impacts stability margins
3. RSLQR gain design and analysis	$\tilde{A}, \tilde{B}, Q, R$	Compute K_{ext} using the ARE. Evaluate the gains in the time domain to determine if performance requirements are met. Examine the design in the frequency domain to make sure the LGCF is not too high and stability margins are adequate
4. Form the measurements for the observer. Observer gain design and analysis	$y_{\text{meas}}, Q_0, R_0, \bar{B}, v$	y_{meas}, L_v

$$\begin{aligned}
 y_{\text{reg}} &= \underbrace{\left(0_{n_u \times n_u} \quad C_{p \text{ reg}}\right)}_{C_{\text{reg}}} \underbrace{\begin{pmatrix} e_y I \\ x_p \end{pmatrix}}_x + \underbrace{D_{p \text{ reg}} u}_{D_{\text{reg}}} \\
 y_{\text{meas}} &= \underbrace{\left(I_{n_u} \quad 0_{n_u \times n_x} \atop 0_{n_u \times n_u} \quad C_{p \text{ meas}}\right)}_{C_{\text{meas}}} \underbrace{\begin{pmatrix} e_y I \\ x_p \end{pmatrix}}_x
 \end{aligned} \tag{6.224}$$

incorporating integral control on the regulated output tracking error $e_y I = y_{\text{reg}} - y_{\text{cmd}}$. This will produce a RSLQR design in which the regulated output y_{reg} track the command vector y_{cmd} . The RSLQR gain design model state-space pair (\tilde{A}, \tilde{B}) in (6.224) is intended to be a low-order model description of the system dynamics. Using this approach, a simple, low-order robust PI output feedback controller can be constructed, which reduces the number of output measurements needed for its implementation. The definition of $y_{\text{meas}} = [e_y I \ y_p \text{ meas}]^T$ and the matrix C_{meas} in (6.224) will be discussed within the observer design step.

In the design process outlined in Table 6.3, the RSLQR controller is designed before the observer, assuming a state feedback architecture. We know from the

Separation Principle that the eigenvalues $\lambda(\tilde{A} - \tilde{B}K_{\text{ext}})$ will describe the closed-loop system response to a command. Once the RSLQR gain matrix K_{ext} is designed to meet requirements, a state observer can then be designed to implement the control using output feedback signals.

Step 2: Form Linear Model for Time Domain and Frequency Domain Analysis In the OBLTR design process, the RSLQR and observer gain matrices are designed with the low-order design model defined in Step 1. In the subsequent gain tuning for both RSLQR and the observer (Steps 3 and 4), the gains are evaluated in both the time domain and the frequency domain using an analysis model that includes all relevant and known subsystem models for the vehicle. This modeling for analysis is very important.

In the RSLQR gain tuning process (Step 3) the gains are increased in magnitude to improve the system speed of response. As the gains get larger, the loop gain-crossover frequency (LGCF) ω_c increases, and when evaluated using this analysis model it may have degraded stability margins, even though the RSLQR design model will show infinite gain margin and at least 60 deg phase margin. Any subsystem that contributes gain change or a phase lag needs to be included in evaluating any particular choice of control gains. Doing this model-based design, will produce a control system capable of controlling the actual vehicle, with the model-predicted time domain performance and frequency domain robustness characteristics.

In Sect. 2.4.2, we introduced control-oriented models for LTI MIMO systems. Following this framework, a state-space model for both plant and controller can be formed explicitly. The open-loop plant model is

$$\begin{aligned}\dot{x}_p &= A_p x_p + B_p u \\ y &= C_p x_p + D_p u\end{aligned}\tag{6.225}$$

and the controller model is

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} y_{\text{cmd}} \\ u &= C_c x_c + D_{c1} y + D_{c2} y_{\text{cmd}}\end{aligned}\tag{6.226}$$

The plant model in (6.225) can be of the same order as the design model in Step 1, or it can include all relevant subsystem models for the vehicle, as we discussed above. The more subsystem models that are included the more accurate the stability margin analysis will be. The controller model in this step will contain only the RSLQR state feedback controller. In Step 4, it will include the full OBLTR control law into the analysis process.

These two systems are connected to form all relevant time domain simulation and frequency domain analysis models used at each step of the design process. This will include a closed-loop system model,

$$\begin{aligned}\dot{x} &= A_{cl} x + B_{cl} y_{\text{cmd}} \\ y &= C_{cl} x + D_{cl} y_{\text{cmd}}\end{aligned}\tag{6.227}$$

an open-loop models at plant-input loop break point to form the loop gain at input L_u , and at the plant-output loop break to form the loop gain at output L_y . Subsequently, both transfer functions will be used for frequency domain analysis.

Forming accurate analysis models can be challenging. Often, the installed performance of subsystems on a vehicle differs from procurement specifications and/or linear models used to describe them. The real hardware once assembled may exhibit more phase lag than the analysis model used in the linear analysis. As our knowledge of these models matures, the gains need to be re-evaluated with the best models available. If they fail to meet requirements then the gain design step in this process needs to be repeated.

Aircraft subsystems can also be nonlinear, whose behavior is only locally captured by a linear analysis model. These nonlinearities need to be modeled in a high-fidelity nonlinear simulation environment so that the best-data-informed analysis of the system stability, performance, and robustness can be predicted. This is an important step in the validation of any control system design. We refer interested readers to Chap. 1 for a discussion on system simulation capabilities that are required to support control design, analysis, verification, and validation processes.

Step 3: RSLQR Gain Design and Analysis In this step, the extended plant design model of (6.224) is used to solve for the LQR state feedback gains K_{ext} . The gains are formed using the RSLQR ARE

$$P \tilde{A} + \tilde{A}^T P - P \tilde{B} R^{-1} \tilde{B}^T P + Q = 0, \quad K_{\text{ext}} = R^{-1} \tilde{B} P, \quad u = -K_{\text{ext}} x \quad (6.228)$$

with $Q = Q^T \geq 0$, $R = R^T > 0$, and (\tilde{A}, \sqrt{Q}) is observable pair. This last requirement makes sure that any unstable modes in the system are penalized in the cost index which guarantees that the resulting feedback control law stabilizes the system. Figure 6.15 illustrates the RSLQR gain tuning process. The RSLQR design model is used to form the gains, and then the gains are analyzed in the time domain and then the frequency domain using the analysis model. The process is as follows:

$$\begin{aligned} (\tilde{A}, \tilde{B}, Q, R) &\rightarrow K_{\text{ext}} \rightarrow \text{Time Domain} \\ &\rightarrow L_u(s), \omega_c, \text{Frequency Domain Robustness} \end{aligned} \quad (6.229)$$

The gain tuning process typically starts with making $R = I_{n_u}$, and numerically increasing the weights on the states until a design constraint is reached. As discussed in Sect. 4.4 (design chart process), a faster response can be obtained by numerically increasing the elements of Q that penalize the command error states. Larger weights produce larger gains, $\|K_{\text{ext}}\|_2 \sim \|Q\|_2/\|R\|_2$. Larger gains will move the closed-loop poles further into the LHP, and will increase the loop gain-crossover frequency (LGCF), ω_c . Unfortunately, as the LGCF increases the control system will drive the actuators with high rates and stability margins will

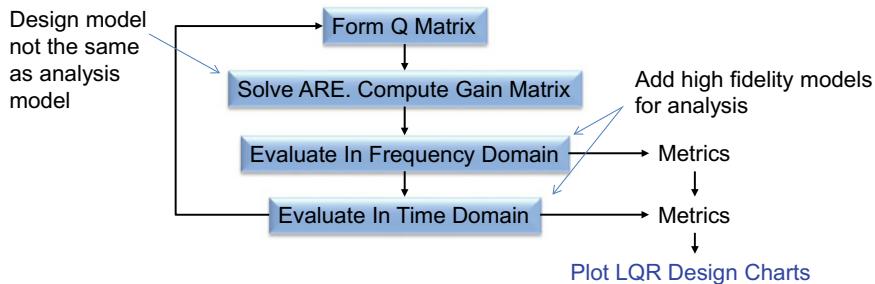


Fig. 6.15 RSLQR design process

suffer. High LGCFs also cause challenges with accommodating sensor data filtering that might be needed to address sensor noise and attenuation of the flexible body dynamics. Thus, challenges in real-world implementation of the control system will be minimized by engineering the lowest LGCF that meets time domain requirements. This should serve as a guiding principle in developing any control system.

In control system design environments like MATLAB[®], a script can easily be written to execute the gain design process illustrated in Fig. 6.15. Logic can be introduced to exit the Q -matrix tuning loop when either time domain or frequency domain requirements are satisfied in order to keep the LGCF appropriate.

In some control applications like piloted aircraft, a fast response to the command is a top requirement. Using the RSLQR in (6.229) which uses integral error control may not provide the ideal solution. In the standard servomechanism architecture, the command error is integrated before control action is obtained. This lag may be unacceptable. For applications where the time response is critical, a feed-forward control augmentation can be added that can significantly reduce the system reaction time while keeping the LGCF low. As already noted, a low frequency LGCF makes it much easier to add filtering on feedback signals to accommodate sensor noise and attenuation of the flexible body dynamics.

In control applications that are sometimes driven by sensors, the command path may contain sensor noise itself. In this type of application, a feedforward compensation would pass the sensor noise directly into the control signal, which would not be desired. In these applications, the servomechanism integrator in (6.229) smooths the noise and can provide the fastest overall response.

Step 4: Observer Gain Design and Analysis In this step we define the measurements to be used by the observer and use Eqs. (6.198)–(6.200) to compute the observer gain matrix L_v . There are several important system-level metrics to consider when designing the observer. They are:

- System stability margins and frequency domain metrics as used in the RSLQR design (see Chap. 4, Example 4.10).
- Observer eigenvalue separation from the RSLQR eigenvalues.

- Analysis of the sensor noise-to-control and control rate transfer function matrix.
- Adaptive angle κ_i .

The Separation Principle in observer-based architectures, (6.74), shows us that the time domain performance of the closed-loop system under OBLTR control is determined by the RSLQR eigenvalue locations. However, the location of the observer eigenvalues can influence transient responses and degrade stability margins. The key attribute in observer design is to create *frequency separation* between the state feedback design and the observer design. The RSLQR eigenvalues are the zeros of

$$\phi_{lqr}(s) = \det(sI_n - \tilde{A} + \tilde{B}K_{\text{ext}}) \quad (6.230)$$

The observer eigenvalues are the zeros of

$$\phi_{\text{obs}}(s) = \det(sI_n - \tilde{A} + L_vC) \quad (6.231)$$

and need to be located sufficiently away (in the LHP) from the RSLQR poles. In some designs this may be difficult to achieve.

From Table 6.3 the parameters needed in the observer ARE to compute the gains L_v are (Q_0, R_0, \bar{B}, v) . A short discussion on how to define these parameters follows.

(Q_0, R_0) Matrices: The matrices $Q_0 \in R^{n_{\text{ext}} \times n_{\text{ext}}}$ and $R_0 \in R^{n_y \times n_y}$ from (6.151) in a Kalman filter design problem represent the covariances of the plant disturbance and measurement noise, respectively. When using LTR ($v \rightarrow 0$) the numerical values used in (Q_0, R_0) are not critical except that the observer needs to have eigenvalues faster than the RSLQR (for all values of v). Simple methods like Bryson's rule [17] can be used to set the initial values, or, in some problems we have found it adequate to just set $Q_0 = I_{n_{\text{ext}}}$ and $R_0 = I_{n_y}$ and adjust specific elements of (Q_0, R_0) to produce a fast observer. In many high-speed design conditions, the control power stability derivatives in \bar{B} can become quite large (these stability derivatives are scaled by dynamic pressure). Thus, in (6.151), the matrix $(\frac{v+1}{v}) \bar{B} \bar{B}^T$ can numerically dominate Q_0 . This can influence the results when tuning LTR ($v \rightarrow 0$).

The observer, by design, will have a loop gain $L_u(s)$ that produces an additional -20 dB per decade of attenuation (filtering) as compared to the RSLQR, $L_{u \text{lqr}}(s)$. This additional filtering can be exploited to aid in attenuating sensor noise and flexible body dynamics. The phase lag that results from this filtering is why observers degrade stability margins, and is why LTR is needed. In the process of applying LTR, (6.222), the gains within the observer gain matrix L_v will get large in magnitude and $L_u(s) \rightarrow_{v \rightarrow 0} L_{u \text{lqr}}(s)$. It is desirable when applying LTR to only recover $L_{u \text{lqr}}(s)$ up to the LGCF ω_g . This will recover/improve stability margins and then preserve the natural loop gain roll-off (attenuation) of the

observer at higher frequencies. This property will be evident in the Bode magnitude analysis of $L_u(s)$. Since $L_v \rightarrow \infty$ as $v \rightarrow 0$, the sensor noise-to-control and sensor noise-to-control rate transfer functions magnitudes should be examined to determine if additional filtering is needed in the implementation of the control system.

\bar{B} Matrix: The \bar{B} matrix is formed using the squaring-up method of Sect. 6.2.4. Select columns B_2 in (6.173) to make $\bar{B} = [B \ B_2]$ to have column rank equal to the row rank of C and to make the extended system minimum phase. Use (6.193) to compute B_2 . Placement of transmission zeros can be performed using pole placement design or via LQR design. When using a pole placement algorithm, specific locations of the transmission zeros are obtained. When using LQR, A_{z22cl} becomes Hurwitz, and the transmission zeros do not have to be explicitly specified.

LTR as $v \rightarrow 0$: LTR uses the matrices (Q_v, R_v) from (6.200) in the observer ARE (6.199) to compute the observer gain matrix L_v . As $v \rightarrow 0$, the eigenvalues of the observer will move further into the LHP providing separation from the control eigenvalues. However, as $v \rightarrow 0$, elements of $L_v \rightarrow \infty$, and sensor noise will be amplified by the observer. Also, as $v \rightarrow 0$, the matrix Q_0 is dominated by $(\frac{v+1}{v}) \bar{B} \bar{B}^T$. The relative size between the matrices (Q_v, R_v) is important as is the relative size between the elements internal to each matrix. Both influence the eigenvalues movement as $v \rightarrow 0$, and both can be adjusted/tuned to achieve a satisfactory loop gain L_u ,

$$L_u(s) = K_{\text{ext}} \left(sI_n - \tilde{A} + \tilde{B}K_{\text{ext}} - L_v C_{\text{meas}} \right)^{-1} L_v C_{\text{meas}} \left(sI_n - \tilde{A} \right)^{-1} \tilde{B} \quad (6.232)$$

that has adequate margins and observer eigenvalues with sufficient separation from the control eigenvalues.

In Wise [18] the observer-Hamiltonian matrix H_{obs} ,

$$H_{\text{obs}} = \begin{bmatrix} \tilde{A} & Q_v \\ C^T R_v^{-1} C & -\tilde{A}^T \end{bmatrix} \quad (6.233)$$

was used to show the relationship between the observer ARE penalty matrices (Q_v, R_v) and their effect on the roots of $\phi_{\text{obs}}(s)$ in (6.231). Using $(\tilde{A}, C_{\text{meas}})$ from (6.195) and \bar{B} is from (6.197), define

$$H_2(s) = C_{\text{meas}} \left(sI_n - \tilde{A} \right)^{-1} \bar{B} \quad (6.234)$$

then

$$\det[sI_n - H_{\text{obs}}] = \phi_{\text{obs}}(s)\phi_{\text{obs}}(-s)$$

$$= \phi_{ol}(s)(-1)^{n_x} \phi_{ol}(-s) \det \left[I_{n_y} + \left(\frac{v+1}{v} \right)^2 H_2(s) H_2^T(-s) \right] \quad (6.235)$$

Thus, as $v \rightarrow 0$, the eigenvalues of the observer are driven to the zeros of the open-loop plant $R_0^{-\frac{1}{2}} C_{\text{meas}} (sI - \tilde{A}) \bar{B}$. The design of the \bar{B} matrix can influence the observer poles, subsequently influencing both the transient response characteristics and the system frequency response, (stability margins).

The adaptive angle κ_v is a measure of the angle between the vectors in (6.166). That is as $v \rightarrow 0$, $\tilde{P}_v B \approx C^T R_0^{-\frac{1}{2}} W$. The matrix $\tilde{P}_v B$ is a column vector in this application, so this asymptotic relation can be analyzed as $v \rightarrow 0$ by computing the angle between these two columns. Cosine of the angle between two vectors is the dot product divided by their magnitudes. We will see when applying LTR that the adaptive angle κ_v significantly improves as $v \rightarrow 0$.

Summary

Dynamics: $\dot{x} = Ax + Bu + w \quad x(t_0) = x_0 \quad x, w \in R^{n_x}, u \in R^{n_u}$

$$y_{\text{meas}} = C_{\text{meas}} x + v \quad y, v \in R^{n_y}$$

$$E\{x\} = \hat{x}, \quad E\{w\} = 0, \quad E\{v\} = 0, \quad \text{cov}(x) = P_v$$

$$E\{ww^T\} = Q_0 \delta(\tau); \quad E\{vv^T\} = R_0 \delta(\tau)$$

Robust Servomechanism LQR:

Command $y_{\text{cmd}} = \text{constant}$.

Controlled output (to follow y_{cmd}): $y_{\text{reg}} = C_p \text{ reg } x + D_p \text{ reg } u$

$$e = y_c - r; \quad z = [e \ \dot{x}]^T, \quad \mu = \dot{u} \quad z \in R^{n_{\text{ext}}}, \quad \mu \in R^{n_u}$$

$$\text{Dynamics: } \dot{z} = \tilde{A}z + \tilde{B}\mu \quad \tilde{A} = \begin{bmatrix} 0 & C_p \text{ reg} \\ 0 & A \end{bmatrix}; \quad \tilde{B} = \begin{bmatrix} D_p \text{ reg} \\ B \end{bmatrix}$$

$$\text{Performance index: } J = \int_0^\infty (z^T Q z + \mu^T R \mu) d\tau$$

(\tilde{A}, \tilde{B}) Stabilizable, $(\tilde{A}, Q^{1/2})$ Detectable

Algebraic Riccati Equation: $P\tilde{A} + \tilde{A}^T P + Q - P\tilde{B}R^{-1}\tilde{B}^T P$

Optimal Control: $\mu = -R^{-1}B^T P x = -K_{\text{ext}} z$

$$u = \int \mu = -K_{\text{ext}} \begin{bmatrix} \int e \\ \hat{x} \end{bmatrix}$$

OBLTR Observer: $\hat{\dot{x}} = \tilde{A}\hat{x} + \tilde{B}u + L_v(y_{\text{meas}} - \hat{y}) \quad \hat{x} \in R^{n_{\text{ext}}}$

Square of the system: $C[\tilde{B} \quad \bar{B}]$ has full rank

$$Q_v = Q_0 + \left(\frac{v+1}{v}\right) \bar{B} \bar{B}^T, \quad R_v = \frac{v}{v+1} R_0$$

$$P_v \tilde{A}^T + \tilde{A} P_v - \left(1 + \frac{1}{v}\right) P_v C^T R_0^{-1} C P_v + Q_0 + \left(1 + \frac{1}{v}\right) \bar{B} \bar{B}^T = 0$$

$$L_v = P_v C^T R_v^{-1}$$

Example 6.3 OBLTR Design for Pitch Dynamics of an Aircraft Table 6.3 lists the design steps we will follow for developing a pitch-axis acceleration command control system. For this example, we will use the same design problem presented in from Chap. 4, Example 4.10. The flight condition is Mach 0.3, 5000 ft altitude, and a trim angle-of-attack α of 5 degrees. The open-loop plant model data (stability and control derivatives), true airspeed, and actuator parameters are

$$\begin{aligned} Z_\alpha/V &= -1.05273 \text{ (1/s)} \\ Z_\delta/V &= -0.0343 \text{ (1/s)} \\ M_\alpha &= -2.3294 \text{ (1/s}^2\text{)} \\ M_q &= -1.03341 \text{ (1/s}^2\text{)} \\ M_\delta &= -1.1684 \text{ (1/s}^2\text{)} \\ V &= 329.127 \text{ (ft/s)} \\ \omega_a &= 2\pi * 13 \text{ (rad/s)} \\ \zeta_a &= 0.6 \end{aligned} \tag{6.236}$$

Step 1: Step 1 begins by defining the RSLQR gain design model, the regulated variable, y_{reg} , and the measurements available for feedback, y_{meas} , to be used in the observer.

Consider the pitch plane dynamics of an unpiloted aircraft given as

$$\begin{aligned} \dot{\alpha} &= \frac{Z_\alpha}{V} \alpha + q + \frac{Z_\delta}{V} \delta_e \\ \dot{q} &= M_\alpha \alpha + M_q q + M_\delta \delta_e \end{aligned} \tag{6.237}$$

It is desired to design an acceleration command $y_{\text{cmd}} = A_z c$ flight control system. We will assume that the command is constant, and will design an RSLQR controller with integral feedback. We will design a constant gain matrix K_z for a single flight condition, and will assume gain scheduling will be used to interpolate the gains between conditions (other design points). Normal acceleration A_z (ft/s²) dynamics are described by

$$A_z = -V \dot{\gamma} = Z_\alpha \alpha + Z_\delta \delta \tag{6.238}$$

The resulting RSLQR design model $\dot{z} = \tilde{A}z + \tilde{B}\mu$ state vector and the control input are

$$\begin{aligned} z &= [e \ \dot{x}^T]^T = [e_{A_z} \ \dot{\alpha} \ \dot{q}]^T \in R^3 \\ \mu &= \dot{\delta}_e \in R \end{aligned} \quad (6.239)$$

where $e_{A_z} = A_z - A_{z_c}$. In matrix form we have

$$\begin{bmatrix} \dot{e}_{A_z} \\ \ddot{\alpha} \\ \ddot{q} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & Z_\alpha & 0 \\ 0 & Z_\alpha/V & 1 \\ 0 & M_\alpha & M_q \end{bmatrix}}_{\tilde{A}} \begin{bmatrix} e_{A_z} \\ \dot{\alpha} \\ \dot{q} \end{bmatrix} + \underbrace{\begin{bmatrix} Z_\delta \\ Z_\delta/V \\ M_\delta \end{bmatrix}}_{\tilde{B}} \dot{\delta}_{e_c} \quad (6.240)$$

Using this model will produce a (Proportional + Integral) state feedback control law for the aircraft.

$$u = -K_1 \int e_{A_z} - K_2 \alpha - K_3 q \quad (6.241)$$

The objective in the design is to provide a fast response and track constant acceleration commands with zero steady-state error, while avoiding using large gains. The RSLQR performance index for this problem is

$$J = \int_0^\infty (z^T Q z + \mu^2 R) d\tau \quad (6.242)$$

Since the control is a scalar we set $R = 1$ (since R can be factored out of the integral in (6.242)). The Q matrix is formed to penalize the error state e_{A_z} in (6.240). This gives

$$z^T Q z = [e_{A_z} \ \dot{\alpha} \ \dot{q}] \begin{bmatrix} q_{11} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_{A_z} \\ \dot{\alpha} \\ \dot{q} \end{bmatrix} = q_{11} e_{A_z}^2, \quad (6.243)$$

Substituting (6.243) into (6.242), gives the performance index as

$$J = \int_0^\infty (q_{11} e_{A_z}^2 + \mu^2) d\tau \quad (6.244)$$

If we check observability of the pair $(\tilde{A}, Q^{1/2})$ we find the system to be observable through this choice of Q .

Step 2: Step 2 from Table 6.3 forms the linear plant model for time domain simulation and frequency domain analysis. The plant model is

$$\begin{aligned}\dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u\end{aligned}\quad (6.245)$$

where $x = [\alpha \ q \ \delta_e \ \dot{\delta}_e]^T$, $u = \delta_{e_c}$, and $y = [A_z \ \alpha \ q \ \delta_e \ \dot{\delta}_e]^T$. The state-space matrices are

$$\begin{bmatrix} A_p & B_p \\ C_p & D_p \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} Z_\alpha/V & 1 & Z_\delta/V & 0 \\ M_\alpha & M_q & M_\delta & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_a^2 & -2\zeta_a\omega_a \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_a^2 \end{bmatrix} \\ \begin{bmatrix} Z_\alpha & 0 & Z_\delta & 0 \\ I_4 & & & \end{bmatrix} & \begin{bmatrix} 0_{5 \times 1} \end{bmatrix} \end{bmatrix} \quad (6.246)$$

The controller model implementing (6.241) is

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} y_{cmd} \\ u &= C_c x_c + D_{c1} y + D_{c2} y_{cmd}\end{aligned}\quad (6.247)$$

where $x_c = \int e_{A_z}$, $y = [A_z \ \alpha \ q \ \delta_e \ \dot{\delta}_e]^T$, and $y_{cmd} = A_{z_c}$. The state-space matrices are

$$\begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} = \begin{bmatrix} [0] & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} & [-1] \\ [-K_1] & \begin{bmatrix} 0 & -K_2 & -K_3 & 0 & 0 \end{bmatrix} & [0] \end{bmatrix} \quad (6.248)$$

Step 3: Step 3 from Table 6.3 uses the extended plant design model of (6.224) to compute the RSLQR state feedback gains K_{ext} . In Chap. 4 we presented a process for creating RSLQR design charts and selecting from these charts the gain design point. For this example, we will reuse the design from Chap. 4, Example 4.10. Populating the RSLQR design model gives

$$\tilde{A} = \begin{bmatrix} 0 & -346.48 & 0 \\ 0 & -1.0527 & 1 \\ 0 & -2.3294 & -1.0334 \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} -11.289 \\ -0.0343 \\ -1.1684 \end{bmatrix} \quad (6.249)$$

The RSLQR (Q , R) matrices in (6.242) are

$$Q = \begin{bmatrix} 0.081113 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad R = 1 \quad (6.250)$$

Solving the ARE gives the following gain matrix

$$K_{\text{ext}} = [0.2848 \ -37.114 \ -9.1957] \quad (6.251)$$

Step 4: The observer design process begins with defining the measurement vector to be used by the observer. Figure 6.12 shows the closed-loop block diagram implementing the OBLTR control law

$$\begin{aligned} \dot{\hat{x}} &= A_{\text{obs}}\hat{x} + B_{\text{cmd}}y_{\text{cmd}} + L_v y_{\text{meas}} \\ u &= -K_{\text{ext}}\hat{x} \end{aligned} \quad (6.252)$$

where $\hat{x} = [\int e_{A_z} \alpha \ q]^T$. In Fig. 6.12, and in (6.195), an integrator is added to the controller to form the RSLQR integral error, and the signal is routed to the observer as a measurement. This defines the measurement vector.

$$y_{\text{meas}} = \begin{bmatrix} \int e_{A_z} \\ q \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{C_{\text{meas}}} \begin{bmatrix} \int e_{A_z} \\ \alpha \\ q \end{bmatrix} \quad (6.253)$$

This technique has significant advantages in terms of filtering the acceleration signal (the integrator acts as a low pass filter attenuating noise and other high frequency signals) without a phase lag penalty.

To complete the observer design we need to define the parameters Q_0 , R_0 , \bar{B} , v that enter into the ARE weighting matrices as

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) \bar{B} \bar{B}^T, \quad R_v = \frac{v}{v+1} R_0 \quad (6.254)$$

In this example we will illustrate how to set elements of (Q_0, R_0) to create an observer faster than the RSLQR eigenvalues. We will fix $R_0 = I_2$ and vary diagonal elements of Q_0 to see which elements influence the observer closed-loop eigenvalues the most. The RSLQR closed-loop eigenvalues $\lambda(\tilde{A} - \tilde{B}K_{\text{ext}})$

are $\lambda_i = \{-3.79, -3.55 \pm 3.98j\}$. To begin, we set $Q_0 = I_3$; $R_0 = I_2$ with $Q_v = Q_0$, $R_v = R_0$ in (6.254) and solve the ARE. This results in

$$L_v = \begin{bmatrix} 2.5559e+01 & -1.6057e-01 \\ -9.4129e-01 & 1.8821e-02 \\ -1.6057e-01 & 3.6461e-01 \end{bmatrix} \quad (6.255)$$

The observer closed-loop eigenvalues $\lambda(\tilde{A} - L_v C_{\text{meas}}) = \{-1.75, -1, 31 \pm 1.32j\}$ are too small, too low in frequency. By making Q_0 larger, the eigenvalues $\lambda(\tilde{A} - L_v C_{\text{meas}})$ will become faster. With $R_v = I_2$, set

$$\begin{aligned} Q_v = qq Q_0_i; \quad Q_{01} &= \text{diag}[100, 1, 1]; \quad Q_{02} = \text{diag}[1, 100, 1]; \\ Q_{03} &= \text{diag}[1, 1, 100]; \end{aligned} \quad (6.256)$$

where qq is a vector of penalty weights defined on a log interval with bounds $[10^{-2}, 10]$ using 100 design points. A small value of $qq(i)$, $qq(1) = 10^{-2}$, will produce an observer design with small, low frequency eigenvalues. For each Q_0_i in (6.256), we loop through the qq vector scaling Q_0_i and compute the resulting $\lambda(\tilde{A} - L_v C_{\text{meas}})$. Then we plot the eigenvalues and examine the resulting root locus (Fig. 6.16) with varying qq and Q_0 in (6.256).

The RSLQR eigenvalues are shown as red circles with the eigenvalues $\lambda(\tilde{A} - L_v C_{\text{meas}})$ as blue x's. Note that the axes change scale. Both the index 1 and 2 plots show eigenvalues on the real axis which are slower than the RSLQR values. The index 3 plot, Q_{03} , shows all of its eigenvalues faster (further into the LHP) than that of the RSLQR. Thus, we will set $Q_0 = Q_{03}$ and $R_0 = I_2$.

The next step in the observer design process is to square up the extended system as described in Sect. 6.2.4, Table 6.1. Following Table 6.1, Assumption 6.1, the

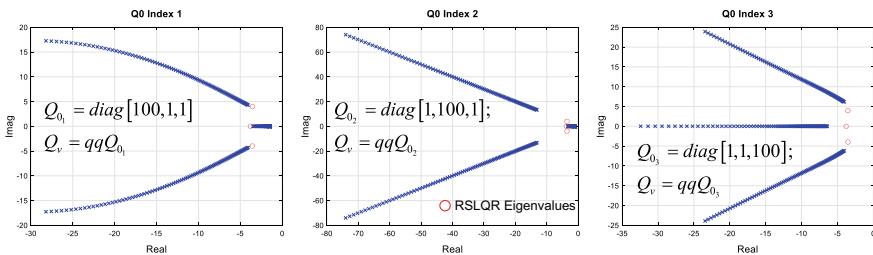


Fig. 6.16 Observer eigenvalues varying Q_0

system

$$\begin{bmatrix} \tilde{A} & \tilde{B} \\ C_{\text{meas}} & 0_{2 \times 1} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & -3.4648e+02 & 0 \\ 0 & -1.0527e+00 & 1.0000e+00 \\ 0 & -2.3294e+00 & -1.0334e+00 \end{bmatrix} & \begin{bmatrix} -1.1289e+01 \\ -3.4300e-02 \\ -1.1684e+00 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} \quad (6.257)$$

is controllable, observable and has no finite transmission zeros (TZs). We see from (6.257) that this is a tall system, ($n_y > n_u$). As outlined in Table 6.1, squaring-up a tall system necessitates finding $(n_y - n_u)$ columns $B_2 \in R^{n_x \times (n_y - n_u)}$ so that the Rosenbrock matrix

$$\text{rank} \begin{bmatrix} \tilde{A} - s_0 I_{n_x} & \bar{B} = (B \ B_2) \\ C & 0_{n_y \times n_y} \end{bmatrix} < n_x + n_y \Rightarrow s_0 \in \mathbb{C}^- \quad (6.258)$$

drops rank at $s_0 \in \mathbb{C}^-$. Following the procedure outlined in Table 6.1, a matrix $B_{z22} \in R^{(n_x - n_y) \times (n_y - n_u)}$ can be found to assign the $(n_x - n_y)$ desired TZs to become the eigenvalues of A_{z22cl} in (6.193). Equivalently, from (6.193),

$$A_{z22cl} = \underbrace{\left(A_{z22} - B_{z21} \tilde{A}_{z121} \right)}_A - \underbrace{B_{z22}}_L \underbrace{\tilde{A}_{z122}}_C \quad (6.259)$$

where $L = B_{z22}$ can be computed using a pole placement algorithm. Using the *place* function in MATLAB®, we compute B_2 to place a TZ at $s_0 = -5.3$. The resulting \bar{B} is

$$\bar{B} = \begin{bmatrix} -1.1289e+01 & 1.1540e+00 \\ -3.4300e-02 & -1.7777e+00 \\ -1.1684e+00 & -1.1150e+01 \end{bmatrix} \quad (6.260)$$

and when substituted in (6.258) produces the desired TZ.

To complete this example, we will evaluate several TZ locations and apply LTR to the system to illustrate how to interpret the frequency domain analysis results needed to select design parameters. Once a TZ is selected, B_2 in (6.258) is computed, and then the frequency domain properties of the system are evaluated (Figs. 6.12 and 6.13).

Figure 6.17 shows the Nyquist plot, Bode magnitude, $\underline{\sigma}(I_u + L_u)$, $1/|T_u|$, $|A_z/A_{zc}| = |T_{y11}|$, and $|e_{A_z}/A_{zc}| = |S_{y11}|$ frequency responses as $v \rightarrow 0$. The $v = 999$ in the legend denotes no LTR (L_v designed only using (Q_0, R_0)). The Nyquist and Bode magnitude plots show how as $v \rightarrow 0$, $L_u(s) \rightarrow L_{u\text{lqr}}(s)$. The observer without LTR clearly has insufficient stability margins. As $v \rightarrow 0$, the singular value stability margins $\sigma_S = \min_\omega \underline{\sigma}(I_u + L_u)$ and $\sigma_T = \min_\omega 1/|T_u|$

significantly improve. Both the complementary sensitivity, $|T_{y_{11}}|$, and sensitivity, $|S_{y_{11}}|$, look good and are not significantly influenced by LTR, as expected.

We know that as $v \rightarrow 0$, $\|L_v\|_2 \rightarrow \infty$. Figure 6.18 shows the magnitude plots of sensor noise to control, $|u/v_{\text{noise}}|$ and sensor noise-to-control rate $|\dot{u}/v_{\text{noise}}|$ frequency responses. The two output measurements are acceleration and pitch rate. We see from Fig. 6.18 that the gyro noise is amplified through the observer. The use of integral error as a measurement in the observer, (6.253), produces significant

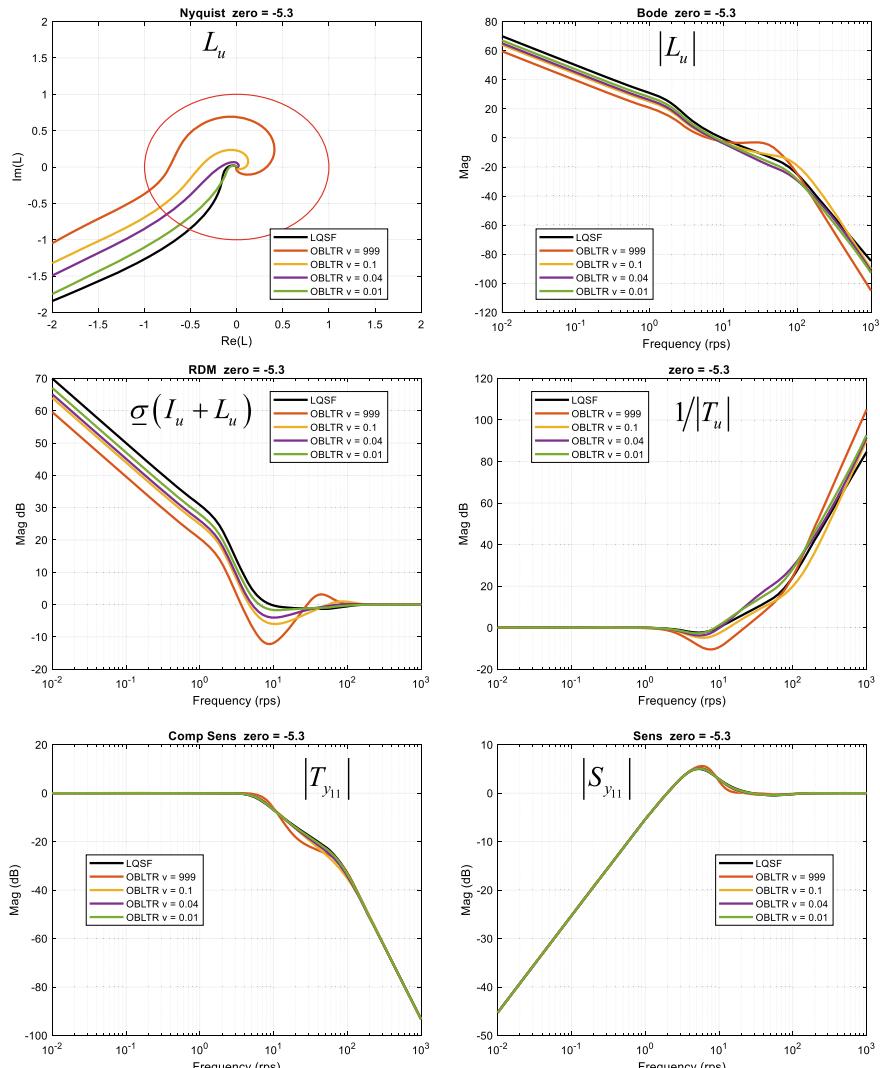


Fig. 6.17 Example 6.2 frequency response analysis at plant input and output

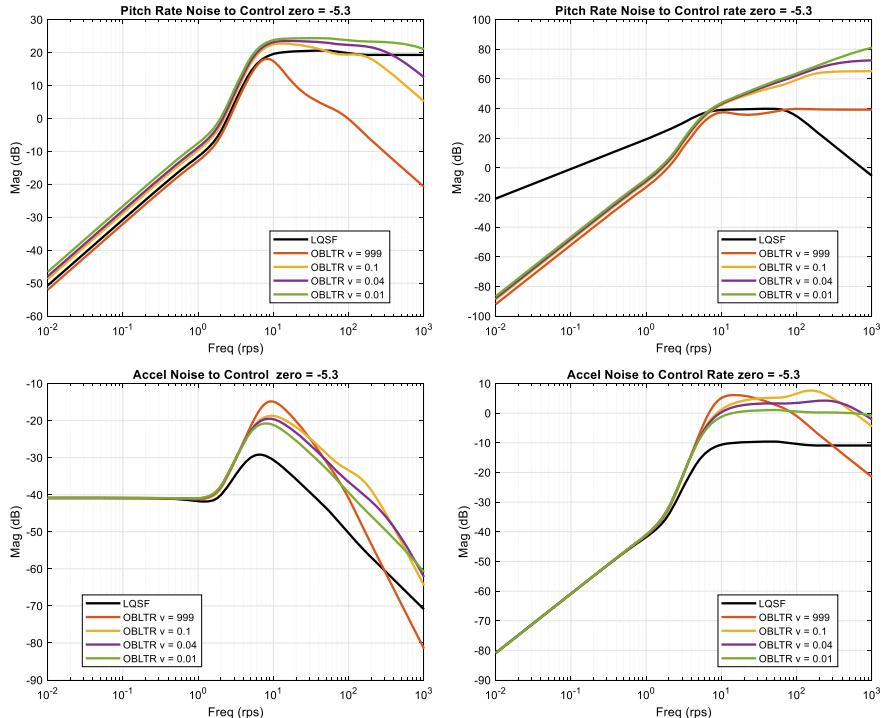


Fig. 6.18 Example 6.2 sensor noise-to-control and control rate magnitude plots

attenuation of the accelerometer noise. These plots show that additional filtering might be needed on the rate gyro signal but not on the accelerometer signal.

Figure 6.19 shows the $\|L_v\|_2$ with max and min eigenvalues of $(\tilde{A} - L_v C_{\text{meas}})$ denoted $\bar{\lambda}(\tilde{A} - L_v C_{\text{meas}})$ and $\underline{\lambda}(\tilde{A} - L_v C_{\text{meas}})$, respectively, and the max eigenvalue of $(\tilde{A} - \tilde{B} K_{\text{ext}})$, $\bar{\lambda}(\tilde{A} - \tilde{B} K_{\text{ext}})$, (which is not a function of v), as well as the adaptive angle κ_v . These are plotted versus LTR parameter, $1/\rho = v/(v+1)$, as $v \rightarrow 0$. The $\|L_v\|_2$ plot shows how the observer gain matrix norm is increasing in size, as $v \rightarrow 0$. The eigenvalue curves show that as $v \rightarrow 0$ the $\bar{\lambda}(\tilde{A} - L_v C_{\text{meas}})$ is getting much faster (see Fig. 6.16). The plot of $\underline{\lambda}(\tilde{A} - L_v C_{\text{meas}})$ compared to $\bar{\lambda}(\tilde{A} - \tilde{B} K_{\text{ext}})$ (red curve) shows that the observer is always faster than the RSLQR, providing frequency separation. We see that as $v \rightarrow 0$, the adaptive angle significantly improves, (becomes smaller). This asymptotic feature of the OBLTR design will play an important role in later chapters in using the observer as a reference model for model reference adaptive control.

Figures 6.17, 6.18 and 6.19 show observer analysis results for a TZ of $s_0 = -5.3$. Experience with the squaring-up procedure has shown there exist trends

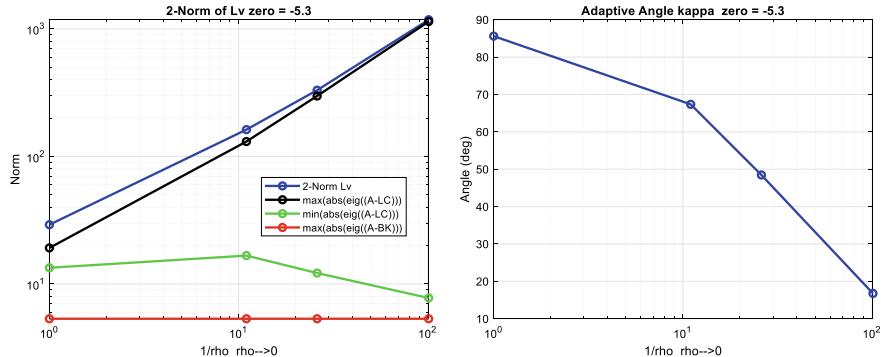


Fig. 6.19 Example 6.2 sensor noise magnitude peak and gain matrix 2-norm comparison

where the TZs are placed and how small $v \rightarrow 0$ needs to be adjusted to recover the RSLQR stability margins. Lower frequency TZs allow for margin recovery with larger values of v . And since $\|L_v\|_2 \rightarrow \infty$ as $v \rightarrow 0$, it is desirable to use larger values of v . Figure 6.20 shows a summary of the observer design and analysis results, while varying TZ location. The σ_S and σ_T are used in computing singular values stability margins are defined in defined in Chap. 3 (the lower two plots). On the plots we have shown arrows indicating trends that are desirable and minimum values. These results show that lower frequency TZs with a larger value of v produce the best results.

For this example, we will select $s_0 = -1$ with $v = 0.04$. The plant model is given in (6.246). The following matrices are used to compute the observer gains,

$$\begin{aligned} \bar{B} &= \begin{bmatrix} -1.1289e+01 & 1.1679e+00 \\ -3.4300e-02 & -3.3659e-01 \\ -1.1684e+00 & -1.1284e+01 \end{bmatrix}, \\ Q_v &= \begin{bmatrix} 3.3500e+03 & -1.5290e-01 & 2.9850e-01 \\ -1.5290e-01 & 3.9761e+00 & 9.9792e+01 \\ 2.9850e-01 & 9.9792e+01 & 3.4461e+03 \end{bmatrix}, \\ R_v &= \begin{bmatrix} 3.8462e-02 & 0 \\ 0 & 3.8462e-02 \end{bmatrix}, \quad L_v = \begin{bmatrix} 3.0064e+02 & -5.4639e+00 \\ -4.7820e+00 & 9.5139e+00 \\ -5.4639e+00 & 2.9818e+02 \end{bmatrix} \end{aligned} \quad (6.261)$$

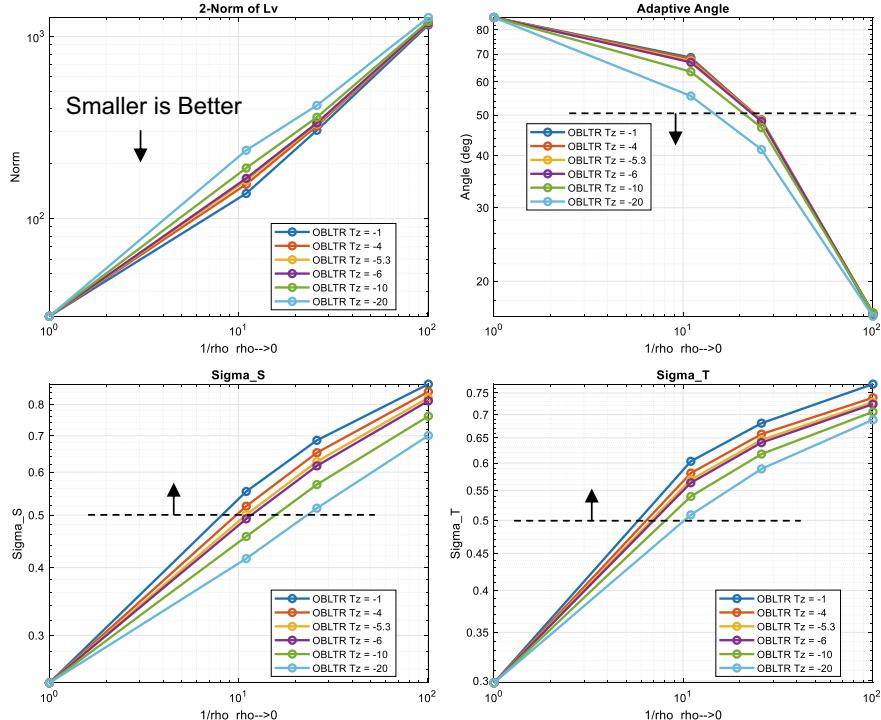


Fig. 6.20 Example 6.2 LTR summary varying transmission zero location

with the resulting controller matrices given as

$$\begin{aligned}
 & \begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_c & D_{c1} & D_{c2} \end{bmatrix} \\
 &= \begin{bmatrix} \begin{bmatrix} 0 & 0_{1 \times 3} \\ L_v(:, 1) (\tilde{A} - \tilde{B} K_{\text{ext}} - L_v C_{\text{meas}}) & \end{bmatrix}_{4 \times 4} & \begin{bmatrix} 1 & 0 & 0 & 0_{1 \times 2} \\ 0_{3 \times 2} & L_v(:, 2) & 0_{3 \times 2} \end{bmatrix}_{4 \times 5} & \begin{bmatrix} -1 \\ -1 \\ 0_{2 \times 1} \end{bmatrix}_{4 \times 1} \\ \begin{bmatrix} 0 & -K_{\text{ext}} \end{bmatrix}_{1 \times 4} & \begin{bmatrix} 0_{1 \times 5} \end{bmatrix} & [0] \end{bmatrix} \\
 &= \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 3.0064e+02 & -2.9742e+02 & -7.6546e+02 & -9.8347e+01 \\ -4.7820e+00 & 4.7918e+00 & -2.3257e+00 & -8.8293e+00 \\ -5.4639e+00 & 5.7966e+00 & -4.5693e+01 & -3.0995e+02 \end{bmatrix}_{1 \times 4} & \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 0 & -5.4639e+00 & 0 \\ 0 & 0 & 9.5139e+00 & 0 \\ 0 & 0 & 2.9818e+02 & 0 \end{bmatrix}_{4 \times 4} & \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & -2.8480e-01 & 3.7114e+01 & 9.1957e+00 \end{bmatrix}_{1 \times 4} & \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}_{4 \times 1} & [0] \end{bmatrix} \quad (6.262)
 \end{aligned}$$

Figure 6.21 shows the closed-loop system step-input response.

The purple curve is under the RSLQR black curve. Figure 6.22 shows the Nyquist and Bode magnitude data for this design.

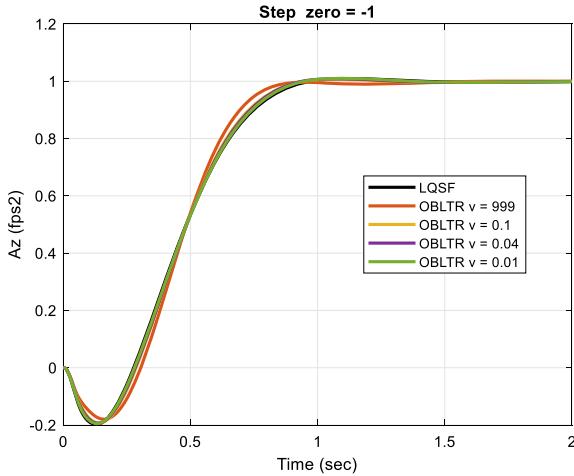


Fig. 6.21 Example 6.2 design point step response

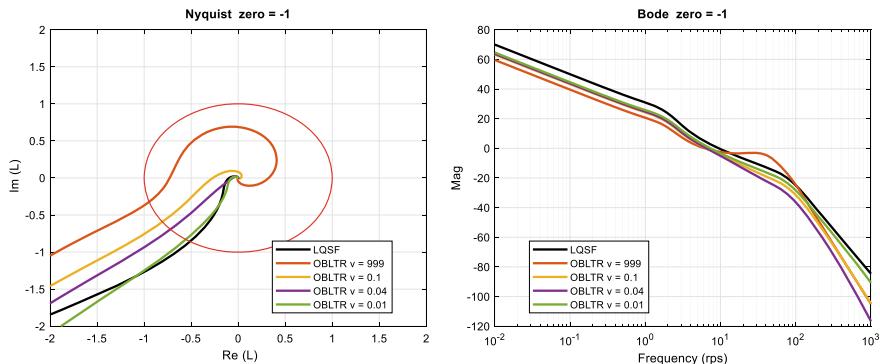


Fig. 6.22 Example 6.2 with low frequency transmission zero $s_0 = -1$

Note how the Bode magnitude preserves the filtering/attenuation of the observer. Table 6.4 shows the analysis results and stability margins.

Overall, the design and analysis data indicate a reasonably well-tuned flight control system with adequate stability margins and command tracking performance. ■

Table 6.4 Example 6.2 design results

S_0 (rps)	v	$\ L_v\ _2$	κ_v (deg)	σ_S	σ_T	GM (dB)	PM (deg)
- 1	0.04	305.2	48.6	0.6867	0.6811	[- 9.93, 10.1]	± 40.1617

6.4 Summary

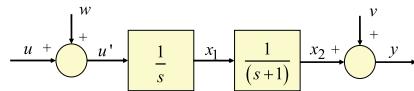
In this chapter, we presented static output feedback and dynamic projective control, followed by the design of full state observers using the OBLTR framework. These design methods in no way capture all output feedback methods available to the engineer. We selected these methods because they have proven to be good design methods in practice, and more importantly for the reader, they demonstrate the insights needed to develop a realistic control system.

The static projective control method has been found to be very effective at designing output feedback controllers. In flight control applications using gain scheduling, these controllers are of low-order, making them seamless to implement.

The OBLTR controllers require a dynamic observer for implementation. In flight control applications where gain scheduling is relied upon to compensate for a large flight envelope, these state observers may introduce small transients as the observer parameters vary with scheduling parameters. The engineer must analyze, simulate and evaluate if these transients are acceptable.

6.5 Exercises

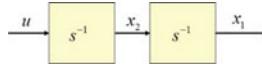
Exercise 6.1 Consider a second-order system shown in the block diagram



where u is the scalar control, w the scalar plant disturbance, y the scalar measurement, and v measurement noise, and x_1 and x_2 are states of the system.

- Derive the following state-space model of the form $\dot{x} = Ax + Bu + Gw$, $y = Cx + v$. List the matrices (A, B, G, C) .
- Using the state-space model from (a), design an optimal observer $\hat{\dot{x}} = A\hat{x} + Bu + L(y - \hat{y})$ using the filter Algebraic Riccati Equation using $Q = \begin{bmatrix} 16 & 0 \\ 0 & 0 \end{bmatrix}$, $R = 1$. List the Riccati solution matrix P and the observer gain matrix L .
- What are the eigenvalues of the observer error dynamics?

Exercise 6.2 Consider the double integrator system given by this block diagram.



- (a) Only x_1 can be measured. Design an optimal observer (FARE) using $Q = \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix}$, $R = 1$ to implement the estimated state feedback control $u = -[k_1 \ k_2] \hat{x}$. List the FARE matrix solution P , the resulting observer gain matrix L .
- (b) Write an expression for the observer error $e = x - \hat{x}$ dynamics. What are the eigenvalues for the error dynamics?
- (c) Write a state-space controller implementing the observer-based controller.

Exercise 6.3 Consider the unstable longitudinal dynamics model, as defined in Example 6.1, where $x = [\alpha \ q \ \delta_e \ \dot{\delta}_e]^T$. The matrices for the control design model $\dot{x} = A_p x + B_p u$ are

$$[A_p \ B_p] = \left[\begin{bmatrix} -1.3046e & 1.0 & -0.21420 & 0 \\ 47.711 & 0 & -104.83 & 0 \\ 0 & 0 & 0 & 1.0 \\ 0 & 0 & -12769. & -135.6 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 12769 \end{bmatrix} \right]$$

- (a) Design a robust servomechanism LQR state feedback control to track a constant α command using the method of Chap. 3. Simulate the state feedback design to show the command tracking.
- (b) It is desired not to feedback the elevator state and rate signals to improve reliability. Use the static projective control method of Sect. 6.1 to project out the actuator dynamics, keeping the dominant eigenstructure for command tracking. Simulate the static projective control design to show the command tracking and compare with (a).
- (c) Compute the eigenstructure for (a) and (b) to show that the dominant eigenvalues are retained. Analyze this design in the frequency domain. Compute Nyquist, Bode, $|I + L_u|$, $1/|T_u|$ frequency responses for (a) and (b) at the plant input. Compute $|S_y|$ and $|T_y|$ frequency responses for (a) and (b) at the plant output for the α loop. Compute the loop gain-crossover frequency and singular value stability margins for the design.

Exercise 6.4 Consider the unstable longitudinal dynamics from Exercise 6.3. The output signals available from the inertial measurement unit are $y = [A_z \ q]^T$. The matrices for the output model $y = C_p x + D_p u$ are

$$[C_p \ D_p] = \begin{bmatrix} -1156.9 & 0 & -189.95 & 0 \\ 0 & 1.0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- (a) Design a robust servomechanism LQR state feedback control to track a constant α command using the method of Chap. 3. (Same controller from Exercise 6.1.) Design a full state Kalman filter observer to estimate the states for feedback using the method outlined in Sect. 6.2. Use the following plant process and measurement noise covariance matrices for the Kalman filter design

$$Q_0 = \begin{bmatrix} 1.0 \times 10^{-8} & 0 & 0 & 0 \\ 0 & 2.5 \times 10^{-7} & 0 & 0 \\ 0 & 0 & 1.0 \times 10^{-8} & 0 \\ 0 & 0 & 0 & 1.0 \times 10^{-6} \end{bmatrix} \begin{bmatrix} (\text{rad})^2/\text{s} \\ (\text{rps})^2/\text{s} \\ (\text{rad})^2/\text{s} \\ (\text{rps})^2/\text{s} \end{bmatrix}$$

$$R_0 = \begin{bmatrix} 6.25 \times 10^{-2} & 0 \\ 0 & 1.0 \times 10^{-6} \end{bmatrix} \begin{bmatrix} (\text{fps})^2 \\ (\text{rps})^2 \end{bmatrix} \quad (6.263)$$

List all matrices used in the design.

- (b) Simulate the LQG design and compare it to the state feedback design.
(c) Analyze this LQG design in the frequency domain. Compute Nyquist, Bode, $|I + L_u|$, $1/|T_u|$ frequency responses for the LQG and state feedback at the plant input. Compute $|S_y|$ and $|T_y|$ frequency responses for (a) and (b) at the plant output for the α loop. Compute the loop gain-crossover frequency and singular value stability margins for both designs. Determine the impact of using the Kalman filter estimator on the stability robustness of the system.
(d) Use the OBLTR method of Sect. 6.2 to recover the frequency domain properties of the state feedback design in the LQG design. Evaluate the design in the frequency domain as in (c). Compute the maximum singular value of the noise-to-control transfer function matrix frequency response to examine the noise amplification in the resulting OBLTR design.

Exercise 6.5 Give examples of functions $f(\varepsilon)$ and $g(\varepsilon)$ that satisfy the following asymptotic relations, as $\varepsilon \rightarrow 0$:

- (a) $f(\varepsilon) = O(1)$
- (b) $f(\varepsilon) = o(1)$
- (c) $f(\varepsilon) = O(g(\varepsilon))$
- (d) $f(\varepsilon) = o(g(\varepsilon))$

Exercise 6.6 Prove asymptotic relations (6.106) and (6.107).

Exercise 6.7 Suppose that in the parameter-dependent Algebraic Riccati Equation (6.116) all quantities are scalars. Analytically solve it to calculate the unique positive root. Use the asymptotic expansion (6.130) and compute the first two terms in the series. Show that the asymptotic relation (6.137) takes place.

Exercise 6.8 Based on Table 6.1, implement/code the squaring-up algorithm. Verify your implementation using the short-period dynamics model from Example 6.2. Use data of your choice to populate the model parameters. For the selected model, verify that the explicit squaring-up solution in Example 6.2 matches the output of your implementation. Select a MIMO tall system, such as lateral-directional aircraft dynamics from Chap. 1, to verify your squaring-up solution implementation.

Exercise 6.9 Design, analyze, and simulate an OBLTR controller for aircraft short-period two-state dynamics with one control input. The regulated output is the aircraft vertical acceleration. The measurements consist of the vehicle pitch rate and the vertical acceleration. Use data from Chap. 1 to populate the model parameters.

Exercise 6.10 Design, analyze, and simulate an OBLTR controller for aircraft lateral-directional three-state dynamics with two control inputs. The regulated output is the aircraft roll rate in stability axis and the lateral acceleration. The measurements consist of the regulated output and the vehicle yaw rate in stability axis. Use data from Chap. 1 to populate the model parameters.

References

1. Hopkins WE Jr, Medanic J, Perkins WR (1981) Output feedback pole placement in the design of suboptimal linear quadratic regulators. *Int J Control* 34:593–612
2. Medanic J, Uskokovic Z (1983) The design of optimal output regulators for linear multivariable systems with constant disturbances. *Int J Control* 37:809–830
3. Tharp HS, Medanic JV, Perkins WR (1986) Robust projective controls for structured perturbations. In: Proceedings of the American control conference, Seattle, WA, pp 1833–1838
4. Luenberger DG (1964) Observing the state of a linear system. *IEEE Trans Mili Electron* 8:74–80
5. Luenberger DG (1966) Observers for multivariable systems. *IEEE Trans Autom Control* 11(2):190–197
6. Luenberger DG (1971) An introduction to observers. *IEEE Trans Autom Control* 16(2):596–602
7. Kwakernaak H, Sivan R (1972) Linear optimal control systems. Wiley, New York
8. Anderson BDO, Moore JB (1990) Optimal control, linear quadratic methods. Dover, Mineola, NY
9. Davidson EJ, Copeland B (1985) Gain margin and time lag tolerance constraints applied to the stabilization problem and robust servomechanism problem. *IEEE Trans Autom Control* 30(3):229–239

10. Lavretsky E (2012) Adaptive output feedback design using asymptotic properties of LQG / LTR controllers. *IEEE Trans Autom Control* 57(6):1587–1591
11. Kevorkian J, Cole JD (1996) Multiple scale and singular perturbation methods. Applied mathematical sciences, vol 114. Springer, New York
12. Murray JD (1984) Asymptotic analysis. Applied mathematical sciences, vol 48. Springer, New York
13. Kailath T (1980) Linear systems. Prentice-Hall Inc., Englewood Cliffs, NJ
14. Misra P (1998) Numerical algorithms for squaring-up non-square systems, Part II: General case. In: Proceedings of American control conference, San Francisco, CA
15. Balas G, Young P (2003) Sensor selection via closed-loop control objectives. *IEEE Trans Autom Control* 7(6):692–704
16. Wal M, Jager B (2001) A review of methods for input/output selection. *Automatica* 37:487–510
17. Bryson EA, Ho Y-C (1975) Applied optimal control. Optimization, estimation and control. Taylor & Francis
18. Wise KA (2018) Design parameter tuning in adaptive observer-based flight control architectures. In: Proceedings of the 2018 AIAA SciTech conference, Kissimmee, FL

Part II

Robust Adaptive Control



Direct Model Reference Adaptive Control: Motivation and Introduction

7

This chapter presents essential concepts for the now-classical model reference adaptive control. We begin with motivational examples from aerospace applications, followed by basic definitions, and a brief description of control-theoretic tools for the design and analysis of state feedback adaptive controllers that are applicable to an aircraft-like general class of multi-input–multi-output systems. Our primary goal here is to motivate, introduce, and outline the material that will be discussed in the remainder of the book.

7.1 Model Reference Control: Motivational Example

In the design of flight control systems, it is essential to provide closed-loop stability, adequate command tracking performance, as well as robustness to model uncertainties, control failures, and environmental disturbances. In the previous chapters, we considered optimal linear quadratic regulator (LQR) control design techniques that were suitable for flight control of aerial systems. These design methods relied on the inherent robustness properties of LQR optimal controllers. It was shown that with a proper selection of the LQR design tuning parameters (Q and R matrices), one could achieve 6 dB gain margin, and at least 60° phase margin, at the system control input break points.

It is also possible to show that LQR optimal controllers can tolerate time-state-dependent nonlinear uncertainties that might be present in the system control

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-38314-4_7.

channels. These uncertainties are called “matched” since they appear only where control inputs exist in the system dynamics. The matching conditions imply that if the system uncertainties were known, a controller would have the ability to cancel them out.

In the presence of matched uncertainties, a deterioration of the system baseline closed-loop performance is inevitable. This is to be expected since the LQR controllers are designed to be robust to the entire class of matched uncertainties. However, they are not tuned to handle any specific uncertainty from this class. In other words, these LQR controllers may become overly conservative.

We pose the question: “Can we restore a given baseline closed-loop performance of the system, while operating under matched uncertainties?” The answer is “Yes”. This is the area where adaptive controllers are highly effective.

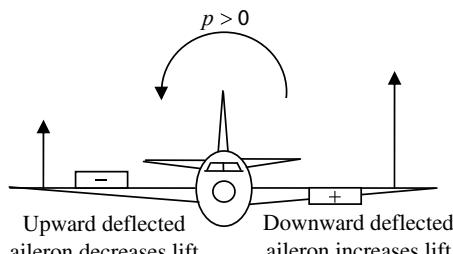
Throughout the chapters of Part II, we shall utilize the concept of a reference model for specifying the desired closed-loop tracking performance. Fixed-gain controllers, as well as adaptive systems, can be constructed using the reference model-based design concept. We shall begin our discussions with a motivational example.

Example 7.1 Fixed-Gain Model Reference Control of Aircraft Roll Dynamics The roll dynamics of a conventional aircraft are controlled using differential motion of ailerons and spoilers. Ailerons are movable surfaces that are mounted outboards on the trailing edge of the wing, where they are placed symmetrically on each side of the wing, with respect to the aircraft centerline (Fig. 7.1).

Deflected differentially (e.g., downward on one side and upward on the other), ailerons have the ability to increase the lift force on the downward deflected portion of the wing and to decrease it on the other side. The two distinct lift forces will create a rolling moment around the aircraft velocity vector placed at the aircraft center of gravity. While ailerons can move up and down, spoilers can only be deflected upward above the trailing edge of the wing to reduce the lift force and thus to aid ailerons in providing roll control. As a result, the aircraft rotates around its velocity vector. In this case, the aircraft roll dynamics can be approximated by a scalar (first-order) ordinary differential equation (ODE) in the form,

$$\dot{p} = L_p p + L_{\delta_a} \delta_a \quad (7.1)$$

Fig. 7.1 Lift forces arising from positive differential aileron deflection cause aircraft to roll counterclockwise (positive roll rate)



where p is the aircraft roll rate in stability axes (radians/s), δ_a is the total differential aileron-spoiler deflection (radians), L_p is the roll damping derivative, and L_{δ_a} is the dimensional rolling moment derivative with respect to differential aileron-spoiler deflection, (the aileron-to-roll control effectiveness). For a conventional open-loop-stable aircraft, the roll damping derivative L_p is negative, unless portions of the wing are stalled, in which case the roll damping may become positive. Positive differential aileron-spoiler deflection is defined to produce positive rolling moment, and as such, the aileron-to-roll control effectiveness L_{δ_a} typically has positive values.

Strictly speaking, the roll dynamics approximation above is valid only for sufficiently small values of p and δ_a . In addition, it is assumed that the aircraft yawing motion is suppressed by the ruddera vertical tail mounted surface. Readers who might be unfamiliar with the flight mechanics nomenclature, may consider (7.1) as a scalar ODE $\dot{x} = a x + b u$, with two constant parameters $a = L_p$, $b = L_{\delta_a}$, whose state and control input are $x = p$ and $u = \delta_a$, respectively.

The control task of interest is to force the aircraft to roll like the reference model,

$$\dot{p}_{\text{ref}} = a_{\text{ref}} p_{\text{ref}} + b_{\text{ref}} p_{\text{cmd}} \quad (7.2)$$

with the prescribed values of $a_{\text{ref}} < 0$ (the desired inverse time constant) and $b_{\text{ref}} > 0$ (the desired DC gain). The reference model (7.2) is driven by the commanded roll rate p_{cmd} and it calculates the reference roll rate p_{ref} . In essence, the reference model (7.2) imbeds and defines the desired closed-loop command tracking performance. The control task amounts to finding δ_a that would force the aircraft roll rate p track any bounded, possibly time varying, reference command p_{ref} . This is the model reference control design task. Sometimes, it is also referred to as the model following control. Using this concept, allows the designer to create controllers whose main task is to asymptotically match a given reference model behavior. Let us now explore details of the model reference control design.

Comparing the roll dynamics (7.1) to that of the reference model (7.2), it is easy to see that a control solution can be formulated in the feedback-feedforward form,

$$\delta_a = \left(\frac{a_{\text{ref}} - L_p}{L_{\delta_a}} \right) p + \left(\frac{b_{\text{ref}}}{L_{\delta_a}} \right) p_{\text{cmd}} \quad (7.3)$$

where $k_p = \frac{a_{\text{ref}} - L_p}{L_{\delta_a}}$ is the roll-rate feedback gain, and $k_{p_{\text{cmd}}} = \frac{b_{\text{ref}}}{L_{\delta_a}}$ is the command feedforward gain. In fact, substituting the controller (7.3) into the roll dynamics (7.1), gives the desired closed-loop system dynamics.

$$\dot{p} = a_{\text{ref}} p + b_{\text{ref}} p_{\text{cmd}} \quad (7.4)$$

In order to formally assess if (7.4) indeed converges to (7.2), we first define the roll-rate tracking error,

$$e = p - p_{\text{ref}} \quad (7.5)$$

and then compute the tracking error dynamics by differentiating e with respect to time, while substituting (7.4) and (7.2).

$$\dot{e} = \dot{p} - \dot{p}_{\text{ref}} = a_{\text{ref}}(p - p_{\text{ref}}) = a_{\text{ref}}e \quad (7.6)$$

Since by definition $a_{\text{ref}} < 0$ (e.g., the reference model is exponentially stable), the error dynamics (7.6) are globally exponentially stable. Therefore, given any initial values $p(0)$ and $p_{\text{ref}}(0)$, the tracking error $e(t)$ will converge to the origin exponentially fast,

$$e(t) = \exp(a_{\text{ref}}t)e(0) \quad (7.7)$$

starting at any initial tracking error value $e(0) = p(0) - p_{\text{ref}}(0)$. So, the aircraft roll rate $p(t)$ will track the reference roll rate $p_{\text{ref}}(t)$, with the exponentially fast decaying tracking error $e(t)$,

$$p(t) = p_{\text{ref}}(t) + \exp(a_{\text{ref}}t)(p(0) - p_{\text{ref}}(0)) \quad (7.8)$$

and this closed-loop tracking performance is valid for any constant or any bounded time-varying command $p_{\text{cmd}} = p_{\text{cmd}}(t)$. The command tracking problem is solved. The corresponding closed-loop system block diagram with the fixed-gain model reference controller (7.3) is shown in Fig. 7.2.

The model reference controller (7.3) is by no means unique in solving the command tracking problem of interest. Other solutions can be found. For example, any controller in the form,

$$\delta_a = k_p p + k_{p_{\text{cmd}}} p_{\text{cmd}} - k_e(p - p_{\text{ref}}) \quad (7.9)$$

solves the same tracking problem, where $k_e \geq 0$ represents the error feedback gain.

However, does the error feedback in (7.9) give any advantage over the original controller (7.3)? In order to answer that question, let us calculate the error dynamics obtained using the modified controller (7.9).

$$\dot{e} = (a_{\text{ref}} - k_e)e \quad (7.10)$$

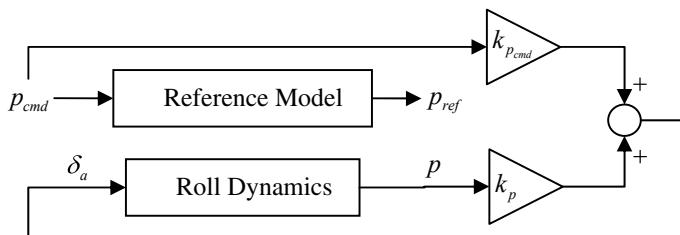


Fig. 7.2 Block diagram of the closed-loop roll dynamics with fixed-gain model reference controller obtained in Example 7.1

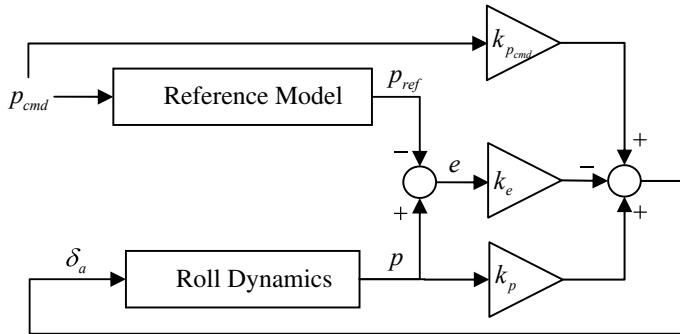


Fig. 7.3 Closed-loop system block diagram with fixed-gain model reference controller and error feedback obtained in Example 7.1

Consequently,

$$p(t) = p_{ref}(t) + \exp((a_{ref} - k_e)t)(p(0) - p_{ref}(0)) \quad (7.11)$$

By definition, the error (7.10) dynamics define the transients that are incurred by the system while tracking a given reference command $p_{ref}(t)$. It is now evident that choosing $k_e > 0$ sufficiently large, will allow the designer to obtain any desired (fast) transient dynamics. This constitutes the primary advantage of using an error feedback gain in the fixed-gain model reference controller (7.9). Figure 7.3 shows the resulting closed-loop system block diagram.

Of course, practical limitations, as well as stability robustness considerations, will place upper and lower limits on the selection of the controller gains. Eventually, these restrictions will dictate the tradeoff between achievable transients in the closed-loop system and adequate stability robustness margins. ■

7.2 Introduction to Direct Model Reference Adaptive Control

In (7.1) the roll control example above, we have assumed that the system dynamics (defined by the aircraft aerodynamics) were completely known. Then, we utilized the roll damping L_p and the aileron control effectiveness L_{δ_a} to design the two fixed-gain model reference controllers, (7.3) and (7.9).

In reality, aerodynamic parameters are rarely known exactly. This type of uncertainty is called parametric. If the true parameters are substantially different from their assumed constant values, controllers such as (7.9) can lead to instabilities in the system. Even when the system remains stable in the presence of parametric uncertainties, its closed-loop tracking performance may deteriorate to a point of becoming unacceptable.

Robustness considerations may not always solve the parameter sensitivity problem. Often, robust controllers will have a conservatism built into their design and as such, they may not be able to provide adequate tracking performance, when operating under specific parametric uncertainties. This leads to the idea of adding a gain adaptation mechanism and arriving at model reference adaptive controllers.

Example 7.2 Model Reference Adaptive Control of Aircraft Roll Dynamics Suppose that the two aerodynamic parameters, L_p and L_{δ_a} , in the roll dynamics (7.1) are constant but otherwise completely unknown, with the exception that we do know the sign of the aileron control effectiveness L_{δ_a} , (it is positive for a conventional aircraft). The control task remains the same as in Example 7.1—we need to find δ_a such that p tracks p_{ref} , which in turn is driven by a bounded possibly time-varying command p_{cmd} .

The main control challenge here is to achieve the desired closed-loop tracking performance, specified by the reference model (7.2), while operating in the presence of constant parametric uncertainties L_p and L_{δ_a} .

In the forthcoming chapters, we will exploit Lyapunov-based methods that allow us to design adaptive controllers with formal guarantees of closed-loop stability, boundedness, and tracking performance. In the meantime, we shall outline main ideas in the design of adaptive systems.

If we knew the roll dynamics model parameters, then a feedback–feedforward controller in the form similar to (7.3)

$$\delta_a = k_p p + k_{p_{\text{cmd}}} p_{\text{cmd}} \quad (7.12)$$

would have solved the tracking problem. Since the system parameters are unknown, the ideal controller gains, k_p and $k_{p_{\text{cmd}}}$, cannot be computed directly as in Example 7.1. Instead, we consider an adaptive controller in the form,

$$\delta_a = \hat{k}_p p + \hat{k}_{p_{\text{cmd}}} p_{\text{cmd}} \quad (7.13)$$

where $(\hat{k}_p, \hat{k}_{p_{\text{cmd}}})$ represent the estimated feedback and feedforward gains, in that order. Substituting (7.13) into (7.1), gives the closed-loop system.

$$\dot{p} = (L_p + L_{\delta_a} \hat{k}_p) p + (L_{\delta_a} \hat{k}_{p_{\text{cmd}}}) p_{\text{cmd}} \quad (7.14)$$

Using parameterization (7.13), the reference model dynamics (7.2) can be equivalently written in terms of the ideal unknown gains, as

$$\dot{p}_{\text{ref}} = \underbrace{(L_p + L_{\delta_a} k_p)}_{a_{\text{ref}}} p_{\text{ref}} + \underbrace{(L_{\delta_a} k_{p_{\text{cmd}}})}_{b_{\text{ref}}} p_{\text{cmd}} \quad (7.15)$$

We now define the gain estimation errors,

$$\Delta k_p = \hat{k}_p - k_p, \quad \Delta k_{p_{\text{cmd}}} = \hat{k}_{p_{\text{cmd}}} - k_{p_{\text{cmd}}} \quad (7.16)$$

and rewrite the closed-loop system (7.14) in the following form

$$\dot{p} = \underbrace{(L_p + L_{\delta_a} k_p)}_{a_{\text{ref}}} p + \underbrace{(L_{\delta_a} k_{p_{\text{cmd}}})}_{b_{\text{ref}}} p_{\text{cmd}} + L_{\delta_a} (\Delta k_p p + \Delta k_{p_{\text{cmd}}} p_{\text{cmd}}) \quad (7.17)$$

Subtracting (7.15) from (7.17), gives the tracking error dynamics.

$$\dot{e} = a_{\text{ref}} e + L_{\delta_a} (\Delta k_p p + \Delta k_{p_{\text{cmd}}} p_{\text{cmd}}) \quad (7.18)$$

There are three error signals in the error dynamics (7.18): (1) the roll-rate tracking error e ; (2) the feedback gain estimation error Δk_p , and (3) the feedforward gain estimation error $\Delta k_{p_{\text{cmd}}}$. We are going to devise adaptive laws for changing the gains $(\hat{k}_p, \hat{k}_{p_{\text{cmd}}})$, such that all these three errors tend to zero, globally and asymptotically.

In order to do that, we first define a scalar function V , representative of the total “kinetic energy” of all the errors in the system.

$$V(e, \Delta k_p, \Delta k_{p_{\text{cmd}}}) = \frac{e^2}{2} + \frac{|L_{\delta_a}|}{2 \gamma_p} \Delta k_p^2 + \frac{|L_{\delta_a}|}{2 \gamma_{p_{\text{cmd}}}} \Delta k_{p_{\text{cmd}}}^2 \quad (7.19)$$

The “energy” function represents a weighted sum of squares of all the errors in the system. This is the so-called Lyapunov function candidate, and the positive constant scalar weights $(\gamma_p, \gamma_{p_{\text{cmd}}})$ will eventually become the rates of adaptation. We can easily evaluate the time derivative of V .

$$\dot{V}(e, \Delta k_p, \Delta k_{p_{\text{cmd}}}) = e \dot{e} + \frac{|L_{\delta_a}|}{\gamma_p} \Delta k_p \dot{\hat{k}}_p + \frac{|L_{\delta_a}|}{\gamma_{p_{\text{cmd}}}} \Delta k_{p_{\text{cmd}}} \dot{\hat{k}}_{p_{\text{cmd}}} \quad (7.20)$$

This is the system “power”. Substituting (7.18) into (7.20), yields the time derivative of V , along the trajectories of the error dynamics (7.18), but without explicit knowledge of these trajectories.

$$\begin{aligned} \dot{V}(e, \Delta k_p, \Delta k_{p_{\text{cmd}}}) &= a_{\text{ref}} e^2 + e L_{\delta_a} (\Delta k_p p + \Delta k_{p_{\text{cmd}}} p_{\text{cmd}}) \\ &\quad + \frac{|L_{\delta_a}|}{\gamma_p} \Delta k_p \dot{\hat{k}}_p + \frac{|L_{\delta_a}|}{\gamma_{p_{\text{cmd}}}} \Delta k_{p_{\text{cmd}}} \dot{\hat{k}}_{p_{\text{cmd}}} \end{aligned} \quad (7.21)$$

Rearranging terms, we further get

$$\dot{V}(e, \Delta k_p, \Delta k_{p_{\text{cmd}}}) = a_{\text{ref}} e^2 + \Delta k_p |L_{\delta_a}| \left(\text{sgn}(L_{\delta_a}) p e + \frac{\dot{\hat{k}}_p}{\gamma_p} \right)$$

$$+ \Delta k_{p_{\text{cmd}}} |L_{\delta_a}| \left(\text{sgn}(L_{\delta_a}) p_{\text{cmd}} e + \frac{\dot{k}_{p_{\text{cmd}}}}{\gamma_{p_{\text{cmd}}}} \right) \quad (7.22)$$

We want the energy function V to dissipate in time. It is then sufficient to require that its derivative \dot{V} (the system power) be non-positive, when evaluated along the system trajectories. The non-positivity of \dot{V} can be easily achieved if we select the following adaptive laws,

$$\begin{aligned} \dot{\hat{k}}_p &= -\gamma_p p e \text{sgn}(L_{\delta_a}) \\ \dot{\hat{k}}_{p_{\text{cmd}}} &= -\gamma_{p_{\text{cmd}}} p_{\text{cmd}} e \text{sgn}(L_{\delta_a}) \end{aligned} \quad (7.23)$$

or, equivalently

$$\begin{aligned} \dot{\hat{k}}_p &= -\gamma_p p e \\ \dot{\hat{k}}_{p_{\text{cmd}}} &= -\gamma_{p_{\text{cmd}}} p_{\text{cmd}} e \end{aligned} \quad (7.24)$$

thus, making the second and the third terms in (7.22) disappear. Then,

$$\dot{V}(e, \Delta k_p, \Delta k_{p_{\text{cmd}}}) = a_{\text{ref}} e^2 \leq 0 \quad (7.25)$$

and consequently, the system kinetic energy V is a non-increasing function of time. This fact immediately implies that all the signals in the error dynamics (7.18), such as $(e, \Delta k_p, \Delta k_{p_{\text{cmd}}})$, are bounded functions of time. Furthermore, since the ideal gains $(k_p, k_{p_{\text{cmd}}})$ are constant, the adaptive gains $(\hat{k}_p, \hat{k}_{p_{\text{cmd}}})$ are also bounded.

The stable (by design) reference model (7.2), when driven by a bounded command p_{cmd} , gives a bounded output p_{ref} . Also, e was proven to be bounded. Then the roll rate p is bounded. Consequently, the control input δ_a in (7.13) and the roll acceleration \dot{p} in the system dynamics (7.1) are bounded. Furthermore, since \dot{p}_{ref} is bounded then \dot{e} is bounded, and so

$$\ddot{V}(e, \Delta k_p, \Delta k_{p_{\text{cmd}}}) = 2 a_{\text{ref}} e \dot{e} \quad (7.26)$$

is a uniformly bounded function of time. The latter implies that \dot{V} is a uniformly continuous function of time.

By the definition (7.19), $V \geq 0$ and because of (7.25), V is a non-increasing function of time. Therefore, V tends to a limit as $t \rightarrow \infty$, where the function limiting value may not necessarily be zero.

We have shown that $0 \leq \lim_{t \rightarrow \infty} V(e(t), \Delta k_p(t), \Delta k_{p_{\text{cmd}}}(t)) < \infty$ and \dot{V} is uniformly continuous. According to Barbalat's lemma (see Chap. 8), these two facts imply that the system power \dot{V} in (7.25), asymptotically tends to zero, which in turn means

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (7.27)$$

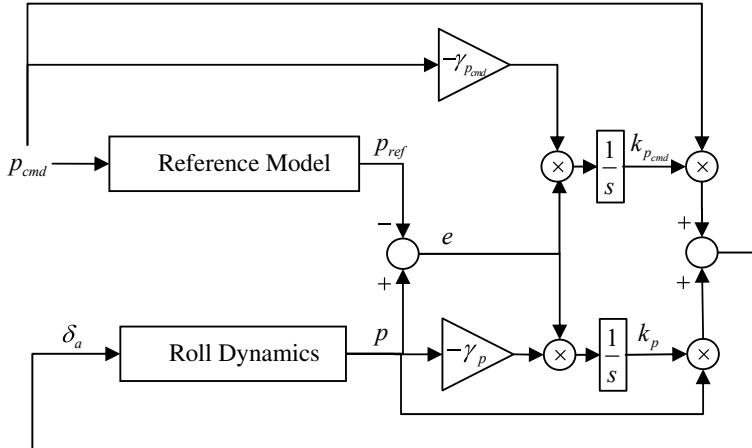


Fig. 7.4 Model reference adaptive controller obtained in Example 7.2

Thus, the adaptive controller (7.13), along with the adaptive laws (7.24), forces p track its reference signal p_{ref} asymptotically and for any initial conditions (globally). At the same time, all signals in the corresponding closed-loop system remain uniformly bounded. These arguments prove closed-loop stability and tracking performance of the closed-loop system with the adaptive controller. The corresponding block diagram is shown in Fig. 7.4.

As seen from the figure, the closed-loop system is comprised of the original roll dynamics (7.1) operating under the adaptive controller (7.13), with the reference model dynamics (7.2), and using the adaptive laws (7.24). Here, the external input is the roll-rate command p_{cmd} .

$$\begin{aligned}\dot{p} &= \left(L_p + L_{\delta_a} \hat{k}_p \right) p + L_{\delta_a} \hat{k}_{p_{\text{cmd}}} p_{\text{cmd}} \\ \dot{p}_{\text{ref}} &= a_{\text{ref}} p_{\text{ref}} + b_{\text{ref}} p_{\text{cmd}} \\ \dot{\hat{k}}_p &= -\gamma_p p(p - p_{\text{ref}}) \\ \dot{\hat{k}}_{p_{\text{cmd}}} &= -\gamma_{p_{\text{cmd}}} p_{\text{cmd}}(p - p_{\text{ref}})\end{aligned}\quad (7.28)$$

Equivalently, this system can be written in terms of the tracking and parameter estimation errors.

$$\begin{aligned}\dot{e} &= (a_{\text{ref}} + L_{\delta_a} \Delta k_p) e + L_{\delta_a} (\Delta k_p p_{\text{ref}} + \Delta k_{p_{\text{cmd}}} p_{\text{cmd}}) \\ \frac{d}{dt}(\Delta k_p) &= -\gamma_p (e + p_{\text{ref}})e \\ \frac{d}{dt}(\Delta k_{p_{\text{cmd}}}) &= -\gamma_{p_{\text{cmd}}} p_{\text{cmd}}e\end{aligned}\quad (7.29)$$

If instead of command tracking, the state regulation is of interest, then $p_{\text{ref}} = p_{\text{cmd}} = 0$, and so $\hat{k}_{p_{\text{cmd}}} = k_{p_{\text{cmd}}} = 0$. In this case, the closed-loop systems (7.28) and (7.29) simplify to the following time-invariant second-order inherently nonlinear dynamics,

$$\begin{aligned}\dot{p} &= \left(L_p + L_{\delta_a} \hat{k}_p \right) p \\ \dot{\hat{k}}_p &= -\gamma_p p^2\end{aligned}\tag{7.30}$$

These relations reveal the essential mechanism of adaptive control. The time-varying adaptive feedback gain $\hat{k}_p(t)$ will monotonically decrease its value until $(L_p + L_{\delta_a} \hat{k}_p)$ becomes negative, and as a result, the roll rate $p(t)$ will asymptotically converge to zero. In (7.30), the constant $\gamma_p > 0$ defines the rate of adaptation in the sense that large values of γ_p will force the adaptive gain $\hat{k}_p(t)$ to decrease faster.

In summary, using energy-based arguments we have shown that the adaptive controller (7.12), (7.24) provides the desired model reference-based closed-loop tracking performance for the system (7.1), while operating in the presence of the parametric uncertainties (L_p, L_{δ_a}) . ■

7.3 Direct Model Reference Adaptive Control of Scalar Linear Systems with Parametric Uncertainties

Let us now generalize and summarize the results obtained in Example 7.2, while restating them for a generic class of scalar linear time-invariant uncertain systems in the form,

$$\dot{x} = a x + b u\tag{7.31}$$

where $x \in R$ is the systems state, $u \in R$ is the control input, and (a, b) represent the parametric uncertainties, (constant and unknown), with the known $\text{sgn } b$.

First, we choose the desired reference model,

$$\dot{x}_{\text{ref}} = a_{\text{ref}} x_{\text{ref}} + b_{\text{ref}} r\tag{7.32}$$

with $a_{\text{ref}} < 0$. This model is driven by any bounded, possibly time varying, reference command r . The model parameters $(a_{\text{ref}}, b_{\text{ref}})$ must be chosen such that x_{ref} tracks r , with the designer specified criteria. For example, one might set $b_{\text{ref}} = -a_{\text{ref}}$ in order to enforce the unity DC gain from r to x_{ref} . Also, the value of $|a_{\text{ref}}|$ can be chosen such that the desired inverse time constant of the reference model is achieved.

Second, we define the model reference adaptive controller, as a linear combination of feedback and feedforward terms,

$$u = \hat{k}_x x + \hat{k}_r r \quad (7.33)$$

where (\hat{k}_x, \hat{k}_r) are the two adaptive gains, whose adaptive law dynamics are constructed similar to (7.24).

$$\begin{aligned} \dot{\hat{k}}_x &= -\gamma_x x(x - x_{\text{ref}}) \text{sgn}(b) \\ \dot{\hat{k}}_r &= -\gamma_r r(x - x_{\text{ref}}) \text{sgn}(b) \end{aligned} \quad (7.34)$$

In (7.34), positive scalars (γ_x, γ_r) are called the rates of adaptation. The larger their values, the faster the system will adapt to the parametric uncertainties.

This particular controller is called “direct” to indicate that the controller gains are adapted in (7.34) directly in order to enforce the desired closed-loop tracking performance. Alternatively, indirect adaptive controllers can be designed to estimate the unknown plant parameters (a, b) online, and then use their estimated values to calculate controller gains.

Finally, using energy-based arguments, we can formally prove that the adaptive controller (7.33), (7.34) provides the desired closed-loop tracking performance, in the sense that the system state x globally asymptotically tracks the state x_{ref} of the reference model (7.32), while keeping all signals in the corresponding closed-loop dynamics uniformly bounded in time.

A few immediate remarks are in order.

- The direct model reference adaptive controller (7.33), (7.34) operates using only available (online measured) signals in the system. The latter consists of: (a) the system state x ; (b) the state of the reference model x_{ref} ; (c) the tracking error $e = x - x_{\text{ref}}$, and (d) The sign of the control effectiveness $\text{sgn } b$.
- All signals in the closed-loop system remain uniformly bounded in time.
- The system state x tracks the state of the reference model x_{ref} , globally and asymptotically. However, a characterization of the system transient dynamics in model reference adaptive control remains an open problem.
- The adaptive parameters (\hat{k}_x, \hat{k}_r) are not guaranteed to converge to their true unknown values (k_x, k_r) , nor they are assured to converge to constant values in any way. All that is known is that these parameters remain uniformly bounded in time. Sufficient conditions for parameter convergence are known as persistency of excitation [1, 2]. It turns out that for a first-order linear system such as (7.1), persistent excitation is guaranteed if the commanded signal $r(t)$ contains at least one sinusoidal component. In this case, the two adaptive gains (\hat{k}_x, \hat{k}_r) will converge to their true constant unknown values, exponentially fast.

7.4 Historical Roots and Foundations of Model Reference Adaptive Control

The adaptive control development was largely motivated in the early 1950s by the design of autopilots for aircraft that operated in a wide flight envelope, with a large range of speeds and altitudes. Different flight conditions caused the aircraft dynamics to change significantly. This phenomenon called for flight controllers that could accommodate drastic changes in the aircraft aerodynamic and propulsive forces and moments. Adaptive control was proposed as one of the design approaches to solving the flight control problem.

The concept of a model reference adaptive system (MRAS) was originally proposed in 1958 by Whitaker et al. at MIT [3, 4]. The main idea behind this concept was to specify the desired command-to-output performance of a servo-tracking system using a differential or a difference equation (the reference model) that would define the ideal response of the system due to external commands. This control concept was later called “explicit model following” and the corresponding architecture became known as the model reference adaptive control, (MRAC).

Shortly after its introduction, the first proof of MRAC closed-loop stability using Lyapunov theory was given in 1965 by Butchart and Shackcloth, at the IFAC Symposium on Adaptive Control [5], and in 1966 by Parks [6].

In the following years, adaptive control theory for a broad class of multi-input–multi-output uncertain dynamical systems was extensively developed and well documented in several now-classical textbooks [1, 2, 7, 8].

7.5 Exercises

Exercise 7.1 Consider the aircraft roll dynamics from Example 7.1. Given the roll damping $L_p = -0.8 (\text{s}^{-1})$ and the aileron effectiveness $L_{\delta_a} = 1.6 (\text{s}^{-1})$, design a fixed-gain model reference controller in the form of (7.3) to recover the reference model dynamics (7.2), with $a_{\text{ref}} = -2$, $b_{\text{ref}} = 2$. Also design a fixed-gain controller with error feedback in the form of (7.9). Choose several bounded time-varying roll-rate commands. Simulate the closed-loop system response, with each of the two controllers active (one at a time). Compare the two controllers and comment on the achieved closed-loop system stability, robustness, tracking, and transient properties.

Exercise 7.2 Derive relations (7.28), (7.29), and (7.30).

Exercise 7.3 Assume that the constant roll dynamics data (L_p, L_{δ_a}) from Example 7.1 are unknown, and that only the sign of L_{δ_a} is known to be positive. Using the same reference model parameters, design an adaptive roll-rate tracking controller in the form of (7.13), (7.23). Choose various roll-rate commands and simulate the resultant closed-loop system performance. Compare fixed-gain versus adaptive controller performances and comment on your results.

Exercise 7.4 Consider a scalar dynamical system described by the first-order differential equation

$$\dot{x} = a x + b u, \quad x(0) = x_0$$

where $a = 2$ and $b = 3$ represent unknown constant parameters. It is assumed that $\text{sgn } b = 1$ is known. The goal is to design a controller such that the system state tracks the state of the reference model,

$$\dot{x}_{\text{ref}} = r(t) - x_{\text{ref}}$$

where $r = r(t)$ is the commanded reference input (a bounded signal). Assuming that the system dynamics are known, design a fixed-gain command tracking controller. Then design a direct model reference adaptive controller. Simulate the closed-loop system dynamics for both controllers, starting from different initial conditions and using three different reference commands: (a) a step-input; (b) a series of steps, and (c) a sum of sinusoids. Tune your adaptive design (i.e., select rates of adaptation). Compare tracking performance of the two closed-loop systems and their corresponding control signals. Comment on your results.

References

1. Narendra KS, Annaswamy AM (2005) Stable adaptive control. Dover, New York
2. Ioannou P, Fidan P (2006) Adaptive control tutorial, SIAM, advances in design and control. SIAM, PA
3. Whitaker HP, Yamron J, Kezer A (1958) Design of model reference adaptive control systems for aircraft, Report R-164. Instrumentation Laboratory, MIT, Cambridge, MA
4. Whitaker HP, Osburn PV, Kezer A (1961) New developments in the design of model reference adaptive control systems, paper 61-39. Institute of the Aerospace Sciences, Easton, PA
5. Butchart RL, Shackcloth B (1965) Synthesis of model reference adaptive control systems by Lyapunov's second method. In: Proceedings 1965 IFAC symposium on adaptive control, Teddington, UK
6. Parks PD (1966) Lyapunov redesign of model reference adaptive systems. IEEE Trans Autom Control 11:362–367
7. Slotine JJE, Li W (1995) Applied nonlinear control. Prentice Hall, New Jersey
8. Krstic M, Kanellakopoulos I, Kokotovic P (1995) Nonlinear and adaptive control design. Wiley, New York



Lyapunov Stability of Motion

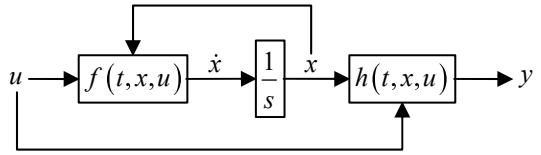
8

The main intent of this chapter is to introduce the essential mathematical tools for stability analysis of continuous finite-dimensional dynamical systems. We begin with an overview of sufficient conditions to guarantee existence and uniqueness of the system solutions, followed by a collection of Lyapunov-based methods for studying stability of the system equilibria and trajectories. The beginning of what is known today as the Lyapunov stability theory can be traced back to the original publication of Alexander Mikhailovich Lyapunov's doctoral thesis "On the general problem of the stability of motion", which he defended at the University of Moscow in 1892. Our interest and emphasis on the Lyapunov's stability methods stem from the fact that these methodologies lay out the much-needed theoretical framework and the foundation for performing design and analysis of adaptive controllers. In this chapter, we review selected (but not inclusive) methods due to Lyapunov. This selection is primarily driven by our interest in justifying the design of stable model reference adaptive controllers, with predictable and guaranteed closed-loop performance, for a wide class of nonlinear nonautonomous multi-input–multi-output dynamical systems.

8.1 Dynamical Systems

A dynamical system may be thought of as a collection of finite or infinite number of interconnected and time-dependent components. The system evolution is driven by an environment where the system operates. When subjected to an external time-dependent input $u(t)$, the system generates an output $y(t)$, which in turn may explicitly depend on the system internal properties, defined by the system states

Fig. 8.1 State-space model
block diagram



$x(t)$. The states describe the system inner-component connections, their dynamical response due to environmental stimulus, and their contributions to the system response.

In what follows, we consider a special class of dynamical systems that can be modeled by a finite number of coupled scalar ordinary differential equations in the form

$$\dot{x} = f(t, x, u) \quad (8.1)$$

In (8.1), $t \in R^+$ denotes time and $f : R \times R^n \times R^m \rightarrow R^n$ is a vector-function. We call (8.1) the system dynamics, refer to $x \in R^n$ as the system state at time t , and define $u \in R^m$ as the control input, (an externally supplied signal). The number of state components n is called the order of the system.

A solution $x(t)$ of (8.1) (if one exists) corresponds to a curve in the system state space R^n , as t varies from an initial time t_0 to infinity. This curve is often referred to as the system state trajectory. Later in this chapter, we will formulate sufficient conditions guaranteeing existence and uniqueness of solutions for dynamical systems such as (8.1), starting from a given set of initial conditions $x(t_0) = x_0$.

In addition to the system dynamics (8.1), a set of algebraic equations may also be given,

$$y = h(t, x, u) \quad (8.2)$$

where $h : R \times R^n \times R^m \rightarrow R^p$ and $y \in R^p$. This is the system output. Together, Eqs. (8.1) and (8.2) form the system state-space model, whose block diagram is shown in Fig. 8.1.

A special case of (8.1) and (8.2) is the linear-in-control system,

$$\begin{aligned} \dot{x} &= f(t, x) + g(t, x) u \\ y &= h(t, x) + d(t, x) u \end{aligned} \quad (8.3)$$

where the functions g and d are of matching dimensions.

Letting $x = (x_1 \ x_2 \ \dots \ x_n)^T$, a particular class of nonlinear continuous-time dynamics is given by the systems in Brunovsky canonical form.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\dots \\ \dot{x}_n &= f(x) + g(x) u \\ y &= h(x) \end{aligned} \quad (8.4)$$

For linear time-variant (LTV) systems, the state-space model (8.1) and (8.2) is

$$\begin{aligned}\dot{x} &= A(t)x + B(t)u \\ y &= C(t)x + D(t)u\end{aligned}\tag{8.5}$$

Finally, the class of linear time-invariant (LTI) systems is written in the familiar form,

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{8.6}$$

whose dynamic properties can be completely characterized by the matrix quadruple (A, B, C, D) .

If the model (8.1) does not contain the control input signal u

$$\dot{x} = f(t, x)\tag{8.7}$$

then the resulting dynamics are called “unforced”. If in addition, the function f does not depend explicitly on t , that is if

$$\dot{x} = f(x)\tag{8.8}$$

then the system unforced dynamics are called autonomous or time-invariant. Systems that explicitly depend on time are nonautonomous (i.e., time-variant).

8.2 Existence and Uniqueness of Solutions

Suppose that we initialize the state of the system (8.7),

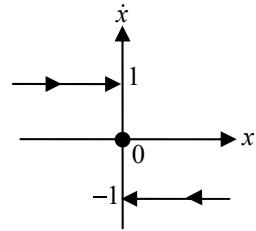
$$x(t_0) = x_0 \in R^n\tag{8.9}$$

at a time instant $t_0 \geq 0$. Together, (8.7)–(8.9) define the Cauchy Problem, or equivalently, the Initial Value Problem (IVP), whose solutions may or may not exist. Moreover, when a solution does exist, it may or may not be unique.

Besides theoretical demands, the questions of existence and uniqueness become quite important for practitioners in simulation, dynamics, and control. For example, if the system (8.7) is constructed to emulate a real process that starts from an initial condition x_0 , we need to know if and when the system unique solution would exist. Otherwise, the resulting simulation data may lead to erroneous conclusions about the underlying process dynamics and control.

Contrary to LTI systems (8.6), existence and uniqueness of solutions for nonlinear equations (8.7) are not always guaranteed. In order to motivate our discussion, we consider several examples.

Fig. 8.2 Phase portrait of the system dynamics in Example 8.1



Example 8.1 The scalar nonlinear dynamics

$$\dot{x} = -\operatorname{sgn} x$$

has the discontinuous (at the origin) right half side, which is defined by the sign function.

$$\operatorname{sgn} x = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

The system phase portrait is easy to draw, and is given in Fig. 8.2.

These data indicate that the system trajectories asymptotically approach the origin from the left or the right side, depending on whether the initial conditions are negative or positive, respectively. The rate of change of the “kinetic energy” for this system is non-positive.

$$\frac{d}{dt} \left(\frac{x^2(t)}{2} \right) = x(t) \dot{x}(t) = -x(t) \operatorname{sgn} x(t) = -|x(t)| \leq 0$$

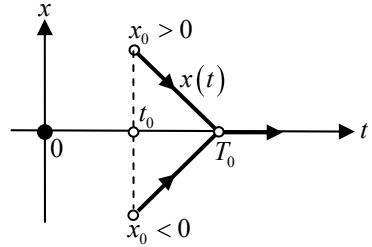
Therefore, the kinetic energy must dissipate, and so it seems that the system trajectories should asymptotically approach the origin, where $\operatorname{sgn} 0 = 0$ by the definition. To further investigate the system behavior, we can integrate the system dynamics on the interval from t_0 to t .

$$x(t) = x_0 - (t - t_0) \operatorname{sgn} x(t)$$

Even though this equation is implicit in x , we can easily sketch its solutions versus time (Fig. 8.3).

First, we note that every solution arrives at zero in finite time $T_0 = t_0 + |x_0|$, and it remains zero for all future times. Second, for every solution with $x_0 > 0$ there is the solution that starts at $(-x_0)$, and it meets the former solution at the same exact time T_0 . Third, the system trajectories are not continuously differentiable at T_0 . In fact, for all $t \geq T_0$ and as the system trajectory evolves along the t -axis, the system solutions will “jitter”. This interesting phenomenon is solely caused by the discontinuity of the system dynamics at the origin. ■

Fig. 8.3 System trajectories in Example 8.1



Example 8.2 Let $k > 0$ be a real number and consider the IVP

$$\dot{x} = x^k, \quad x(t_0) = x_0$$

Using separation of variables, we can write the solution of this system,

$$x^{1-k}(t) = x_0^{1-k} + (1 - k)(t - t_0)$$

and make several observations.

- For $0 < k < 2$, the system does not have solutions that start at $x_0 < 0$.
- Suppose $k = 2$, $x_0 = 1$, and $t_0 = 0$. Then $x(t) = \frac{1}{1-t}$. This solution grows unbounded “blows up” in finite time $T = 1$, and it is not defined for $t \geq T$.
- For $k = \frac{2}{3}$ and $x_0 = t_0 = 0$, the IVP has not one but two solutions: $x(t) = \frac{t^3}{27}$ and $x(t) \equiv 0$. Let $T > 0$ denote a constant. As it turns out, this IVP has infinitely many solutions.

$$x(t) = \begin{cases} 0, & 0 \leq t \leq T \\ \frac{1}{27}(t-T)^3, & t > T \end{cases}$$
■

All of the above examples imply that both existence and uniqueness of IVP solutions for the dynamical system (8.7) depend on certain properties of the vector-function $f(t, x)$.

We begin with a theorem that states sufficient conditions for the IVP problem to admit a solution which may not necessarily be unique [1].

Theorem 8.1 (Peano) If $f(t, x) : R \times R^n \rightarrow R^n$ is continuous in a closed region,

$$B = \{(t, x) : |t - t_0| \leq T, \|x - x_0\| \leq r\} \subseteq R \times R^n \quad (8.10)$$

where T, r are some strictly positive constants, and $\|\bullet\|$ is the Euclidean vector norm (see (8.12)), then there exists $t_0 < t_1 \leq T$ such that the IVP (8.7), (8.9) has at least one continuously differentiable solution $x(t)$ on the interval $[t_0, T]$. ■

The assumed continuity of $f(t, x)$ in its arguments ensures that there is at least one solution of the IVP. Note however that this theorem does not guarantee the uniqueness of the solution. The key constraint that yields uniqueness is the so-called Lipschitz condition, whereby $f(t, x)$ satisfies the inequality,

$$\|f(t, x) - f(t, y)\| \leq L \|x - y\| \quad (8.11)$$

for all (t, x) and (t, y) in some neighborhood of (t_0, x_0) with a finite constant $L > 0$. In (8.11) and everywhere else throughout the book, $\|x\|$ denotes the Euclidean vector norm of $x \in R^n$,

$$\|x\| = \begin{cases} \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, & 1 \leq p < \infty \\ \max_{1 \leq i \leq n} |x_i|, & p = \infty \end{cases} \quad (8.12)$$

Lipschitz-based sufficient conditions for the unique existence of IVP solutions are stated below and without proof [2].

Theorem 8.2 (Local Existence and Uniqueness) *Let $f(t, x) : R \times R^n \rightarrow R^n$ be piecewise continuous in t and satisfy the Lipschitz condition (8.11),*

$$\forall x, y \in B = \{x \in R^n : \|x - x_0\| \leq r\}, \quad \forall t \in [t_0, t_1] \quad (8.13)$$

Then, there exists some $\delta > 0$ such that the IVP for the state equation $\dot{x} = f(t, x)$ with $x(t_0) = x_0$ has a unique solution over $[t_0, t_0 + \delta]$. ■

Notice that the Lipschitz condition (8.11) is assumed to be valid locally in a neighborhood of (t_0, x_0) from a compact (closed and bounded) set B , as it is defined in (8.13).

We can try to extend the interval of existence and uniqueness over a given time interval $[t_0, t_0 + \delta]$ by taking $t_0 \triangleq t_0 + \delta$ as the new initial time and $x_0 \triangleq x(t_0 + \delta)$ as the new initial state. If the conditions of the theorem are satisfied at $(t_0 + \delta, x(t_0 + \delta))$ then there exists $\delta_2 > 0$ such that the IVP has a unique solution over $[t_0 + \delta, t_0 + \delta + \delta_2]$ that passes through the point $(t_0 + \delta, x(t_0 + \delta))$. We can now piece together the two solutions to establish the existence of a unique solution over the larger interval $[t_0, t_0 + \delta + \delta_2]$. This idea can be repeated to keep extending the IVP solution, arriving at the maximal IVP solution, which is defined on the maximal interval $[t_0, t_0 + \delta_{\max}]$, with finite or infinite δ_{\max} . It is interesting to note that if δ_{\max} is finite then the respective maximal solution tends to infinity [3], as the following example demonstrates.

Example 8.3 The unique solution of the scalar IVP,

$$\dot{x} = 1 + x^2, \quad x(0) = 0$$

is $x(t) = \tan t$. Its maximal interval of existence is finite with $\delta_{\max} = \frac{\pi}{2}$, and predictably, $\lim_{t \rightarrow \frac{\pi}{2}} x(t) \rightarrow \infty$, that is this solution becomes unbounded in finite time. ■

In process modeling applications, we are primarily interested in constructing IVP-s whose solutions are unique and exist for all $t \geq t_0$. The global uniqueness and existence requirements would ensure at least soundness of our models, but not necessarily their validity. The latter would have to be verified by correlating model data with the application process under consideration.

The next theorem states that if the system dynamics function f satisfies global Lipschitz conditions then the corresponding IVP has a unique solution over arbitrarily large time interval [2].

Theorem 8.3 (Global Existence and Uniqueness) *Suppose that a vector-function $f(t, x) : R \times R^n \rightarrow R^n$ is piecewise continuous in t and globally Lipschitz in x ,*

$$\|f(t, x) - f(t, y)\| \leq L \|x - y\|, \quad \forall x, y \in R^n, \quad \forall t \in [t_0, t_1] \quad (8.14)$$

with a finite constant $L > 0$. Then the IVP (8.7)–(8.9) has a unique solution over $[t_0, t_1]$, where the final time t_1 may be arbitrarily large. ■

We immediately note that the above stated global Lipschitz condition (8.14) is sufficient but not necessary as the next example shows.

Example 8.4 The system dynamics function in the scalar IVP,

$$\dot{x} = -x^3, \quad x(0) = x_0$$

is not globally Lipschitz, yet the system has the unique solution,

$$x(t) = \frac{x_0}{\sqrt{2x_0^2 t + 1}}$$

which is defined for any initial condition x_0 globally and for all time $t \geq 0$. ■

The next theorem is of particular interest to us. It presents sufficient conditions for extending IVP solutions indefinitely. Its detailed proof can be found in [2].

Theorem 8.4 (Global Existence and Uniqueness on Unbounded Time Interval) *Let $f(t, x) : R \times R^n \rightarrow R^n$ be piecewise continuous in t , locally Lipschitz in x for all $t \geq 0$ and all x in a domain $D \subset R^n$. Let $W \subset D$ be a compact subset of D , $x_0 \in W$, and suppose it is known that every solution of the IVP (8.7)–(8.9) lies entirely in W . Then there is a unique solution that is defined for all $t \geq t_0$.* ■

In the forthcoming chapters, we will use Lyapunov's methods to check if system trajectories evolve inside a compact set. We will be able to do that without solving the system differential equation. Lyapunov's analysis methods generalize and extend the notion of energy, from mechanical systems to generic dynamics. For adaptive systems, we will show that suitable energy functions can be formed as sum of squares of the system state components. Then we would compute the system power—the time derivative of the energy function, evaluated along the system trajectories. We will argue that if the system power is non-positive, that is if the system energy dissipates, then every trajectory is bounded and exists globally for all time.

For now, let us illustrate the energy-based analysis using the dynamics from Example 8.4. Toward that end, we shall utilize the system “kinetic” energy,

$$V(x) = \frac{x^2}{2}$$

and compute its time derivative along the system dynamics (the system power).

$$\frac{d}{dt} \left(\frac{x^2}{2} \right) = x \dot{x} = x (-x^3) = -x^4 \leq 0$$

Since the power function is non-positive then the energy must decrease and consequently, the system state must be bounded for all time. Therefore, and according to Theorem 8.4, the system dynamics must have a unique solution starting from any initial condition at $t_0 = 0$ and extending indefinitely, for all $t \geq 0$. ■

Energy-based methods and Theorem 8.4 become extremely useful especially when the system dynamics cannot be integrated to obtain its IVP solutions explicitly.

Example 8.5 Consider the autonomous scalar dynamics $\dot{x} = f(x)$, whose phase portrait is shown in Fig. 8.4.

We assume that $f(0) = f(a) = f(b) = 0$, $f(x) > 0$ for all $x < 0$, and that the function is locally Lipschitz in x . Other than that, the function shape and its

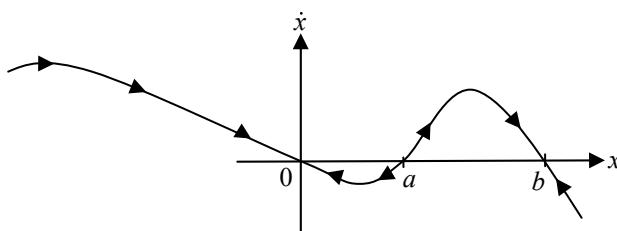


Fig. 8.4 Phase portrait of the system from Example 8.5

values are assumed to be completely unknown, and as such these dynamics cannot be analytically integrated to compute the system solutions in their explicit form.

We now pose several questions and give their answers to demonstrate that explicit knowledge of IVP solutions is not required at all in order to assess if the system unique solutions exist. In addition, we will also assess their interval of existence.

Question: Will the IVP with a nonzero initial condition $x(0) \neq 0$ has a unique solution?

Answer: Since f is locally Lipschitz then existence and uniqueness of the IVP solutions directly follows from Theorem 8.2.

Question: Is this solution defined for all time?

Answer: Anchored in Theorem 8.4, we can either employ energy-based arguments to show that all trajectories are bounded, or we can simply examine the system phase portrait shown in Fig. 8.4. Analyzing the latter, it becomes clear that starting from any nonzero initial condition, all trajectories of this system will enter the interval $[0, b]$ in finite time. Therefore, all these solutions are bounded, and because of Theorem 8.4, the IVP unique solutions are defined globally, for all $t \geq 0$. ■

We have surveyed and discussed several well-known theorems concerning existence and uniqueness of IVP solutions for nonautonomous continuous dynamical systems. Basically, existence of IVP solutions is provided if the system dynamics are continuous in its arguments. However, in order to guarantee uniqueness, we have called upon the Lipschitz assumption, (local or global). As it turns out, the Lipschitz condition, even when local, is quite restrictive, since the set of all Lipschitz-continuous functions represents a very small (called “meager”) subset of all continuous functions. Such an observation might lead to a conjecture that only a very small set of IVP-s have unique solutions. Fortunately, this conjecture is incorrect. In 1932, the Polish mathematician Witold Orlicz proved that the set of all functions for which IVP-s have unique solutions is very large, (a complement of a meager set). Orlicz’ theorem states that “almost all” differential equations with continuous right-hand sides have unique solutions. On the other hand, the set of IVP-s, for which we can formally characterize uniqueness of their solutions, is “almost nothing”. This compelling argument suggests that there are very many classes of non-Lipschitz IVP-s with unique solutions that are yet to be discovered. Further details on the subject and the proof of Orlicz’ theorem can be found in [3, Appendix A].

8.3 System Equilibrium

One of the central concepts in systems and control theory is the concept of an equilibrium point. We will focus our discussions on nonautonomous unforced dynamical systems,

$$\dot{x} = f(t, x) \quad (8.15)$$

with the vector-function $f : [0, \infty) \times D \rightarrow R^n$ which is piecewise continuous in t and locally Lipschitz in x , and with a domain $D \subset R^n$ that contains the origin $x = 0$.

Definition 8.1 The origin in R^n is an equilibrium point for the unforced nonautonomous system (8.15) at $t_0 = 0$, if

$$f(t, 0) = 0, \quad \forall t \geq 0 \quad (8.16)$$

It is not difficult to show that there is no loss of generality in using the origin and the zero initial time in the definition above. In fact, suppose we define a nonzero vector $x^* \in R^n$ to be an equilibrium point of (8.15) at a nonzero initial time $t = t_0$.

$$f(t, x^*) = 0, \quad \forall t \geq t_0$$

We can redefine time $\tau = t - t_0$, introduce the new state,

$$z(\tau) = x(\tau + t_0) - x^*$$

and arrive at the transformed system dynamics,

$$\boxed{\frac{d z(\tau)}{d \tau}} = \frac{d x(\tau + t_0)}{d t} = f(\tau + t_0, z(\tau) + x^*) = \boxed{g(\tau, z(\tau))}$$

with $g(0, 0) = f(t_0, x^*) = 0$. Thus, we have shifted the equilibrium point to the origin, and the initial time to zero.

This idea can be further generalized. Suppose that we are given a trajectory $x^*(t)$ of (8.15) that starts at $t = t_0$.

$$\dot{x}^*(t) = f(t, x^*(t)), \quad t \geq t_0$$

We can again redefine time $y(\tau) = x(\tau + t_0) - x^*(\tau + t_0)$, introduce the new state,

$$z(\tau) = x(\tau + t_0) - x^*(\tau + t_0)$$

and rewrite the system dynamics,

$$\begin{aligned} \left[\frac{dz(\tau)}{d\tau} \right] &= \frac{dx(\tau + t_0)}{dt} - \frac{dx^*(\tau + t_0)}{dt} = f(\tau + t_0, z(\tau) + x(\tau + t_0)) \\ &\quad - f(\tau + t_0, z(\tau) + x^*(\tau + t_0)) = \boxed{g(\tau, z(\tau))} \end{aligned}$$

with $g(0, 0) = 0$. Consequently, analyzing the redefined dynamics around the origin, as an equilibrium point, while starting at $t_0 = 0$, allows to determine the original system behavior around the original nonzero equilibrium x^* . This modification also allows us to assess the system relative dynamics with respect to a time-dependent trajectory $x^*(t)$, starting at an arbitrary initial time instant $t_0 \geq 0$.

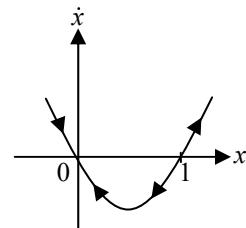
A dynamical system can have multiple equilibrium points. Some of these equilibrium points might be isolated from each other, whilst others might form a continuum of equilibrium points. In either case, it is worth noting that whenever the system starts at an equilibrium point, it will remain there for all future times. The converse is also true and can be formally proven.

Example 8.6 The LTI system $\dot{x} = Ax$ has an isolated equilibrium point at $x = 0$ if and only if $\det A \neq 0$, (A has no zero eigenvalues). Otherwise, the system has a continuum of equilibrium points. These are the only possible equilibrium patterns that a linear time-invariant system may have. ■

Example 8.7 A nonlinear system can have multiple isolated equilibrium points. Consider the Bernoulli equation $\dot{x} = x(x - 1)$. It has two isolated equilibrium points, $x^* = 0$ and $x^* = 1$. The system phase portrait is shown in Fig. 8.5.

Clearly, all trajectories that start in the open interval $(-\infty, 1)$ will converge to the origin, while all other trajectories will diverge to $+\infty$. This phenomenon is typical for nonlinear dynamics, where depending on the initial conditions, the system exhibits completely different behaviors. It is also clear that the system equilibrium at the origin is asymptotically stable (formal definition will be given later) in the sense that all trajectories that start in the open interval $(-\infty, 1)$ will converge back to the origin without leaving the interval. The other equilibrium $x^* = 1$ is unstable, meaning that there are trajectories that start arbitrarily close to 1, yet they move away from this equilibrium point. ■

Fig. 8.5 Phase portrait of a Bernoulli equation from Example 8.7



8.4 Lyapunov Stability Definitions

The concept of Lyapunov stability is one of the most prominent and fundamental in dynamics and control. It is primarily concerned with analyzing behavior of system trajectories near equilibrium, but without explicit computation of those solutions.

Theoretical foundations of what is known today as the Lyapunov stability theory are due to the Russian mathematician Alexander Mikhailovich Lyapunov (1857–1918). In 1892 at the University of Moscow, Lyapunov presented and subsequently defended his doctoral thesis “On the general problem of the stability of motion”, where he had introduced basic definitions and fundamental theorems for studying the stability of solutions for a broad class of differential equations.

In 1908, Lyapunov’s work was translated into French, reprinted by Princeton University Press in 1947, and gained wide acceptance in the West in the 1960’s. Today, Lyapunov stability theory represents an indispensable tool that enables engineers and scientists analyze nonlinear systems and design controllers with stable and predictable performance.

System stability can be interpreted as a continuity of the system trajectories, with respect to initial conditions, over infinite time interval. The key words here are “over infinite time interval”. They highlight the difference between the notions of the stability and continuity on initial conditions. It is well-known that solutions of Lipschitz-continuous differential equations continuously depend on the system initial conditions [1–3]. However, the notion of stability requires that this continuity property holds infinitely in time.

Let $x(t; x_0)$ denote a solution of (8.15) with the initial condition $x(t_0) = x_0$. Suppose that this solution is unique and exists on a finite, possibly open-ended, interval $[t_0, T)$. The continuity property of $x(t; x_0)$ due to changes in x_0 can be described as follows: Given any positive constant $\varepsilon > 0$, there must exist a sufficiently small positive constant $\delta > 0$, such that for all perturbed initial conditions $x_0 + \Delta x_0$ with $|\Delta x_0| \leq \delta$, the corresponding perturbed solution $x(t; x_0 + \Delta x_0)$ deviates from the original by no more than ε , that is $\|x(t; x_0 + \Delta x_0) - x(t; x_0)\| \leq \varepsilon$, for all $t_0 \leq t < T$. Figure 8.6 illustrates the continuity property for a scalar system.

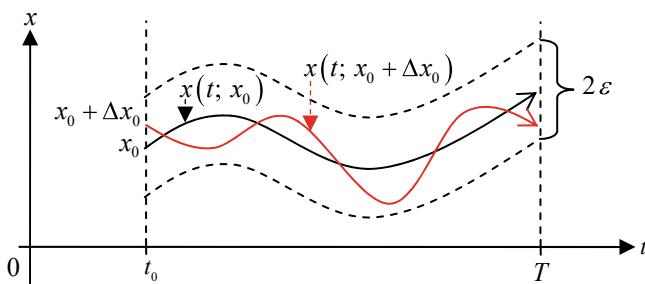


Fig. 8.6 Continuity of system solutions with respect to initial conditions

On the finite interval $[t_0, T]$, the perturbed trajectory $x(t; x_0 + \Delta x_0)$ will evolve within the (2ϵ) -strip relative to the original unperturbed solution $x(t; x_0)$, as long as the perturbed initial condition $(x_0 + \Delta x_0)$ is located within the (2δ) -strip of x_0 .

In practice, we are often interested in analyzing system solutions that are defined on an infinite interval $[t_0, \infty)$. Will in this case the perturbed solution stay close to the original or will it deviate from the latter? A simple example demonstrates that both cases can occur.

Example 8.8 Starting at $t_0 = 0$ and from the initial condition $x_0 = \frac{1}{a}$, the linear time-invariant system,

$$\dot{x} = ax - 1$$

has the steady-state solution $x(t; \frac{1}{a}) = \frac{1}{a}$. If the initial condition is perturbed by Δx_0 then the corresponding solution is

$$x\left(t; \frac{1}{a} + \Delta x_0\right) = \Delta x_0 e^{at} + \frac{1}{a}$$

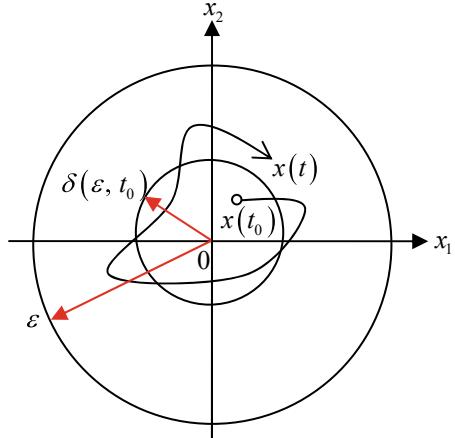
Clearly, if $a < 0$ then for any $\epsilon > 0$, $|x(t; \frac{1}{a} + \Delta x_0) - x(t; \frac{1}{a})| = |\Delta x_0 e^{at}| \leq \epsilon$, as long as $|\Delta x_0| \leq \delta = \epsilon$, and this relation is valid for all $t \geq 0$. So, for any initial condition from the (2ϵ) -strip, the corresponding perturbed solution will remain within the same strip, which is centered around the steady-state solution $x(t) = \frac{1}{a}$. Note that in addition, the perturbed trajectory asymptotically approaches the original steady-state solution, as time tends to infinity. However, if $a > 0$ then no matter how small Δx_0 is, the perturbed trajectory will become arbitrarily large in time, and as a result, it will deviate from the steady-state solution. ■

A solution of (8.15) with the continuity property defined on an infinite interval is called stable. Otherwise, it is unstable.

Definition 8.2 (*Stability of Equilibrium in the Sense of Lyapunov*) The equilibrium point $x^* = 0$ of the nonautonomous unforced dynamics (8.15) is stable if for any $\epsilon > 0$ and $t_0 \geq 0$ there exists $\delta(\epsilon, t_0) > 0$ such that for all initial conditions $\|x(t_0)\| < \delta$ and for all $t \geq t_0 \geq 0$, the corresponding system trajectories are bounded, as in $\|x(t)\| < \epsilon$. The equilibrium is uniformly stable if it is stable and δ does not depend on t_0 . Finally, the equilibrium is unstable if it is not stable.

Using logical symbols such as \wedge “and”, \forall “for any”, \exists “there exists”, and \Rightarrow “implies”, we can formally define the meanings of stable, uniformly stable, and unstable equilibria, (note that the equilibrium under consideration is the origin in R^n).

Fig. 8.7 Geometric interpretation of Lyapunov stability for two-dimensional dynamics



Stable

$$\forall \varepsilon > 0 \quad \forall t_0 > 0 \quad \exists \delta(\varepsilon, t_0) > 0 \quad \forall t \geq t_0 \quad \|x(t_0)\| < \delta(\varepsilon, t_0) \Rightarrow \|x(t)\| < \varepsilon$$

Uniformly Stable

$$\forall \varepsilon > 0 \quad \forall t_0 > 0 \quad \exists \delta(\varepsilon) > 0 \quad \forall t \geq t_0 \quad \|x(t_0)\| < \delta(\varepsilon) \Rightarrow \|x(t)\| < \varepsilon$$

Unstable

$$\exists \varepsilon > 0 \quad \exists t_0 > 0 \quad \forall \delta > 0 \quad \exists T \geq t_0 \quad \|x(t_0)\| < \delta \wedge \|x(T)\| > \varepsilon \quad (8.17)$$

For two-dimensional dynamics, Lyapunov stability of the origin admits a simple geometrical interpretation (Fig. 8.7).

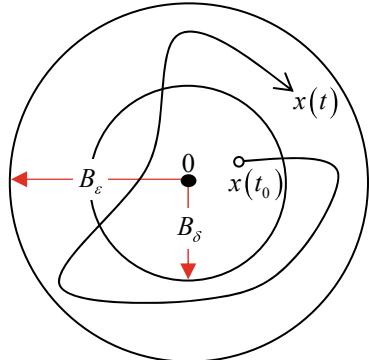
The origin is stable if given a sphere with a radius ε , one can find a smaller sphere whose radius is $\delta \leq \varepsilon$, such that all trajectories that start in the smaller sphere will continue to evolve within the larger sphere, for all $t \geq t_0$. The origin is uniformly stable if δ is independent of t_0 . Finally, the origin is unstable if there exists an ε -sphere and an initial time t_0 , such that no matter how close to the origin a trajectory starts, it will exit this sphere at some finite time T .

Such a geometrical explanation of Lyapunov stability can be easily extended to n -dimensional dynamics, such as (8.15) (Fig. 8.8).

In essence, Lyapunov stability of the origin means that given an outer-sphere $B_\varepsilon = \{x \in R^n : \|x\| \leq \varepsilon\}$ in the system state space R^n , one can find an inner-sphere $B_\delta = \{x \in R^n : \|x\| \leq \delta\} \subset B_\varepsilon$, such that any trajectory that starts in the inner-sphere B_δ will evolve inside the outer-sphere B_ε , for all future times.

A unique feature of nonlinear dynamical systems is their ability to display a completely different behavior in various domains. For example, systems that are stable in a neighborhood of the origin may become unstable, or go to a different

Fig. 8.8 Geometric interpretation of Lyapunov stability in n -dimensional state space



equilibrium, if their initial conditions are chosen outside of this neighborhood. For these reasons, we need to be able to clearly distinguish between local and global stability.

The local feature of Lyapunov stability definitions (8.17) is understood in the sense that for a given outer-sphere B_ε , one must find a set of initial conditions (an inner-sphere B_δ) such that the resulting trajectories stay within the outer-sphere B_ε . It is easy to see that if the origin is stable then for an outer-sphere B_{ε_1} of a bigger radius $\varepsilon_1 > \varepsilon$, the same inner-sphere B_δ can be used to show stability of the system equilibrium. Suppose that the radius of the inner-sphere can be increased indefinitely, as the radius of the outer-sphere increases. In other words, let us suppose that $\delta(\varepsilon, t_0) \rightarrow \infty$, as $\varepsilon \rightarrow \infty$. This would indicate that the set of initial conditions that lead to stable trajectories, is getting bigger. Eventually, one can declare that starting anywhere in R^n , a trajectory will not deviate too far from where it began and as a result, the stability property becomes global.

Definition 8.3 (Global Stability) The origin is globally stable if it is stable and $\lim_{\varepsilon \rightarrow \infty} \delta(\varepsilon, t_0) = \infty$.

Dependence of the system trajectories on a selected initial time t_0 is yet another unique feature of nonautonomous systems. This is in contrast to autonomous dynamics $\dot{x} = f(x)$, whose solutions depend only on the time difference from the initial condition $(t - t_0)$. For nonautonomous systems, stability of an equilibrium will in general be dependent on the selected initial time t_0 . That is why we had to introduce the notion of uniform stability. Also in the definition (8.17), we emphasized that there would exist equilibriums whose stability may or may not depend on the system initial conditions near the equilibrium. We had to also characterize the notion of instability which was merely a logic negation of the stability concept. The next example illustrates differences between the notions of stability and uniform stability.

Example 8.9 Stable but not uniformly stable equilibrium Consider the linear time-dependent dynamics,

$$\dot{x}(t) = 2t(3\sin(t) - 1)x(t)$$

with the initial condition $x(t_0)$. The system solution is

$$\begin{aligned} x(t) &= x(t_0) \exp \left[\int_{t_0}^t 2\tau(3\sin(\tau) - 1) d\tau \right] \\ &= x(t_0) \exp(6\sin t - 6t\cos t - t^2 - 6\sin t_0 + 6t_0\cos t_0 + t_0^2) \end{aligned}$$

The obvious inequality,

$$6\sin t - 6t\cos t - t^2 \leq 6 + \underbrace{(6t - t^2)}_{\leq 9} \leq 15$$

implies

$$|x(t)| \leq |x(t_0)| \underbrace{\exp(15 - 6\sin t_0 + 6t_0\cos t_0 + t_0^2)}_{c(t_0)} = |x(t_0)| c(t_0)$$

Clearly, the origin is the system equilibrium. Is it stable? Since,

$$|x(t)| \leq |x(t_0)| c(t_0)$$

it is evident that for any given positive ε , we can select $\delta(\varepsilon, t_0) = \frac{\varepsilon}{c(t_0)}$, and immediately verify that for all $|x(t_0)| < \delta$, the relation

$$|x(t)| \leq |x(t_0)| c(t_0) \leq \frac{\varepsilon}{c(t_0)} c(t_0) = \varepsilon$$

takes place for all $t \geq t_0$. According to (8.17), we have proved stability of the origin. Is this equilibrium uniformly (in t_0) stable? In order to answer that question, we need to study sensitivity of the system solutions due to changes in t_0 . Toward that end, let $t_0 = 2k\pi$, where k is a fixed positive integer. We can examine $x(t)$ at $t = t_0 + \pi = (2k + 1)\pi$, and get

$$x((2k + 1)\pi) = x(2k\pi) \exp((4k + 1)(6 - \pi)\pi)$$

or, equivalently

$$\frac{x((2k + 1)\pi)}{x(2k\pi)} = \exp((4k + 1)(6 - \pi)\pi) > 1, \quad \forall k \geq 1$$

So, the sequence $x(2k\pi)$ tends to infinity, as $k = 1, 2, \dots, \infty$. In other words, there is an unboundedly increasing sequence of initial time instants $t_0(k) = 2k\pi$ which leads to an unboundedly increasing sequence of the initial values for the system solutions $x(t_0(k)) = x(2k\pi) \xrightarrow{k \rightarrow \infty} \infty$. Therefore, given any $\varepsilon > 0$, there is no $\delta(\varepsilon)$ independent of t_0 that would satisfy the uniform stability definition in (8.17). ■

As shown in Example 8.9, in addition to being stable, perturbed trajectories may asymptotically converge back to the equilibrium. This observation naturally leads to the definitions of (a) asymptotic stability; (b) uniform asymptotic stability, and (c) global uniform asymptotic stability.

Definition 8.4 (Asymptotic Stability) The equilibrium point $x^* = 0$ of (8.15) is asymptotically stable if it is stable and there exists a positive constant $c = c(t_0)$ such that $x(t) \rightarrow 0$ as $t \rightarrow \infty$, for all $\|x(t_0)\| \leq c$.

Definition 8.5 (Uniform Asymptotic Stability) The equilibrium point $x^* = 0$ of (8.15) is uniformly asymptotically stable if it is uniformly stable and there exists a positive constant c , independent of t_0 , such that $x(t) \rightarrow 0$ as $t \rightarrow \infty$, for all $\|x(t_0)\| \leq c$, uniformly in t_0 , where the limit uniformity is understood in the following sense.

$$\exists c \forall \eta > 0 \exists T(\eta) \forall t \geq t_0 + T(\eta) \forall \|x(t_0)\| \leq c \Rightarrow \|x(t)\| \leq \eta$$

Definition 8.6 (Global Uniform Asymptotic Stability) The origin is globally uniformly asymptotically stable if it is uniformly asymptotically stable and $\lim_{\varepsilon \rightarrow \infty} \delta(\varepsilon) = \infty$.

Achieving uniform asymptotic stability is a highly desirable property in any control design, since asymptotically stable systems are able to maintain their closed-loop performance in the presence of perturbations and disturbances. We shall see that, in general, adaptive controllers achieve uniform stability and force the system tracking errors to converge to zero, asymptotically in time. This key property is lesser than uniform asymptotic stability but it is greater than uniform stability that is in addition to being uniformly stable, certain signals (such as tracking errors) in the closed-loop system asymptotically tend to zero, while others are kept uniformly stable and bounded.

8.5 Lyapunov Stability Theorems

In his seminal work on stability of motion, A. M. Lyapunov introduced two theorems, known as Lyapunov's indirect (first) and direct (second) methods, for assessing stability of nominal solutions that arise in dynamical systems, which are governed by a finite number of coupled ordinary differential equations. Lyapunov's methods provide verifiable sufficient conditions for stability of a nominal

trajectory. Moreover, neither method requires an explicit knowledge of the system solutions.

Lyapunov's indirect method allows one to draw conclusions about the stability of an equilibrium point (the origin) for a nonlinear autonomous n -dimensional system $\dot{x} = f(x)$. The method is based on the linearization of the system dynamics around an equilibrium. In order for the original nonlinear system to be locally stable in the sense of Lyapunov, it is sufficient to show that the system Jacobian matrix $A = \left. \frac{\partial f(x)}{\partial x} \right|_{x=0}$ has all its eigenvalues $\{\lambda_i\}_{i=1, 2, \dots, n}$ in the complex open left half plane: $\operatorname{Re} \lambda_i < 0$, $\forall i = 1, 2, \dots, n$. If $\operatorname{Re} \lambda_i > 0$ for at least one eigenvalue of A then the origin is unstable. If A has eigenvalues on the $j\omega$ -axis then the indirect method of Lyapunov does not apply. Further details, including formal proofs, can be found in [1–3]. From the control design point of view, the indirect method of Lyapunov provides the much-needed theoretical foundation for application of linearization-based controllers in nonlinear systems.

Our main interest will be focused on Lyapunov's direct method. Specifically, we will discuss the method formulation and its applications to analyzing uniform stability of nonautonomous systems (8.15). We begin with the definitions of positive definite and negative definite (semidefinite) functions. Subsequently, we will utilize these functions to constructively determine stability of an equilibrium point.

Definition 8.7 (Positive Definite and Semidefinite Functions) A scalar function $V(x) : R^n \rightarrow R$ of a vector argument $x \in R^n$ is called locally positive definite (semidefinite) if $V(0) = 0$, and there exists a constant $r > 0$ such that $V(x) > 0$ ($V(x) \geq 0$), for all nonzero $x \in R^n$ from the r -neighborhood of the origin $B_r = \{x \in R^n : \|x\| \leq r\}$. The function is said to be globally positive definite if $B_r = R^n$.

Definition 8.8 (Negative Definite and Semidefinite Functions) A scalar function $V(x) : R^n \rightarrow R$ of a vector argument $x \in R^n$ is called locally (globally) negative definite (semidefinite) if the function $(-V(x))$ is locally (globally) positive definite (semidefinite).

Example 8.10 Sign definite and semidefinite functions Consider a scalar function of a scalar argument: $V(x) = x^2 (9 - x^2)$. A graphical sketch of this function is shown in Fig. 8.9.

It is easy to see that this function is locally positive definite on the open interval $(-3, 3)$, and it becomes positive semidefinite on the closed interval $[-3, 3]$. On the other hand, the function $V(x) = x^2$ is globally positive definite. Furthermore, if $P \in R^{n \times n}$ is a symmetric positive definite (semidefinite) matrix then the function $V(x) = x^T P x$ is globally positive definite (semidefinite), while $W(x) = -x^T P x$ represents a globally negative definite (semidefinite) function. ■

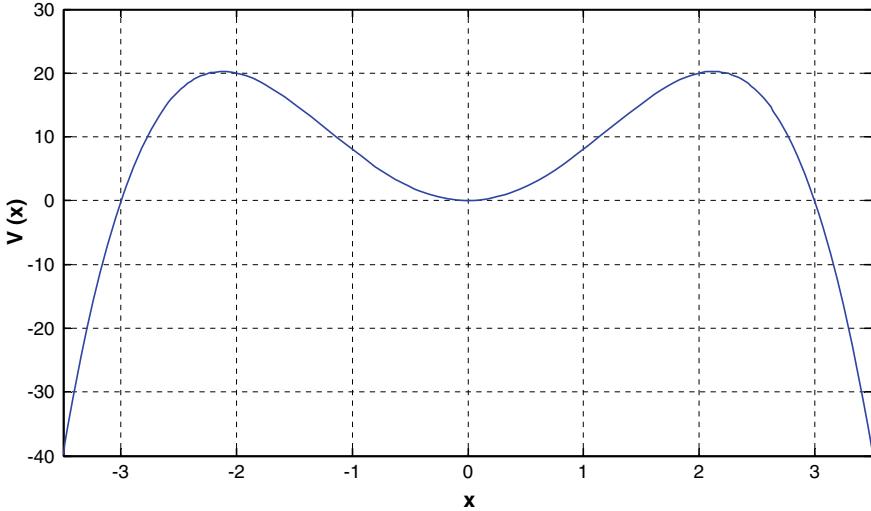


Fig. 8.9 Locally positive definite-function from Example 8.10

Next, we introduce the concept of the time derivative of a scalar function along the trajectories of a differential equation. Suppose that we are given a scalar continuously differentiable function $V(x)$, whose vector argument $x(t) \in R^n$ represents a time-varying trajectory of the nonautonomous system (8.15). We can compute the time derivative of $V(x(t))$ along the system solution $x(t)$,

$$\boxed{\dot{V}(x)} = \sum_{i=1}^n \frac{\partial V}{\partial x_i} \dot{x}_i = \sum_{i=1}^n \frac{\partial V}{\partial x_i} f_i(t, x) = \boxed{\nabla V(x) f(t, x)} \quad (8.18)$$

where $\nabla V(x) = \left(\frac{\partial V}{\partial x_1}, \frac{\partial V}{\partial x_2}, \dots, \frac{\partial V}{\partial x_n} \right)$ is the vector-row gradient of $V(x)$ with respect to x . We immediately note that the time derivative of $V(x)$ along the trajectories of (8.15) depends not only on the function $V(x)$ but also on the system dynamics under consideration. Changing the latter while keeping the same V will in general yield a different $\dot{V}(x)$. We are now fully equipped to formulate the direct (second) method of Lyapunov.

Theorem 8.5 (Lyapunov's Direct Method for Assessing Uniform Stability of Nonautonomous Systems) *Let $x^* = 0 \in R^n$ be an equilibrium point for the nonautonomous dynamics (8.15), whose initial conditions are drawn from a domain $D \subset R^n$, with $x^* \in D$ and $t_0 = 0$. Suppose that on the domain D there exists a continuously differentiable locally positive-definite function $V(x) : D \rightarrow R$, whose time derivative along the system trajectories is locally negative semidefinite,*

$$\dot{V}(x) = \nabla V(x) f(t, x) \leq 0 \quad (8.19)$$

for all $t \geq 0$ and for all $x \in D$. Then the system equilibrium $x^* = 0$ is locally uniformly stable in the sense of Lyapunov. If in (8.19) $\dot{V}(x) < 0$ for all nonzero x and for all $t \geq 0$ (the time derivative along the system trajectories is locally negative definite) then the origin is locally uniformly asymptotically stable. ■

We shall immediately note that Lyapunov's direct method presents sufficient conditions for appraising uniform stability, (formal proof can be found in [2, Th. 4.8, pp. 151–153]). These sufficient conditions are expressed in terms of a locally positive-definite function $V(x)$, which is often called a Lyapunov function candidate. If in addition, the strict inequality (8.19) holds then $V(x)$ becomes what is commonly referred to as a Lyapunov function. In terms of these concepts, Theorem 8.5 states that the origin is a uniformly stable if given the system dynamics, a Lyapunov function can be found. Conversely, if a Lyapunov function candidate does not satisfy the sufficient for stability requirement (8.19), no definite conclusions can be drawn and the search for a suitable Lyapunov function must continue.

Let us briefly discuss a geometric interpretation of Lyapunov's direct method. Choose a sufficiently small positive constant c we can ensure that the level set $V_c = \{x \in D : V(x) = c\}$ of the Lyapunov function $V(x)$ resides inside D (see Fig. 8.10).

Then, it is possible to show that the interior set $\Omega_c = \{x \in D : V(x) \leq c\}$, whose boundary is V_c , is closed and bounded (i.e., compact). For any $x \in V_c$, the gradient row-vector $\nabla V(x)$ points perpendicular to the tangent hyperplane that touches the level set at x . Also, the inequality (8.19) implies that at any given time t and for any $x \in V_c$, the angle between the gradient vector $\nabla V(x)$ and the system dynamics $f(t, x)$ is no less than $\frac{\pi}{2}$. Therefore, the system trajectory will not leave Ω_c . Moreover, since $V(x(t))$ is non-increasing then $x(t)$ will remain in

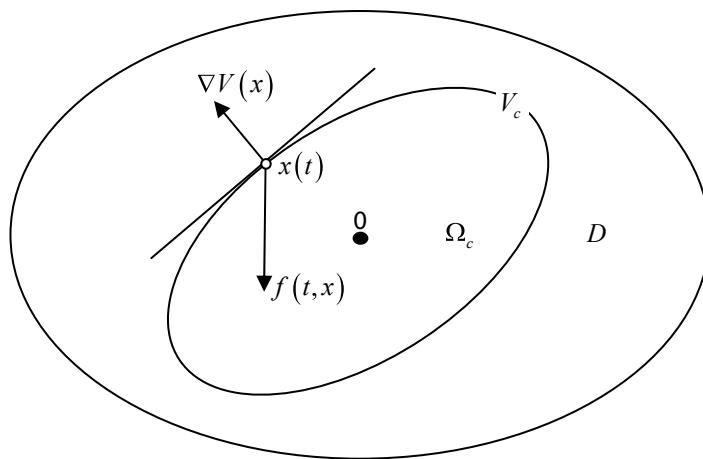
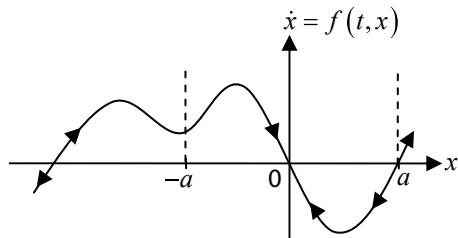


Fig. 8.10 Geometrical interpretation of Lyapunov's direct method

Fig. 8.11 System dynamics from Example 8.11



this set for all future times. If in addition it is assumed that $\dot{V}(x) < 0$ then the system trajectories, starting anywhere in D , will evolve by entering a sequence of diminishing level sets ($V_{c_1} \supset \dots \supset V_{c_k} \supset \dots$) with ($c > c_1 > \dots > c_k > \dots$), and as a result, these solutions will asymptotically approach the origin.

The Lyapunov function $V(x)$ can now be viewed as an “energy-like” function for testing stability of a system. If the values of V do not increase along the system trajectories, then the origin is uniformly stable. If V strictly decreases then in addition to being stable, the system trajectories will approach the origin asymptotically.

Example 8.11 Consider the scalar system,

$$\dot{x} = f(t, x)$$

where $f(t, x)$ is locally Lipschitz on an open interval $(-a, a)$, $f(t, 0) = 0$, and $x f(t, x) < 0$ for all $t \geq 0$ and all nonzero $x \in (-a, a)$, that is the graph of $f(t, x)$ is located in the second and the fourth quadrants, uniformly in t and for all x from $(-a, a)$. The system dynamics are shown in Fig. 8.11.

It is clear, that starting anywhere within the open interval $(-a, a)$, the system solutions will asymptotically converge to the origin. Let us now use Lyapunov’s direct method to show that the origin is uniformly asymptotically stable. Toward that end, we consider a quadratic Lyapunov function candidate in the form $V(x) = x^2$. Its time derivative along the system trajectories is strictly negative for all nonzero $x \in (-a, a)$.

$$\dot{V}(x) = 2x\dot{x} = 2x f(t, x) < 0$$

Consequently, $V(x)$ is a Lyapunov function and, according to Theorem 8.5, the origin is locally uniformly asymptotically stable. Of course, we already knew the answer since the system dynamics were scalar and the phase plane analysis method was readily applicable.

Suppose that the same system is n -dimensional and assume that the vector-field $f(t, x)$ satisfies $x^T f(t, x) < 0$, uniformly in t and for all x from a domain $D \subset R^n$. We can use a quadratic Lyapunov function in the form $V(x) = x^T x$, show that its time derivative along the system trajectories is negative.

$$\dot{V}(x) = 2x^T \dot{x} = 2x^T f(t, x) < 0$$

and thus, prove the uniform asymptotic stability property of the origin. Note that in this case, the phase plane analysis does not apply. ■

Lyapunov functions are by no means unique. Recalling the scalar dynamics in Example 8.11, let us assume that the system is autonomous. We can prove asymptotic stability using the same Lyapunov function as before. In order to show that it is not unique, let us consider the following Lyapunov function candidate

$$V(x) = - \int_0^x f(y) dy$$

Since $x f(x) < 0$ for all nonzero x , $V(x)$ is positive definite and $V(0) = 0$. Therefore, it represents a Lyapunov function candidate. The function time derivative along the system trajectories is negative,

$$\dot{V}(x) = -f(x) \dot{x} = -f^2(x) < 0$$

for all nonzero $x \in (-a, a)$. Consequently, the origin is uniformly asymptotically stable. ■

The uniform asymptotic stability property calls for a subset of D . Starting there, the system solutions will converge to the origin. This subset is called the region of attraction. We shall study the case when the system domain and the region of attraction both equal R^n . This will lead to the concept of global uniform asymptotic stability.

Definition 8.9 If the region of attraction of a uniformly asymptotically stable equilibrium is R^n , then the equilibrium is said to be globally uniformly asymptotically stable.

The next definition leads to a verifiable condition for a Lyapunov function to yield global uniform stability properties.

Definition 8.10 A Lyapunov function candidate $V(x) : R^n \rightarrow R$ defined such that $\lim_{\|x\| \rightarrow \infty} V(x) = \infty$ is called radially unbounded.

Let $V_c = \{x \in R^n : V(x) = c\}$ denote a level set of a radially unbounded Lyapunov function candidate $V(x) : R^n \rightarrow R$, and let $\Omega_c = \{x \in R^n : V(x) \leq c\}$ be the union of the interior set of V_c , and V_c itself. Consider a converging sequence $\lim_{n \rightarrow \infty} x_n = a$, with all x_n from Ω_c . Then the limit point a must also be in Ω_c . In fact, since $V(x)$ is continuous on R^n and $V(x_n) \leq c$ for all $n = 1, 2, \dots$, we get $c \geq \lim_{n \rightarrow \infty} V(x_n) = V(a)$, and consequently $a \in \Omega_c$. We have proved that every converging sequence in Ω_c has its limit point in the same set. Hence, Ω_c is a closed set. Moreover, we can prove that Ω_c is bounded. This fact can be shown by contradiction. Suppose that Ω_c is unbounded. Then there must exist a sequence

of points $\{x_n\} \in \Omega_c$, whose limit is infinity. Since $V(x)$ is continuous and radially unbounded then $c \geq \lim_{n \rightarrow \infty} V(x_n) = \infty$, which is an obvious contradiction to the argument. Therefore, Ω_c is a bounded set. Since it is also closed and belongs to R^n , Ω_c is compact.

The next theorem states that if a radially unbounded Lyapunov function can be found then the local uniform (asymptotic) stability properties from Theorem 8.5 become global. When applied to autonomous systems, this results is also known as Barbashin–Krasovskii–LaSalle theorem [1, 2].

Theorem 8.6 *Let $x = 0$ be an equilibrium point for (8.15). Let $V(x) : R^n \rightarrow R$ be a radially unbounded Lyapunov function of the system. Then the system equilibrium is globally uniformly asymptotically stable.* ■

Simple examples of radially unbounded Lyapunov function candidates include quadratic functions of the form $V(x) = x^T P x$, where $P \in R^{n \times n}$ is a symmetric positive-definite matrix.

Example 8.12 The rotational motion of a rigid aircraft in three-dimensional space is governed by the following system of ordinary differential equations,

$$J \dot{\omega} = -[\omega \times J \omega] + M$$

where $\omega = (p \ q \ r)^T$ is the body angular velocity vector, with the roll (p), the pitch (q), and the yaw (r) velocity components, $J \in R^{3 \times 3}$ is the aircraft inertia matrix,

$$J = \begin{pmatrix} J_{xx} & 0 & -J_{xz} \\ 0 & J_{yy} & 0 \\ -J_{xz} & 0 & J_{zz} \end{pmatrix}$$

with positive components $(J_{xx}, J_{yy}, J_{zz}, J_{xz})$, and $M \in R^3$ is the vector of aerodynamic/propulsive moments, computed with respect to the vehicle center of gravity. We assume that

$$\det J = J_{yy} (J_{xx} J_{zz} - J_{xz}^2) > 0$$

and also suppose that the moment vector M represents the system control input. The control task is to select M such that the aircraft rotational dynamics become globally uniformly asymptotically stable. We begin by considering a quadratic Lyapunov function candidate in the form

$$V(\omega) = \omega^T J \omega$$

This is indeed a Lyapunov function candidate since $V(0) = 0$ and J is symmetric and positive definite. We proceed to compute the time derivative of $V(\omega)$ along the trajectories of the aircraft rotational dynamics.

$$\dot{V}(\omega) = 2\omega^T J \dot{\omega} = 2\omega^T (-[\omega \times J \omega] + M) = 2\omega^T M$$

According to Theorem 8.5, we need $\dot{V}(\omega) < 0$. This can be easily achieved if we select the control input as a weighted negative feedback on ω ,

$$M = -P\omega$$

with a symmetric positive-definite matrix of weights $P \in R^{3 \times 3}$. Then for any nonzero angular velocity $\omega \in R^3$,

$$\dot{V}(\omega) = -2\omega^T P\omega < 0$$

and so, the origin is uniformly asymptotically stable. Moreover, since $V(\omega)$ is radially unbounded, the achieved closed-loop uniform asymptotic stability property is global. This example illustrates both the practicality and the effectiveness of Lyapunov's direct method. Not only we were able to assert the desired stability property but we did so by using the "inverse" Lyapunov design arguments that is we chose our control input to enforce the sufficient conditions of Theorem 8.5. ■

Example 8.13 For the linear time-invariant (LTI) n -dimensional dynamics,

$$\dot{x} = Ax$$

with a Hurwitz (stable) matrix $A \in R^{n \times n}$, consider a quadratic Lyapunov function candidate $V(x) = x^T P x$, where $P \in R^{n \times n}$ is a symmetric positive-definite matrix. Let $Q \in R^{n \times n}$ be another symmetric positive-definite matrix. The time derivative of $V(x)$ along the system solutions is

$$\dot{V}(x) = x^T P \dot{x} + \dot{x}^T P x = x \left(P A + A^T P \right) x$$

If we can make this derivative negative for all nonzero $x \in R^n$, then we would prove global uniform asymptotic stability of the origin. In order to do that, we define P to be the solution of the so-called Lyapunov algebraic equation.

$$P A + A^T P = -Q$$

It turns out that given any symmetric positive definite Q , the Lyapunov algebraic equation has the unique symmetric positive-definite solution $P = P^T > 0$ if and only if A is Hurwitz [3]. Then,

$$\dot{V}(x) = -x^T Q x < 0$$

for all nonzero $x \in R^n$, which immediately proves global uniform asymptotic stability of the origin.

Evidently, since the system is linear and time-invariant, we could have proven asymptotic stability by simply noting that A is Hurwitz. Nevertheless, the Lyapunov's arguments allow us to establish an important link between the stability of LTI systems and the Lyapunov's direct method. This link is given by the Lyapunov algebraic equation, and the latter will become the key design component for adaptive controllers. ■

8.6 Uniform Ultimate Boundedness

The concepts of stability in the sense of Lyapunov are formulated with respect to an equilibrium or a nominal trajectory. Often, systems are designed to operate in the presence of disturbances and other uncertainties. As a result, the “ideal” definition of an equilibrium may not apply. Consider the nonautonomous system,

$$\dot{x} = f(t, x) + \xi(t), \quad x(t_0) = x_0 \quad (8.20)$$

subject to a bounded disturbance $\xi(t) \in R^n$, with $\|\xi(t)\| \leq \xi_{\max}$. Suppose that $f : [0, \infty) \times D \rightarrow R^n$ is piecewise continuous in t , locally Lipschitz in x on $[0, \infty) \times D$, and $D \subset R^n$ is a domain that contains the origin $x = 0$. Also, suppose that $f(t, 0) = 0$, $\forall t \geq 0$. It is easy to see that no matter how small the disturbance bound ξ_{\max} is, the origin is no longer an equilibrium point of the system. Nevertheless, we can still use Lyapunov's direct method to study the system behavior outside of the sphere $B_{\xi_{\max}} = \{x \in R^n : \|x\| \leq \xi_{\max}\}$, as if the origin is the system equilibrium. The main idea is to find a Lyapunov-like function $V(x)$ for all x outside of a bigger sphere $B_r \supset B_{\xi_{\max}}$, and then show that in finite time T the system trajectories enter B_r and remain there for all $t \geq T$. This thought will eventually lead us to the concept of uniform ultimate boundedness (UUB).

Example 8.13 Consider the scalar nonautonomous dynamics,

$$\dot{x} = -x + \xi(t), \quad x(t_0) = x_0 > \xi_{\max} > 0$$

where $\xi(t)$ is a time-varying unknown disturbance bounded by ξ_{\max} . Clearly, the system has no equilibrium points. The system solutions can easily be found.

$$x(t) = e^{-(t-t_0)} x_0 + \int_{t_0}^t e^{-(t-\tau)} \xi(\tau) d\tau$$

We can also compute an upper bound,

$$|x(t)| \leq e^{-(t-t_0)} |x_0| + \left(1 - e^{-(t-t_0)}\right) \xi_{\max} \leq e^{-(t-t_0)} |x_0| + \xi_{\max}$$

and show that for any initial condition x_0 and any given $r > 0$, there must exist a finite time $0 \leq T(x_0, r) < \infty$, such that $|x(t)| \leq \xi_{\max} + r$, for all $t \geq t_0 + T(x_0, r)$. In fact, if $|x_0| \leq r$ then $|x(t)| \leq e^{-(t-t_0)} r + \xi_{\max} \leq r + \xi_{\max}$, and so $T(x_0, r) = 0$. If, on the other hand, $|x_0| > r$, then it is sufficient to choose $T(x_0, r) = \ln \frac{|x_0|}{r}$. This simple argument shows that the system trajectories enter a neighborhood of the origin $B_r = \{x \in R : |x| \leq r\}$ in finite time $T(x_0, r)$, and continue to evolve within the neighborhood afterward.

Alternatively, we can also exploit Lyapunov's direct method to show uniform ultimate boundedness of the system solutions. Let us utilize $V(x) = x^2$, which in this case is not a Lyapunov function candidate, since the origin is not an equilibrium of the system. Nevertheless, we proceed to calculate the function time derivative along the system trajectories.

$$\begin{aligned}\dot{V}(x) &= 2x \dot{x} = 2x(-x + \xi(t)) = -2x^2 + 2x\xi(t) \leq -2x^2 \\ &\quad + 2\xi_{\max}|x| = -2|x|(|x| - \xi_{\max})\end{aligned}$$

Given any positive constant $\varepsilon > 0$, it is evident that

$$\dot{V}(x) < 0, \quad \forall |x| \geq \xi_{\max} + \varepsilon$$

Let $r = \xi_{\max} + \varepsilon$. Then the time derivative of V is negative outside of the closed interval $B_r = [-r, r]$. Next, we are going to show that all solutions that start outside of B_r will reenter the interval within a finite time, and will remain their forward in time. Since \dot{V} is negative for all $|x| \geq r$ then the solutions starting inside B_r will remain there. Hence, these trajectories are uniformly bounded in time, that is $|x(t)| \leq r, \forall t \geq t_0$. Starting from any $|x_0| > r$, \dot{V} is strictly negative in the annulus set $\{r^2 \leq V(x) \leq x_0^2\}$, which implies that in this set $V(x(t))$ will continue to decrease monotonically until the solution enters $B_r = \{|x| \leq r\} = \{V(x) \leq r^2\}$, at some finite time $T(x_0, r)$. From that time on, the solution will evolve within B_r since \dot{V} is strictly negative on its boundary $V(x) = r^2$. So again, we conclude that the system solutions are UUB with the ultimate bound $|x(t)| \leq r$. Similar to proving stability, the main advantage of applying Lyapunov's direct method to establish UUB of trajectories is the fact that the method does not require the knowledge of an explicit form of the system solutions. ■

We now give a formal definition of the UUB concept as it is stated in [2].

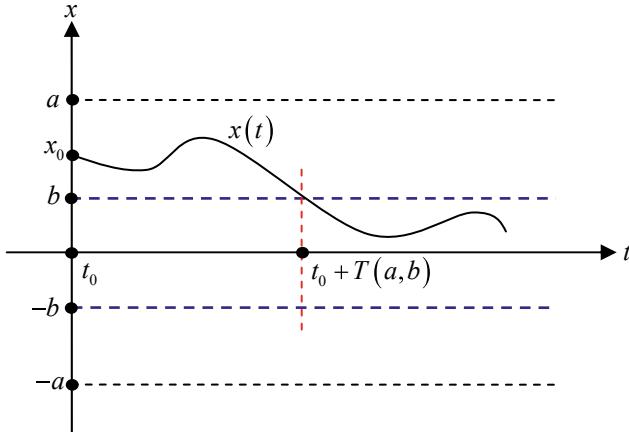


Fig. 8.12 Graphical interpretation of the UUB concept for nonautonomous dynamics

Definition 8.11 The solutions of (8.20) are uniformly ultimately bounded with ultimate bound b if there exist positive constants b and c , independent of $t_0 \geq 0$, and for every $a \in (0, c)$, there is $T = T(a, b)$, independent of t_0 , such that

$$\|x(t_0)\| \leq a \Rightarrow \|x(t)\| \leq b, \quad \forall t \geq t_0 + T \quad (8.21)$$

These solutions are said to be globally uniformly ultimately bounded if (8.21) holds for arbitrarily large a .

Graphical interpretation of the UUB concept is shown in Fig. 8.12.

In the definition above, the term “uniform” indicates that the bound b does not depend on t_0 . The term “ultimate” means that boundedness holds after the lapse of a finite time T . The constant c defines a neighborhood of the origin, independent of t_0 , such that all trajectories starting in the neighborhood will remain bounded in time. If c can be chosen arbitrarily large then the local UUB property becomes global.

The notion of UUB can be considered as a “milder” form of stability in the sense of Lyapunov (SISL). A brief comparison between the SISL and the UUB concepts is given below.

- SISL is defined with respect to an equilibrium, while UUB is not.
- Asymptotic SISL is a strong property that is very difficult to achieve in practical dynamical systems.
- SISL requires the ability to keep the state arbitrarily close to the system equilibrium by starting sufficiently close to it. This is still too strong a requirement for practical systems operating in the presence of uncertainties and unknown disturbances.
- A bound b in the UUB concept cannot be made arbitrarily small by starting closer to the system equilibrium (if it has one) or to the origin.

Next, we present a Lyapunov-based analysis of UUB properties. Suppose that for a given continuously differentiable positive-definite function $V(x)$, we can choose two finite positive constants $0 < \varepsilon < c < \infty$, such that the sets $\Omega_\varepsilon = \{V(x) \leq \varepsilon\}$ and $\Omega_c = \{V(x) \leq c\}$ are closed and bounded, (i.e., compact). This would be true if, for example, $V(x) = x^T P x$ and P is a symmetric positive-definite matrix. Consider the annulus set in R^n ,

$$\Lambda = \{x \in R^n : \varepsilon \leq V(x) \leq c\} = \Omega_c - \Omega_\varepsilon$$

and presume that the time derivative of $V(x(t))$ along the trajectories of the nonautonomous dynamical system (8.20) is strictly negative definite inside Λ ,

$$\dot{V}(x(t)) < 0, \quad \forall x \in \Lambda, \quad \forall t \geq t_0$$

Then a trajectory that starts in the annulus would have to move in a direction where $V(x(t))$ is decreasing. Since the annulus boundary consists of the function level sets, the trajectory would be trapped between the two sets and it would have to move toward the origin. Thus, inside the annulus, the system solution behaves as if the origin is a uniformly asymptotically stable equilibrium, which it is not.

Starting from an initial condition $x_0 = x(t_0)$ at a time instant $t_0 \geq 0$, the corresponding system trajectory $x(t)$ will evolve such that the function $V(x(t))$ decreases until the trajectory enters (in finite time T) the set Ω_ε , where it will remain afterward.

This argument proves the UUB property of the system solutions, with the ultimate bound $b = \max_{x \in \Omega_\varepsilon} \|x\| = \max_{x \in \partial\Omega_\varepsilon} \|x\|$, achieved on the boundary $\partial\Omega_\varepsilon$ of the set Ω_ε . The three sets Λ , Ω_c , Ω_ε and the UUB bound b are shown in Fig. 8.13. ■

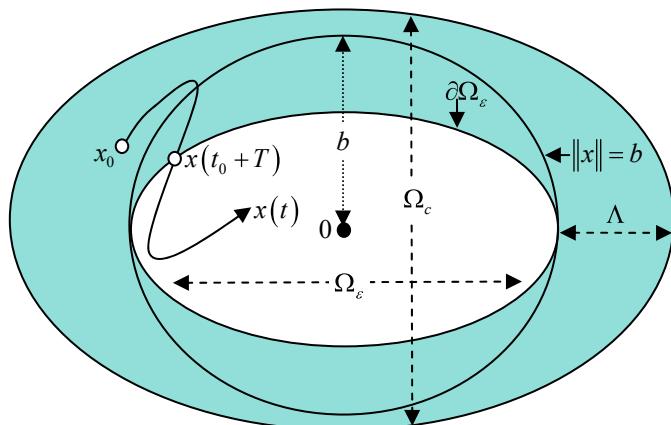


Fig. 8.13 Compact sets in the UUB analysis

Example 8.14 Let D denote a domain in R^n , where the system dynamics are defined as,

$$\dot{x} = A x + B \varepsilon(t, x), \quad x(t_0) = x_0 \quad (8.22)$$

with the state $x \in R^n$, a Hurwitz matrix $A \in R^{n \times n}$, a constant matrix $B \in R^{n \times m}$, and with a bounded function $\varepsilon(t, x) : R \times R^n \rightarrow R^m$, $\|\varepsilon(t, x)\| \leq \varepsilon_{\max}$, which is assumed to hold for all $t \geq t_0$ and $x \in D$. Let us choose $Q = Q^T > 0$ and consider a quadratic positive-definite function in the form,

$$V(x) = x^T P x \quad (8.23)$$

where $P = P^T > 0$ is the unique positive-definite symmetric solution of the algebraic Lyapunov equation.

$$P A + A^T P = -Q \quad (8.24)$$

Such a solution exists for any symmetric positive definite Q , since A is Hurwitz. Due to the latter, it is intuitively clear that the trajectories of (8.22) are UUB. Let us formally prove it.

The time derivative of V along the system trajectories satisfies the following relation for all ($t \geq t_0$, $x \in D$),

$$\begin{aligned} \dot{V}(x) &= -x^T Q x + 2x^T P B \varepsilon(t, x) \leq -\|x\| (\lambda_{\min}(Q) \|x\| \\ &\quad - 2\lambda_{\max}(P) \|B\| \varepsilon_{\max}) \end{aligned} \quad (8.25)$$

where $\lambda_{\min}(Q)$, $\lambda_{\max}(P)$ are the minimum and the maximum eigenvalues of Q and P , respectively. From (8.25), it follows that $\dot{V}(x) < 0$ for all x that are located outside of the compact set,

$$B_r = \left\{ x \in D : \|x\| \leq 2 \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)} \|B\| \varepsilon_{\max} = r \right\} \quad (8.26)$$

where we have assumed a sufficiently small $\varepsilon_{\max} > 0$ for the inclusion $B_r \subset D$ to hold.

We can define the maximal level set of $V(x)$ in D .

$$\Omega_{\max} = \max_c \{x \in D : V(x) = c\} = \{x \in D : V(x) = c_{\max}\} \quad (8.27)$$

If the domain D is bounded then $c_{\max} > 0$ is finite. This follows from the fact that $V(x)$ is a continuous quadratic function of x and as such, its maximum on a bounded domain exists and finite. On the other hand, if D is unbounded then $c_{\max} = \infty$, and consequently, Ω_{\max} is unbounded as well. Either way, existence of this set is guaranteed.

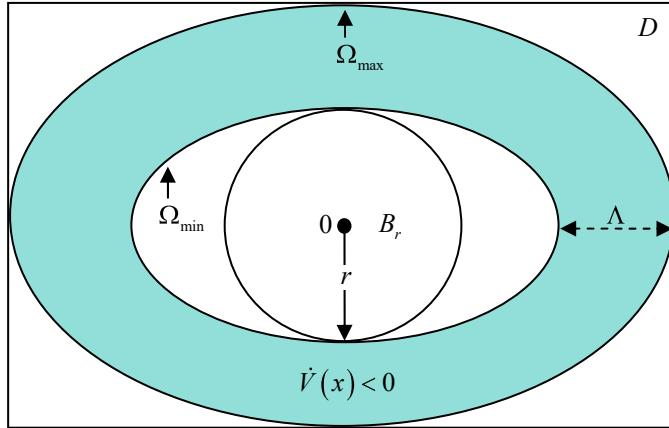


Fig. 8.14 Level sets from Example 8.14

Let us also define the minimal level set of $V(x)$ that contains B_r ,

$$\Omega_{\min} = \min_c \{x \in B_r : V(x) = c\} = \{x \in \partial B_r : V(x) = c_{\min}\} \quad (8.28)$$

where ∂B_r denotes the boundary set of B_r . Existence of Ω_{\min} is guaranteed since B_r is compact and $V(x)$ is a continuous function with its minimum value achieved on the set boundary.

According to (8.25), $\dot{V}(x) < 0$ for all x from the annulus

$$\Lambda = \{x \in R^n : c_{\min} \leq V(x) \leq c_{\max}\} \quad (8.29)$$

Figure 8.14 shows inclusion of the level sets.

Consequently, any trajectory that starts in Λ will have to enter the interior set of Ω_{\min} in finite time T , and it will remain there for all $t \geq t_0 + T$. This proves the UUB property of the system trajectories.

Next, we are going to estimate the corresponding ultimate bound b . In order to do this, we introduce the smallest sphere that contains Ω_{\min} (see Fig. 8.15).

$$B_R = \min_c \{x \in \Omega_{\min} : \|x\| \leq c\} \quad (8.30)$$

Since for all $x \in R^n$,

$$\lambda_{\min}(P) \|x\|^2 \leq \underbrace{x^T P x}_{V(x)} \leq \lambda_{\max}(P) \|x\|^2 \quad (8.31)$$

then for all $x \in B_r$,

$$V(x) \leq \lambda_{\max}(P) r^2 \quad (8.32)$$

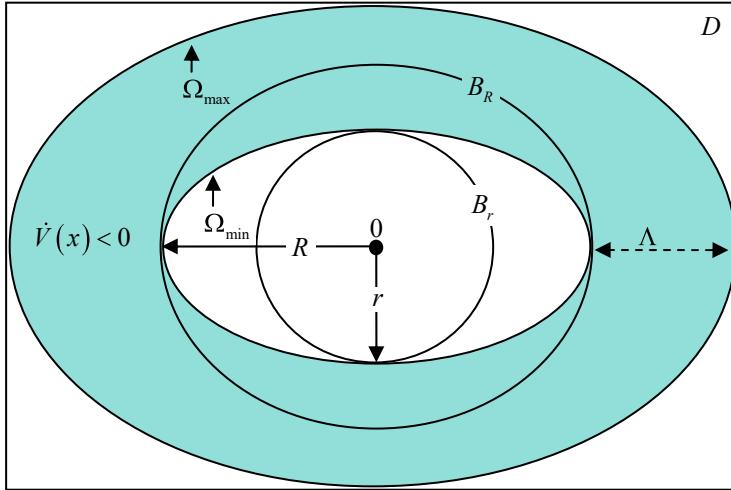


Fig. 8.15 Level sets and spheres from Example 8.14

and, so

$$\Omega_{\min} = \{x \in \partial B_r : V(x) = \lambda_{\max}(P) r^2 = c_{\min}\} \quad (8.33)$$

For all $x \in \Omega_{\min}$, we have

$$\lambda_{\min}(P) \|x\|^2 \leq x^T P x = \lambda_{\max}(P) r^2 \quad (8.34)$$

and therefore,

$$\|x\|^2 \leq \frac{\lambda_{\max}(P)}{\lambda_{\min}(P)} r^2 = R^2 \quad (8.35)$$

In other words, the radius of the smallest sphere B_R that surrounds Ω_{\min} , is

$$R = r \sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}} \quad (8.36)$$

Clearly, $R > r$ as it is shown in Fig. 8.15. Finally, substituting the definition of r from (8.26) into (8.36), we get the ultimate bound for the system trajectories.

$$b = R = 2 \|B\| \varepsilon_{\max} \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)} \sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}} \quad (8.37)$$

Table 8.1 UUB related assumptions and conclusions for the system from Example 8.14

Plant dynamics	$\dot{x} = Ax + B\varepsilon(t, x), \quad x(t_0) = x_0$
Lyapunov equation	$P A + A^T P = -Q$
Lyapunov-like function	$V(x) = x^T P x$
Assumptions	A is Hurwitz $\ \varepsilon(t, x)\ \leq \varepsilon_{\max}, \quad \forall x \in D \subset R^n$ $\left\{x \in D : \ x\ \leq 2 \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)} \ B\ \varepsilon_{\max}\right\} \subset D$
UUB	$\ x(t)\ \leq 2 \ B\ \varepsilon_{\max} \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)} \sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}, \quad \forall t \geq t_0 + T$

Table 8.1 presents all of the key assumptions and relations that were utilized to establish the UUB result.

In summary, we have established the UUB property of the system (8.22). Our analysis was based on Lyapunov's direct method. Specifically, we have shown that all trajectories that start in Ω_{\max} , will enter the interior of Ω_{\min} in finite time, and will evolve inside of this set afterward, with the ultimate bound (8.37). ■

8.7 Barbalat's Lemma

We now turn our attention to n -dimensional nonautonomous systems of the form,

$$\dot{x} = f(t, x), \quad x(t_0) = x_0, \quad f(t, 0) = 0 \quad (8.38)$$

with the vector-field $f : R \times D \rightarrow R^n$ defined on a domain $D \subset R^n$. In order to ensure existence and uniqueness of the system solutions, we assume that $f(t, x)$ is piecewise continuous in t and locally Lipschitz-continuous in x , uniformly in t .

Suppose that we have a Lyapunov function candidate $V(x) : D \rightarrow R$, whose time derivative along the trajectories of (8.38) satisfies,

$$\dot{V}(x) = \nabla V(x) f(t, x) \leq -W(x) \leq 0 \quad (8.39)$$

for all $x \in D$, where $W(x) : D \rightarrow R$ is a continuous positive semidefinite function on D . Then according to Lyapunov's direct method, the origin is stable but not necessarily asymptotically stable, since $W(x)$ is not strictly positive definite. Let,

$$E = \{x \in D : W(x) = 0\} \quad (8.40)$$

be a set of points in D where W is zero. Outside of E , $\dot{V}(x) < 0$. So, one may conjecture that the system trajectories that start outside of E will have to approach E , as time tends to infinity. This property (if it holds) would be equivalent to the

LaSalle's invariance theorem [1–3], which is valid for autonomous systems only. Moreover, if our conjecture holds and if $E = \{0\}$ then the origin would become asymptotically stable. Before we go any further, let us consider an example.

Example 8.15 In adaptive control, we will often encounter nonautonomous systems, such as

$$\begin{aligned}\dot{e} &= -e + (\theta - \theta_*) \varphi(t) \\ \dot{\theta} &= -e \varphi(t)\end{aligned}$$

where θ_* is a constant and $\varphi(t)$ is a bounded function of time t . This system has multiple equilibrium points of the form $(0, \theta_*)^T$. As it turns out later on, this particular system represents closed-loop tracking error dynamics of an adaptive controller for a first-order plant. We would like to prove that the error tends to zero, $e(t) \xrightarrow[t \rightarrow \infty]{} 0$, while the parameter $\theta(t)$ remains uniformly bounded in time. Consider a radially unbounded quadratic Lyapunov function candidate in the form,

$$V(e, \theta) = e^2 + (\theta - \theta_*)^2$$

and compute its time derivative along the system trajectories.

$$\begin{aligned}\dot{V}(e, \theta) &= 2e\dot{e} + 2(\theta - \theta_*)\dot{\theta} = 2e(-e + (\theta - \theta_*)\varphi(t)) + 2(\theta - \theta_*)(-e\varphi(t)) \\ &= -2e^2 \leq 0\end{aligned}$$

So $V(x(t))$ is decreasing, as a function of time, and therefore, both $e(t)$ and $\theta(t)$ are uniformly bounded. Note that V will continue to decrease until $e \neq 0$. Since V is lower-bounded and decreasing, it must tend to a limit, which may not necessarily be zero. If we can prove that in addition to the function having a limit, its derivative tends to zero, we could argue that since the derivative is proportional to e^2 , then e tends to zero as well. The property that relates functions with a limit and their derivatives is given by Barbalat's lemma. Essentially, the lemma states that if a time-dependent function tends to a limit and if its time derivative is uniformly continuous then the derivative tends to zero. We shall formulate Barbalat's lemma and then return to complete the example. ■

We begin with the definition of uniform continuity for a scalar function.

Definition 8.12 (Uniform Continuity) A function $f(t) : R \rightarrow R$ is said to be uniformly continuous if

$$\forall \varepsilon > 0 \quad \exists \delta = \delta(\varepsilon) > 0 \quad \forall |t_2 - t_1| \leq \delta \Rightarrow |f(t_2) - f(t_1)| \leq \varepsilon$$

Note that t_1 and t_2 play a symmetric role in the definition above. The uniform continuity concept should be compared to the definition of continuity at a point t , where $\delta = \delta(\varepsilon, t)$ becomes t -dependent.

It is not difficult to show that for a scalar continuously differentiable function to become uniformly continuous it is sufficient to verify that the function derivative is bounded (Exercise 8.9). This fact becomes important during stability proofs for adaptive controllers.

We now state Barbalat's lemma, whose formal proof can be found in [2].

Lemma 8.1 (Barbalat) *Let $f : R \rightarrow R$ be a uniformly continuous function on $[0, \infty)$. Suppose that $\lim_{t \rightarrow \infty} \int_0^t f(\tau) d\tau$ exists and is finite. Then $\lim_{t \rightarrow \infty} f(t) = 0$.* ■

It is interesting to note that Barbalat's lemma is in some ways analogous to the well-known fact for converging infinite series $\sum_{k=1}^{\infty} a_k$, where $\lim_{k \rightarrow \infty} a_k = 0$ represents a necessary condition for the series to have a finite value.

For continuously differentiable functions, Barbalat's lemma can be restated as follows.

Lemma 8.2 *Let $f : R \rightarrow R$ be continuously differentiable on $[0, \infty)$, and suppose that $\lim_{t \rightarrow \infty} f(t)$ exists and is finite. If the function derivative $\dot{f}(t)$ is uniformly continuous on $[0, \infty)$ then $\lim_{t \rightarrow \infty} \dot{f}(t) = 0$.* ■

An immediate and a very practical corollary of Barbalat's lemma can now be stated.

Corollary 8.1 *If a scalar function $f : R \rightarrow R$ is twice continuously differentiable on $[0, \infty)$, has a finite limit, $\lim_{t \rightarrow \infty} f(t) < \infty$, and the function second derivative is bounded then $\lim_{t \rightarrow \infty} \ddot{f}(t) = 0$.* ■

In general, the fact that derivative of a function tends to zero does not imply that the function itself has a limit. Also, the converse is not true. In fact, as the following examples show, there are no generic relations between functions and their derivatives.

Example 8.16 As $t \rightarrow \infty$, $f(t) = \sin(\ln t)$ does not have a limit, yet its derivative $\dot{f}(t) = \frac{\cos(\ln t)}{t}$ tends to zero. On the other hand, $f(t) = e^{-t} \sin(e^{2t})$ does tend to zero as $t \rightarrow \infty$. However, its derivative $\dot{f}(t) = -e^{-t} \sin(e^{2t}) + e^t \cos(e^{2t})$ tends to infinity. ■

Example 8.15 (continued) Previously, we have shown that the time derivative of the Lyapunov function candidate $V(e, \theta) = e^2 + (\theta - \theta_*)^2$ along the system trajectories

was negative semidefinite: $\dot{V}(e, \theta) = -2e^2 \leq 0$. The second time derivative of V is: $\ddot{V}(e, \theta) = -4e\dot{e} = -4e(-e + (\theta - \theta_*)\varphi(t))$. Since $\varphi(t)$ is bounded by hypothesis, and $e(t)$ and $\theta(t)$ were shown to be bounded, it is clear that $\ddot{V}(x(t))$ is uniformly bounded. Hence, $\dot{V}(x(t))$ is uniformly continuous. Also, it was shown that $V(x(t))$, as a function of time, tends to a limit. Then, by Barbalat's lemma, $\dot{V}(x(t)) \xrightarrow[t \rightarrow \infty]{} 0$, which in turn indicates that $e(t)$ tends to zero, as $t \rightarrow \infty$. ■

We now return to completion of stability analysis for the system (8.38), where we have assumed that a Lyapunov function was found to satisfy the inequality in (8.39). We proceed with Lyapunov-based arguments. Since $V(x(t))$ is bounded from below and $\dot{V}(x(t)) \leq 0$ then the function has a limit, as $t \rightarrow \infty$, and the system state x is uniformly bounded. Next, we show that $\dot{V}(x(t))$ is uniformly continuous. We cannot differentiate an inequality such as (8.39). Instead, we integrate it from t_0 to t .

$$V(x(t)) - V(x(t_0)) \leq - \int_{t_0}^t W(x(\tau)) d\tau \leq 0 \quad (8.41)$$

Rearranging terms, gives

$$\int_{t_0}^t W(x(\tau)) d\tau \leq V(x(t_0)) - V(x(t)) \leq V(x(t_0)) < \infty \quad (8.42)$$

and, consequently

$$\lim_{t \rightarrow \infty} \int_{t_0}^t W(x(\tau)) d\tau < \infty \quad (8.43)$$

Since $x(t)$ is uniformly bounded and $f(t, x(t))$ is Lipschitz-continuous uniformly in x then $x(t)$ is uniformly continuous in t . Moreover, in view of the fact that $W(x(t))$ is continuous in x , $W(x(t))$ becomes uniformly continuous in t . This property, coupled with (8.43), allows for direct application of Barbalat's lemma, which in this case states that $\lim_{t \rightarrow \infty} W(x(t)) = 0$. In other words, the system trajectories asymptotically approach the set E defined in (8.40), uniformly in time. We have just proved a special case of LaSalle–Yoshizawa theorem [1], with a Lyapunov function that did not explicitly depend on time.

Theorem 8.7 (LaSalle–Yoshizawa) *Starting anywhere in a domain D , all trajectories of the nonautonomous dynamics (8.38), with a Lyapunov function satisfying (8.39), uniformly asymptotically approach the set E from (8.40).* ■

Example 8.17 In adaptive control design, we will encounter n -dimensional nonautonomous systems in the form,

$$\begin{aligned}\dot{e} &= A e + b \underbrace{\left(\hat{K} - K \right)}_{\Delta K}^T \Phi(t) \\ \dot{\hat{K}} &= -\gamma \Phi(t) e^T P b\end{aligned}$$

where $e \in R^n$ is the system tracking error, $\hat{K} \in R^N$ is the adaptive N -dimensional vector of gains, γ is a constant positive adaptation rate, $K \in R^N$ is a constant vector of ideal (unknown) gains, $A \in R^{n \times n}$ is Hurwitz, $b \in R^n$ is a constant vector chosen such that the pair (A, b) is controllable, $\Phi(t) \in R^N$ is the so-called “regressor vector”, which is assumed to be uniformly bounded and Lipschitz-continuous in t . Finally, $P \in R^{n \times n}$ is the unique symmetric positive-definite solution of the algebraic Lyapunov equation,

$$P A + A^T P = -Q$$

with a symmetric positive-definite matrix Q .

These dynamics can be viewed as a generalization of the scalar system that was presented and analyzed in Example 8.15. Our immediate goal is to prove uniform boundedness of all signals and global uniform stability of the origin. Let us consider a quadratic radially unbounded Lyapunov function candidate in the form,

$$V(e, \Delta K) = e^T P e + \frac{1}{\gamma} \Delta K^T \Delta K$$

and compute its time derivative along the system trajectories.

$$\begin{aligned}\dot{V}(e, \Delta K) &= \dot{e}^T P e + e^T P \dot{e} + 2 \frac{1}{\gamma} \Delta K^T \dot{\hat{K}} = \left(A e + b \Delta K^T \Phi \right)^T P e \\ &\quad + e^T P \left(A e + b \Delta K^T \Phi \right) - 2 \Delta K^T \Phi e^T P b = -e^T Q e \leq 0\end{aligned}$$

According to Lyapunov's direct method, this inequality implies global uniform stability of the origin, as well as uniform boundedness of $e(t)$ and $\Delta K(t)$. Then because of the system dynamics, $\dot{e}(t)$ is also uniformly bounded, and so the second time derivative of the Lyapunov function

$$\ddot{V}(e, \Delta K) = -2 e^T Q \dot{e}$$

is uniformly bounded. Therefore, $\dot{V}(e(t), \Delta K(t))$ is uniformly continuous in t . At the same time, since $V(e(t), \Delta K(t)) \geq 0$ and $\dot{V}(e(t), \Delta K(t)) \leq 0$, then the

Lyapunov function itself tends to a limit. Lastly, applying Barbalat's lemma (in the form of Corollary 8.1), gives

$$\lim_{t \rightarrow \infty} [e^T(t) P e(t)] = \lim_{t \rightarrow \infty} [\dot{V}(e(t), \Delta K(t))] = 0$$

and, consequently $\lim_{t \rightarrow \infty} \|e(t)\| = 0$, that is the system tracking error globally uniformly and asymptotically tends to the origin, while the rest of the signals remain uniformly bounded. In the forthcoming chapters, this key property will enable us to design stable robust adaptive controllers with predictable closed-loop performance. ■

8.8 Summary and Historical Remarks

Theoretical foundations of stability theory for a general class of nonlinear differential equations were developed and published by Alexander Mikhailovich Lyapunov in his doctoral thesis “On the general problem of the stability of motion”, which he defended at the University of Moscow in 1892. Lyapunov’s stability, along with its extensions due to LaSalle, Yoshizawa, Barbashin, and Krasovskii, provided the necessary framework for the development of adaptive control. For dynamical systems without equilibrium, the notion of uniform ultimate boundedness was introduced and analyzed using Lyapunov’s second method.

We would like to emphasize yet again that Barbalat’s lemma constitutes the corner stone for proving stability of adaptive systems. This lemma allows to assert asymptotic stability of the system tracking error based on two facts: (a) The error is square integrable, and (b) The error time derivative is uniformly bounded. Both statements come from application of Lyapunov’s second method to examine stability of the system error dynamics. Barbalat’s lemma has been independently derived by many authors, but the original work was attributed to Barbalat by V. M. Popov in his book “Hyperstability of control systems”, published by Springer-Verlag in 1973.

8.9 Exercises

Exercise 8.1 Starting at different initial conditions, simulate the system dynamics in Example 8.1. Comment on the system behavior near and at the origin.

Exercise 8.2 Derive the system solution in Example 8.1 and prove the stated three properties.

Exercise 8.3 Derive the system solution in Example 8.4 and draw the system phase portrait. Given an initial condition, find a local Lipschitz constant. Prove that the system dynamics is not globally Lipschitz.

Exercise 8.4 Prove that trajectories of any scalar autonomous ODE (assuming that they exist) are monotonic functions of time.

Exercise 8.5 Prove the statement from Example 8.6.

Exercise 8.6 For a scalar nonautonomous differential equation in the form $\dot{x} = -a(t)x$, define sufficient conditions on $a(t)$, so that the equilibrium of the scalar dynamics is: (a) stable; (b) asymptotically stable, and (c) uniformly asymptotically stable.

Exercise 8.7 Consider the system,

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -g(x_1)(x_1 + x_2)$$

where g is locally Lipschitz and $g(y) \geq 1$ for all $y \in R$. Verify that

$$V(x) = \int_0^{x_1} y g(y) dy + x_1 x_2 + x_2^2$$

is globally positive definite and radially unbounded. Use $V(x)$ to show that the system equilibrium point $x_* = 0$ is globally asymptotically stable.

Exercise 8.8 There are theoretical extensions that deal with existence and uniqueness of IVP-s whose system dynamics are discontinuous in x . Show that the IVP

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_2 - \operatorname{sgn}(x_1 + x_2) \end{cases}, \quad x(0) = x_0$$

does not satisfy the sufficient conditions for existence and uniqueness of its solution. Nevertheless, a solution does exist. Simulate the system starting from different initial conditions. Construct phase portrait of the system and argue that (a) the manifold $c(x) = x_1 + x_2 = 0$ is the system global attractor, (b) all system trajectories reach this manifold in finite time, and (c) the solution “slides” down the manifold toward the origin.

Exercise 8.9 Prove that a scalar continuously differentiable function is uniformly continuous if the function derivative is bounded. Using this fact, prove Corollary 8.1.

Exercise 8.10 Consider the n -dimensional LTI controllable system,

$$\dot{x} = A x + b \left(u - K_x^T x \right)$$

with a Hurwitz matrix A . Suppose that $K_x \in R^{n \times 1}$ is constant and unknown. Let $r(t) \in R$ denote a bounded external command for the system output $y = C x$ to follow. The system control input u is chosen as,

$$u = \hat{K}(t) x + K_r r(t)$$

where $K_r = -C A^{-1} b$ is the command feedforward gain, and $\hat{K}(t)$ is the time-variant state feedback gain, whose dynamics are given by the adaptive laws,

$$\dot{\hat{K}} = -\gamma x e^T P b$$

with a positive scalar γ and using the unique positive-definite solution $P \in R^{n \times n}$ of the Lyapunov algebraic equation $P A + A^T P = -Q$, where $Q \in R^{n \times n}$ is symmetric and positive definite. Let,

$$\dot{x}_{\text{ref}} = A x_{\text{ref}} + b K_r r$$

define the desired dynamics. The system tracking error is $e = x - x_{\text{ref}}$. Write down the tracking error dynamics. Formulate the total closed-loop dynamics by combining the tracking error dynamics with the adaptive laws. Prove that for any bounded command r , any constant positive adaptation rate γ , and any symmetric positive-definite matrix Q , the tracking error $e(t)$ tends to zero globally and asymptotically, that is the system state x tracks the desired state x_{ref} , with diminishing errors. Argue that in this case, the system output $y = C x$ tracks the command $r(t)$ with bounded errors and in the presence of any constant uncertain vector parameter K_x . (Hint: Use the stability arguments from Example 8.17.)

Exercise 8.11 Using the system and control equations from Exercise 8.10, choose $n = 2$, select your own data, and simulate the corresponding closed-loop system dynamics. Verify the theoretical predictions of stability and tracking, while driving the desired dynamics with various external bounded commands. Demonstrate (via simulation tests) the closed-loop system tracking performance in the presence of a constant uncertain parameter K_x . Discuss your results.

References

1. Haddad WM, Chellaboina V (2008) Nonlinear dynamical systems and control. A Lyapunov-based approach. Princeton University Press, Princeton, NJ
2. Khalil H (1996) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, NJ
3. Vidyasagar M (1993) Nonlinear systems analysis. Prentice Hall, New Jersey



State Feedback Direct Model Reference Adaptive Control

9

This chapter presents basic design concepts and analysis methods in the development of direct model reference adaptive controllers for uncertain systems with continuous dynamics and full state measurements. We begin our discussions with scalar systems and gradually transition to adaptive controllers for multi-input–multi-output dynamics with matched parametric uncertainties. In order to gain insights into the intricacy of these nonlinear systems and their expected behavior, we will consider several design examples of increasing complexity.

9.1 Introduction

For over fifty years, adaptive systems have decisively remained in the mainstream of controls and dynamics research. As a result, adaptive control has grown to become a well-formed scientific discipline. One of the reasons for the continuing popularity and rapid growth of adaptive control is its clearly defined goal—to enable control of dynamical systems that operate in the presence of unknown parameters.

Adaptive control research was initiated in the early 1950s. At that time, the interest in adaptive systems was primarily driven by the design of autopilots for high-performance aircraft. This was no surprise since newly designed aerial platforms required control solutions that would provide stable and predictable flight operations throughout the aircraft's vast envelope, ranging from subsonic to supersonic and even to hypersonic regions.

The last decade has witnessed the development of a coherent theory for adaptive control, which has led to many practical applications in the areas such as aerospace, robotics, chemical processes, ship steering, bioengineering, and many others.

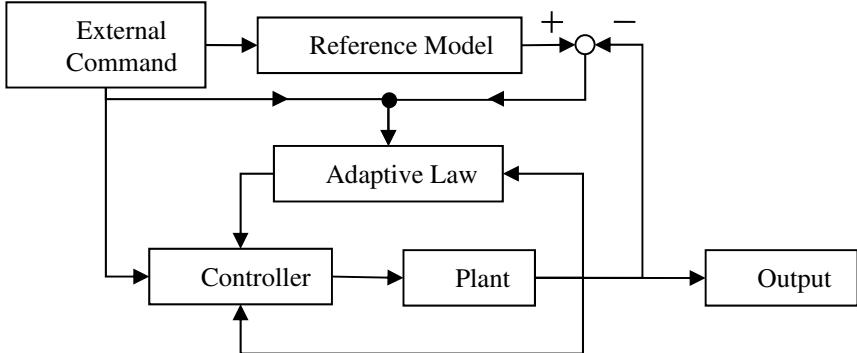


Fig. 9.1 MRAC closed-loop block diagram

A few historical remarks are in order. The original concept of a model reference adaptive system was proposed by Whitaker et al. in [1, 2]. The main idea behind this concept was to specify the desired command-to-output performance of a servotrack system that would eventually define the ideal response of the system output due to external commands. This control concept was later called the “explicit model following”, and the corresponding architecture became known as the model reference adaptive control, or in short MRAC. Soon after its introduction, the first proof of MRAC closed-loop stability using Lyapunov theory was given in 1965 by Butchart and Shackcloth [3] and also in 1966 by Parks [4]. In the years that followed, adaptive control theory for a broad class of dynamical uncertain systems was developed and well documented in several now-classical textbooks [5–8].

A generic block diagram of a system (plant) operating under MRAC controller is shown in Fig. 9.1.

In essence, an MRAC system consists of a controller whose parameters (gains) are updated online using an adaptive law. The latter operates on the system output and on an external command (a.k.a the reference input). The command also drives the reference model that specifies the desired trajectories for the system to follow. The difference between the reference model output and the system output constitutes the tracking error, which subsequently is sent to the adaptive law for online parameter adjustments. Finally, the controller computes its commands based on the reference input, the system output, and the online adjusted parameters from the adaptive law. Per design, the adaptive controller forces the system output to follow the desired external commands, while operating in the presence of the plant uncertainties. For itself, the controller main objective is to maintain consistent performance of the closed-loop system in the presence of uncertainties and unknown variations in plant parameters.

When the true plant parameters are unknown, one might attempt to estimate control gains online using available measurements. This approach is referred to as the “direct”. Alternatively, the gains can be approximated online by solving

system design equations that relate the plant uncertainties to the known signals in the system. This is called the “indirect” method. MRAC systems can be designed using either direct or indirect approaches. There are also design methods available that merge the two, leading to combined (direct + indirect) MRAC architectures.

In this chapter, our focus will be on the design, analysis, and evaluation of direct MRAC systems for continuous plants with uncertain dynamics and full state measurements.

9.2 Command Tracking

We shall consider command tracking algorithms for continuous dynamic plants,

$$\begin{aligned}\dot{x} &= f(t, x, u, \Theta, \xi) \\ y &= h(t, x, u, \Theta, \xi)\end{aligned}\tag{9.1}$$

with vector-parametric constant uncertainties Θ and with bounded environmental disturbances $\xi(t)$. In (9.1), $x \in R^n$ denotes the system state, $u \in R^m$ is the control input, and $y \in R^m$ is the regulated output. It is assumed that the entire system state vector x is available for control synthesis. In other words, the system state can be measured online.

The problem of tracking a command involves the design of the system control input u so that the regulated output $y(t)$ tracks a given bounded reference signal $r(t) \in R^m$, in the presence of the system uncertainties Θ and environmental disturbances $\xi(t)$. Specifically, we are looking for a control input that would force the output tracking error

$$e_y(t) = y(t) - r(t)\tag{9.2}$$

to become sufficiently small, as $t \rightarrow \infty$. Moreover, it is required that during tracking, all the signals in the corresponding closed-loop system remain uniformly bounded in time.

If $e_y(t) \xrightarrow[t \rightarrow \infty]{} 0$ then we assert that asymptotic output tracking has been achieved. In general, asymptotic tracking may not be feasible, and in that case, our goal would be to achieve uniform ultimate boundedness of the tracking error,

$$\|e_y(t)\| \leq \varepsilon, \quad \forall t \geq T\tag{9.3}$$

where $\varepsilon > 0$ is the desired tracking tolerance, T is a finite time instant, and $\|\cdot\|$ denotes a vector norm. A brief review of vector norms was given in Chap. 1.

9.3 Direct MRAC Design for Scalar Systems

We begin with a scalar plant whose dynamics are of the form,

$$\dot{x} = a x + b(u + f(x)) \quad (9.4)$$

where x is the system state, u is the control input, while a and b represent unknown constant parameters. We assume that the sign of b is known, which is equivalent to saying that the system is controllable. The system dynamics depend on the unknown function $f(x)$ defined as a linear combination of N known basis functions $\varphi_i(x)$ with N unknown constants θ_i ,

$$f(x) = \sum_{i=1}^N \theta_i \varphi_i(x) = \theta^T \Phi(x) \quad (9.5)$$

where $\Phi(x) = (\varphi_1(x) \dots \varphi_N(x))^T \in R^N$ denotes the known regressor vector, whose components $\varphi_i(x)$ are assumed to be Lipschitz-continuous in x . So, the scalar model we consider here is

$$\dot{x} = a x + b(u + \theta^T \Phi(x)) \quad (9.6)$$

A stable reference model is given. Its dynamics are described by a first-order differential equation in the form,

$$\dot{x}_{\text{ref}} = a_{\text{ref}} x_{\text{ref}} + b_{\text{ref}} r(t) \quad (9.7)$$

where $a_{\text{ref}} < 0$ and b_{ref} are the desired constants and $r(t)$ is the reference input command. The reference model parameters must be chosen to represent the desired response due to bounded commands. For example, the designer may select $b_{\text{ref}} = -a_{\text{ref}}$ so that the DC gain of the reference dynamics becomes unity, and then select a_{ref} such that the reference system time constant is as small as desired.

The control objective of interest is to asymptotically track the state x_{ref} of the reference model (9.7), which can be driven by any bounded command $r(t)$. In other words, we need to design a control law $u(t)$, such that the state tracking error $e(t) = x(t) - x_{\text{ref}}(t)$ globally uniformly asymptotically tends to zero, as $t \rightarrow \infty$, while all signals in the corresponding closed-loop system remain uniformly ultimately bounded in time.

The required command tracking task must be accomplished in the presence of $(N + 2)$ unknown constant parameters $\{a, b, \theta_1, \dots, \theta_N\}$.

First, we define the “ideal” control solution, as if the unknown parameters were known. The ideal control is composed using the (feedback + feedforward) architecture,

$$u_{\text{ideal}} = k_x x + k_r r - \theta^T \Phi(x) \quad (9.8)$$

where k_x and k_r represent the ideal feedback and feedforward gains, respectively. Substituting (9.8) into (9.6) gives the system closed-loop dynamics.

$$\dot{x} = (a + b k_x)x + b k_r r(t) \quad (9.9)$$

Comparing (9.9) with the desired reference model dynamics (9.7), it follows that the ideal gains k_x and k_r must satisfy the following two algebraic equations:

$$a + b k_x = a_{\text{ref}} \quad b k_r = b_{\text{ref}} \quad (9.10)$$

These relations are called the matching conditions. It is clear that for scalar plants, the unknown ideal gains, k_x and k_r , in (9.10) always exist. As we shall see later, this will not be the case for multi-dimensional dynamics.

Based on (9.8), we propose a tracking control solution in the form,

$$u = \hat{k}_x x + \hat{k}_r r - \hat{\theta}^T \Phi(x) \quad (9.11)$$

where the feedback gain \hat{k}_x , the feedforward gain \hat{k}_r , and the estimated vector of parameters $\hat{\theta}$ will be determined to achieve global uniform asymptotic tracking of the reference model trajectories. Toward that end, we substitute (9.11) into the system dynamics (9.6),

$$\dot{x} = (a + b \hat{k}_x)x + b \left(\hat{k}_r r - (\hat{\theta} - \theta)^T \Phi(x) \right) \quad (9.12)$$

and rewrite the latter using the matching conditions (9.10),

$$\dot{x} = a_{\text{ref}} x + \underbrace{b k_r r}_{b_m} + b \underbrace{(\hat{k}_x - k_x)}_{\Delta k_x} x + b \underbrace{(\hat{k}_r - k_r)}_{\Delta k_r} r - b \underbrace{(\hat{\theta} - \theta)^T \Phi(x)}_{\Delta \theta} \quad (9.13)$$

where

$$\Delta k_x = \hat{k}_x - k_x, \quad \Delta k_r = \hat{k}_r - k_r, \quad \Delta \theta = \hat{\theta} - \theta \quad (9.14)$$

denote parameter estimation errors. Then, the closed-loop dynamics of the system tracking error signal

$$e(t) = x(t) - x_{\text{ref}}(t) \quad (9.15)$$

can be obtained by subtracting (9.7) from (9.13).

$$\dot{e}(t) = a_{\text{ref}} e + b(\Delta k_x x + \Delta k_r r - \Delta \theta^T \Phi(x)) \quad (9.16)$$

We are going to choose adaptive gains $(\hat{k}_x, \hat{k}_r, \hat{\theta})$ to enforce global uniform asymptotic stability of the origin. This will be accomplished through the inverse Lyapunov design approach, where we would choose a Lyapunov function candidate, and then select adaptive laws such that the function time derivative becomes non-positive, when evaluated along the trajectories of the error dynamics (9.16). As a result, the tracking error would asymptotically converge to the origin, and so the system state would asymptotically track the state of the reference model.

Let us consider a quadratic Lyapunov function candidate in the form,

$$V(e, \Delta k_x, \Delta k_r, \Delta \theta) = e^2 + |b| \left(\gamma_x^{-1} \Delta k_x^2 + \gamma_r^{-1} \Delta k_r^2 + \Delta \theta^T \Gamma_\theta^{-1} \Delta \theta \right) \quad (9.17)$$

where scalars $\gamma_x > 0$, $\gamma_r > 0$ and a constant symmetric positive-definite matrix $\Gamma_\theta \in R^{N \times N}$ are the rates of adaptation. Taking the time derivative of V , along the trajectories of (9.16), gives

$$\begin{aligned} \dot{V}(e, \Delta k_x, \Delta k_r, \Delta \theta) &= 2e \dot{e} + 2|b| \left(\gamma_x^{-1} \Delta k_x \dot{\hat{k}}_x + \gamma_r^{-1} \Delta k_r \dot{\hat{k}}_r + \Delta \theta^T \Gamma_\theta^{-1} \dot{\hat{\theta}} \right) \\ &= 2e(a_{\text{ref}} e + b(\Delta k_x x + \Delta k_r r - \Delta \theta^T \Phi(x))) \\ &\quad + 2|b| \left(\gamma_x^{-1} \Delta k_x \dot{\hat{k}}_x + \gamma_r^{-1} \Delta k_r \dot{\hat{k}}_r + \Delta \theta^T \Gamma_\theta^{-1} \dot{\hat{\theta}} \right) \\ &= 2a_{\text{ref}} e^2 + 2|b| \left(\Delta k_x \left(x e \text{sgn}(b) + \gamma_x^{-1} \dot{\hat{k}}_x \right) \right) \\ &\quad + 2|b| \left(\Delta k_r \left(r e \text{sgn}(b) + \gamma_r^{-1} \dot{\hat{k}}_r \right) \right) \\ &\quad + 2|b| \Delta \theta^T \left(-\Phi(x)e \text{sgn}(b) + \Gamma_\theta^{-1} \dot{\hat{\theta}} \right) \end{aligned} \quad (9.18)$$

In order to enforce closed-loop stability, it is sufficient to choose adaptive laws such that $\dot{V}(e, \Delta k_x, \Delta k_r, \Delta \theta) \leq 0$. Indeed, if we select,

$$\begin{aligned} \dot{\hat{k}}_x &= -\gamma_x x e \text{sgn}(b) \\ \dot{\hat{k}}_r &= -\gamma_r r e \text{sgn}(b) \\ \dot{\hat{\theta}} &= \Gamma_\theta \Phi(x) e \text{sgn}(b) \end{aligned} \quad (9.19)$$

then the time derivative of V , computed along the trajectories of (9.16), becomes negative semidefinite,

$$\dot{V}(e, \Delta k_x, \Delta k_r, \Delta \theta) = 2 \underbrace{a_{\text{ref}} e(t)^2}_{<0} \leq 0 \quad (9.20)$$

which immediately implies that the signals $(e, \Delta k_x, \Delta k_r, \Delta \theta)$ are uniformly bounded in time. The latter, coupled with the fact that (x_{ref}, r) are bounded and θ is a constant vector, means that the system state x and the estimated vector of parameters $\hat{\theta}$ are uniformly bounded. Moreover, since the components $\varphi_i(x)$ of the

regressor vector $\Phi(x)$ are Lipschitz-continuous functions of x , which was proven to be bounded, then the regressor components themselves are uniformly bounded. Hence, the control signal u in (9.11) is uniformly bounded as well. Consequently, both \dot{x} and \dot{x}_{ref} are uniformly bounded.

Differentiating (9.20) results in

$$\ddot{V}(e, \Delta k_x, \Delta k_r, \Delta \theta) = 4 a_{\text{ref}} e(t) \dot{e}(t) \quad (9.21)$$

Therefore, \ddot{V} is bounded, and consequently, \dot{V} is a uniformly continuous function of time. Since V is lower bounded and \dot{V} is negative semidefinite then V , as a function of time, must have a finite limit. We can now use Barbalat's lemma to arrive at,

$$\lim_{t \rightarrow \infty} \dot{V}(t) = 0 \quad (9.22)$$

and because of (9.20), we conclude that the tracking error $e(t)$ tends to zero asymptotically, as $t \rightarrow \infty$.

Since the Lyapunov function (9.17) is radially unbounded and it does not depend explicitly on time, the attained asymptotic stability property is global and uniform, that is, the closed-loop tracking error dynamics are globally uniformly asymptotically stable. The command tracking problem is solved. We now recap our formally proven results in Theorem 9.1.

Theorem 9.1 *For the uncertain scalar dynamical system (9.6), with the controller (9.11), and the adaptive laws (9.19), the system state $x(t)$ asymptotically tracks the state $x_{\text{ref}}(t)$ of the reference model (9.7), driven by any bounded command $r(t)$, while all the signals in the closed-loop system remain uniformly bounded. Moreover, the closed-loop tracking error dynamics (9.16) are globally uniformly asymptotically stable.* ■

MRAC design equations for a scalar plant are summarized in Table 9.1.

It is necessary to make a remark about dynamic behavior of the estimated parameters $\hat{\theta}$. The fact that the system tracking error e asymptotically tends to

Table 9.1 Direct MRAC design summary for a scalar plant

Open-loop plant	$\dot{x} = a x + b(u + \theta^T \Phi(x))$
Reference model	$\dot{x}_{\text{ref}} = a_{\text{ref}} x_{\text{ref}} + b_{\text{ref}} r$
Tracking error	$e = x - x_{\text{ref}}$
Control input	$u = \hat{k}_x x + \hat{k}_r r - \hat{\theta}^T \Phi(x)$
Direct MRAC laws	$\begin{aligned} \dot{\hat{k}}_x &= -\gamma_x x e \operatorname{sgn}(b) \\ \dot{\hat{k}}_r &= -\gamma_r r e \operatorname{sgn}(b) \\ \dot{\hat{\theta}} &= \Gamma_\theta \Phi(x) e \operatorname{sgn}(b) \end{aligned}$

zero does not automatically imply that $\hat{\theta}$ converges to its ideal unknown parameter vector θ . What is certain is that the estimated parameters will remain uniformly bounded during tracking. Nevertheless, there are cases when parameter convergence will take place alongside the desired tracking. A sufficient condition for parameter convergence is given by the persistency of excitation (PE) [5–7], which imposes certain restrictions on the commanded signal $r(t)$. We shall define and discuss PE conditions at a later time.

As in any other control design method, MRAC has its own “tuning knobs”. They are the rates of adaptations, represented by two positive constants (γ_x, γ_r) and a symmetric positive-definite matrix Γ_θ . As seen from (9.19), the larger the rates the faster the adaptive laws will evolve. One may conjecture that large rates would result in better and faster closed-loop tracking performance. This is partially true. Indeed, large rates of adaptation will yield fast tracking. However, this will also lead to undesirable oscillations during transient times, when the system regulated output is trying to get closer to its command. The tradeoff between fast tracking and smooth transients presents a design challenge.

Example 9.1 Helicopter Pitch Dynamics and Control During Hover Unlike a fixed-wing airplane, angular motion control of a helicopter is achieved by tilting its main rotor and, as a result, altering the direction of the rotor thrust vector. This action induces a change in angular moments acting on the vehicle and results in pitch, roll, and yaw angular motion.

In hover, the helicopter pitch dynamics depend primarily on the vehicle pitch rate q and on the applied (by a pilot or an automatic system) longitudinal control input δ , which is equivalent in its effect (also induces a pitching motion) to an elevator for a fixed-wing vehicle. Assuming constant thrust, while neglecting small forward and vertical speed components, pitch dynamics of a helicopter during hover can be approximated by the following scalar differential equation:

$$\dot{q} = M_q q + M_\delta(\delta + f(q))$$

where M_q represents the vehicle pitch damping and M_δ is the elevator effectiveness. The system also depends on the unknown function $f(q)$, which models inherent uncertainties in the helicopter dynamics, both linear and nonlinear.

For simulation purposes, we assume model parameters (unknown constants) that are representative of a hovering transport helicopter: $M_q = -0.61$ (rad/s) and $M_\delta = -6.65$ (rad/s²). We also define,

$$f(q) = \underbrace{-0.01}_\theta \underbrace{\tanh\left(\frac{360}{\pi}q\right)}_{\Phi(q)} = \theta \Phi(q)$$

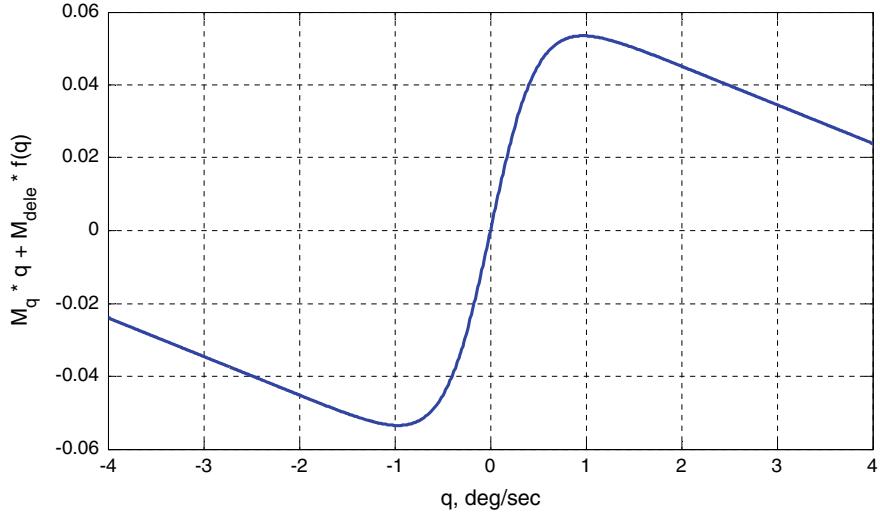


Fig. 9.2 Helicopter open-loop pitch dynamics from Example 9.1

where $\theta = -0.01$ is unknown and $\Phi(q)$ is the known regressor, and arrive at the helicopter dynamics

$$\dot{q} = -0.61q - 6.65 \left(\delta - 0.01 \tanh\left(\frac{360}{\pi}q\right) \right)$$

Clearly, the origin of the open-loop ($\delta = 0$) pitch dynamics becomes locally unstable, as shown in Fig. 9.2.

Such a system would certainly require active control for stabilization and command tracking.

Our particular selection of the system parameters in this example is purely academic. It merely supports the main objective here—to design an MRAC system and to show its efficacy in coping with linear and nonlinear uncertainties of various forms and shapes. Toward that end, we use MRAC design equations from Table 9.1 and construct the following adaptive pitch controller,

$$\delta = \hat{k}_q q + \hat{k}_{q_{cmd}} q_{cmd} - \hat{\theta}^T \Phi(q)$$

with the adaptive laws,

$$\begin{aligned}\dot{\hat{k}}_q &= \gamma_q q(q - q_{ref}) \\ \dot{\hat{k}}_{q_{cmd}} &= \gamma_{q_{cmd}} q_{cmd}(q - q_{ref}) \\ \dot{\hat{\theta}} &= -\Gamma_\theta \Phi(q)(q - q_{ref})\end{aligned}$$

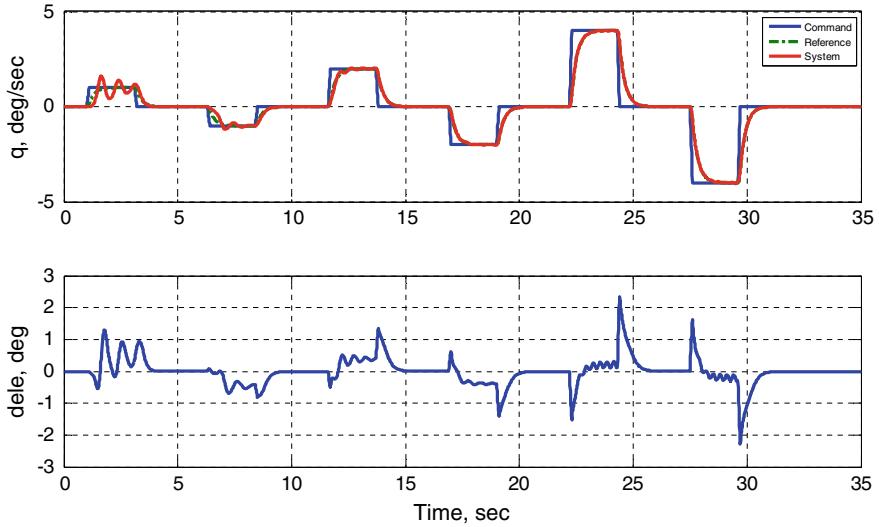


Fig. 9.3 Closed-loop pitch rate command tracking performance in Example 9.1

where q_{ref} is the desired pitch rate signal generated by the reference model,

$$\dot{q}_{\text{ref}} = 4(q_{\text{cmd}} - q_{\text{ref}})$$

driven by any bounded time-varying pitch rate command $q_{\text{cmd}} = q_{\text{cmd}}(t)$. Here, we have selected $a_{\text{ref}} = -b_{\text{ref}} = -4$.

After several design iterations, we have chosen the rates of adaptation to be: $\gamma_q = \gamma_{q_{\text{cmd}}} = 6000$, $\Gamma_\theta = 8$. Figure 9.3 shows the closed-loop system response (pitch rate, degree/seconds) and the required control effort (elevator deflection, degree) for tracking a series of step-input commands of different magnitudes.

During this event, the adaptive parameters (solid green, Fig. 9.4) remain bounded and approach their true unknown values (dashed blue, Fig. 9.4).

The observed parameter convergence is not guaranteed to always take place. For example, suppose that the same system is required to track a sinusoidal command (Fig. 9.5).

In this case, the adaptive parameters do not converge to their ideal values (Fig. 9.6). However, they do remain uniformly bounded, as expected.

It is also interesting to compare the adaptive elevator input (Fig. 9.3) against the ideal signal generated by the fixed-gain controller.

$$\delta_{\text{ideal}} = \frac{1}{M_\delta} ((a_{\text{ref}} - M_q)q + b_{\text{ref}} q_{\text{cmd}}) - f(q)$$

Figure 9.7 shows the comparison data.

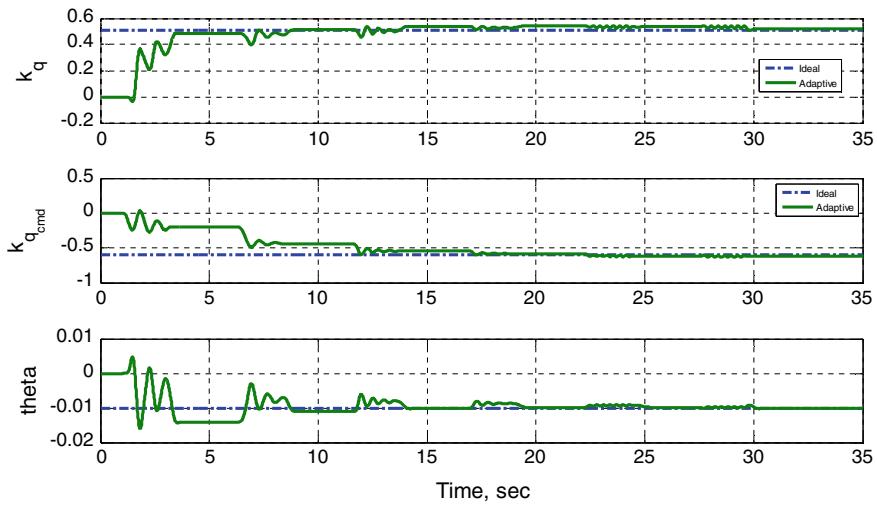


Fig. 9.4 Adaptive parameters from Example 9.1 converge to their ideal values

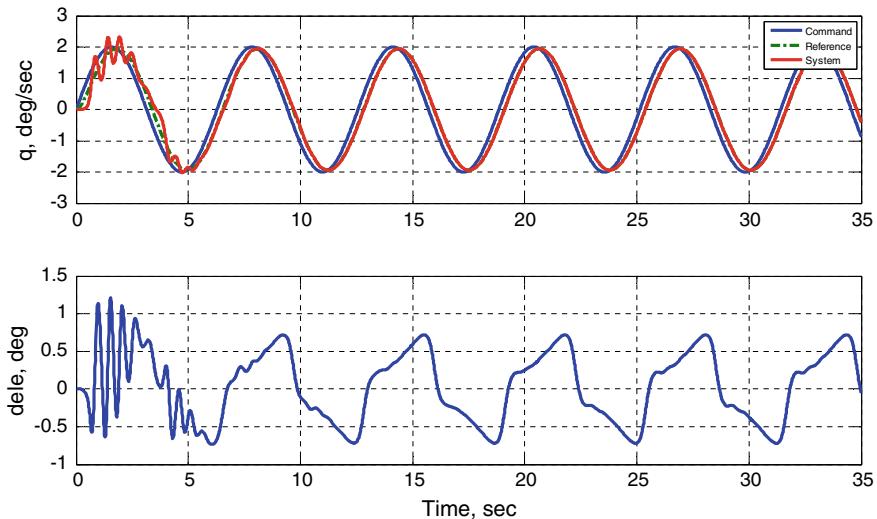


Fig. 9.5 Sinusoidal command tracking from Example 9.1

In spite of clearly visible similarities, the two control signals also exhibit subtle differences. Even after the transients have subsided, the MRAC signal tends to oscillate (Fig. 9.8), while the fixed-gain controller does not.

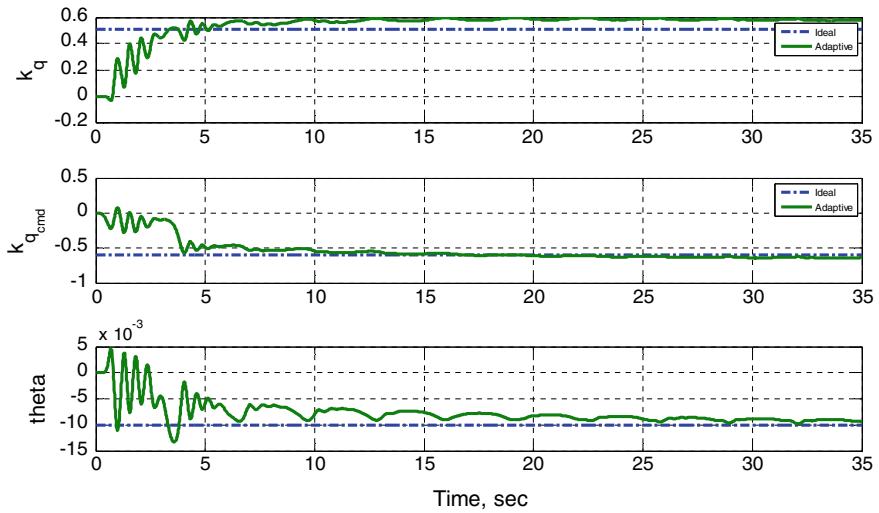


Fig. 9.6 Adaptive parameters during sinusoidal command tracking from Example 9.1

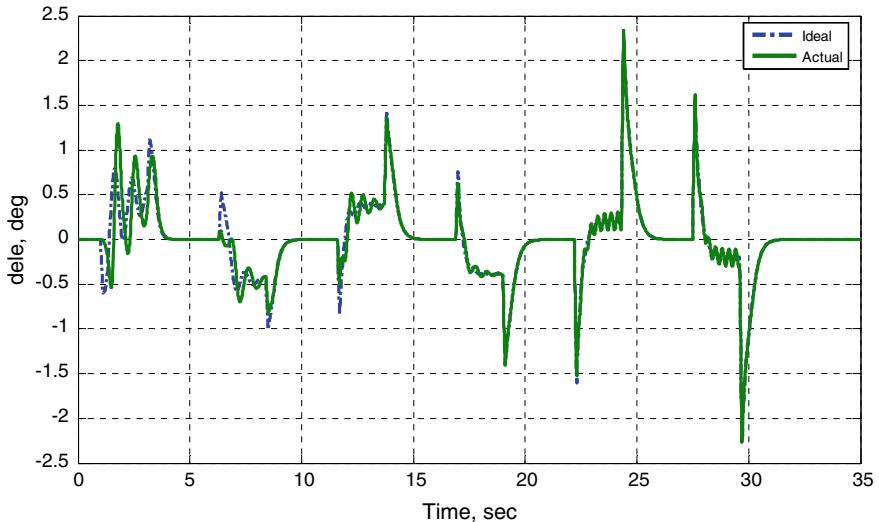


Fig. 9.7 Adaptive and ideal fixed-gain controllers from Example 9.1

Summarizing our discussions, we end this example with an observation that in the two simulation scenarios considered, the pitch rate MRAC system was able to provide adequate closed-loop command tracking performance, while operating in the presence of linear and nonlinear uncertainties. ■

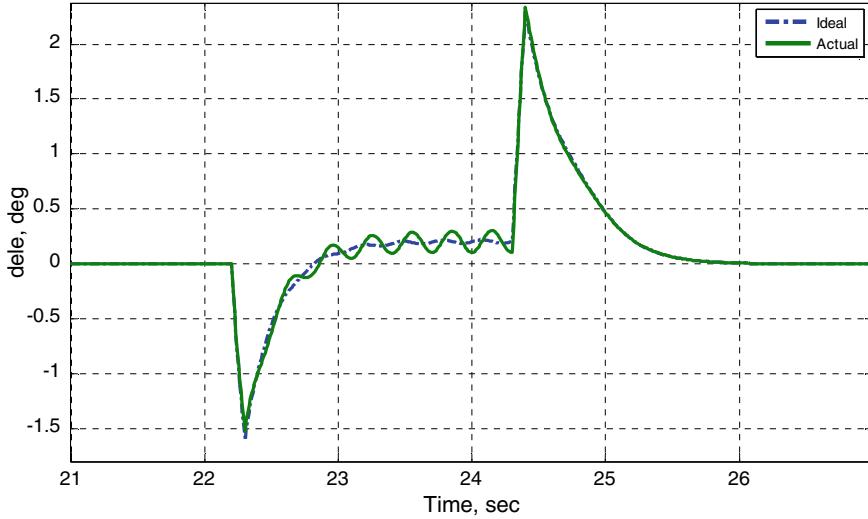


Fig. 9.8 MRAC signal oscillations in Example 9.1

9.4 Dynamic Inversion MRAC Design for Scalar Systems

Dynamic inversion (DI) control for systems with known dynamics represents a well-known method [9]. In this section, we will demonstrate the design of a DI-based MRAC system for the scalar uncertain dynamical system,

$$\dot{x} = a x + b u + \underbrace{\theta^T \Phi(x)}_{f(x)} \quad (9.23)$$

with two unknown constants (a, b) and with an unknown function $f(x)$ in the form of (9.5). Again we assume that the constant vector of ideal parameters θ is not known, while the regressor components $\varphi_i(x)$ represent a known set of Lipschitz-continuous basis functions. We also assume that $\text{sgn } b$ is known and that the system is controllable, that is, $|b| \geq b_{\min} > 0$, where b_{\min} represents a known lower bound of $|b|$.

The reference model dynamics are given by (9.7), and the design task remains the same—find a control input u to force the system state x asymptotically track the state of the reference model, which is in turn driven by any bounded time-varying command r .

This particular control problem was addressed and solved in the previous section, where we derived a direct MRAC system. Here, we shall present an alternative solution and then compare the two approaches.

We begin by rewriting the system dynamics,

$$\dot{x} = \hat{a}x + \hat{b}u + \hat{f}(x) - \underbrace{(\hat{a} - a)}_{\Delta a}x - \underbrace{(\hat{b} - b)}_{\Delta b}u - \underbrace{(\hat{f}(x) - f(x))}_{\Delta f(x)} \quad (9.24)$$

where \hat{a}, \hat{b} are the estimated values, and

$$\hat{f}(x) = \sum_{i=1}^N \hat{\theta}_i \varphi_i(x) = \hat{\theta}^T \Phi(x) \quad (9.25)$$

is the function approximator. All these quantities will be constructed during the design process. Also in (9.24), Δa , Δb , and $\Delta f(x)$ represent the parameter and the function approximation errors, respectively. Using (9.25) gives the function approximation error.

$$\Delta \hat{f}(x) = \hat{f}(x) - f(x) = \sum_{i=1}^N \underbrace{(\hat{\theta}_i - \theta_i)}_{\Delta \theta_i} \varphi_i(x) = \Delta \theta^T \Phi(x) \quad (9.26)$$

Following the DI method, let us consider a controller in the form

$$u = \frac{1}{\hat{b}} \left((a_{\text{ref}} - \hat{a})x + b_{\text{ref}} r - \hat{\theta}^T \Phi(x) \right) \quad (9.27)$$

Substituting (9.27) into the second term of (9.24) yields

$$\dot{x} = a_{\text{ref}}x + b_{\text{ref}}r - \Delta a x - \Delta b u - \Delta \theta^T \Phi(x) \quad (9.28)$$

With the tracking error signal e defined in (9.15), we can now compute the system tracking error dynamics. Subtracting (9.7) from (9.28) gives

$$\dot{e} = a_{\text{ref}}e - \Delta a x - \Delta b u - \Delta \theta^T \Phi(x) \quad (9.29)$$

Consider the following Lyapunov function candidate:

$$V(e, \Delta a, \Delta b, \Delta \theta) = e^2 + \gamma_a^{-1} \Delta a^2 + \gamma_b^{-1} \Delta b^2 + \Delta \theta^T \Gamma_\theta^{-1} \Delta \theta \quad (9.30)$$

where $\gamma_a > 0$, $\gamma_b > 0$, $\Gamma_\theta = \Gamma_\theta^T > 0$ are the adaptation rates. The time derivative of V , evaluated along the trajectories of the error dynamics (9.29), can be computed as

$$\begin{aligned}
\dot{V}(e, \Delta a, \Delta b, \Delta \theta) &= 2e\dot{e} + 2\left(\gamma_a^{-1}\Delta a \dot{a} + \gamma_b^{-1}\Delta b \dot{\hat{b}} + \Delta \theta^T \Gamma_\theta^{-1} \dot{\hat{\theta}}\right) \\
&= 2e(a_{\text{ref}}e - \Delta a x - \Delta b u - \Delta \theta^T \Phi(x)) \\
&\quad + 2\left(\gamma_a^{-1}\Delta a \dot{a} + \gamma_b^{-1}\Delta b \dot{\hat{b}} + \Delta \theta^T \Gamma_\theta^{-1} \dot{\hat{\theta}}\right) \\
&= 2a_{\text{ref}}e^2 + \Delta a\left(\gamma_a^{-1}\dot{\hat{a}} - x e\right) \\
&\quad + \Delta b\left(\gamma_b^{-1}\dot{\hat{b}} - u e\right) + \Delta \theta^T\left(\Gamma_\theta^{-1}\dot{\hat{\theta}} - \Phi(x)e\right) \quad (9.31)
\end{aligned}$$

Based on (9.31) and in order to make $\dot{V} \leq 0$, the adaptive laws are chosen as

$$\dot{a} = \gamma_a x e, \quad \dot{\hat{b}} = \gamma_b u e, \quad \dot{\hat{\theta}} = \Gamma_\theta \Phi(x)e \quad (9.32)$$

Then

$$\dot{V}(e, \Delta a, \Delta b, \Delta \theta) = 2a_{\text{ref}}e^2 \leq 0 \quad (9.33)$$

and consequently, the four signals $(e, \Delta a, \Delta b, \Delta \theta)$ are uniformly bounded. Since $r(t)$ is bounded and $a_{\text{ref}} < 0$, then x_{ref} is also uniformly bounded, and because of that the system state x as well as the three estimated signals $(\hat{a}, \hat{b}, \hat{\theta})$ is uniformly bounded.

In order to claim uniform boundedness of u from (9.27), we need to protect the controller from “blowing up” due to the division by \hat{b} . In other words, we need to modify the adaptive laws (9.32) and enforce boundedness of the estimated parameter \hat{b} .

Let us consider the following modification of the second equation in the adaptive laws (9.32):

$$\dot{\hat{b}} = \begin{cases} \gamma_b u e, & \text{if } |\hat{b}| > b_{\min} \vee [\hat{b} = b_{\min} \operatorname{sgn} b \wedge (u e) \operatorname{sgn} b > 0] \\ 0, & \text{if } |\hat{b}| = b_{\min} \wedge (u e) \operatorname{sgn} b < 0 \end{cases} \quad (9.34)$$

The main motive here is to stop adaptation of \hat{b} if the parameter reaches its lower absolute limit value b_{\min} , with a nonzero time derivative $\dot{\hat{b}}$. In this case, we would prevent the estimated parameter \hat{b} from crossing the known lower absolute value bound b_{\min} .

Let us argue that the modification (9.34) does indeed prevents \hat{b} from crossing its allowable bound, and at the same time, it preserves closed-loop system stability.

Suppose that $\operatorname{sgn} b > 0$. Then according to (9.34), it is easy to see that starting with any initial condition $\hat{b}(0) > b_{\min}$, the estimated parameter will satisfy the desired lower bound relation $\hat{b}(t) \geq b_{\min}$, for all future times. In addition, we must verify that the proposed modification (9.34) does not adversely affect closed-loop stability of the tracking error dynamics (9.29). In particular, we need to ensure that

the inequality (9.33) remains in effect. For this to be true it is sufficient to show that

$$\Delta b \left(\gamma_b^{-1} \dot{\hat{b}} - u e \right) \leq 0 \quad (9.35)$$

Let us argue that with the adaptive law modification (9.34), the above relation does indeed hold.

When $\hat{b} > b_{\min}$, the adaptive law (9.34) is the same as the corresponding law in (9.32) and, therefore, $\dot{V} \leq 2 a_{\text{ref}} e^2 \leq 0$. Suppose that there exists $0 \leq T < \infty$ such that $\hat{b}(T) = b_{\min}$. Since $b \geq b_{\min}$ then $\Delta b(T) = \hat{b}(T) - b = b_{\min} - b \leq 0$. If $u(T)e(T) \geq 0$ then $\dot{V} = 2 a_{\text{ref}} e^2 \leq 0$, while $\dot{\hat{b}}(T) = \gamma_b u e \geq 0$ implying that $\hat{b}(t)$ increases locally for $t \geq T$. On other hand, if $u(T)e(T) < 0$ then according to (9.34), at $t = T$: $\Delta b \left(\gamma_b^{-1} \dot{\hat{b}} - u e \right) = - \underbrace{\Delta b}_{\leq 0} \underbrace{u e}_{\leq 0} \leq 0$, and so again $\dot{V} \leq 2 a_{\text{ref}} e^2 \leq 0$.

This proves the desired properties of (9.34) when $\text{sgn } b > 0$. For $\text{sgn } b < 0$, formal arguments are similar and therefore they will be left as an exercise for the reader.

The adaptive law modification (9.34) enforces the non-positive sign of \dot{V} , and as such, it contributes to achieving closed-loop system stability. The parameter adaptation dynamics in (9.34) represent a special case of the Projection Operator [6], whose continuous version will be introduced in Chap. 11.

With the proposed adjustment (9.34), the DI-based adaptive laws (9.32) become

$$\begin{aligned} \dot{\hat{a}} &= \gamma_a x e \\ \dot{\hat{b}} &= \begin{cases} \gamma_b u e, & \text{if } |\hat{b}| > b_{\min} \vee [\hat{b} = b_{\min} \text{ sgn } b \wedge (u e) \text{ sgn } b > 0] \\ 0, & \text{if } |\hat{b}| = b_{\min} \wedge (u e) \text{ sgn } b < 0 \end{cases} \\ \dot{\hat{\theta}} &= \Gamma_\theta \Phi(x) e \end{aligned} \quad (9.36)$$

Next, we are going to formally prove that the DI-based adaptive controller in (9.27) provides global uniform asymptotic tracking of the reference model state. Since $\dot{V} \leq 0$ then $(e, \Delta a, \Delta b, \Delta \theta)$ are uniformly bounded. The latter implies that $(x, \hat{a}, \hat{b}, \hat{\theta})$ are also uniformly bounded. Due to (9.34), $\hat{b} \geq b_0$ and consequently u is uniformly bounded, and so is \dot{x} . Since r is bounded, then \dot{x}_{ref} is bounded, and consequently \dot{e} is bounded as well. Because of (9.34)

$$\dot{V}(e, \Delta a, \Delta b, \Delta \theta) \leq -2|a_{\text{ref}}|e^2 \leq 0 \quad (9.37)$$

for all $t \geq 0$. Since V is positive definite and its derivative is semi-negative definite, then V converges to a limit, as a function of time. Integrating both sides of (9.37) yields

$$V(t) - V(0) \leq -2|a_{\text{ref}}| \int_0^t e^2(\tau) d\tau \leq 0 \quad (9.38)$$

or, equivalently

$$\int_0^t e^2(\tau) d\tau \leq \frac{1}{2|a_{\text{ref}}|} (V(0) - V(t)) < \infty \quad (9.39)$$

Let $W(t) = \int_0^t e^2(\tau) d\tau$. From (9.39) it follows that $W(t)$ tends to a finite limit, as $t \rightarrow \infty$. At the same time, its time derivative is $\dot{W}(t) = e^2(t)$, and so its second time derivative is bounded: $\ddot{W}(t) = 2e\dot{e}(t) < \infty$. Then $\dot{W}(t)$ is uniformly continuous. Finally, we can apply Barbalat's lemma to conclude that $\lim_{t \rightarrow \infty} \dot{W}(t) = 0$, which immediately implies $\lim_{t \rightarrow \infty} e^2(t) = 0$. The tracking problem is solved, and the DI-based MRAC design is summarized in Table 9.2.

Let us now illustrate the DI-based MRAC procedure by redesigning the pitch controller using the helicopter pitch dynamics data from Example 9.1.

Table 9.2 DI-based MRAC design summary

Open-loop plant	$\dot{x} = ax + bu + \theta^T \Phi(x)$
Reference model	$\dot{x}_{\text{ref}} = a_{\text{ref}}x_{\text{ref}} + b_{\text{ref}}r$
Tracking error	$e = x - x_{\text{ref}}$
Control input	$u = \frac{1}{b} \left((a_{\text{ref}} - \hat{a})x + b_{\text{ref}}r - \hat{\theta}^T \Phi(x) \right)$
DI-based MRAC laws	$\dot{\hat{a}} = \gamma_a x e$ $\dot{\hat{b}} = \begin{cases} \gamma_b u e, & \text{if } \hat{b} > b_{\min} \vee [\hat{b} = b_{\min} \operatorname{sgn} b \wedge (u e) \operatorname{sgn} b > 0] \\ 0, & \text{if } \hat{b} = b_{\min} \wedge (u e) \operatorname{sgn} b < 0 \end{cases}$ $\dot{\hat{\theta}} = \Gamma_\theta \Phi(x)e$

Example 9.2 DI-Based MRAC Design for Helicopter Pitch Dynamics For the helicopter pitch dynamics (see Example 9.1),

$$\dot{q} = -0.61 q - 6.65 \delta + \underbrace{\left(\underbrace{0.0665}_{\theta} \tanh \left(\frac{360}{\pi} q \right) \right)}_{f(q)}$$

the DI-based MRAC system is constructed using the design equations from Table 9.2. The resulting adaptive pitch controller,

$$\delta = \frac{1}{\hat{b}} \left((a_{\text{ref}} - \hat{a})q + b_{\text{ref}} q_{\text{cmd}} - \hat{\theta}^T \Phi(q) \right)$$

with the reference model parameters $a_{\text{ref}} = -b_{\text{ref}} = -4$, and with the adaptive laws,

$$\begin{aligned} \dot{\hat{a}} &= \gamma_a (q - q_{\text{ref}}) \\ \dot{\hat{b}} &= \begin{cases} \gamma_b \delta(q - q_{\text{ref}}), & \text{if } \hat{b} < -b_{\min} \vee [\hat{b} = -b_{\min} \wedge (\delta(q - q_{\text{ref}})) < 0] \\ 0, & \text{if } \hat{b} = -b_{\min} \wedge (\delta(q - q_{\text{ref}})) > 0 \end{cases} \\ \dot{\hat{\theta}} &= \Gamma_\theta \Phi(q)(q - q_{\text{ref}}) \end{aligned}$$

was given the task to track the same exact step-inputs commands from Example 9.1. We assumed $|M_\delta| \geq b_{\min} = 1$ to be the known lower bound and selected $\hat{b}(0) = -4$ to represent the initial value for the estimated elevator effectiveness. With the adaptation rates $\gamma_a = \gamma_b = \Gamma_\theta = 200$, the closed-loop system pitch rate response and the corresponding elevator input are shown in Fig. 9.9.

Similar to Fig. 9.3, these simulation data also show adequate command tracking performance and achievable control input (elevator) values. Comparison of the DI-based MRAC signal with the ideal fixed-gain controller (Fig. 9.10) reveals transient oscillations in the MRAC signal.

Moreover, it is interesting to note that the estimated parameters are nowhere near their ideal unknown values (Fig. 9.11).

For example, the estimated pitch damping $\hat{a} = \hat{M}_q$ remains predominantly near zero, while its true value $a = M_q$ is negative and much larger than this estimate. Nevertheless, as predicted by the design, all of the estimated parameters stay uniformly bounded in time, while the system state tracks the state of the desired reference model. ■

After reviewing the two simulation examples presented in this section, the reader should be able to appreciate inherent nonlinear features of MRAC systems. Even for scalar dynamics, these controllers may yield transient oscillations and adaptive gain values that do not resemble the true unknowns in the system dynamics. However, these “undesirable” features are not in conflict with the formally

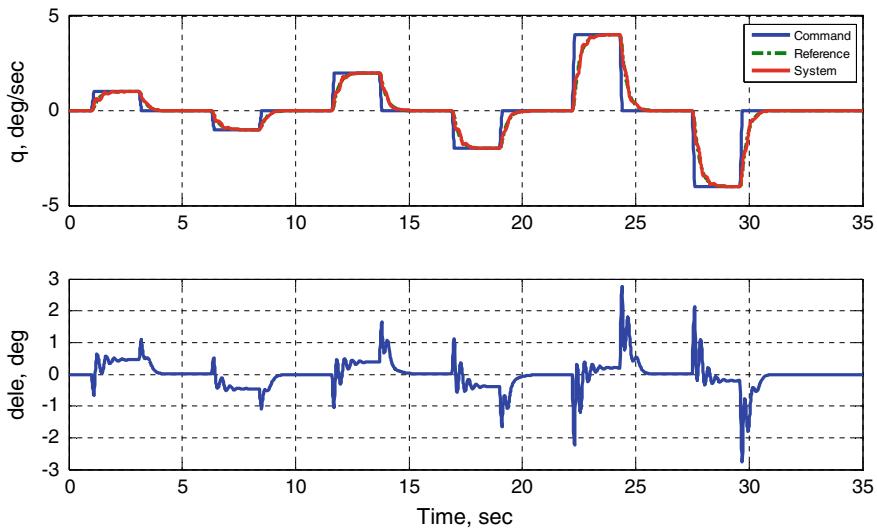


Fig. 9.9 Closed-loop tracking performance in Example 9.2

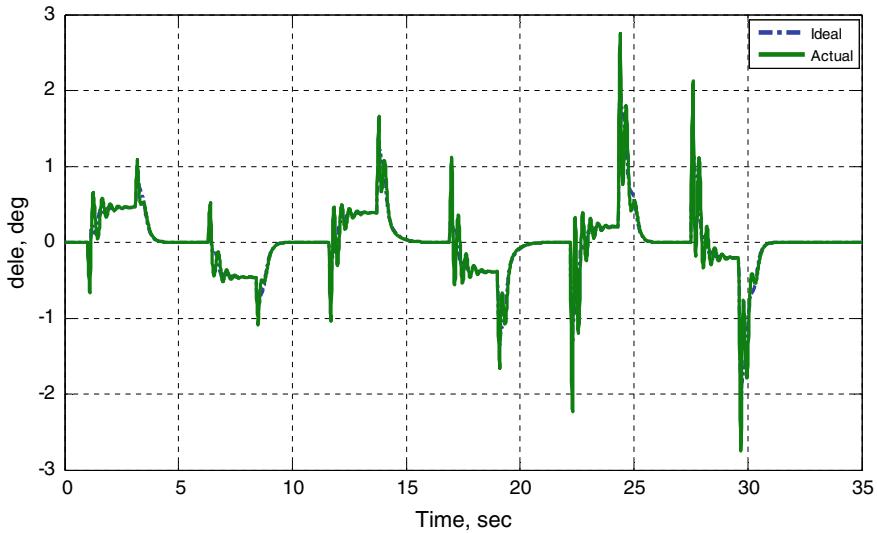


Fig. 9.10 DI-based MRAC and ideal fixed-gain controller from Example 9.2

derived MRAC design. Achieving smooth transients or having adaptive parameters converge to their ideal values were not formulated as the design goals. Only asymptotic command tracking was of interest, and that goal was fully achieved despite the system uncertainties.

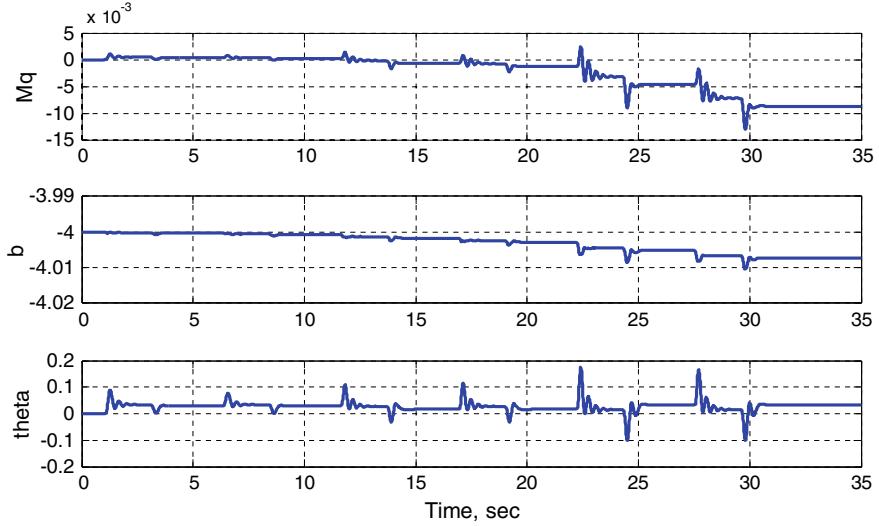


Fig. 9.11 Estimated parameters in Example 9.2

9.5 MRAC Design for Multi-Input–Multi-Output Systems

In this section, we will extend applicability of the MRAC design from scalar dynamics to multi-input–multi-output (MIMO) nonlinear systems in the form,

$$\dot{x} = A x + B \Lambda(u + f(x)) \quad (9.40)$$

where $x \in R^n$ is the system state, $u \in R^m$ is the control input, $B \in R^{n \times m}$ is the known control matrix, while $A \in R^{n \times n}$ and $\Lambda \in R^{m \times m}$ are unknown constant matrices. In addition, it is assumed that Λ is diagonal, its elements λ_i are strictly positive, and the pair $(A, B \Lambda)$ is controllable. The uncertainty in Λ is introduced to model control failures or modeling errors, in the sense that there may exist uncertain control gains or the designer may have incorrectly estimated the system control effectiveness.

In (9.40), the unknown possibly nonlinear vector function $f(x) : R^n \rightarrow R^m$ represents the system matched uncertainty. It is assumed that each individual component $f_i(x)$ of $f(x)$ can be written as a linear combination of N known locally Lipschitz-continuous basis functions $\varphi_i(x)$, with unknown constant coefficients. So, we write

$$f(x) = \Theta^T \Phi(x) \quad (9.41)$$

where $\Theta \in R^{N \times m}$ is a constant matrix of the unknown coefficients and $\Phi(x) = (\varphi_1(x) \dots \varphi_N(x))^T \in R^N$ is the known regressor vector.

We are interested in the design of a MIMO state feedback adaptive control law such that the system state x globally uniformly asymptotically tracks the state $x_{\text{ref}} \in R^n$ of the reference model,

$$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} r(t) \quad (9.42)$$

where $A_{\text{ref}} \in R^{n \times n}$ is Hurwitz, $B_{\text{ref}} \in R^{n \times m}$, and $r(t) \in R^m$ is the external bounded command vector.

We also require that during tracking, all signals in the closed-loop system remain uniformly bounded. Thus, given any bounded command $r(t)$, the control input u needs to be chosen such that the state tracking error

$$e(t) = x(t) - x_{\text{ref}}(t) \quad (9.43)$$

globally uniformly asymptotically tends to zero, that is,

$$\lim_{t \rightarrow \infty} \|x(t) - x_{\text{ref}}(t)\| = 0 \quad (9.44)$$

If matrices A and Λ were known, one could have calculated and applied the ideal fixed-gain control law,

$$u = K_x^T x + K_r^T r - \Theta^T \Phi(x) \quad (9.45)$$

and obtain the closed-loop system

$$\dot{x} = (A + B \Lambda K_x^T)x + B \Lambda K_r^T r \quad (9.46)$$

Comparing (9.46) with the desired reference dynamics (9.42), it follows that for existence of a controller in the form of (9.45), the ideal unknown control gains, K_x and K_r , must satisfy the matching conditions.

$$\begin{aligned} A + B \Lambda K_x^T &= A_{\text{ref}} \\ B \Lambda K_r^T &= B_{\text{ref}} \end{aligned} \quad (9.47)$$

Assuming that these matching conditions hold, it is easy to see that using (9.45) yields the closed-loop system which is exactly the same as the reference model. Consequently, for any bounded reference input signal $r(t)$, the fixed-gain controller (9.45) provides global uniform asymptotic tracking performance.

Let us at once note that given $(A, B, \Lambda, A_{\text{ref}}, B_{\text{ref}})$, there is no guarantee that the ideal gains K_x, K_r exist such that the matching conditions (9.47) are satisfied. In other words, the control law (9.45) may not be able to meet the design objective. However often in practice, the structure of A is known, and the reference model matrices $A_{\text{ref}}, B_{\text{ref}}$ are chosen so that the system (9.47) has at least one ideal solution pair (K_x, K_r) .

Assuming that K_x, K_r in (9.47) do exist, we consider the following control law:

$$u = \hat{K}_x^T x + \hat{K}_r^T r - \hat{\Theta}^T \Phi(x) \quad (9.48)$$

where $\hat{K}_x \in R^{n \times m}$, $\hat{K}_r \in R^{m \times m}$, $\hat{\Theta} \in R^{N \times n}$ are the estimates of the ideal unknown matrices K_x, K_r, Θ , respectively. These estimated parameters will be generated online through the inverse Lyapunov analysis. Substituting (9.48) into (9.40), the closed-loop system dynamics can be written as

$$\dot{x} = \left(A + B \Lambda \hat{K}_x^T \right) x + B \Lambda \left(\hat{K}_r^T r - \left(\hat{\Theta} - \Theta \right)^T \Phi(x) \right) \quad (9.49)$$

Subtracting (9.42) from (9.49), we compute the closed-loop dynamics of the n -dimensional tracking error vector $e(t) = x(t) - x_{\text{ref}}(t)$.

$$\dot{e} = \left(A + B \Lambda \hat{K}_x^T \right) x + B \Lambda \left(\hat{K}_r^T r - \left(\hat{\Theta} - \Theta \right)^T \Phi(x) \right) - A_{\text{ref}} x_{\text{ref}} - B_{\text{ref}} r \quad (9.50)$$

With the matching conditions (9.47) in place, we further get

$$\begin{aligned} \dot{e} &= \left(A_{\text{ref}} + B \Lambda \left(\hat{K}_x - K_x \right) \right) x - A_{\text{ref}} x_{\text{ref}} \\ &\quad + B \Lambda \left(\hat{K}_r - K_r \right) r - B \Lambda \left(\hat{\Theta} - \Theta \right)^T \Phi(x) \\ &= A_{\text{ref}} e + B \Lambda \left[\left(\hat{K}_x - K_x \right)^T x + \left(\hat{K}_r - K_r \right)^T r - \left(\hat{\Theta} - \Theta \right)^T \Phi(x) \right] \end{aligned} \quad (9.51)$$

Let $\Delta K_x = \hat{K}_x - K_x$, $\Delta K_r = \hat{K}_r - K_r$, and $\Delta \Theta = \hat{\Theta} - \Theta$ represent the parameter estimation errors. In terms of the latter, the tracking error dynamics become

$$\dot{e} = A_{\text{ref}} e + B \Lambda \left[\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x) \right] \quad (9.52)$$

We introduce rates of adaptation: $\Gamma_x = \Gamma_x^T > 0$, $\Gamma_r = \Gamma_r^T > 0$, $\Gamma_\Theta = \Gamma_\Theta^T > 0$. Going back to analyzing stability of the tracking error dynamics (9.52), let us consider a globally radially unbounded quadratic Lyapunov function candidate in the form

$$\begin{aligned} V(e, \Delta K_x, \Delta K_r, \Delta \Theta) &= e^T P e \\ &\quad + \text{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \Delta K_x + \Delta K_r^T \Gamma_r^{-1} \Delta K_r + \Delta \Theta^T \Gamma_\Theta^{-1} \Delta \Theta \right] \Lambda \right) \end{aligned} \quad (9.53)$$

where $P = P^T > 0$ satisfies the algebraic Lyapunov equation,

$$P A_{\text{ref}} + A_{\text{ref}}^T P = -Q \quad (9.54)$$

for some $Q = Q^T > 0$. Then the time derivative of V , evaluated along the trajectories of (9.52), can be calculated.

$$\begin{aligned}
 \dot{V} &= \dot{e}^T P e + e^T P \dot{e} + 2 \text{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{K}_x + \Delta K_r^T \Gamma_r^{-1} \dot{K}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \dot{\Theta} \right] \Lambda \right) \\
 &= (A_{\text{ref}} e + B \Lambda (\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x)))^T P e \\
 &\quad + e^T P (A_{\text{ref}} e + B \Lambda (\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x))) \\
 &\quad + 2 \text{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{K}_x + \Delta K_r^T \Gamma_r^{-1} \dot{K}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \dot{\Theta} \right] \Lambda \right) \\
 &= e^T (A_{\text{ref}} P + P A_{\text{ref}}) e + 2 e^T P B \Lambda (\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x)) \\
 &\quad + 2 \text{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{K}_x + \Delta K_r^T \Gamma_r^{-1} \dot{K}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \dot{\Theta} \right] \Lambda \right) \tag{9.55}
 \end{aligned}$$

Using (9.54) further yields

$$\begin{aligned}
 \dot{V} &= -e^T Q e + \left[2 e^T P B \Lambda \Delta K_x^T x + 2 \text{tr} \left(\Delta K_x^T \Gamma_x^{-1} \dot{K}_x \Lambda \right) \right] \\
 &\quad + \left[2 e^T P B \Lambda \Delta K_r^T r + 2 \text{tr} \left(\Delta K_r^T \Gamma_r^{-1} \dot{K}_r \Lambda \right) \right] \\
 &\quad + \left[-2 e^T P B \Lambda \Delta \Theta^T \Phi(x) + 2 \text{tr} \left(\Delta \Theta^T \Gamma_\Theta^{-1} \dot{\Theta} \Lambda \right) \right] \tag{9.56}
 \end{aligned}$$

Via the vector trace identity (defined in Chap. 1),

$$\begin{aligned}
 \underbrace{e^T P B \Lambda}_{a^T} \underbrace{\Delta K_x^T x}_{b} &= \text{tr} \left(\underbrace{\Delta K_x^T x}_{b} \underbrace{e^T P B \Lambda}_{a^T} \right) \\
 \underbrace{e^T P B \Lambda}_{a^T} \underbrace{\Delta K_r^T r}_{b} &= \text{tr} \left(\underbrace{\Delta K_r^T r}_{b} \underbrace{e^T P B \Lambda}_{a^T} \right) \\
 \underbrace{e^T P B \Lambda}_{a^T} \underbrace{\Delta \Theta^T \Phi(x)}_{b} &= \text{tr} \left(\underbrace{\Delta \Theta^T \Phi(x)}_{b} \underbrace{e^T P B \Lambda}_{a^T} \right) \tag{9.57}
 \end{aligned}$$

Substituting (9.57) into (9.56) results in

$$\begin{aligned}
 \dot{V} &= -e^T Q e + 2 \text{tr} \left(\Delta K_x^T \left[\Gamma_x^{-1} \dot{K}_x + x e^T P B \right] \Lambda \right) \\
 &\quad + 2 \text{tr} \left(\Delta K_r^T \left[\Gamma_r^{-1} \dot{K}_r + r e^T P B \right] \Lambda \right) \\
 &\quad + 2 \text{tr} \left(\Delta \Theta^T \left[\Gamma_\Theta^{-1} \dot{\Theta} - \Phi(x) e^T P B \right] \Lambda \right) \tag{9.58}
 \end{aligned}$$

If the adaptive laws are selected as,

$$\begin{aligned}\dot{\hat{K}}_x &= -\Gamma_x x e^T P B \\ \dot{\hat{K}}_r &= -\Gamma_r r(t) e^T P B \\ \dot{\hat{\Theta}} &= \Gamma_\Theta \Phi(x) e^T P B\end{aligned}\tag{9.59}$$

then the time derivative of V in (9.58) becomes globally negative semidefinite.

$$\dot{V} = -e^T Q e \leq 0\tag{9.60}$$

Therefore, the closed-loop error dynamics are uniformly stable. So, the tracking error $e(t)$ and the parameter estimation errors $\Delta K_x(t)$, $\Delta K_r(t)$, $\Delta \Theta(t)$ are uniformly bounded, and so are the parameter estimates $\hat{K}_x(t)$, $\hat{K}_r(t)$, $\hat{\Theta}(t)$. Since $r(t)$ is bounded and A_{ref} is Hurwitz then $x_{\text{ref}}(t)$ and $\dot{x}_{\text{ref}}(t)$ are bounded. Hence, the system state $x(t)$ is uniformly bounded, and the control input $u(t)$ in (9.48) is bounded as well. The latter implies that $\dot{x}(t)$ is bounded, and thus $\dot{e}(t)$ is bounded. Furthermore, the second time derivative of $V(t)$

$$\ddot{V} = -2 e^T Q \dot{e}\tag{9.61}$$

is bounded, and so $\dot{V}(t)$ is uniformly continuous. Since in addition, $V(t)$ is lower bounded and $\dot{V}(t) \leq 0$, then using Barbalat's lemma, gives $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$. We have formally proven that the state tracking error $e(t)$ tends to the origin globally, uniformly, and asymptotically: $\lim_{t \rightarrow \infty} \|x(t) - x_{\text{ref}}(t)\| = 0$. The MIMO command tracking problem for the system dynamics (9.40) is solved. We now formulate our obtained results as a theorem.

Theorem 9.2 *Given MIMO dynamics (9.40) with a control uncertainty Λ and a matched unknown function $f(x)$ from (9.41), the MRAC system (9.48)–(9.59) enforces global uniform asymptotic tracking performance of the reference model dynamics (9.42), driven by any bounded time-varying command $r(t)$. Moreover, all signals in the corresponding closed-loop system remain uniformly bounded in time.* ■

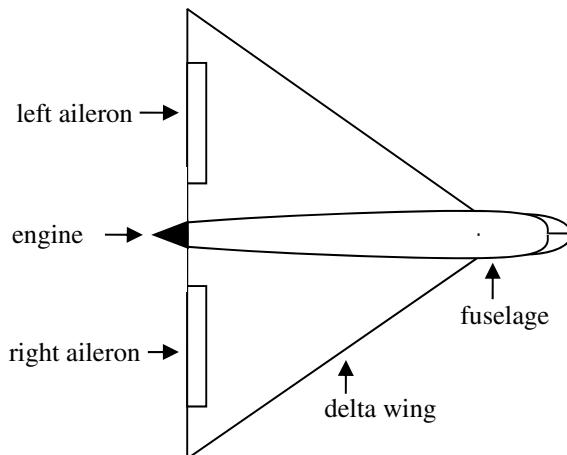
Table 9.3 summarizes the MIMO MRAC design equations.

Example 9.3 MRAC Control of Delta Wing Dynamics at High Angle of Attack A sketch of an aircraft equipped with a delta wing is shown in Fig. 9.11 (Fig. 9.12).

The wing sweeps sharply back from the fuselage with the angle between the wing leading edge often as high as 80° , and the angle between the fuselage and the trailing edge of the wing at around 90° . Delta wings are known to be unstable, especially at high angle of attack (the angle between the aircraft velocity vector and the fuselage centerline). Yet their primary advantage is aerodynamic efficiency in high-speed flight [10].

Table 9.3 MIMO MRAC design equations

Open-loop plant	$\dot{x} = A x + B \Lambda(u + \Theta^T \Phi(x))$
Reference model	$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} r$
Model matching conditions	$A + B \Lambda K_x^T = A_{\text{ref}}, B \Lambda K_r^T = B_{\text{ref}}$
Tracking error	$e = x - x_{\text{ref}}$
Control input	$u = \hat{K}_x^T x + \hat{K}_r^T r - \hat{\Theta}^T \Phi(x)$
Algebraic Lyapunov equation	$P A_{\text{ref}} + A_{\text{ref}}^T P = -Q$
MIMO MRAC laws	$\dot{\hat{K}}_x = -\Gamma_x x e^T P B$ $\dot{\hat{K}}_r = -\Gamma_r r(t) e^T P B$ $\dot{\hat{\Theta}} = \Gamma_\Theta \Phi(x) e^T P B$

Fig. 9.12 Delta wing aircraft

A delta wing aircraft flying at high angle of attack is open-loop unstable in roll. This instability is called the “wing-rock phenomenon”. It is induced by unsteady aerodynamic effects acting on the delta wing asymmetrically. As a result, the aircraft undergoes an unstable rocking motion that needs to be actively controlled.

In this example, we consider a delta wing aircraft whose roll dynamics can be regulated by ailerons—the movable surfaces that are located symmetrically on the outboard portions of the aircraft left and right-wing segments. Moving the left aileron down (positive deflection) and the right one up (negative deflection) induces the right-wing-down rolling motion (positive roll rate) of the aircraft. The difference between the left and right aileron positions is called the “differential aileron”. This is the primary control input signal for regulating the aircraft roll dynamics.

We shall make use of a generic delta wing-rock dynamic model in the form,

$$\begin{aligned}\dot{\varphi} &= p \\ \dot{p} &= \theta_1 \varphi + \theta_2 p + (\theta_3 |\varphi| + \theta_4 |p|)p + \theta_5 \varphi^3 + \theta_6 \delta_a\end{aligned}$$

where φ is the aircraft roll angle (rad), p is the roll rate (rad/s), and δ_a is the differential aileron (control input, rad). The unknown constant parameters are

$$\theta_1 = -0.018, \quad \theta_2 = 0.015, \quad \theta_3 = -0.062, \quad \theta_4 = 0.009, \quad \theta_5 = 0.021, \quad \theta_6 = 0.75$$

Rewriting the model in the form of (9.40) gives

$$\underbrace{\begin{pmatrix} \dot{\varphi} \\ \dot{p} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0 & 1 \\ \theta_1 & \theta_2 \end{pmatrix}}_A \underbrace{\begin{pmatrix} \varphi \\ p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_B \underbrace{\theta_6 \Lambda}_{\text{A}} \left(\underbrace{\delta_a}_u + \underbrace{\frac{1}{\theta_6} ((\theta_3 |\varphi| + \theta_4 |p|)p + \theta_5 \varphi^3)}_{f(x) = \Theta^T \Phi(x)} \right)$$

where the uncertain state-dependent function $f(x)$ is represented by a constant unknown parameter vector Θ and the known regressor vector $\Phi(x)$.

$$f(x) = \underbrace{\frac{1}{\theta_6} (\theta_3 \theta_4 \theta_5)}_{\Theta^T} \underbrace{(|\varphi|p \ |p|p \ \varphi^3)}_{\Phi(x)}^T = -0.0827 |\varphi|p + 0.012 |p|p + 0.028 \varphi^3$$

The system control effectiveness $\Lambda = \theta_6$ is assumed to be constant and unknown. Also unknown are the second-row coefficients (θ_1, θ_2) in A , as well as the state-dependent function $f(x)$.

The reference roll dynamics are defined by the second-order transfer function,

$$\frac{\varphi_{\text{ref}}}{\varphi_{\text{cmd}}} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

which represents the desired command-to-response roll angle behavior (using the Laplace transform). Here, φ_{ref} is the reference roll angle, φ_{cmd} is the commanded roll angle, and (ω_n, ξ) are the desired natural frequency and the damping ratio, respectively. Let $p_{\text{ref}} = \dot{\varphi}_{\text{ref}}$ denote the reference roll rate. In state-space form, the reference roll dynamics can be easily written as

$$\underbrace{\begin{pmatrix} \dot{\varphi}_{\text{ref}} \\ \dot{p}_{\text{ref}} \end{pmatrix}}_{\dot{x}_{\text{ref}}} = \underbrace{\begin{pmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{pmatrix}}_{A_{\text{ref}}} \underbrace{\begin{pmatrix} \varphi_{\text{ref}} \\ p_{\text{ref}} \end{pmatrix}}_{x_{\text{ref}}} + \underbrace{\begin{pmatrix} 0 \\ \omega_n^2 \end{pmatrix}}_{B_{\text{ref}}} \underbrace{\varphi_{\text{cmd}}}_r$$

Clearly, the matching conditions (9.47) hold. In fact, from

$$\begin{pmatrix} 0 & 1 \\ \theta_1 & \theta_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \theta_6 K_x^T = \begin{pmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \theta_6 K_r^T = \begin{pmatrix} 0 \\ \omega_n^2 \end{pmatrix}$$

it follows that the ideal unknown feedback and feedforward gains are

$$K_x^T = -\frac{1}{\theta_6} (\omega_n^2 + \theta_1 2\xi\omega_n + \theta_2), \quad K_r^T = \frac{\omega_n^2}{\theta_6}$$

In this example, we have selected $\omega_n = 1$ rad/s and $\xi = 0.7$. So, the ideal gains are $K_x = (-1.3093 - 1.8867)^T$ and $K_r = 1.3333$.

One can verify that the open-loop system, with $\delta_a = 0$, has an unstable equilibrium at the origin and a limit cycle near $\varphi = 35^\circ$. The limit cycle attracts all open-loop system trajectories that start on the inside of its boundary, and it repels all the trajectories with the initial conditions on the outside (see Fig. 9.13).

We use the design equations from Table 9.3 to construct an MRAC system. The design “tuning knobs” consist of symmetric positive-definite matrices Q , Γ_x , Γ_r , and Γ_θ , with the last three quantities representing adaptation rates for the adaptive

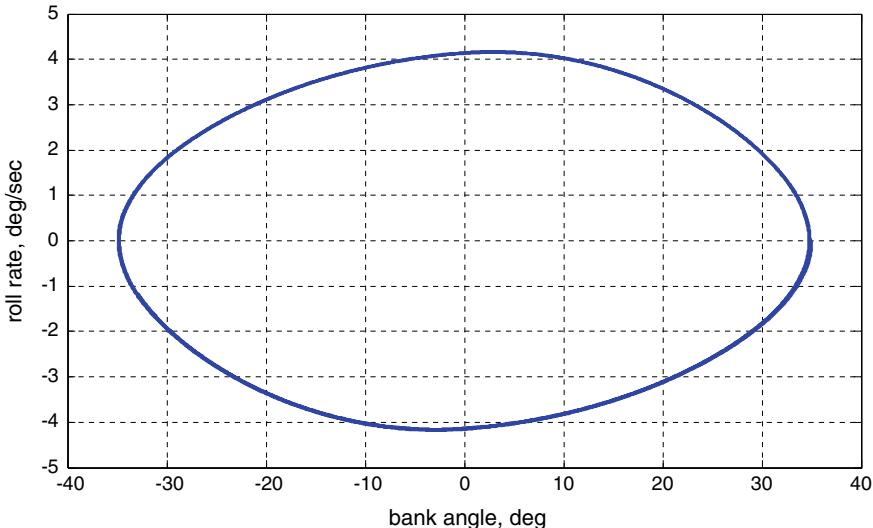


Fig. 9.13 Open-loop limit cycle for delta wing roll dynamics in Example 9.3

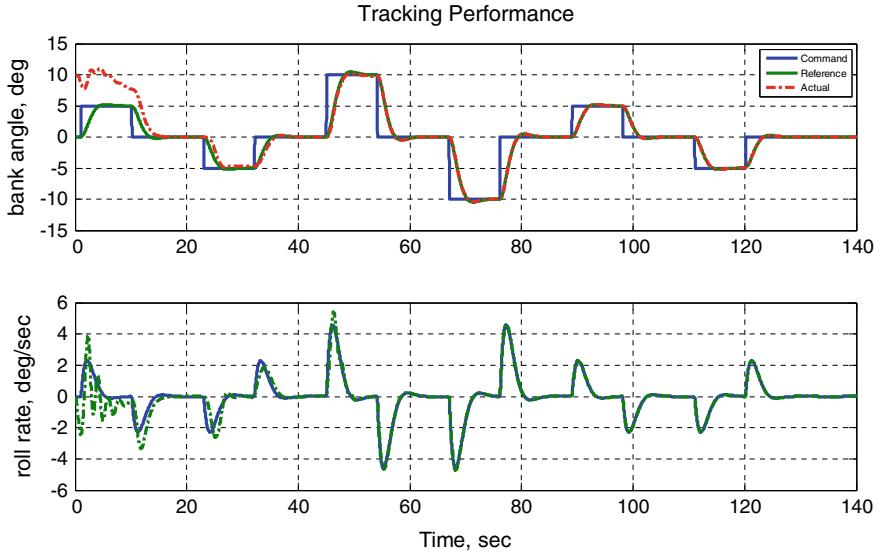


Fig. 9.14 Closed-loop system tracking performance in Example 9.3

parameters \hat{K}_x , \hat{K}_r , and $\hat{\Theta}$, respectively. After several iterations, we have selected the following data:

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}, \quad \Gamma_x = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}, \quad \Gamma_r = 100, \quad \Gamma_\theta = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{pmatrix}$$

Figure 9.14 shows the system closed-loop response in tracking a series of step-input commands, with the initial bank angle set to 10° .

The system tracking error quickly dissipates (Fig. 9.15), while the required control input (differential aileron position) stays within achievable and reasonable limits.

It is interesting to observe (see Fig. 9.16) that in this case, the estimated feedback and feedforward gains (\hat{K}_x, \hat{K}_r) converge to their true unknown values.

However, the estimated parameters $(\hat{\theta}_3, \hat{\theta}_4, \hat{\theta}_5)$ that correspond to the nonlinear regressor components have dissimilar tendencies: The first two are quite different from their ideal counterparts, while the third one does converge to its ideal value (see Fig. 9.17).

Once again, we would like to remark that in general, parameter convergence is not guaranteed by an MRAC controller (see Theorem 9.2). Only uniform boundedness of all signals in the closed-loop system is certain. In order to emphasize this point, we encourage the reader to rerun this exact design but with a different bank angle

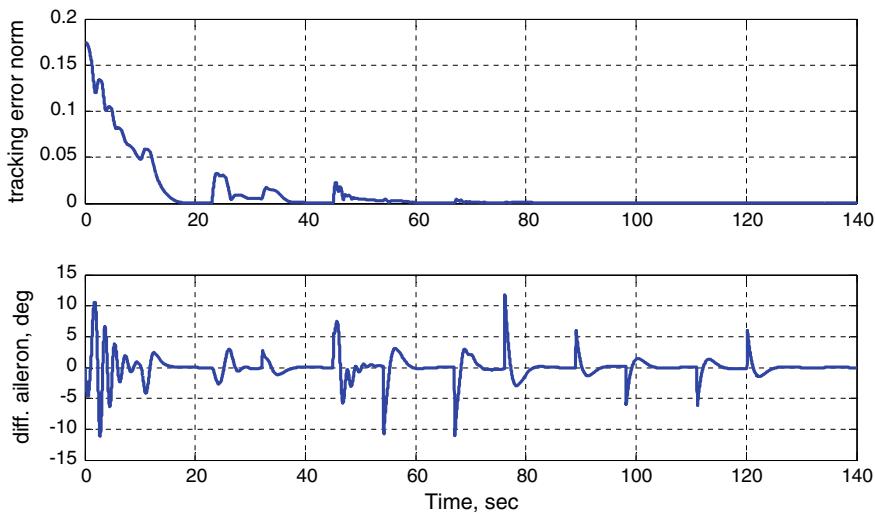


Fig. 9.15 Tracking error and differential aileron in Example 9.3

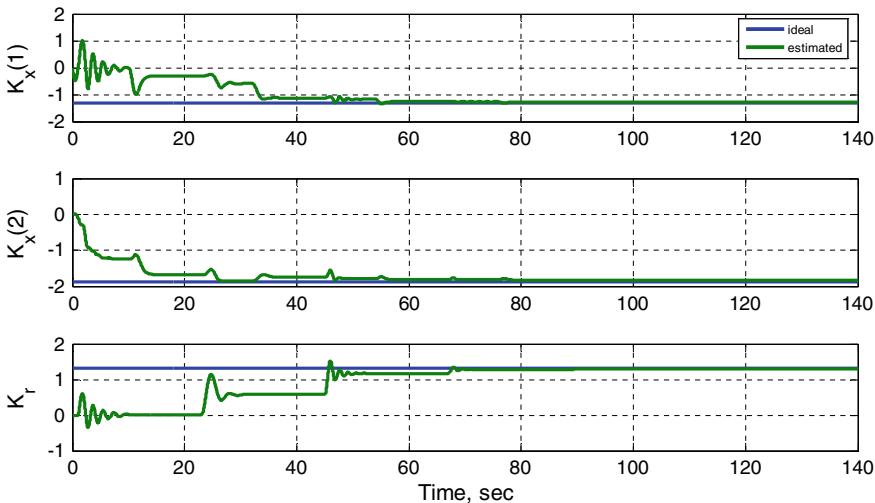


Fig. 9.16 Estimated feedback and feedforward gains in Example 9.3

command profile. For example, choosing $\varphi_{\text{cmd}} = 0.1745 \sin t$ will result in all-adaptive parameters being very different from their ideal values, yet the closed-loop system tracking performance will remain acceptable. ■

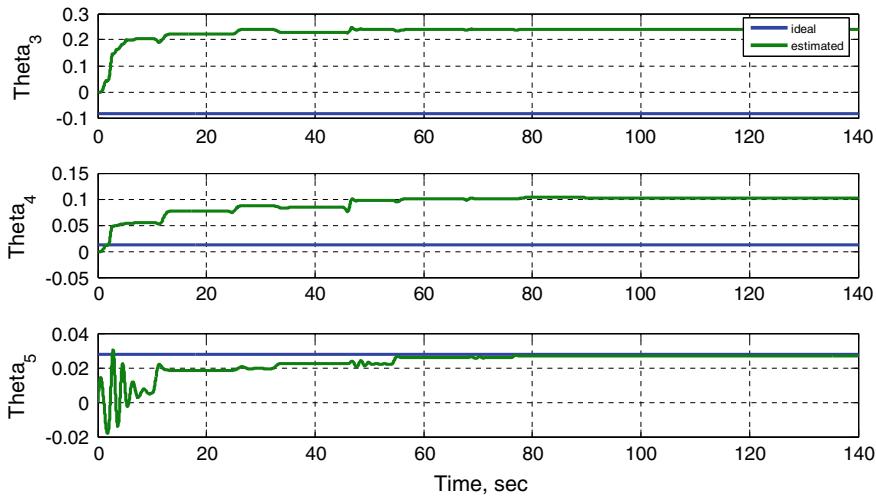


Fig. 9.17 Estimated parameters for nonlinear regressor components in Example 9.3

9.6 Summary

Adaptive control was initiated, inspired, and originally motivated by aerospace applications in the 1950s. The interest in the design of adaptive self-tuning controllers for practical systems that operate in uncertain environment has never diminished. The last decade has witnessed many successful demonstrations of adaptive control technology in aerospace, robotics, auto industry, and bioengineering.

Adaptive controllers have one common goal—to enable a dynamical system to track external commands, while operating in realistic and often uncertain environment. In this chapter, we have shown how to achieve this goal for a specific class of multi-input–multi-output dynamical systems with matched uncertainties. We have also demonstrated efficacy of adaptive control through simulation examples such as: (a) helicopter pitch dynamics in hover and (b) wing-rock dynamics of a delta wing at high angle of attack. All simulation data confirmed our theoretical predictions for MRAC in achieving desired tracking performance and keeping all signals in the corresponding closed-loop system uniformly bounded in time.

9.7 Exercises

Exercise 9.1 Prove that if some of the diagonal elements λ_i of the unknown diagonal matrix Λ in the system dynamics (9.40) are negative and the signs of all of them are known, then the adaptive laws

$$\begin{aligned}\dot{\hat{K}}_x &= -\Gamma_x x e^T P B \operatorname{sgn} \Lambda \\ \dot{\hat{K}}_r &= -\Gamma_r r(t) e^T P B \operatorname{sgn} \Lambda \\ \dot{\hat{\Theta}} &= \Gamma_\Theta \Phi(x) e^T P B \operatorname{sgn} \Lambda\end{aligned}\tag{9.62}$$

solve the MIMO tracking problem, where $\operatorname{sgn} \Lambda = \operatorname{diag}[\operatorname{sgn} \lambda_1, \dots, \operatorname{sgn} \lambda_m]$.

Exercise 9.2 Implement and simulate the system from Example 9.1. Test the MRAC controller (redesign, if needed) in the presence of various uncertainties and external commands of your choice. Comment on the system tracking performance. Discuss adaptive parameter dynamics and convergence of the estimated parameters to their true values.

Exercise 9.3 Repeat all tasks from Exercise 9.2 for the DI-based MRAC system in Example 9.2. Compare closed-loop performance of the two controllers, including their respective control efforts.

Exercise 9.4 Repeat all tasks from Exercise 9.2 for the MRAC controller and the system in Example 9.3. Find external commands that would cause the adaptive parameters to (a) Converge to their true values; (b) Converge to some constant values; and (c) Not have limits. Is there a benefit for the adaptive parameters to converge to their true values? For all these cases, compare and discuss the system tracking performance.

References

- Whitaker HP, Yamron J, Kezer A (1958) Design of model reference adaptive control systems for aircraft. Report R-164, Instrumentation Laboratory, MIT, Cambridge, MA
- Whitaker HP, Osburn PV, Kezer A (1961) New developments in the design of model reference adaptive control systems. Paper 61-39, Institute of the Aerospace Sciences, Easton, PA
- Butchart RL, Shackcloth B (1965) Synthesis of model reference adaptive control systems by Lyapunov's second method. In: Proceedings 1965 IFAC symposium on adaptive control, Teddington, UK
- Parks PD (1966) Lyapunov redesign of model reference adaptive systems. IEEE Trans Autom Control 11:362–367

5. Narendra KS, Annaswamy AM (2005) Stable adaptive control. Dover, New York
6. Ioannou P, Fidan P (2006) Adaptive control tutorial. In: SIAM, Advances in design and control, SIAM, PA
7. Slotine JJE, Li W (1995) Applied nonlinear control. Prentice Hall, Hoboken
8. Krstic M, Kanellakopoulos I, Kokotovic P (1995) Nonlinear and adaptive control design. Wiley, New York
9. Khalil H (1996) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, p 07458
10. Hsu C, Lan CE (1985) Theory of wing rock. *J Aircr* 22(10):920–924



Model Reference Adaptive Control with Integral Feedback Connections

10

In this chapter, we design adaptive command tracking controllers using concepts from linear integral control. Such linear controllers can asymptotically reject constant disturbances and, at the same time, track constant commands with zero steady-state errors. In our attempt to further extend robustness and tracking properties of the linear integral controllers, we will develop and analyze an adaptive augmentation method to combine a baseline linear (Proportional + Integral) state feedback controller with an MRAC system. The adaptive augmentation design approach paves the way to transitioning adaptive controllers into industrial applications where legacy integral controllers are common practice.

10.1 Introduction

We begin by considering a class of MIMO uncertain systems in the form

$$\dot{x}_p = A_p x_p + B_p \Lambda \left(u + \overbrace{\Theta^T \Phi(x_p)}^{f(x_p)} \right) \quad (10.1)$$

where $x_p \in R^{n_p}$ is the system state vector, $u \in R^m$ is the control input,

$$f(x_p) = \Theta^T \Phi(x_p) \in R^m \quad (10.2)$$

is the linear-in-parameters state-dependent matched uncertainty, $\Theta \in R^{N \times m}$ is the matrix of unknown constant parameters, and $\Phi(x_p) \in R^N$ is the known N -dimensional regressor vector, whose components are locally Lipschitz-continuous

functions of x_p . Also in (10.1), $B_p \in R^{n \times m}$ is constant and known, $A_p \in R^{n \times n}$ is constant and unknown, and $\Lambda \in R^{m \times m}$ is a constant diagonal unknown matrix with positive diagonal elements. We assume that the pair $(A_p, (B_p \Lambda))$ is controllable.

The control goal of interest is bounded command tracking, that is we need to design u such that the system regulated output

$$y = C_p x_p \in R^m \quad (10.3)$$

tracks any bounded possibly time-varying command $y_{\text{cmd}}(t) \in R^m$, with bounded errors and in the presence of the system uncertainties $\{A_p, \Lambda, \Theta_d\}$, where the system output matrix C_p is known and constant.

Let,

$$e_y(t) = y(t) - y_{\text{cmd}}(t) \quad (10.4)$$

denote the system output tracking error. Augmenting (10.1) with the integrated output tracking error,

$$\left(e_{yI}(t) = \int_0^t e_y(\tau) d\tau \right) \Leftrightarrow \left(e_{yI} = \frac{e_y}{s} \right) \quad (10.5)$$

yields the extended open-loop dynamics,

$$\dot{x} = A x + B \Lambda (u + f(x_p)) + B_{\text{ref}} y_{\text{cmd}} \quad (10.6)$$

where $x = \begin{pmatrix} e_{yI}^T & x_p^T \end{pmatrix}^T \in R^n$ is the extended system state vector, whose dimension is $n = n_p + m$. The extended open-loop system matrices are

$$A = \begin{pmatrix} 0_{m \times m} & C_p \\ 0_{n_p \times m} & A_p \end{pmatrix}, \quad B = \begin{pmatrix} 0_{m \times m} \\ B_p \end{pmatrix}, \quad B_{\text{ref}} = \begin{pmatrix} -I_{m \times m} \\ 0_{n_p \times m} \end{pmatrix} \quad (10.7)$$

and

$$y = \underbrace{\begin{pmatrix} 0_{m \times m} & C_p \end{pmatrix}}_C x = C x \quad (10.8)$$

represents the extended system controlled output. We will require preservation of controllability for the extended pair of matrices $(A, (B \Lambda))$ in (10.7). It is not difficult to show that the extended pair is controllable if and only if the original pair $(A_p, (B_p \Lambda))$ is controllable and $\det \begin{pmatrix} A_p & B_p \Lambda \\ C_p & 0_{m \times m} \end{pmatrix} \neq 0$.

To summarize, we are interested in the state feedback output regulation problem for a generic class of MIMO uncertain dynamical systems in the form,

$$\begin{aligned}\dot{x} &= A x + B \Lambda \left(u + \Theta^T \Phi(x_p) \right) + B_{\text{ref}} y_{\text{cmd}}(t) \\ y &= C x\end{aligned}\tag{10.9}$$

with known constant matrices $A \in R^{n \times n}$, $B \in R^{n \times m}$, $B_{\text{ref}} \in R^{n \times m}$, $C \in R^{m \times n}$, an unknown constant diagonal positive-definite matrix $\Lambda \in R^{m \times m}$, and an unknown matrix of constant parameters $\Theta \in R^{N \times m}$.

The control goal is to force the system regulated output $y(t) \in R^{m \times 1}$ to track any bounded time-varying reference signal $y_{\text{cmd}}(t) \in R^{m \times 1}$, with bounded errors and in the presence of constant parametric uncertainties (A, Λ, Θ) . We shall also require that the rest of the signals in the corresponding closed-loop system remain uniformly bounded in time.

10.2 Control Design

We commence with the assumption about the existence of an adaptive solution to the MIMO command tracking problem of interest.

Assumption 10.1 (Model Matching Conditions) Given a reference Hurwitz matrix A_{ref} and an unknown positive-definite diagonal constant matrix Λ , there exist a constant (possibly unknown) gain matrix $K_x \in R^{n \times m}$, such that

$$A_{\text{ref}} = A + B \Lambda K_x^T\tag{10.10}$$

Using (10.10), we can rewrite the open-loop extended system dynamics (10.9) in the form,

$$\dot{x} = A_{\text{ref}} x + B \Lambda \left(u - K_x^T x + \Theta^T \Phi(x_p) \right) + B_c y_{\text{cmd}}\tag{10.11}$$

and then choose,

$$u = \hat{K}_x^T x - \hat{\Theta}^T \Phi(x_p)\tag{10.12}$$

where $\hat{K}_x(t) \in R^{n \times m}$ and $\hat{\Theta}(t) \in R^{N \times m}$ are adaptive gains whose dynamics will be defined later. Substituting (10.12) into (10.11), yields

$$\begin{aligned}\dot{x} &= A_{\text{ref}} x + B \Lambda \left(\underbrace{\left(\hat{K}_x - K_x \right)^T}_{\Delta K_x} x - \underbrace{\left(\hat{\Theta} - \Theta \right)^T}_{\Delta \Theta} \Phi(x_p) \right) + B_{\text{ref}} y_{\text{cmd}} \\ &= A_{\text{ref}} x + B \Lambda \left(\Delta K_x^T x - \Delta \Theta^T \Phi(x_p) \right) + B_{\text{ref}} y_{\text{cmd}}\end{aligned}\quad (10.13)$$

Based on (10.13), we consider the following reference model,

$$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} y_{\text{cmd}}, \quad y_{\text{ref}} = C x_{\text{ref}} \quad (10.14)$$

It is easy to verify that the transfer function $G_{\text{ref}}(s)$ from y_{cmd} to y_{ref} ,

$$y_{\text{ref}} = \underbrace{\left[C(sI_{n \times n} - A_{\text{ref}})^{-1} B_{\text{ref}} \right]}_{G_{\text{ref}}(s)} y_{\text{cmd}} \quad (10.15)$$

has the unity DC gain, where s is the Laplace variable. This feature formally prescribes the desired output regulation behavior for constant external commands.

We define the state tracking error,

$$e = x - x_{\text{ref}} \quad (10.16)$$

and then subtract (10.14) from (10.13) to obtain the tracking error dynamics.

$$\dot{e} = A_{\text{ref}} e + B \Lambda \left(\Delta K_x^T x - \Delta \Theta^T \Phi(x_p) \right) \quad (10.17)$$

We now proceed with the Lyapunov-based approach, eventually leading to the design of stable adaptive laws and a verifiable closed-loop system tracking performance. Toward that end, let us consider a radially unbounded Lyapunov function candidate,

$$\begin{aligned}V(e, \Delta K_x, \Delta \Theta) &= e^T P_{\text{ref}} e + \text{trace}(\Delta K_x^T \Gamma_x^{-1} \Delta K_x \Lambda) \\ &\quad + \text{trace}(\Delta \Theta^T \Gamma_{\Theta}^{-1} \Delta \Theta \Lambda)\end{aligned}\quad (10.18)$$

where $\Gamma_x = \Gamma_x^T > 0$, $\Gamma_{\Theta} = \Gamma_{\Theta}^T > 0$ are rates of adaptation, and $P_{\text{ref}} = P_{\text{ref}}^T > 0$ is the unique symmetric positive-definite solution of the algebraic Lyapunov equation,

$$P A_{\text{ref}} + A_{\text{ref}}^T P = -Q \quad (10.19)$$

with some $Q = Q^T > 0$. The time derivative of V , along the trajectories of (10.17), is

$$\begin{aligned}\dot{V}(e, \Delta K_x, \Delta \Theta) &= -e^T Q e + 2 e^T P B \Lambda \left(\Delta K_x^T x - \Delta \Theta^T \Phi(x_p) \right) \\ &\quad + 2 \text{trace} \left(\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x \Lambda \right) + 2 \text{trace} \left(\Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \Lambda \right)\end{aligned}\quad (10.20)$$

Applying the vector trace identity (valid for any two co-dimensional vectors a and b),

$$a^T b = \text{trace} \left(b a^T \right) \quad (10.21)$$

results in

$$\begin{aligned}\dot{V}(e, \Delta K_x, \Delta \Theta) &= -e^T Q e + 2 \text{trace} \left(\Delta K_x^T \left\{ \Gamma_x^{-1} \dot{\hat{K}}_x + x e^T P B \right\} \Lambda \right) \\ &\quad + 2 \text{trace} \left(\Delta \Theta^T \left\{ \Gamma_\Theta^{-1} \dot{\hat{\Theta}} - \Phi(x_p) e^T P B \right\} \Lambda \right)\end{aligned}\quad (10.22)$$

If adaptive laws are selected in the form,

$$\begin{aligned}\dot{\hat{K}}_x &= -\Gamma_x x e^T P B \\ \dot{\hat{\Theta}} &= \Gamma_\Theta \Phi e^T P B\end{aligned}\quad (10.23)$$

then,

$$\dot{V}(e, \Delta K_x, \Delta \Theta) = -e^T Q e \leq 0 \quad (10.24)$$

which in turn, proves uniform ultimate boundedness of $(e, \Delta K_x, \Delta \Theta)$.

Let L_2 and L_∞ define the set of all square integrable and bounded functions in a Euclidean space of interest, respectively.

Relation (10.24) implies that the tracking error signal is square integrable: $e \in L_2$. Since $y_{\text{cmd}} \in L_\infty$ then $x_{\text{ref}} \in L_\infty$, and consequently $x \in L_\infty$. Since the ideal (unknown) parameters (K_x, Θ) are constant and their estimation errors $(\Delta K_x, \Delta \Theta)$ are bounded, then the corresponding estimated values are bounded as well, that is, $(\hat{K}_x, \hat{\Theta}) \in L_\infty$.

Since all components of the regressor vector $\Phi(x_p)$ are locally Lipschitz-continuous functions of $x_p \in L_\infty$, then the regressor components are also bounded. Hence, $u \in L_\infty$ and $\dot{x} \in L_\infty$. Thus $\dot{e} \in L_\infty$, which implies that $\ddot{V} \in L_\infty$. Therefore, \dot{V} is a uniformly continuous function of time.

Since V is lower bounded, $\dot{V} \leq 0$, and \dot{V} is uniformly continuous then V tends to a limit, while its derivative \dot{V} tends to zero, (Barbalat's lemma). Consequently, the tracking error, e tends to zero asymptotically, as $t \rightarrow \infty$. Moreover, since the Lyapunov function (10.18) is radially unbounded, then the asymptotic convergence is global, that is the closed-loop tracking error dynamics are globally asymptotically stable.

Table 10.1 Design summary for MRAC with integral action

Open-loop plant	$\dot{x}_p = A_p x_p + B_p \Lambda(u + \Theta^T \Phi(x_p))$ $y = C_p x_p$
Integrated output tracking error and extended state	$\dot{e}_{I,y} = y - y_{\text{cmd}}, \quad x = (e_{yI}^T \ x_p^T)^T$
Open-loop extended plant	$\dot{x} = Ax + B \Lambda(u + \Theta^T \Phi(x_p)) + B_{\text{ref}} y_{\text{cmd}}$ $y = C x$
Reference model	$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} y_{\text{cmd}}, \quad y_{\text{ref}} = C x_{\text{ref}}$
Tracking error	$e = x - x_{\text{ref}}$
Control input	$u = \hat{K}_x^T x - \hat{\Theta}^T \Phi(x_p)$
Algebraic Lyapunov equation	$PA_{\text{ref}} + A_{\text{ref}}^T P = -Q$
MRAC laws	$\dot{\hat{K}}_x = -\Gamma_x x e^T P B$ $\dot{\hat{\Theta}} = \Gamma_\Theta \Phi(x_p) e^T P B$

We have shown that the system state x globally asymptotically tracks the state x_{ref} of the reference model, and therefore the system output $y = C x$ globally asymptotically tracks the reference model output $y_{\text{ref}} = C x_{\text{ref}}$. At the same time, the reference model dynamics are chosen such that y_{ref} tracks an external bounded command $y_{\text{cmd}}(t)$, with bounded errors. Therefore, y must also track y_{cmd} with bounded errors. The MIMO command tracking problem is solved.

The equations summary is given in Table 10.1 and the end result is stated in Theorem 10.1.

Theorem 10.1 Consider the uncertain system dynamics in (10.19), operating under the MRAC controller (10.12), with the adaptive laws (10.23). Suppose that the matching condition (10.10) holds. Let the reference model (10.14) be driven by a bounded external command $y_{\text{cmd}}(t)$. Then for any symmetric positive-definite matrices $(\Gamma_x, \Gamma_\Theta, Q_{\text{ref}})$, all signals in the closed-loop system,

$$\begin{cases} \dot{x} = Ax + B \Lambda(\hat{K}_x^T x - \hat{\Theta}^T \Phi(x_p) + \Theta^T \Phi(x_p)) + B_{\text{ref}} y_{\text{cmd}}(t) \\ \dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} y_{\text{cmd}}(t) \\ \dot{\hat{K}}_x = -\Gamma_x x (x - x_{\text{ref}})^T P B \\ \dot{\hat{\Theta}} = \Gamma_\Theta \Phi(x_p) (x - x_{\text{ref}})^T P B \end{cases} \quad (10.25)$$

are uniformly ultimately bounded in time, where P represents the unique symmetric positive-definite solution of the algebraic Lyapunov equation (10.19). Moreover, the tracking error signal $e = x - x_{\text{ref}}$ is uniformly ultimately bounded, square integrable, and tends to the origin globally and asymptotically, that is, $\lim_{t \rightarrow \infty} \|e(t)\| = 0$. ■

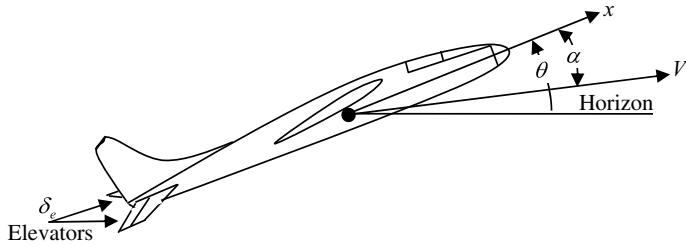


Fig. 10.1 Aircraft short-period motion in Example 10.1

In order to illustrate both usefulness and practicality of MRAC systems with integral feedback, we shall consider an example.

Example 10.1 Aircraft Short-Period Dynamics and Control Longitudinal motion of a conventional aircraft is controlled by engine throttles and elevators (movable trailing edge tail surfaces, δ_e). While throttles are the primary inputs for regulating airspeed, deflecting elevators up or down will change the aircraft pitch rate $q = \dot{\theta}$ and as a consequence, its orientation with respect to the horizon (the pitch angle, θ). At the same time, elevator movements will directly influence the aircraft angle of attack α , (the angle between the velocity vector and the fuselage longitudinal axis x). Figure 10.1 shows an aircraft sketch, with all the relevant degrees-of-freedom.

Coupled relations between α and q , driven by the elevator deflection δ_e , constitute the aircraft short-period dynamics. Assuming fixed throttle setting and constant airspeed, the aircraft short-period dynamics can be approximated by a second-order differential equation in the form,

$$\underbrace{\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix}}_{\dot{x}_p} = \underbrace{\begin{pmatrix} \frac{Z_\alpha}{V} & 1 + \frac{Z_q}{V} \\ M_\alpha & M_q \end{pmatrix}}_{A_p} \underbrace{\begin{pmatrix} \alpha \\ q \end{pmatrix}}_{x_p} + \underbrace{\begin{pmatrix} \frac{Z_\delta}{V} \\ M_\delta \end{pmatrix}}_{B_p} \Lambda \left(\underbrace{\delta_e}_{u} + f(x_p) \right) \quad (10.26)$$

where α (rad) is the aircraft angle of attack, q (rad/s) is the pitch rate, V (ft/s) is the true airspeed (assumed constant), δ_e (rad) is the elevator deflection (the control input), and $(Z_\alpha, Z_q, Z_\delta, M_\alpha, M_q, M_\delta)$ are the aircraft stability derivatives [1, 2]. In (10.26), $\Lambda > 0$ represents a loss-of-control effectiveness, and $f(x_p)$ is the matched uncertainty in the system dynamics.

We make use of generic transport aircraft (DC-8) cruise data from [1, p. 712] to populate the short-period model,

$$A_p = \begin{pmatrix} -0.8060 & 1.0 \\ -9.1486 & -4.59 \end{pmatrix}, \quad B_p = \begin{pmatrix} -0.04 \\ -4.59 \end{pmatrix}$$

and define the aircraft angle of attack α to be the system regulated output.

$$y = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{C_p} x_p = \alpha$$

The open-loop system has its eigenvalues in the left half plane. So, the short-period dynamics are open-loop stable, ($\omega_n = 3.58$ rad/s, $\zeta = 0.753$).

This model is augmented with the integrated output (angle of attack) tracking error. The extended open-loop matrices are

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -0.8060 & 1 \\ 0 & -9.1486 & -4.59 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ -0.04 \\ -4.59 \end{pmatrix}, \quad B_{\text{ref}} = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \quad C = (0 \ 1 \ 0)$$

Assuming that these linear data are known, our next step is to construct a suitable reference model. We can use a linear control design technique to accomplish this task. Such an approach would allow us to automatically satisfy the required matching conditions (10.10), and at the same time, to construct a reference model with the desired transient characteristics. We choose the linear quadratic regulator (LQR) method [3], as our baseline control design tool. Since the open-loop dynamics are already stable and sufficiently fast, we pick LQR weight matrices

$$Q_{\text{LQR}} = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad R_{\text{LQR}} = 1$$

and arrive at the desired reference model, which represents the baseline closed-loop short-period dynamics,

$$A_{\text{ref}} = A + B \underbrace{\begin{pmatrix} 3.1623 & 1.1016 & 0.2152 \end{pmatrix}}_{-K_{\text{LQR}}^T} = \begin{pmatrix} 0 & 1 & 0 \\ -0.1328 & -0.8522 & 0.9910 \\ -14.5149 & -14.2048 & -5.5779 \end{pmatrix}$$

achieved via the LQR control feedback.

$$u_{\text{LQR}} = -K_{\text{LQR}}^T x$$

The reference model natural frequency and damping are quite close to those of the open-loop: ($\omega_n \text{LQR} = 3.57$ rad/s, $\zeta_{\text{LQR}} = 0.734$). The integrator pole represents the closed-loop system dominant eigenvalue. The pole is placed at $\lambda = -1.1873$ to enable adequate tracking performance with a reasonable control (elevator deflection) effort (see Fig. 10.2).

In this case, the reference and the actual system responses coincide with each other. This is to be expected since the system is simulated without uncertainties.

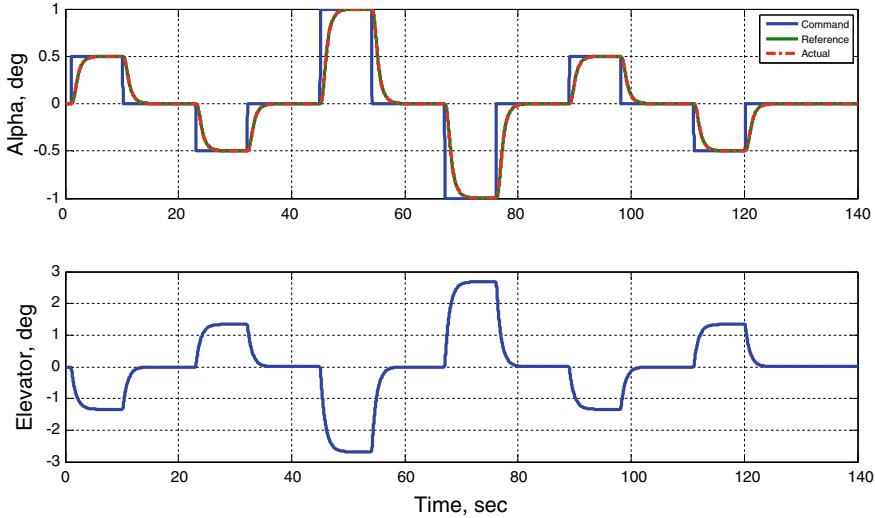


Fig. 10.2 Closed-loop baseline (no uncertainties) tracking performance in Example 10.1

Next, we introduce linear state-dependent uncertainties into the system dynamics,

$$f(x_p) = f(\alpha, q) = k_\alpha \alpha + k_q q$$

and choose $\Lambda = 0.5$, $k_\alpha = 1.5 M_\alpha$, and $k_q = 0.5 M_q$. Our particular selection corresponds to simultaneous changes in: (a) The control effectiveness M_δ ; (b) The static stability M_α , and (c) The pitch damping M_q . These uncertainties are intentionally chosen to destabilize the LQR closed-loop short-period open-loop dynamics. In fact, with these uncertainties, the perturbed open-loop dynamics become

$$\begin{aligned} \dot{x} &= A x + B \Lambda (u + k_\alpha \alpha + k_q q) \\ &= A_{\text{ref}} x + B \Lambda \left(u + \Lambda^{-1} K_{\text{LQR}}^T x + k_\alpha \alpha + k_q q \right) \end{aligned}$$

and so, the ideal unknown controller gains are

$$K_x \text{ ideal} = -\left(\Lambda^{-1} K_{\text{LQR}}^T + (0 \ k_\alpha \ k_q) \right)^T = (6.3246 \ 15.9261 \ 2.7254)^T$$

In other words, if we knew the uncertainties then the linear feedback

$$u_{\text{ideal}} = K_x^T \text{ ideal } x$$

would have enforced the desired reference dynamics.

The next step is to design a MRAC system in order to recover the desired closed-loop performance, without any information about the parametric uncertainties. Since $f(x_p)$ is linear in x_p , only the adaptive gains \hat{K}_x are required. Here, the design “tuning knobs” consist of two symmetric positive-definite (3×3) matrices Q and Γ_x . After several iterations, we have selected

$$Q = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{pmatrix}, \quad \Gamma_x = \begin{pmatrix} 2000 & 0 & 0 \\ 0 & 2000 & 0 \\ 0 & 0 & 200 \end{pmatrix}$$

Our iterative design focus was on reducing unwanted transient oscillations, while providing adequate command tracking performance. Utilizing MRAC design equations from Table 10.1 and with the uncertainties turned on, the corresponding simulated closed-loop system tracking performance data are shown in Fig. 10.3.

Clearly, the MRAC design is able to recover the baseline closed-loop dynamics. However, the control effort is significantly larger than before, and the uncertainties are the driving factor. Dynamics of the corresponding adaptive gains are shown in Fig. 10.4.

The three gains approach their ideal values. This is a “bonus”, since parameter convergence is not guaranteed by the MRAC design.

In order to demonstrate good tracking without parameter convergence, we select $y_{cmd} = \sin(0.1 t)$ and simulate the closed-loop system without any other changes. As expected, the output tracking performance remains of good quality (Fig. 10.5).

On the other hand, the adaptive gains are different from the ideal values K_x ideal that are defined by the corresponding matching conditions (Fig. 10.6).

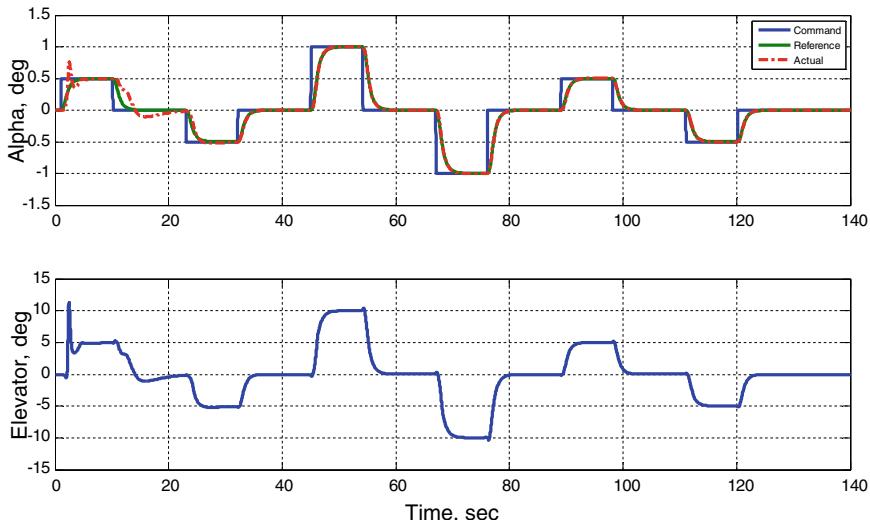


Fig. 10.3 Closed-loop performance recovery under MRAC system in Example 10.1

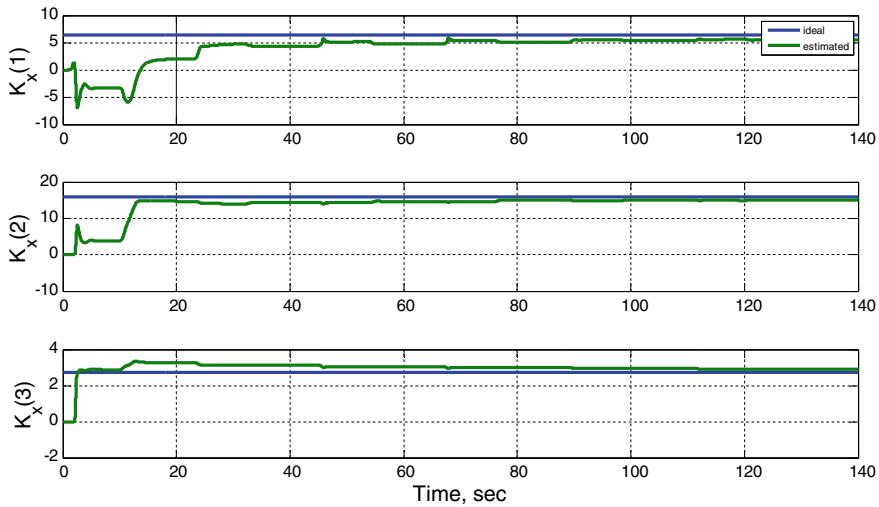


Fig. 10.4 Evolution of adaptive gains in Example 10.1

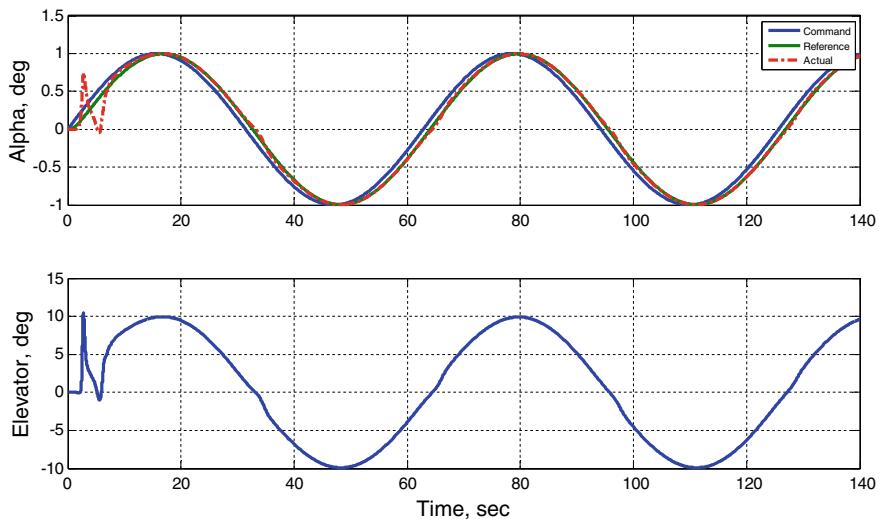


Fig. 10.5 Output tracking of a sinusoidal command in Example 10.1

Parameter convergence in adaptive control depends on the persistency of excitation (PE) conditions [4, 5]. Basically, the external command needs to “persistently excite” the closed-loop system dynamics. For linear dynamical systems with linear-in-parameters uncertainties (such as those considered in this example) the PE

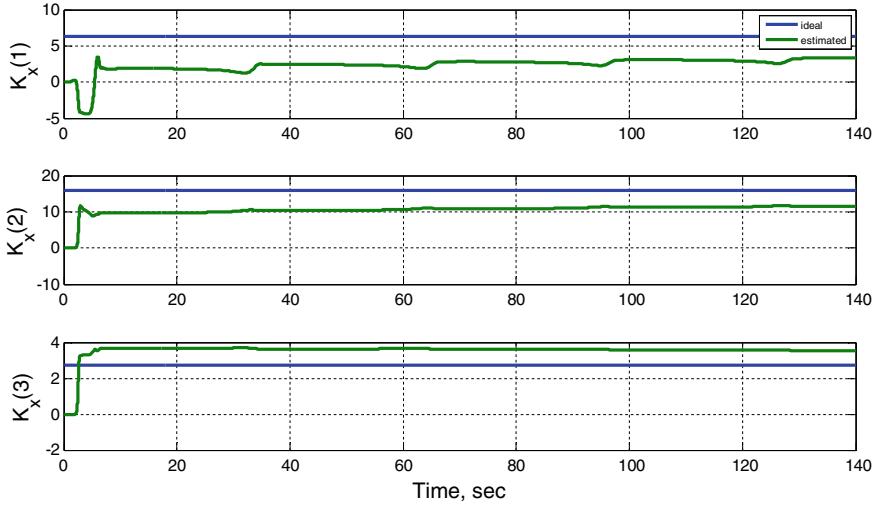


Fig. 10.6 Evolution of adaptive gains during tracking of a sinusoid in Example 10.1

conditions are satisfied if the system external command is chosen as a sum of sinusoids with different frequencies. Then, a single frequency would give exponential convergence of two adaptive gains to their corresponding unknown constant ideal values. For nonlinear systems, this rule is no longer holds and the generic PE conditions are hard to verify numerically. ■

10.3 MRAC Augmentation of an Optimal Baseline Controller

The adaptive design developed in the previous section can be modified to augment a baseline linear controller with the (Proportional + Integral) (PI) feedback architecture. The rational for using an augmentation approach (as oppose to all adaptive) stems from the fact that in most realistic applications, a system may already have a baseline controller, which often is designed to contain proportional as well as integral feedback connections. Such a baseline controller would have been intended to operate under nominal conditions (no uncertainties), where it would asymptotically reject constant unknown disturbances and track constant commands with zero errors. If adding uncertainties destroys the expected baseline closed-loop performance then one might attempt to recover the desired performance by augmenting the baseline controller with an adaptive element.

We consider the same class of n -dimensional MIMO nonlinear systems with m controls (as defined in (10.1)), whose plant dynamics are linearly parameterized, the uncertainties satisfy matching conditions, and the system state is measurable (i.e., available on-line for control synthesis). The system dynamics are,

$$\dot{x}_p = A_p x_p + B_p \Lambda (u + f(x_p)) \quad (10.27)$$

where n_p and m are the dimensions of the system state x_p and of the control u , respectively. Also we assume that $A_p \in R^{n_p \times n_p}$ and $B_p \in R^{n_p \times m}$ are known, while $\Lambda \in R^{m \times m}$ is an unknown diagonal matrix with strictly positive diagonal elements λ_i . The pair $(A_p, (B_p \Lambda))$ is presumed controllable, and the constant uncertainty Λ is introduced to model possible imperfections in the system control channels.

The unknown nonlinear function $f(x_p) : R^{n_p} \rightarrow R^m$ represents the system matched uncertainty. It is assumed that this function can be written as a linear combination of N known basis functions, with unknown constant coefficients.

$$f(x_p) = \Theta^T \Phi(x_p) \quad (10.28)$$

In (10.28), $\Theta \in R^{N \times m}$ is the unknown constant matrix of ideal parameters, and $\Phi(x_p) \in R^N$ represents the known locally Lipschitz-continuous regressor vector. Thus, we consider a generic class of MIMO systems in the form,

$$\dot{x}_p = A_p x_p + B_p \Lambda (u + \Theta^T \Phi(x_p)) \quad (10.29)$$

with the regulated output,

$$y = C_p x_p + D_p \Lambda (u + \Theta^T \Phi(x_p)) \quad (10.30)$$

where $C_p \in R^{m \times n_p}$ and $D_p \in R^{m \times m}$ are known and constant.

Let $y_{\text{cmd}}(t) \in R^m$ denote a bounded command for the system output $y \in R^m$ to follow. This task is to be accomplished using the system control input $u \in R^m$, in the form of a full state feedback.

We define the output tracking error,

$$e_y(t) = y(t) - y_{\text{cmd}}(t) \quad (10.31)$$

its integral e_{yI} ,

$$\dot{e}_{yI} = e_y = y - y_{\text{cmd}} \quad (10.32)$$

and formulate the extended open-loop dynamics,

$$\underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0_{m \times m} & C_p \\ 0_{n_p \times m} & A_p \end{pmatrix}}_A \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} D_p \\ B_p \end{pmatrix}}_B \Lambda (u + f(x_p)) + \underbrace{\begin{pmatrix} -I_{m \times m} \\ 0_{n_p \times m} \end{pmatrix}}_{B_{\text{ref}}} y_{\text{cmd}} \quad (10.33)$$

or, equivalently

$$\dot{x} = A x + B \Lambda (u + \Theta^T \Phi(x_p)) + B_{\text{ref}} y_{\text{cmd}} \quad (10.34)$$

In terms of (10.34), the system regulated output y in (10.30) can be written as

$$y = \underbrace{\begin{pmatrix} 0 & C_p \end{pmatrix}}_C \underbrace{\begin{pmatrix} e_y I \\ x_p \end{pmatrix}}_x + \underbrace{D_p}_D \Lambda \left(u + \Theta^T \Phi(x_p) \right) = C x + D \Lambda \left(u + \Theta^T \Phi(x_p) \right) \quad (10.35)$$

The control problem of interest is bounded tracking in the presence of the system constant parametric uncertainties Λ and Θ . Specifically, we need to design the control input u , so that the system regulated output y tracks any bounded time-varying command y_{cmd} , with bounded tracking errors, while the rest of the signals in the corresponding closed-loop dynamics remain bounded.

We begin with the design of a baseline linear controller. Setting $\Lambda = I_{m \times m}$, $\Theta = 0_{N \times m}$ in (10.28), results in the linear baseline open-loop dynamics.

$$\begin{aligned} \dot{x} &= A x + B u + B_{\text{ref}} y_{\text{cmd}} \\ y &= C x + D u \end{aligned} \quad (10.36)$$

Assuming constant command y_{cmd} , we can use the linear quadratic regulator (LQR) method, with Proportional + Integral (PI) feedback connections, to design the baseline LQ optimal control law, in the form of an LQR PI servomechanism. This design is outlined below.

We first calculate the optimal stabilizing controller for

$$\dot{z} = A z + B v \quad (10.37)$$

where

$$z = \dot{x} = \begin{pmatrix} \dot{e}_y I \\ \dot{x}_p \end{pmatrix}, \quad v = \dot{u} \quad (10.38)$$

and the control input v is designed to minimize the linear quadratic cost index,

$$J(v) = \int_0^\infty \left(z^T Q z + v^T R v \right) dt \quad (10.39)$$

with the appropriately selected symmetric positive-definite matrices Q and R . It is well-known that the corresponding optimal LQR solution is given in feedback form

$$v = \dot{u} = - \underbrace{R^{-1} B^T P}_{K_x^T} z = - \begin{pmatrix} K_I & K_P \end{pmatrix} \begin{pmatrix} \dot{e}_y I \\ \dot{x}_p \end{pmatrix} \quad (10.40)$$

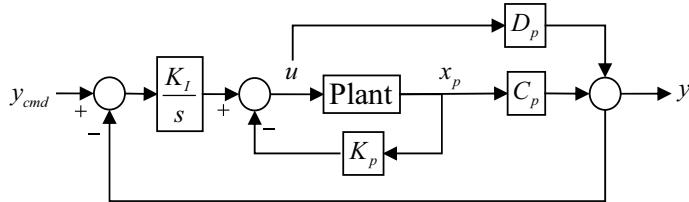


Fig. 10.7 Baseline servomechanism LQR PI control block diagram

In (10.40), P is the unique symmetric positive-definite solution of the Algebraic Riccati Equation,

$$A^T P + P A + Q - P B R^{-1} B^T P = 0 \quad (10.41)$$

which is solved using an appropriately chosen $Q = Q^T \geq 0$. Integrating (10.40), yields the baseline LQR PI controller,

$$u_{bl} = -K_x^T x = -K_I e_{yI} - K_P x = K_I \frac{(y_{cmd} - y)}{s} - K_P x_p \quad (10.42)$$

where the optimal gain matrix

$$K_x^T = (K_I \ K_P) \quad (10.43)$$

is partitioned into the integral gain K_I and the proportional gain K_P . The corresponding baseline LQR PI control block diagram is shown in Fig. 10.7.

In the presence of the system uncertainties Λ and Θ , the baseline tracking performance will often deteriorate. In order to restore the expected baseline behavior, we augment the baseline system with an adaptive element. This process consists of: (a) the reference model definition; (b) the tracking dynamics formulation, and (c) the design of adaptive laws.

First, we define the reference model to represent the baseline closed-loop system dynamics, which are obtained by substituting the baseline controller (10.42) into the linear system (10.36). The resulting reference model dynamics become,

$$\dot{x}_{ref} = A_{ref} x_{ref} + B_{ref} y_{cmd}, \quad y_{ref} = C_{ref} x_{ref} \quad (10.44)$$

where

$$A_{ref} = A - B K_x^T, \quad C_{ref} = C - D K_x^T \quad (10.45)$$

and A_{ref} is Hurwitz by design.

Then, we synthesize the total control input as the sum of the baseline LQR PI component (10.42) and its adaptive augmentation u_{ad} (to be constructed),

$$u = \underbrace{-K_x^T x + u_{ad}}_{u_{bl}} = u_{bl} + u_{ad} \quad (10.46)$$

Substituting (10.46) into the original system dynamics (10.34), gives

$$\begin{aligned} \dot{x} &= A_{\text{ref}} x + B \Lambda \left(u_{ad} + \underbrace{\left(I_{m \times m} - \Lambda^{-1} \right) u_{bl}}_{\overline{\Theta}^T \overline{\Phi}(u_{bl}, x_p)} + \Theta^T \Phi(x_p) \right) + B_{\text{ref}} y_{\text{cmd}} \\ y &= C_{\text{ref}} x + D \Lambda \left(u_{ad} + \overline{\Theta}^T \overline{\Phi}(u_{bl}, x_p) \right) \end{aligned} \quad (10.47)$$

or equivalently,

$$\begin{aligned} \dot{x} &= A_{\text{ref}} x + B \Lambda \left(u_{ad} + \overline{\Theta}^T \overline{\Phi}(u_{bl}, x_p) \right) + B_{\text{ref}} y_{\text{cmd}} \\ y &= C_{\text{ref}} x + D \Lambda \left(u_{ad} + \overline{\Theta}^T \overline{\Phi}(u_{bl}, x_p) \right) \end{aligned} \quad (10.48)$$

with the redefined regressor vector,

$$\overline{\Phi}(u_{bl}, x_p) = (u_{bl}^T \Phi^T(x_p))^T \quad (10.49)$$

and with the extended matrix of unknown/ideal parameters.

$$\overline{\Theta} = (K_u^T \Theta^T)^T \quad (10.50)$$

The adaptive component u_{ad} is chosen to dominate the system matched uncertainty $\overline{\Theta}^T \overline{\Phi}(u_{bl}, x_p)$,

$$u_{ad} = -\hat{\overline{\Theta}}^T \overline{\Phi}(u_{bl}, x_p) \quad (10.51)$$

where $\hat{\overline{\Theta}} \in R^{(n+N) \times m}$ is the matrix of adaptive parameters. Substituting (10.51) into (10.48), results in

$$\begin{aligned}\dot{x} &= A_{\text{ref}} x - B \Lambda \underbrace{\left(\hat{\Theta}^T - \bar{\Theta} \right)}_{\Delta \bar{\Theta}} \bar{\Phi} + B_{\text{ref}} y_{\text{cmd}} \\ y &= C_{\text{ref}} x - D \Lambda \Delta \bar{\Theta}^T \bar{\Phi}\end{aligned}\quad (10.52)$$

where

$$\Delta \bar{\Theta} = \hat{\Theta} - \bar{\Theta} \quad (10.53)$$

is the matrix of parameter estimation errors. We now introduce the state tracking error,

$$e = x - x_{\text{ref}} \quad (10.54)$$

and calculate the tracking error dynamics by subtracting the reference system dynamics (10.44) from the extended open-loop system dynamics (10.52).

$$\dot{e} = A_{\text{ref}} e - B \Lambda \Delta \bar{\Theta}^T \bar{\Phi} \quad (10.55)$$

In order to design MRAC laws and at the same time enforce closed-loop stability of the error dynamics, we consider a radially unbounded quadratic Lyapunov function candidate such as,

$$V(e, \Delta \bar{\Theta}) = e^T P_{\text{ref}} e + \text{trace} \left(\Delta \bar{\Theta}^T \Gamma_{\bar{\Theta}}^{-1} \Delta \bar{\Theta} \Lambda \right) \quad (10.56)$$

where elements of $\Gamma_{\bar{\Theta}} = \Gamma_{\bar{\Theta}}^T > 0$ represent rates of adaptation, and $P_{\text{ref}} = P_{\text{ref}}^T > 0$ is the unique symmetric positive-definite solution of the algebraic Lyapunov equation,

$$A_{\text{ref}}^T P_{\text{ref}} + P_{\text{ref}} A_{\text{ref}} = -Q_{\text{ref}} \quad (10.57)$$

with some appropriately chosen matrix $Q_{\text{ref}} = Q_{\text{ref}}^T > 0$. Time-differentiating V , along the trajectories of (10.55), gives

$$\dot{V}(e, \Delta \bar{\Theta}) = -e^T Q_{\text{ref}} e - 2e^T P_{\text{ref}} B \Lambda \Delta \bar{\Theta}^T \bar{\Phi} + 2 \text{trace} \left(\Delta \bar{\Theta}^T \Gamma_{\bar{\Theta}}^{-1} \dot{\hat{\Theta}} \Lambda \right) \quad (10.58)$$

Applying the vector trace identity,

$$a^T b = \text{trace} \left(b a^T \right) \quad (10.59)$$

further yields

$$\dot{V}(e, \Delta\bar{\Theta}) = -e^T Q_{\text{ref}} e + 2 \text{trace} \left(\Delta\bar{\Theta}^T \left\{ \Gamma_{\bar{\Theta}}^{-1} \dot{\hat{\Theta}} - \bar{\Phi} e^T P_{\text{ref}} B \right\} \Lambda \right) \quad (10.60)$$

If adaptive laws are selected in the form,

$$\dot{\hat{\Theta}} = \Gamma_{\bar{\Theta}} \bar{\Phi}(u_{bl}, x_p) e^T P_{\text{ref}} B \quad (10.61)$$

then,

$$\dot{V}(e, \Delta\bar{\Theta}) = -e^T Q_{\text{ref}} e \leq 0 \quad (10.62)$$

which immediately proves uniform ultimate boundedness of $(e, \Delta\bar{\Theta})$.

Moreover, it follows from (10.62) that the tracking error signal is square integrable, $e \in L_2$. Since $y_{\text{cmd}} \in L_\infty$ then $x_{\text{ref}} \in L_\infty$, and consequently, $x \in L_\infty$ and $(u_{bl}, x_p) \in L_\infty$. Since the ideal (unknown) matrix of parameters $\bar{\Theta}$ is constant and the estimation errors $\Delta\bar{\Theta}$ are bounded, then their estimated values are bounded as well, that is, $\hat{\Theta} \in L_\infty$. Since components of the regressor vector $\bar{\Phi}(u_{bl}, x_p)$ are locally Lipschitz continuous, and $(u_{bl}, x_p) \in L_\infty$ then the regressor components are bounded. Hence, $u \in L_\infty$ and $\dot{x} \in L_\infty$. Thus $\dot{e} \in L_\infty$, which implies that $\ddot{V} \in L_\infty$. Therefore, \dot{V} is a uniformly continuous function of time. Since V is lower bounded, $\dot{V} \leq 0$, and \dot{V} is uniformly continuous then V tends to a limit, while its derivative \dot{V} tends to zero (see Barbalat's lemma, Chap. 8). Consequently, the tracking error e tends to zero asymptotically, as $t \rightarrow \infty$.

Moreover, since the Lyapunov function (10.56) is radially unbounded, then the asymptotic convergence is global, that is the closed-loop tracking error dynamics (10.55) are globally asymptotically stable.

Using the error dynamics (10.55), it is easy to check that $\ddot{e} \in L_\infty$. Then $\dot{e}(t)$ is uniformly continuous. Since in addition $e(t)$ tends to zero then using Barbalat's lemma we conclude that $\lim_{t \rightarrow \infty} \|\dot{e}(t)\| = 0$. Consequently,

$$\lim_{t \rightarrow \infty} \left\| \Delta\bar{\Theta}^T(t) \bar{\Phi}(u_{bl}(t), x(t)) \right\| = 0 \quad (10.63)$$

and

$$y = C x - D \underbrace{\Lambda \left(\Delta\bar{\Theta}^T \bar{\Phi} \right)}_{\rightarrow 0} \rightarrow C_{\text{ref}} x \rightarrow C_{\text{ref}} x_{\text{ref}} = y_{\text{ref}} \quad (10.64)$$

We have proven that for any bounded command y_{cmd} , the closed-loop system output from (10.52) globally asymptotically tracks the reference model output from (10.44), as $t \rightarrow \infty$. At the same time, the reference model dynamics (10.44) are chosen such that y_{ref} tracks any external bounded command $y_{\text{cmd}}(t)$, with bounded

errors. Therefore, y must shall also track y_{cmd} with bounded errors. The MIMO command tracking problem is solved.

The adaptive laws (10.61) can be written in terms of the system original parameters. Partition,

$$\Gamma_{\bar{\Theta}} = \begin{pmatrix} \Gamma_u & 0_{n \times m} \\ 0_{N \times m} & \Gamma_{\Theta} \end{pmatrix} \quad (10.65)$$

where $(\Gamma_u, \Gamma_{\Theta})$ denote rates of adaptation for uncertainties that correspond to x and $\Phi(x_p)$. Using (10.49), (10.50), (10.65), the adaptive laws (10.61) become

$$\begin{aligned} \dot{\hat{K}}_u &= \Gamma_u u_{bl} e^T P_{\text{ref}} B \\ \dot{\hat{\Theta}} &= \Gamma_{\Theta} \Phi(x_p) e^T P_{\text{ref}} B \end{aligned} \quad (10.66)$$

Also, the (LQR PI Baseline + Adaptive) total control input (10.46) is,

$$u = u_{bl} + u_{ad} = \boxed{-K_x^T x}_{u_{bl} = \text{Baseline}} + \boxed{-\hat{K}_u^T u_{bl} - \hat{\Theta}^T \Phi(x_p)}_{u_{ad} = \text{Adaptive Augmentation}} \quad (10.67)$$

or, equivalently

$$\begin{aligned} u &= (I_{m \times m} - \hat{K}_u^T) u_{bl} - \hat{\Theta}^T \Phi(x_p) = -(I_{m \times m} - \hat{K}_u^T) K_x^T x - \hat{\Theta}^T \Phi(x_p) \\ &= (I_{m \times m} - \hat{K}_u^T) \left(K_I \frac{(y_{\text{cmd}} - y)}{s} - K_P x_p \right) - \hat{\Theta}^T \Phi(x_p) \end{aligned} \quad (10.68)$$

Table 10.2 summarizes the developed adaptive augmentation procedure of an LQR PI baseline controller.

By design, this controller does not have a feedforward component. Also note that in the adaptive laws (10.66), the parameter initial values are arbitrary and as such, they can be set to zero. The following flight control design example illustrates the developed methodology.

Example 10.2 Adaptive augmentation design for DC-8 short-period dynamics In Example 10.1 we designed a baseline optimal (LQR PI) controller for regulating short-period dynamics of the DC-8 transport aircraft. Our reference model was selected to represent the closed-loop system that was achieved under the baseline controller. Matched uncertainties were introduced to destabilize the baseline system. After that, we constructed an MRAC controller to recover the desired reference closed-loop performance, with the uncertainties turned on.

We now take a different approach. Instead of using an all-adaptive control solution, we demonstrate how to achieve the same closed-loop performance recovery by utilizing an adaptive augmentation design from Table 10.2. Such an approach would

Table 10.2 MRAC augmentation of a LQR PI baseline system

Open-loop plant	$\dot{x}_p = A_p x_p + B_p \Lambda(u + \Theta^T \Phi(x_p))$ $y = C_p x_p + D_p \Lambda(u + \Theta^T \Phi(x_p))$
Integrated output tracking error and extended state	$\dot{e}_{I,y} = y - y_{\text{cmd}}, \quad x = (e_{y,I}^T \ x_p^T)^T$
Open-loop extended plant	$\dot{x} = A x + B \Lambda(u + \Theta^T \Phi(x_p)) + B_{\text{ref}} y_{\text{cmd}}$ $y = C x + D \Lambda(u + \Theta^T \Phi(x_p))$
Reference model	$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} y_{\text{cmd}}$ $y_{\text{ref}} = C_{\text{ref}} x_{\text{ref}}$
Tracking error	$e = x - x_{\text{ref}}$
Riccati equation for LQR PI controller	$A^T P + P A - P B R^{-1} B^T P + Q = 0$
Baseline control input	$u_{bl} = -R^{-1} P B x$
Lyapunov equation for adaptive laws	$P_{\text{ref}} A_{\text{ref}} + A_{\text{ref}}^T P_{\text{ref}} = -Q_{\text{ref}}$
Total control input	$u = (I_{m \times m} - \hat{K}_u^T) u_{bl} - \hat{\Theta}^T \Phi(x_p)$
MRAC laws	$\dot{\hat{K}}_u = \Gamma_u u_{bl} e^T P_{\text{ref}} B$ $\dot{\hat{\Theta}} = \Gamma_{\Theta} \Phi(x_p) e^T P_{\text{ref}} B$

allow us to retain the baseline controller, instead of performing a complete redesign of the system.

After a few design iterations, we have selected appropriate values for adaptive tuning “knobs” (Table 10.3).

Using the design equations from Table 10.2, the system closed-loop dynamics are simulated with the uncertainties from Example 10.1. Figure 10.8 shows the results.

In comparison to the all-adaptive solution (see Fig. 10.3), the adaptive augmentation design also yields adequate tracking performance and a similar to the previous case control activity. There are also three adaptive gains, whose dynamics along with their corresponding ideal (unknown) values, are shown in Fig. 10.9.

Notwithstanding parameter convergence, the adaptive gains are well-behaved and remain bounded throughout the maneuver, as predicted by the theory. ■

Let us now elaborate on the usefulness of an augmentation-based control design approach. In control engineering applications, a control designer is often faced with a pre-existing controller, which constitutes and provides the baseline (i.e., expected)

Table 10.3 Adaptive augmentation parameters for DC-8 short-period dynamics in Example 10.2

Q matrix for adaptive laws	$Q_{\text{ref}} = \text{diag}(100, 100, 100)$
Rates of adaptation	$\Gamma_u = \Gamma_{\Theta} = 800$
Regressor vector	$\Phi(x_p) = (\alpha \ q)^T$

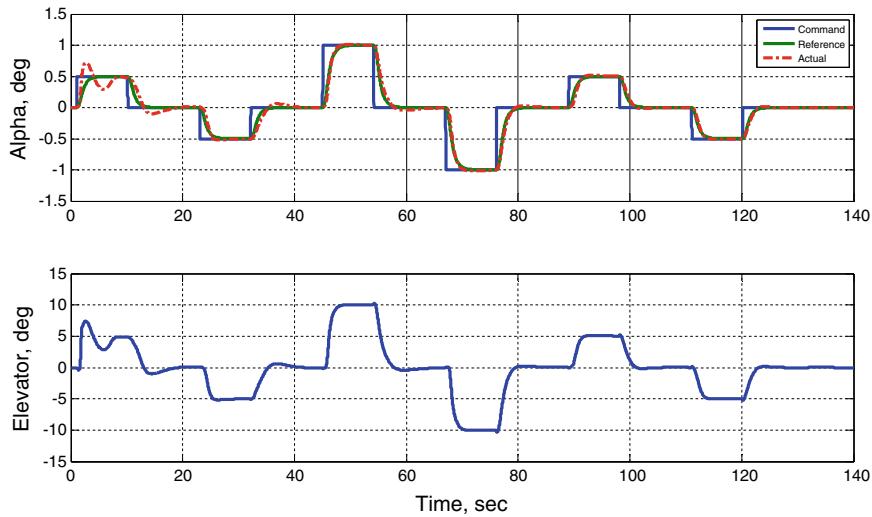


Fig. 10.8 Tracking performance and control effort in Example 10.2

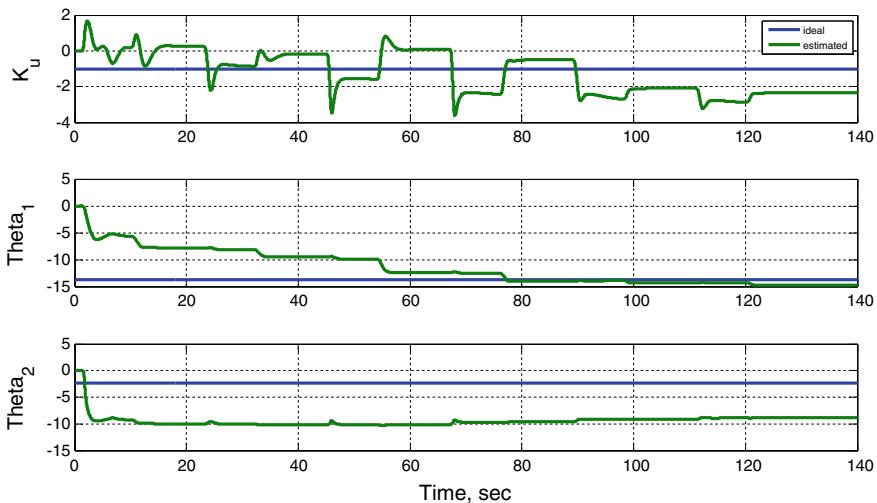


Fig. 10.9 Adaptive gains in Example 10.2

closed-loop tracking performance. Because of that, the control task at hand is to enhance the baseline system performance instead of replacing it with yet another system. Our adaptive augmentation procedure aims exactly at solving this particular task. Using control-theoretic arguments, we have developed a (Baseline + Adaptive) control system, capable of restoring the desired tracking characteristics when

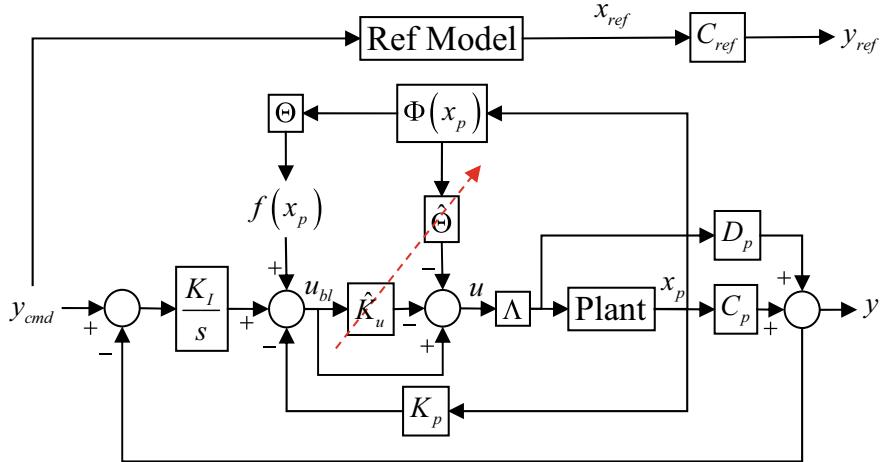


Fig. 10.10 Block diagram: Adaptive augmentation of a baseline PI controller

matched uncertainties are prevalent in the system dynamics. Without the uncertainties, the system resorts to the baseline controller, while its adaptive component becomes inactive.

The overall (Baseline + Adaptive) control block diagram is shown in Fig. 10.10.

The red-dotted line in the figure denotes the adaptive nature of the gains \hat{K}_u and $\hat{\Theta}$, whose dynamics are driven by the tracking error $e = x - x_{ref}$, and according to the adaptive laws (10.66), where x is the state of the extended system (10.33). Per design, the (Baseline + Adaptive) controller will force the system output y asymptotically track the reference model output y_{ref} , in spite of the matched uncertainties $f(x_p) = \Theta^T \Phi(x_p)$ and the unknown control gain Λ .

We have constructed the reference model to represent the desired closed-loop system operating under the baseline PI controller

$$u_{bl} = -K_P x_p - K_I \left(\frac{y - y_{cmd}}{s} \right)$$

If there are no uncertainties and if the adaptive gains are initialized at zero, the tracking error will vanish asymptotically. Consequently, the adaptive gains will be constant and small. Then, the adaptive component

$$u_{ad} = -\hat{K}_u^T u_{bl} - \hat{\Theta}^T \Phi(x_p)$$

will become small as well, and as a result, the system will operate mostly under the baseline controller.

In the presence of uncertainties, the adaptive component becomes active, and it will provide an incremental signal (augmentation) to the baseline PI controller. In

other words, anytime when the tracking error is sufficiently large, the total control signal is,

$$u = u_{bl} + u_{ad}$$

which represents the (Baseline + Adaptive) architecture shown in Fig. 10.10.

It is interesting to note that if the system uncertainties fade away after being active, the adaptive gains will “freeze” and their values will remain constant until the tracking error becomes nonzero again. However, the adaptive component will not be necessarily zero. In fact, the adaptive signal u_{ad} becomes representative of a nonlinear controller with fixed gains and as such, it will continue to add nonzero values to the baseline controller u_{bl} . ■

10.4 Summary

We have demonstrated how to embed fixed-gain linear integral controllers into MRAC design. This leads to adaptive systems with integral action, and provides a capability of tracking time-varying bounded commands without feedforward connections. We have also illustrated the design steps and its associated benefits using short-period dynamics of a generic transport aircraft.

In essence, we have offered a design procedure to combine a baseline linear (proportional + integral) controller with an MRAC system. The specific MRAC augmentation method discussed in this chapter, allows a designer to merge a linear baseline system with an adaptive controller, yet without “canceling” the former. Such an architecture is relevant in industrial applications where stability, performance, and robustness of pre-existing baseline controllers can be enhanced through direct adaptation. This would result in the preservation and a recovery of the system baseline closed-loop performance in the presence of significant uncertainties that may exist in the process dynamics.

10.5 Exercises

Exercise 10.1 Verify that the transfer function $G_{\text{ref}}(s) = C(sI_{n \times n} - A_{\text{ref}})^{-1}B_{\text{ref}}$ in (10.15), (10.44) has the unity DC gain, that is, $G_{\text{ref}}(0) = -C A_{\text{ref}}^{-1} B_{\text{ref}} = I_{m \times m}$.

Exercise 10.2 Table 10.2 presents an adaptive augmentation design. Show that an alternative way to construct an adaptive augmentation of a baseline linear controller $u_{bl} = -K_x^T x$ is to start with the adaptive controller (10.12) and then initialize the adaptive gain \hat{K}_x from (10.23) such that $\hat{K}_x(0) = -K_x$.

Exercise 10.3 Prove (10.63).

Exercise 10.4 For the delta wing dynamics from Example 9.3, assume that A and B matrices are known. The system uncertainties are represented by Λ and $f(x)$. The system regulated output is the bank angle φ . Design a baseline LQR PI controller u_{bl} , and then augment it with an MRAC signal u_{ad} , via equations from Table 10.2. Use Table 10.1 to design a pure adaptive controller and initialize its adaptive state gains at their corresponding baseline (LQR PI) values. Simulate both controllers. Compare and discuss their tracking performance and the associated control efforts.

References

1. McRuer D, Ashkenas I, Graham D (1990) Aircraft dynamics and automatic control. Princeton University Press, Princeton, NJ
2. Stevens BL, Lewis FL (1992) Aircraft control and simulation. Wiley
3. Anderson BDO, Moore JB (2007) Optimal control. Linear quadratic methods. Dover Publications Inc., Mineola, NY
4. Narendra KS, Annaswamy AM (2005) Stable adaptive control. Dover, New York
5. Ioannou P, Fidan P (2006) Adaptive control tutorial. Advances in design and control. SIAM, PA



Robust Adaptive Control

11

This chapter is devoted to the design of adaptive controllers for dynamical systems that operate in the presence of parametric uncertainties and bounded noise. Four MRAC design modifications for robustness are discussed: (a) the dead-zone; (b) the σ -modification; (c) the e -modification, and (d) the Projection Operator. We argue that out of the four modifications, the dead-zone and the Projection Operator are essential for any MRAC system designed to predictably operate in a realistic environment.

11.1 MRAC Design in the Presence of Bounded Disturbances

Our starting point is the MIMO dynamical system,

$$\dot{x} = A_{\text{ref}} x + B \Lambda \left(u + \Theta^T \Phi(x) \right) + B_{\text{ref}} y_{\text{cmd}} + \xi(t) \quad (11.1)$$

whose regulated output is

$$y = C_{\text{ref}} x \quad (11.2)$$

The system is operating in the presence of a uniformly bounded time-dependent disturbance $\xi(t) \in R^n$,

$$\|\xi(t)\| \leq \xi_{\max} \quad (11.3)$$

with its known and constant upper bound $\xi_{\max} \geq 0$. The system matched uncertainties are represented by a diagonal positive definite matrix $\Lambda \in R^{m \times m}$ and a constant matrix $\Theta \in R^{N \times m}$. We assume that the constant matrices

$(A_{\text{ref}}, B, B_{\text{ref}}, C_{\text{ref}})$ are known, the pair $(A_{\text{ref}}, B \Lambda)$ is controllable, and A_{ref} is Hurwitz.

The control objective is to design a state feedback MRAC system to enable bounded tracking of the reference model dynamics,

$$\begin{aligned}\dot{x}_{\text{ref}} &= A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} y_{\text{cmd}} \\ y_{\text{ref}} &= C_{\text{ref}} x_{\text{ref}}\end{aligned}\quad (11.4)$$

with the output y_{ref} . The reference model is driven by a bounded time-dependent command $y_{\text{cmd}} \in R^m$, and the control goal consists of finding a state feedback controller u to force the system output y track a command y_{cmd} , in the presence of the system parametric uncertainties and while keeping the rest of the signals uniformly bounded in time.

Based on (11.1) we choose the control input to be,

$$u = -\hat{\Theta}^T \Phi(x) \quad (11.5)$$

where $\hat{\Theta} \in R^{N \times m}$ is the matrix of adaptive parameters to be determined at a later time. Substituting (11.5) into (11.1), gives

$$\dot{x} = A_{\text{ref}} x - B \Lambda \Delta \Theta^T \Phi(x) + B_{\text{ref}} y_{\text{cmd}} + \xi(t) \quad (11.6)$$

where

$$\Delta \Theta = \hat{\Theta} - \Theta \quad (11.7)$$

is the matrix of the parameter estimation errors. Let,

$$e = x - x_{\text{ref}} \quad (11.8)$$

be the state tracking error. Subtracting the reference model dynamics (11.4) from that of the system (11.1), yields the tracking error dynamics

$$\dot{e} = A_{\text{ref}} e - B \Lambda \Delta \Theta^T \Phi(x) + \xi(t) \quad (11.9)$$

A radially unbounded quadratic Lyapunov function candidate is selected in the familiar form

$$V(e, \Delta \Theta) = e^T P e + \text{trace}(\Delta \Theta^T \Gamma_{\Theta}^{-1} \Delta \Theta \Lambda) \quad (11.10)$$

where $\Gamma_{\Theta} = \Gamma_{\Theta}^T > 0$ denotes constant rates of adaptation, and $P = P^T > 0$ is the unique symmetric positive definite solution of the algebraic Lyapunov equation,

$$P A_{\text{ref}} + A_{\text{ref}}^T P = -Q \quad (11.11)$$

with $Q = Q^T > 0$. Time-differentiating V , along the trajectories of (11.9), gives

$$\begin{aligned}\dot{V}(e, \Delta\Theta) &= -e^T Q e - 2e^T P B \Lambda \Delta\Theta^T \Phi(x) + 2e^T P \xi(t) \\ &\quad + 2 \operatorname{trace}(\Delta\Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \Lambda)\end{aligned}\quad (11.12)$$

Applying the vector trace identity,

$$a^T b = \operatorname{trace}(b a^T) \quad (11.13)$$

further yields

$$\dot{V}(e, \Delta\Theta) = -e^T Q e + 2 \operatorname{trace}(\Delta\Theta^T \left\{ \Gamma_\Theta^{-1} \dot{\hat{\Theta}} - \Phi e^T P B \right\} \Lambda) + 2e^T P \xi(t) \quad (11.14)$$

Suppose that we use the same adaptive laws as in the previous sections, that is

$$\dot{\hat{\Theta}} = \Gamma_\Theta \Phi(x) e^T P B \quad (11.15)$$

Then,

$$\dot{V}(e, \Delta\Theta) = -e^T Q e + 2e^T P \xi(t) \leq -\lambda_{\min}(Q) \|e\|^2 + 2 \|e\| \lambda_{\max}(P) \xi_{\max} \quad (11.16)$$

and consequently, $\dot{V} < 0$ outside of the set,

$$E_0 = \left\{ (e, \Delta\Theta) : \|e\| \leq 2 \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)} \xi_{\max} = e_0 \right\} \quad (11.17)$$

According to [6] (Theorem 4.18, p. 172), trajectories $e(t)$ of the error dynamics (11.9) enter a compact set $(\Omega_0 \supset E_0) \subset R^n$ in finite time, and will remain there for all future times. However, Ω_0 is not compact in the $(e, \Delta\Theta)$ -space. In fact, Ω_0 is unbounded, since the parameter estimation errors $\Delta\Theta$ are not restricted at all. Therefore, inside Ω_0 , \dot{V} can become positive and, as a consequence, the parameter errors $\Delta\Theta$ can grow unbounded, even though the tracking error norm remains finite at all times. This phenomenon is known as the “parameter drift”. It is caused by the disturbance term $\xi(t)$. This argument shows that the MRAC laws (11.15) are not robust to bounded disturbances, no matter how small the latter are.

11.2 MRAC Design Modifications for Robustness

In this section, we introduce three design modifications to enforce robustness of MRAC laws in the presence of unmatched disturbances, such as bounded process noise. These modifications are: (a) the dead-zone; (b) the ϵ -modification, and (c) the σ -modification.

11.2.1 The Dead-Zone Modification

In order to enforce robustness, we consider adaptive laws with the dead-zone modification

$$\dot{\hat{\Theta}} = \begin{cases} \Gamma_\Theta \Phi(x) e^T P B, & \text{if } \|e\| > e_0 \\ 0_{N \times m}, & \text{if } \|e\| \leq e_0 \end{cases} \quad (11.18)$$

Proposed by B. B. Peterson and K. S. Narendra in [1], the dead-zone modification stops the adaptation process when the norm of the tracking error becomes smaller than the prescribed value e_0 . This assures uniform ultimate boundedness (UUB) of $\Delta\Theta$ (in addition to UUB of e). We are going to formally prove this claim.

Suppose that $\|e\| > e_0$ then the adaptive law is defined by (11.15), and it results in the upper bound (11.16). Consequently, $e(t)$ enters Ω_0 in finite time T and will reside within the set for all $t \geq T$. From that time forward, the adaptive parameter dynamics are frozen, that is $\dot{\hat{\Theta}}(t + T) = 0_{N \times m}$. This proves UUB of the error dynamics (11.9), and it also proves boundedness (but not necessarily UUB) of the adaptive parameter estimation errors, $\|\Delta\Theta(t)\| < \infty$, uniformly in time.

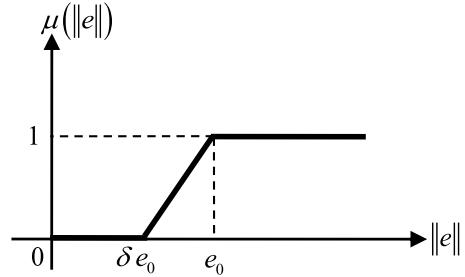
The tracking error bound e_0 in (11.17) depends on the eigenvalue ratio $\frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}$. It is not too difficult to show (see [8], pp. 92–93) that the minimum of this ratio is achieved for $Q = I_{n \times n}$. Thus, the computable tracking error upper bound is proportional to $2\lambda_{\max}(P)\xi_{\max}$, where $P = P^T > 0$ is the unique solution of the Lyapunov equation $P A_{\text{ref}} + A_{\text{ref}}^T P = -I_{n \times n}$. However, even when the disturbance vanishes, with the dead-zone modification being active, asymptotic stability of the tracking error cannot be recovered.

The dead-zone modification is not Lipschitz, and as such it may cause chattering (high frequency oscillations) and other undesirable effects, especially when the tracking error is at or near the dead-zone boundary. A smooth version of the dead-zone modification was introduced by J.-J. E. Slotine and J. A. Coetsee in [2]. Motivated by this idea, we choose a constant $0 < \delta < 1$, and consider a Lipchitz-continuous modulation function in the form,

$$\mu(\|e\|) = \max\left(0, \min\left(1, \frac{\|e\| - \delta e_0}{(1 - \delta)e_0}\right)\right) \quad (11.19)$$

A sketch of this function is shown in Fig. 11.1.

Fig. 11.1 The dead-zone modulation function



Adaptive laws with the continuous dead-zone modification are defined as,

$$\dot{\hat{\Theta}} = \Gamma_\Theta \Phi(x) \mu(\|e\|) e^T P B \quad (11.20)$$

With these laws of adaptation, one can use Lyapunov-based arguments to prove bounded tracking and UUB of all signals [1].

Example 11.1 MRAC with the Dead-zone Modification for Aircraft Roll Dynamics We shall illustrate an MRAC design with the dead-zone modification using the aircraft roll dynamics (a scalar system from Example 10.1)

$$\dot{p} = L_p p + L_{\delta_a} \delta_a + \xi(t)$$

subjected to a bounded environmental disturbance $\xi(t)$, which in this case may represent the rotational component of a gust. Also in the model, p is the aircraft roll rate (rad/s), δ_a is the differential aileron deflection, (rad), L_p is the aerodynamic roll damping (s^{-1}), and L_{δ_a} is the aileron effectiveness (s^{-1}).

For a mid-size airplane cruising at high altitude, typical values of the aerodynamic parameters are: $L_p = -0.8$, $L_{\delta_a} = 1.6$. These are the two constant unknowns in the system. The goal is to design an MRAC state feedback/feedforward controller with the dead-zone modification and to enable bounded tracking of the reference model,

$$\dot{p}_{\text{ref}} = A_{\text{ref}} p_{\text{ref}} + B_{\text{ref}} p_{\text{cmd}}$$

which is subsequently driven by a bounded roll-rate command $p_{\text{cmd}}(t)$.

The roll dynamics can be easily rewritten in the form of (11.1),

$$\begin{aligned} \dot{p} = & A_{\text{ref}} p + \underbrace{B}_{1} \underbrace{\frac{L_{\delta_a}}{\Lambda > 0}}_{\text{u}} \left(\underbrace{\delta_a}_{u} + \underbrace{\left(\frac{L_p - A_{\text{ref}}}{L_{\delta_a}} \right) p - \frac{B_{\text{ref}}}{L_{\delta_a}} p_{\text{cmd}}}_{\Theta^T \Phi(p, p_{\text{cmd}})} \right) \\ & + B_{\text{ref}} p_{\text{cmd}} + \xi(t) \end{aligned}$$

where $\Theta^T = \frac{1}{L_{\delta_a}}(L_p - A_{\text{ref}} - B_{\text{ref}})$ is the vector of unknown constant parameters, and $\Phi^T = (p \ p_{\text{cmd}})$ is the known regressor vector, which depends on the system state p and the external command p_{cmd} . This model differs from (11.1) where the regressor is a state-dependent function. Even so, it is not difficult to repeat Lyapunov-based stability arguments and show that the same adaptive laws (11.18) apply, with the state and command dependent regressor vector $\Phi = \Phi(x, y_{\text{cmd}})$.

Therefore, according to (11.5) and (11.18), the MRAC roll-rate tracking controller computes differential aileron deflections in the form,

$$\delta_a = -\hat{k}_p p - \hat{k}_{p_{\text{cmd}}} p_{\text{cmd}}$$

where $\hat{\Theta}^T = (\hat{k}_p \ \hat{k}_{p_{\text{cmd}}})$ are the adaptive gains, whose dynamics are specified by the adaptive laws shown below, with the discontinuous dead-zone modification.

$$\begin{aligned}\dot{\hat{k}}_p &= \begin{cases} \gamma_p p (p - p_{\text{ref}}), & \text{if } |p - p_{\text{ref}}| > e_0 \\ 0, & \text{if } |p - p_{\text{ref}}| \leq e_0 \end{cases} \\ \dot{\hat{k}}_{p_{\text{cmd}}} &= \begin{cases} \gamma_{p_{\text{cmd}}} p_{\text{cmd}} (p - p_{\text{ref}}), & \text{if } |p - p_{\text{ref}}| > e_0 \\ 0, & \text{if } |p - p_{\text{ref}}| \leq e_0 \end{cases}\end{aligned}$$

For simulation, we have selected the following parameters

$$A_{\text{ref}} = -B_{\text{ref}} = -2, \quad \gamma_p = \gamma_{p_{\text{cmd}}} = 100$$

The rotational gust component $\xi(t)$ was modeled as a random process noise, uniformly distributed on the interval $\frac{\pi}{180} [-10 \ 10]$.

For a step-input roll-rate command of 10 deg/s and without the dead-zone modification, i.e., setting $e_0 = 0$, the system closed-loop tracking performance and the MRAC control effort (the aileron deflection) are adequate (Fig. 11.2).

As expected, the norm of the system tracking error is not zero and it is primarily driven by the process noise $\xi(t)$. However, the adaptive parameters exhibit the undesirable drift phenomenon (Fig. 11.3).

Rerunning the same case but with the dead-zone tolerance $e_0 = 0.0524$, we maintain good tracking performance (Fig. 11.4).

At the same time, the dead-zone modification prevents the adaptive parameters from drifting (Fig. 11.5).

As seen from Fig. 11.5, the adaptive parameters tend to their ideal unknown values. This can be attributed to an apparent level of persistency of excitation in the system dynamics, which is induced by the process noise. ■

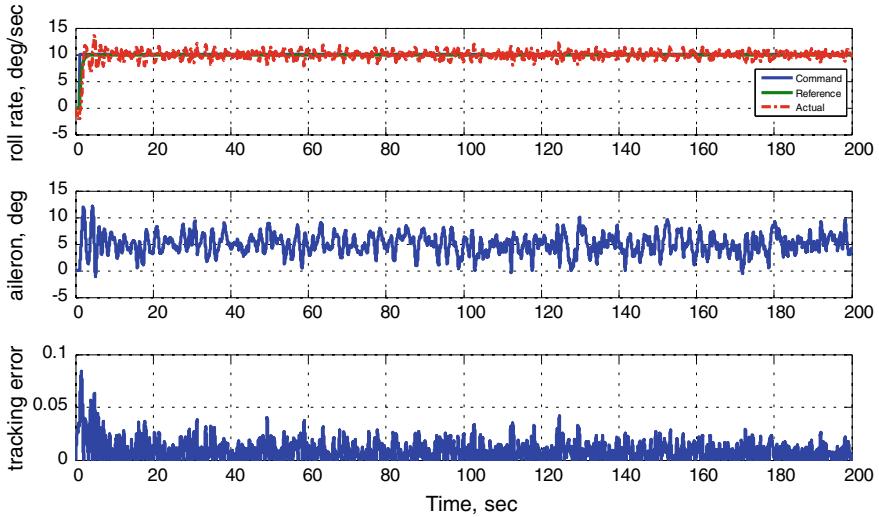


Fig. 11.2 Step-input roll-rate tracking without the dead-zone modification in Example 11.1

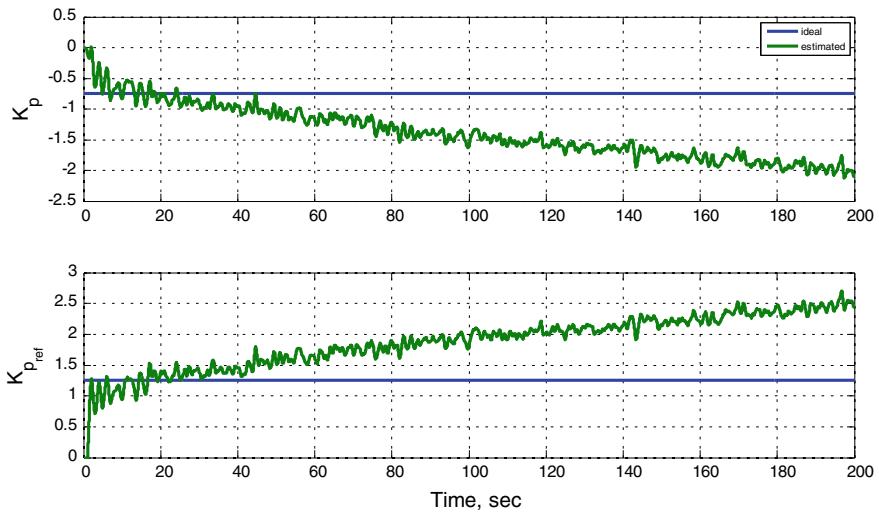


Fig. 11.3 Parameter drift without the dead-zone modification in Example 11.1

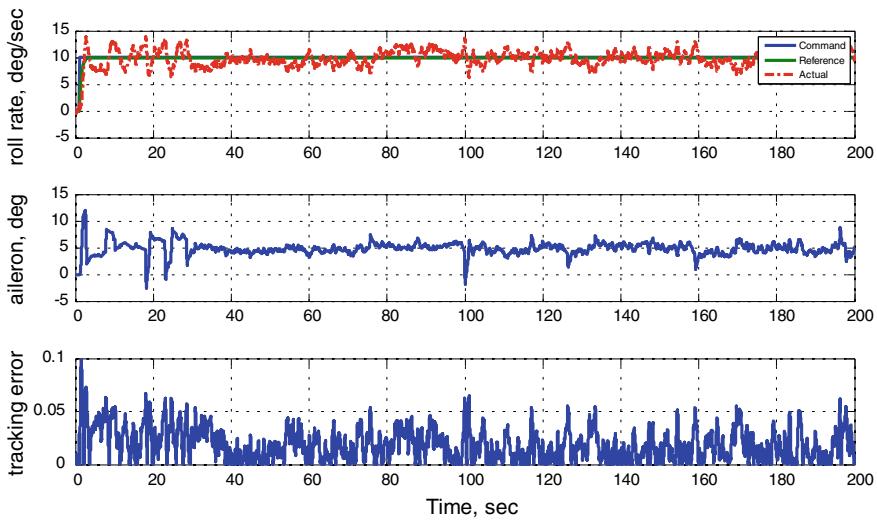


Fig. 11.4 Step-input roll-rate tracking with the dead-zone modification in Example 11.1

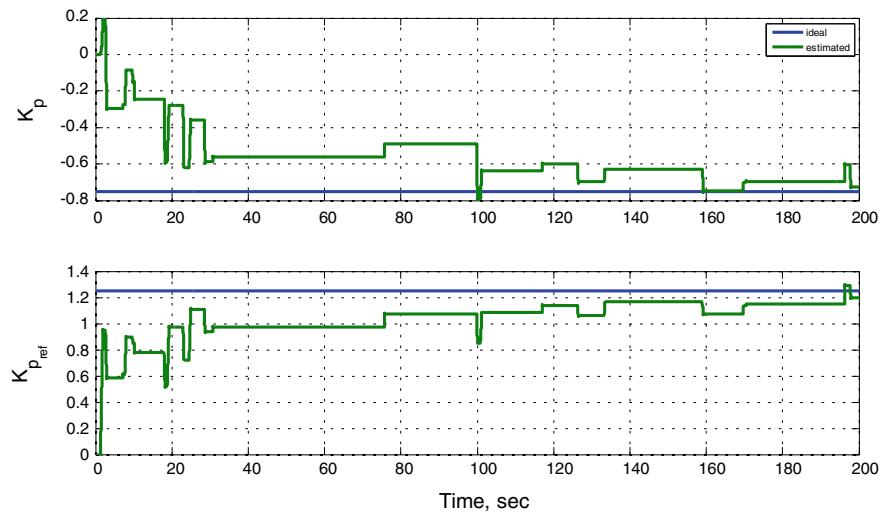


Fig. 11.5 Adaptive parameters with the dead-zone modification in Example 11.1

11.2.2 The σ -Modification

Earlier, we have assumed prior knowledge of an upper bound ξ_{\max} for the system disturbance $\xi(t)$. The σ -modification scheme, developed by Ioannou and Kokotovic [3], does not require any prior information on the system disturbance upper bounds. The adaptive law with the σ -modification is

$$\dot{\hat{\Theta}} = \Gamma_\Theta \left(\Phi(x) e^T P B - \sigma \hat{\Theta} \right) \quad (11.21)$$

where σ is a strictly positive constant. In essence, this modification adds damping to the ideal adaptive law (11.15).

In order to prove UUB of all signals, we again consider the Lyapunov function candidate (11.10) and compute its time derivative along the trajectories of the tracking error dynamics (11.9).

$$\begin{aligned} \dot{V}(e, \Delta\Theta) &= -e^T Q e + 2 \operatorname{trace} \left(\Delta\Theta^T \left\{ \Gamma_\Theta^{-1} \dot{\hat{\Theta}} - \Phi e^T P B \right\} \Lambda \right) + 2 e^T P \xi(t) \\ &= -e^T Q e - 2\sigma \operatorname{trace} \left(\Delta\Theta^T \underbrace{\hat{\Theta}}_{\Theta + \Delta\Theta} \Lambda \right) + 2 e^T P \xi(t) \\ &= -e^T Q e - 2\sigma \operatorname{trace} \left(\Delta\Theta^T \Delta\Theta \Lambda \right) - 2\sigma \operatorname{trace} \left(\Delta\Theta^T \Theta \Lambda \right) \\ &\quad + 2 e^T P \xi(t) \end{aligned} \quad (11.22)$$

By definition

$$\operatorname{trace} \left(\Delta\Theta^T \Delta\Theta \Lambda \right) = \sum_{i=1}^N \sum_{j=1}^m \Delta\Theta_{ij}^2 \Lambda_{ii} \geq \|\Delta\Theta\|_F^2 \Lambda_{\min} \quad (11.23)$$

where $\|\Delta\Theta\|_F^2 = \sum_{i=1}^n \sum_{j=1}^m \Delta\Theta_{ij}^2$ is the Frobenius norm of $\Delta\Theta$, and Λ_{\min} is the minimum diagonal element of Λ . Moreover, using the Schwarz inequality, gives

$$\left| \operatorname{trace} \left(\Delta\Theta^T \Theta \Lambda \right) \right| \leq \left\| \Delta\Theta^T \Theta \right\|_F \|\Lambda\|_F \leq \|\Delta\Theta\|_F \|\Theta\|_F \|\Lambda\|_F \quad (11.24)$$

Substituting (11.23) and (11.24) into (11.22), results in

$$\begin{aligned} \dot{V}(e, \Delta\Theta) &\leq -\lambda_{\min}(Q) \|e\|^2 + 2 \|e\| \lambda_{\max}(P) \xi_{\max} - 2\sigma \|\Delta\Theta\|_F^2 \Lambda_{\min} \\ &\quad + 2\sigma \|\Delta\Theta\|_F \|\Theta\|_F \|\Lambda\|_F \end{aligned} \quad (11.25)$$

Using completion of squares, the sum of the first and the second terms in (11.25) can be transformed into

$$\begin{aligned} -\lambda_{\min}(Q) \|e\|^2 + 2\|e\|\lambda_{\max}(P)\xi_{\max} &= -\lambda_{\min}(Q) \left(\|e\| - \frac{\lambda_{\max}(P)\xi_{\max}}{\lambda_{\min}(Q)} \right)^2 \\ &+ \frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} \end{aligned}$$

Similarly, the sum of the third and the fourth terms in (11.25) can be written as

$$\begin{aligned} -2\sigma\|\Delta\Theta\|_F^2\Lambda_{\min} + 2\sigma\|\Delta\Theta\|_F\|\Theta\|_F\|\Lambda\|_F \\ = -2\sigma\Lambda_{\min}\left(\|\Delta\Theta\|_F - \frac{1}{2}\|\Theta\|_F\frac{\|\Lambda\|_F}{\Lambda_{\min}}\right)^2 + \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}} \end{aligned}$$

Substituting these two expressions back into (11.25), gives

$$\begin{aligned} \dot{V}(e, \Delta\Theta) &\leq -\lambda_{\min}(Q) \left(\|e\| - \frac{\lambda_{\max}(P)\xi_{\max}}{\lambda_{\min}(Q)} \right)^2 + \frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} \\ &- 2\sigma\Lambda_{\min}\left(\|\Delta\Theta\|_F - \frac{1}{2}\|\Theta\|_F\frac{\|\Lambda\|_F}{\Lambda_{\min}}\right)^2 + \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}} \end{aligned} \quad (11.26)$$

Hence, $\dot{V}(e, \Delta\Theta) < 0$ if at least one of the following two relations take place

$$\begin{aligned} \lambda_{\min}(Q) \left(\|e\| - \frac{\lambda_{\max}(P)\xi_{\max}}{\lambda_{\min}(Q)} \right)^2 - \frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} - \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}} &> 0 \\ \text{OR} \\ 2\sigma\Lambda_{\min}\left(\|\Delta\Theta\|_F - \frac{1}{2}\|\Theta\|_F\frac{\|\Lambda\|_F}{\Lambda_{\min}}\right)^2 - \frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} - \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}} &> 0 \end{aligned} \quad (11.27)$$

or equivalently, when

$$\begin{aligned} \|e\| &> \sqrt{\frac{1}{\lambda_{\min}(Q)} \left(\frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} + \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}} \right)} + \frac{\lambda_{\max}(P)\xi_{\max}}{\lambda_{\min}(Q)} = c_1 \\ \text{OR} \\ \|\Delta\Theta\|_F &> \sqrt{\frac{1}{2\sigma\Lambda_{\min}} \left(\frac{\lambda_{\max}^2(P)\xi_{\max}^2}{\lambda_{\min}(Q)} + \sigma\frac{\|\Theta\|_F^2\|\Lambda\|_F^2}{2\Lambda_{\min}} \right)} + \frac{1}{2}\|\Theta\|_F\frac{\|\Lambda\|_F}{\Lambda_{\min}} = c_2 \end{aligned} \quad (11.28)$$

In other words, $\dot{V}(e, \Delta\Theta) < 0$ outside of the compact (closed and bounded) set $\Omega \subset (R^n \times R^{N \times m})$ defined below.

$$\Omega = \{(e, \Delta\Theta) : (\|e\| \leq c_1) \wedge (\|\Delta\Theta\|_F \leq c_2)\} \quad (11.29)$$

This argument immediately proves UUB of all signals in the closed-loop dynamics. In particular, (11.29) proves UUB tracking of the external command $y_{cmd}(t)$ by the system output $y(t)$. Note that in this case, command tracking is achieved in the presence of parametric uncertainties (Λ, Θ) and nonparametric bounded time-varying disturbances $\xi(t)$. Next, we illustrate the σ -modification features and benefits for the scalar roll dynamics from Example 11.1.

Example 11.2 MRAC with the σ -modification for Aircraft Roll Dynamics Continuing with the roll dynamics model from Example 11.1, we utilize (11.21) and write the adaptive laws with the σ -modification

$$\begin{aligned}\dot{\hat{k}}_p &= \gamma_p \left(p(p - p_{ref}) - \sigma \hat{k}_p \right) \\ \dot{\hat{k}}_{p_{cmd}} &= \gamma_{p_{cmd}} \left(p_{cmd}(p - p_{ref}) - \sigma \hat{k}_{p_{cmd}} \right)\end{aligned}$$

We then select $\sigma = 0.1$, and simulate the same roll-rate step-input response as in Example 11.1, but with the σ -modification turned on. This design also gives adequate roll-rate command tracking performance (Fig. 11.6).

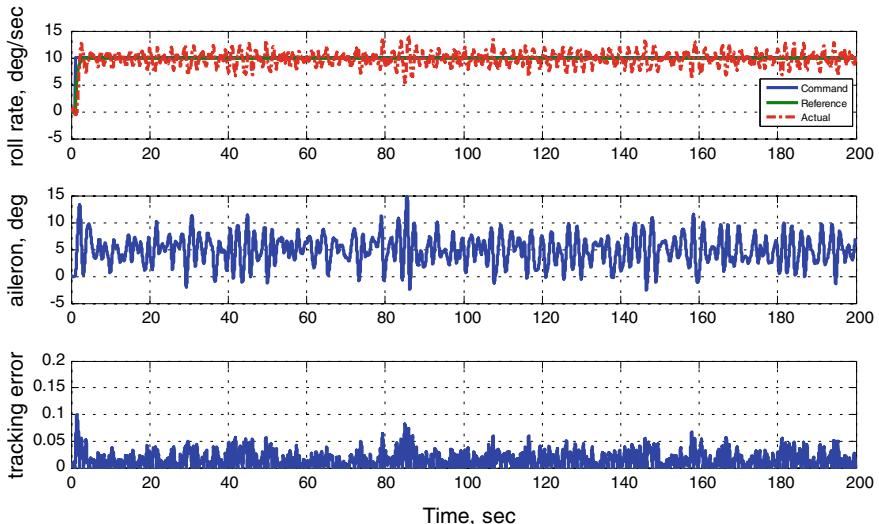


Fig. 11.6 Step-input roll-rate tracking with the σ -modification in Example 11.2

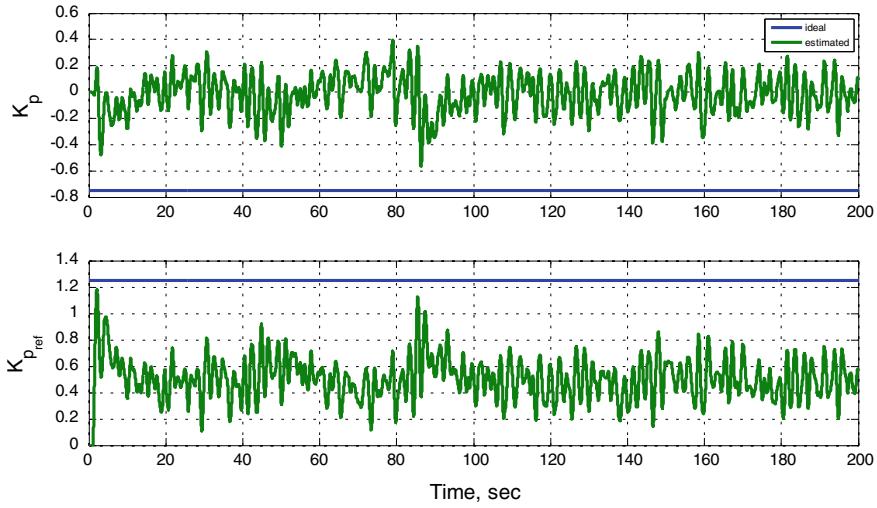


Fig. 11.7 Adaptive gains with the σ -modification in Example 11.2

The data are comparable to the simulation results achieved using the dead-zone modification in Example 11.1 (see Fig. 11.4). In addition, the corresponding adaptive gains are bounded (Fig. 11.7), and potential drift due to noise tendencies are completely prevented.

We make a note that in this case, the adaptive gains are oscillatory which may not be desirable. The oscillations are driven by the process noise and the data reveal noise sensitivity of the adaptive law dynamics. ■

11.2.3 The e -Modification

There are performance related drawbacks to applying the σ -modification. When the tracking error becomes small, the adaptive law dynamics (11.21) can be approximately written as $\dot{\Theta} \approx -\Gamma_\Theta \sigma \hat{\Theta}$. Hence for small tracking errors, the adaptive parameters tend to return to the origin, that is they “unlearn” the gain values that caused the tracking error to become small in the first place. Furthermore, even if the disturbance $\xi(t)$ is removed from the system dynamics (11.1), and if the reference command y_{cmd} is persistently exciting [4], the parameter errors $\Delta\Theta(t)$ do not converge to the origin.

In order to overcome these undesirable effects, K. S. Narendra and A. M. Annaswamy introduced the e -modification [4]. Originally called the e_1 -modification, the method’s main idea is to replace the constant damping gain σ in (11.21) with a term proportional to a linear combination of the system tracking errors, such as $\|\epsilon^T P B\|$. The rational for using an error-dependent damping is

that it tends to 0, as the regulated output error diminishes. The adaptive laws with e -modification are

$$\dot{\hat{\Theta}} = \Gamma_{\Theta} \left(\Phi(x) e^T P B - \sigma \|e^T P B\| \hat{\Theta} \right) \quad (11.30)$$

As seen from (11.30), the e -modification adds a tracking error-dependent damping $\sigma \|e^T P B\|$ to the adaptive dynamics.

Using these laws, one can compute the time derivative of the Lyapunov function candidate (11.10), along the trajectories of the tracking error dynamics (11.9), and then repeat similar derivations that lead to (11.26). The only difference here is that instead of a constant parameter σ , we have an error-dependent damping term $\sigma \|e^T P B\|$. This fact allows to arrive at a compact set [4], outside of which $\dot{V}(e, \Delta\Theta) < 0$. Once again, we can claim UUB of all trajectories. This completes the stability analysis for the e -modification with a guaranteed UUB-type output tracking performance.

Example 11.3 MRAC with the e -modification for Aircraft Roll Dynamics We now apply the e -modification design to the roll dynamics that was introduced in Example 11.1 and subsequently reused in Example 11.2. According to (11.30), the adaptive laws with the e -modification are

$$\begin{aligned}\dot{\hat{k}}_p &= \gamma_p \left(p(p - p_{\text{ref}}) - \sigma |p - p_{\text{ref}}| \hat{k}_p \right) \\ \dot{\hat{k}}_{p_{\text{cmd}}} &= \gamma_{p_{\text{cmd}}} \left(p_{\text{cmd}}(p - p_{\text{ref}}) - \sigma |p - p_{\text{ref}}| \hat{k}_{p_{\text{cmd}}} \right)\end{aligned}$$

So now, the damping term $\sigma |p - p_{\text{ref}}|$ depends on the tracking error $e = p - p_{\text{ref}}$, and it will diminish if e becomes small.

Figure 11.8 shows the system closed-loop tracking performance, with e -modification gain $\sigma = 1$.

Once again, we obtained adequate step-input command tracking in the presence of noise. The results are comparable to those shown in Figs. 11.4 and 11.6. The corresponding adaptive parameters are shown in Fig. 11.9.

It is interesting to note that in this simulation scenario, the e -modification kept the adaptive parameters uniformly bounded and in addition, it also forced them to approach their ideal values. However, such a tendency would not be possible without persistency of excitation induced by the process noise into the system dynamics. ■

It is easy to see that for large tracking errors, the dead-zone, the σ -modification, and the e -modification slow down (i.e., dampen) the adaptation. Often, such an effect is considered detrimental since it may contradict the control goal of reducing the tracking error as fast as possible.

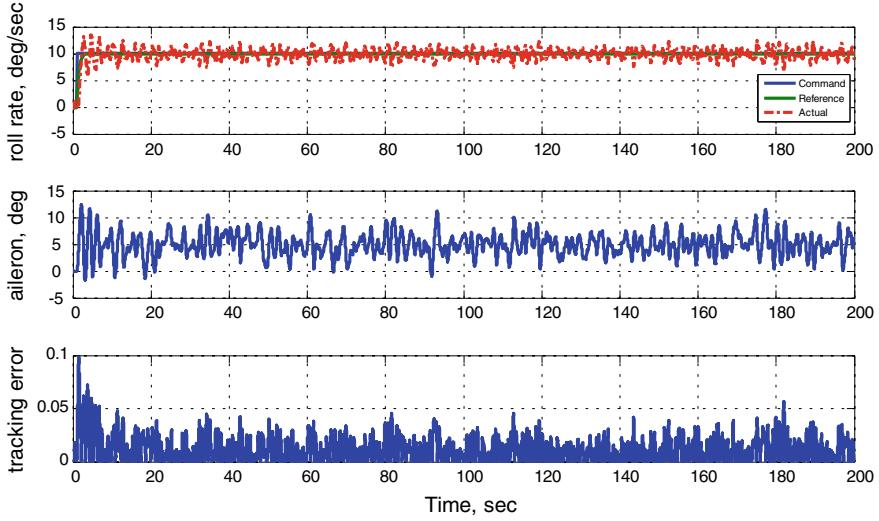


Fig. 11.8 Step-input roll-rate tracking with the e -modification in Example 11.3

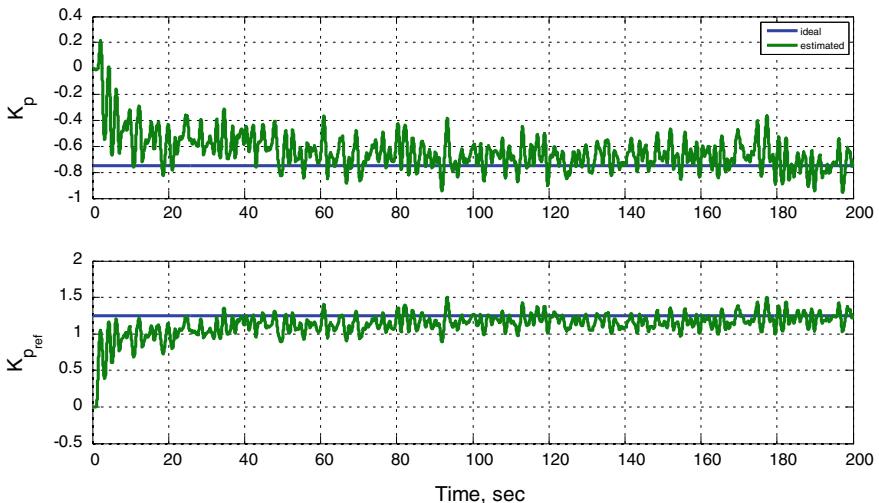


Fig. 11.9 Adaptive parameters with the e -modification in Example 11.3

11.3 The Projection Operator

In this section, we shall introduce a Lipschitz-continuous version of the Projection Operator [7, 11]. This concept is essential for enabling the adaptive laws (11.15) to achieve robustness with respect to parametric and nonparametric uncertainties

that might exist in the system dynamics. We show that the Projection Operator tolerates fast adaptation, enforces uniform boundedness of the adaptive parameters, and maintains closed-loop stability of the corresponding error dynamics and of the original system. The selected version of the Projection Operator can be thought of as a direct extension of a projection-like modification that was originally proposed by G. Kreisselmeier and K. S. Narendra in [9].

To reiterate, our overall design goal is to continuously modify adaptive laws (11.15) in order to maintain negative semidefiniteness of the trace term in the Lyapunov function time derivative in (11.14),

$$\text{trace}\left(\Delta\Theta^T \left\{\Gamma_\Theta^{-1} \dot{\hat{\Theta}} - \Phi e^T P B\right\} \Lambda\right) \leq 0 \quad (11.31)$$

and at the same time, to keep the adaptive parameters $\hat{\Theta}(t)$ uniformly bounded in time. These two design objectives will be achieved through the introduction of the Projection Operator into the adaptive law dynamics.

We begin with basic definitions of convex sets and functions. These concepts will facilitate proper introduction of the Projection Operator.

Definition 11.1 A subset $\Omega \subset R^n$ is convex if

$$[\forall x, y \in \Omega \subset R^n] \Rightarrow [\lambda x + (1 - \lambda) y = z \in \Omega], \quad \forall 0 \leq \lambda \leq 1 \quad (11.32)$$

Relation (11.32) states that if two points belong to a convex subset Ω then all the points on the connecting line also belong to Ω .

Definition 11.2 A function $f : R^n \rightarrow R$ is convex on R^n , if

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y), \quad \forall 0 \leq \lambda \leq 1, \quad \forall x, y \in R^n \quad (11.33)$$

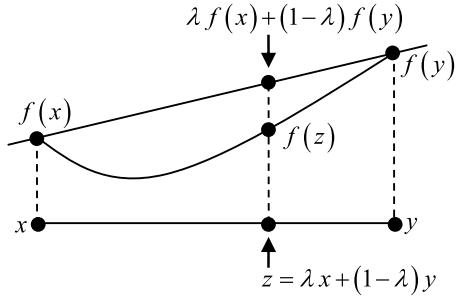
Inequality (11.33) is illustrated in Fig. 11.10. It shows that the graph of a convex function must be located below the straight line, which connects the two corresponding function values. ■

Lemma 11.1 Let $f(x) : R^n \rightarrow R$ be convex. Then for any constant $\delta > 0$ the subset $\Omega_\delta = \{\theta \in R^n | f(\theta) \leq \delta\}$ is convex. ■

Proof of Lemma 11.1 Let $\theta_1, \theta_2 \in \Omega_\delta$. Then $f(\theta_1) \leq \delta$ and $f(\theta_2) \leq \delta$. Since $f(x)$ is convex then for any $0 \leq \lambda \leq 1$,

$$f\left(\underbrace{\lambda \theta_1 + (1 - \lambda) \theta_2}_{\theta}\right) \leq \lambda \underbrace{f(\theta_1)}_{\leq \delta} + (1 - \lambda) \underbrace{f(\theta_2)}_{\leq \delta} \leq \lambda \delta + (1 - \lambda) \delta = \delta$$

Fig. 11.10 Graph of a convex function



Therefore, $f(\theta) \leq \delta$ and, consequently, $\theta \in \Omega_\delta$ which completes the proof. ■

Lemma 11.2 Let $f(x) : R^n \rightarrow R$ be a differentiable convex function. Choose a constant $\delta > 0$ and consider the subset

$$\Omega_\delta = \{ \theta \in R^n \mid f(\theta) \leq \delta \} \subset R^n$$

Let $\theta^* \in \Omega_\delta$ and assume that $f(\theta^*) < \delta$, that is θ^* is an interior point (i.e., not on the boundary) of Ω_δ . Also, let $\theta \in \Omega_\delta$ and assume that $f(\theta) = \delta$, that is θ lays on the boundary of Ω_δ . Then the following inequality holds

$$(\theta^* - \theta)^T \nabla f(\theta) \leq 0 \quad (11.34)$$

where $\nabla f(\theta) = \left(\frac{\partial f(\theta)}{\partial \theta_1} \dots \frac{\partial f(\theta)}{\partial \theta_n} \right)^T \in R^n$ is the gradient vector of f evaluated at θ . ■

Relation (11.34) is illustrated in Fig. 11.11. It shows that the gradient vector of a function, evaluated at the boundary of a convex level set generated by this function, always points away from the set.

Proof of Lemma 11.2 Since $f(x)$ is convex then,

$$f(\lambda \theta^* + (1 - \lambda) \theta) \leq \lambda f(\theta^*) + (1 - \lambda) f(\theta)$$

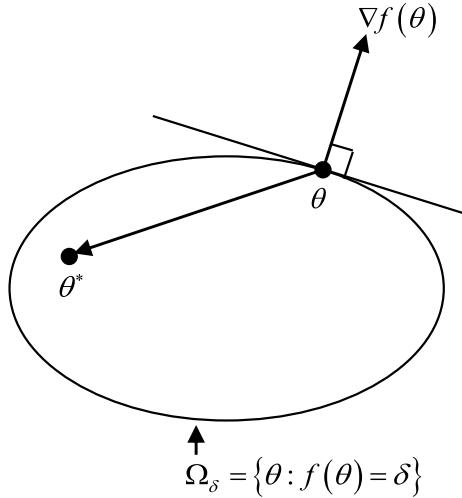
or equivalently

$$f(\theta + \lambda (\theta^* - \theta)) \leq f(\theta) + \lambda (f(\theta^*) - f(\theta))$$

Consequently, for any nonzero $0 < \lambda \leq 10 < \lambda \leq 1$

$$\frac{f(\theta + \lambda (\theta^* - \theta)) - f(\theta)}{\lambda} \leq \underbrace{f(\theta^*)}_{<\delta} - \underbrace{f(\theta)}_{\delta} < \delta - \delta = 0$$

Fig. 11.11 Gradient vector on the boundary of a convex set



Taking the limit as $\lambda \rightarrow 0$, yields relation (11.34) and completes the proof. ■

Suppose that a parameter vector θ , belongs to a convex set Ω_0 .

$$\Omega_0 = \{\theta \in R^n \mid f(\theta) \leq 0\} \quad (11.35)$$

Let us introduce another convex set

$$\Omega_1 = \{\theta \in R^n \mid f(\theta) \leq 1\} \quad (11.36)$$

Then, it becomes obvious that $\Omega_0 \subseteq \Omega_1$.

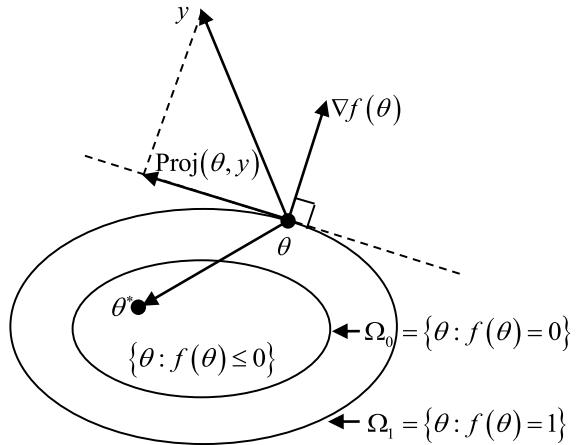
We may now define the continuous Projection Operator,

$$\text{Proj}(\theta, y) = \begin{cases} y - \frac{\Gamma \nabla f(\theta) (\nabla f(\theta))^T}{\|\nabla f(\theta)\|_{\Gamma}^2} y f(\theta), & \text{if } [f(\theta) > 0 \wedge y^T \nabla f(\theta) > 0] \\ y, & \text{if not} \end{cases} \quad (11.37)$$

where $\Gamma \in R^{n \times n}$ is any constant symmetric positive definite matrix, and $\|\nabla f\|_{\Gamma}^2 = (\nabla f)^T \Gamma \nabla f$ is the weighed Euclidean squared norm of ∇f .

Let us graphically illustrate the Projection Operator in (11.37). To simplify the discussion, we set Γ to be the identity matrix. As seen from the definition (11.37), $\text{Proj}(\theta, y)$ does not alter the vector y if θ belongs to the convex set Ω_0 from (11.35). In the annulus set $\{0 \leq f(\theta) \leq 1\}$, the Projection Operator subtracts a vector normal to the boundary $\{f(\theta) = \lambda\}$ from y . As a result, we get a smooth transformation from the original vector field y for $\lambda = 0$ to the tangent

Fig. 11.12 The projection operator



to the boundary vector for $\lambda = 1$. The Projection Operator concept is shown in Fig. 11.12.

For an arbitrary positive definite symmetric matrix Γ , a similar sketch can be drawn.

Next, we derive an important convex property of the Projection Operator.

Lemma 11.3 *For any symmetric positive definite matrix $\Gamma \in R^{n \times n}$,*

$$(\theta - \theta^*)^T (\Gamma^{-1} \text{Proj}(\theta, \Gamma y) - y) \leq 0 \quad (11.38)$$

Proof of Lemma 11.3 Using (11.34) and (11.37), gives

$$\begin{aligned} & (\theta - \theta^*)^T (\Gamma^{-1} \text{Proj}(\theta, \Gamma y) - y) \\ &= \left\{ \begin{array}{ll} \overbrace{-\frac{(\theta - \theta^*)^T \nabla f}{\|\nabla f\|_\Gamma^2} \overbrace{[(\nabla f)^T \Gamma y]}^{>0}}^{>0} \overbrace{\frac{f}{f}}^{>0}, & \text{if } [f > 0 \wedge y^T \Gamma \nabla f > 0] \\ 0, & \text{if not} \end{array} \right\} < 0 \end{aligned} \quad (11.39)$$

and the proof is complete. ■

We now state and prove yet another result of conceptual importance to the forthcoming development of adaptive controllers.

Lemma 11.4 *Let $f(\theta)$ be a convex continuously differentiable map from R^n to R . Using (11.37) consider the n -dimensional dynamics,*

$$\dot{\theta} = \text{Proj}(\theta, y) \quad (11.40)$$

where $\theta \in R^n$ is the system state, and $y \in R^n$ is a time-varying piecewise continuous vector. Then starting from any initial condition $\theta(0) = \theta_0$ within the set,

$$\Omega_0 = \{\theta \in R^n \mid f(\theta) \leq 0\} \quad (11.41)$$

the system trajectory $\theta(t)$ will remain in the set,

$$\Omega_1 = \{\theta \in R^n \mid f(\theta) \leq 1\} \quad (11.42)$$

for all $t \geq 0$. ■

Proof of Lemma 11.4 Existence and uniqueness of the system (11.40) solutions are provided by the fact that the Projection Operator is locally Lipschitz in θ , while the system external input $y(t)$ is piecewise continuous in time.

To prove the lemma, we need to show that the following relation holds

$$\underbrace{[f(\theta_0) \leq 0]}_{\theta_0 \in \Omega_0} \Rightarrow \underbrace{[f(\theta(t)) \leq 1]}_{\theta(t) \in \Omega_1}, \quad \forall t \geq 0 \quad (11.43)$$

Toward that end, we evaluate the time derivative of $f(\theta(t))$ along the trajectories of the system dynamics (11.40). Based on the definition (11.37), we obtain

$$\begin{aligned} \dot{f}(\theta) &= (\nabla f(\theta))^T \text{Proj}(\theta, y) \\ &= \begin{cases} (\nabla f(\theta))^T y (1 - f(\theta)), & \text{if } [f(\theta) > 0 \wedge y^T \nabla f(\theta) > 0] \\ (\nabla f(\theta))^T y, & \text{if not} \end{cases} \end{aligned} \quad (11.44)$$

Consequently,

$$\begin{aligned} \dot{f}(\theta) &> 0, & \text{if } [0 < f(\theta) < 1 \wedge y^T \nabla f(\theta) > 0] \\ \dot{f}(\theta) &= 0, & \text{if } [f(\theta) = 1 \wedge y^T \nabla f(\theta) > 0] \\ \dot{f}(\theta) &\leq 0, & \text{if } [f(\theta) \leq 0 \vee y^T \nabla f(\theta) \leq 0] \end{aligned} \quad (11.45)$$

The first and the second relations in (11.45) imply that if $f(\theta(0)) > 0$ then $f(\theta(t))$ monotonically increases in time for all $t \geq 0$, but it will never exceed 1. Also, the third condition in (11.45) states that if $f(\theta(0)) \leq 0$ then $f(\theta(t))$ is monotonically decreasing for all $t \geq 0$. Therefore, irrespective of initial values (as long as they are negative), $f(\theta(t)) \leq 1$ for all $t \geq 0$, which completes the proof of the lemma. ■

The next example shows how to use the Projection Operator to construct actuator models with position and rate constraints.

Example 11.4 Actuator Dynamics with Position and Rate Constraints In control engineering applications, one often needs to account for mechanical, hydraulic, or electrical control actuation devices. Their dynamics are frequently modeled by a scalar system,

$$\tau \dot{u} = u_{\text{cmd}} - u$$

where u_{cmd} is the actuator commanded position, u is the actuator achieved position, and τ is the actuator time constant. Since these devices have inherent position limits, the latter must be introduced into the model and analyzed appropriately.

In this example, we shall demonstrate how to create a dynamic model of an actuator with position constraints $|u| \leq u_{\text{max}}$. Let ε be a constant such that $\varepsilon \in (0, 1)$. We introduce,

$$\bar{u}_{\text{max}} = \frac{u_{\text{max}}}{\sqrt{1 + \varepsilon}}$$

and then embed the actuator constraints into the Projection Operator definition (11.37), by selecting a convex function in the form

$$f(u) = \frac{u^2 - \bar{u}_{\text{max}}^2}{\varepsilon \bar{u}_{\text{max}}^2} = \frac{(1 + \varepsilon)u^2 - u_{\text{max}}^2}{\varepsilon u_{\text{max}}^2}$$

In this case, the two convex sets from (11.35) and (11.36), become

$$\begin{aligned}\Omega_0 &= \{u \in R : f(u) \leq 0\} = \left\{u \in R : |u| \leq \frac{u_{\text{max}}}{\sqrt{1 + \varepsilon}}\right\} \\ \Omega_1 &= \{u \in R : f(u) \leq 1\} = \{u \in R : |u| \leq u_{\text{max}}\}\end{aligned}$$

Using (11.37), we can now define the following projection-based first-order actuator model with position constraints.

$$\begin{aligned}\dot{u} &= \text{Proj}\left(u, \frac{u_{\text{cmd}} - u}{\tau}\right) \\ &= \begin{cases} \left(\frac{u_{\text{cmd}} - u}{\tau}\right)(1 - f(u)), & \text{if } [f(u) > 0 \wedge (u_{\text{cmd}} - u)u > 0] \\ \left(\frac{u_{\text{cmd}} - u}{\tau}\right), & \text{if not} \end{cases}\end{aligned}$$

According to Theorem 11.1, starting anywhere within the “conservative” position limits $(\pm \frac{u_{\text{max}}}{\sqrt{1 + \varepsilon}})$, the actuator achieved position $u(t)$ will never exceed the original limits $(\pm u_{\text{max}})$, even if it is commanded to do so. In other word, there is no need to limit the commanded position. No matter what the actuator command is, the achieved position will remain within the pre-specified limits.

One can make further modifications to the derived actuator model and enforce rate limit constraints ($\pm \dot{u}_{\max}$) in addition to position limits. For example, the following model,

$$\dot{u} = \dot{u}_{\max} \operatorname{sat}\left(\frac{1}{\dot{u}_{\max}} \operatorname{Proj}\left(u, \frac{u_{\text{cmd}} - u}{\tau}\right)\right)$$

uses the saturation function $y = \operatorname{sat}(x) = \max(-1, \min(x, 1))$, along with the Projection Operator. It is easy to see that these two modifications will keep both the actuator position and its rate contained within their desired limits.

Frequently in practice, actuator requirements are specified in terms of their natural frequencies and damping ratios. This leads to consideration of a second-order actuator model in the form,

$$\ddot{u} + 2\xi\omega\dot{u} + \omega^2 u = \omega^2 u_{\text{cmd}}$$

where (ω, ξ) denote the actuator natural frequency and its damping ratio, correspondingly. Rewriting the model in state space, gives

$$\underbrace{\begin{pmatrix} \dot{u}_1 \\ \dot{u}_2 \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0 & 1 \\ -\omega^2 & -2\xi\omega \end{pmatrix}}_A \underbrace{\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}}_x + \underbrace{\begin{pmatrix} 0 \\ \omega^2 \end{pmatrix}}_B u_{\text{cmd}}$$

where $u_1 = u$ is the actuator achieved position, and $u_2 = \dot{u}$ is the respective rate.

In order to impose position and rate constraints $\pm(u_{\max}, \dot{u}_{\max})$, we shall again use the Projection Operator (11.37) and modify the actuator dynamics as follows:

$$\dot{x} = \operatorname{Proj}(x, A x + B u_{\text{cmd}})$$

For this model, a convex function $f(x) = f(u, \dot{u})$, which defines the Projection Operator domain, can be selected as

$$f(u, \dot{u}) = \frac{(1 + \varepsilon) \left(\frac{u^2}{u_{\max}^2} + \frac{\dot{u}^2}{\dot{u}_{\max}^2} \right) - 1}{\varepsilon}$$

This leads to the following two convex sets

$$\begin{aligned} \Omega_0 &= \{f(u, \dot{u}) \leq 0\} = \left\{ |u| \leq \frac{u_{\max}}{\sqrt{1 + \varepsilon}} \wedge |\dot{u}| \leq \frac{\dot{u}_{\max}}{\sqrt{1 + \varepsilon}} \right\} \\ \Omega_1 &= \{f(u, \dot{u}) \leq 1\} = \{|u| \leq u_{\max} \wedge |\dot{u}| \leq \dot{u}_{\max}\} \end{aligned}$$

From Lemma 11.4, we can assert that starting with any initial conditions from Ω_0 , which satisfy the actuator position and rate bounds, the actuator model will produce trajectories evolving within the prescribed bounds in Ω_1 . ■

In the next section, we shall employ the Projection Operator (11.37) to construct provably-stable adaptive laws in the form,

$$\begin{aligned}\dot{\theta} &= \text{Proj}(\theta, \Gamma y) \\ &= \Gamma \begin{cases} y - \frac{\nabla f(\nabla f)^T}{\|\nabla f\|_\Gamma^2} \Gamma y f, & \text{if } [f > 0 \wedge y^T \Gamma \nabla f > 0] \\ y, & \text{if not} \end{cases}\end{aligned}\quad (11.46)$$

where θ denotes the estimated parameter vector, whose dynamics are driven by the time-varying external vector $y = y(t)$.

Based on (11.46), we can introduce a matrix version of the Projection Operator, when both Y and Θ are matrices of the same dimensions.

$$Y = (\vec{y}_1 \dots \vec{y}_N) \in R^{n \times N}, \quad \Theta = (\vec{\theta}_1 \dots \vec{\theta}_N) \in R^{n \times N} \quad (11.47)$$

In this case, the Projection Operator is defined column-wise.

$$\text{Proj}(\Theta, \Gamma Y) = \left(\text{Proj}(\vec{\theta}_1, \Gamma \vec{y}_1) \dots \text{Proj}(\vec{\theta}_N, \Gamma \vec{y}_N) \right) \quad (11.48)$$

We can also generalize the convex inequality (11.38).

$$\begin{aligned}\text{tr}(\Delta \Theta^T (\Gamma^{-1} \text{Proj}(\hat{\Theta}, \Gamma Y) - Y)) \\ = \sum_{j=1}^m \underbrace{\left(\hat{\Theta} - \Theta \right)_j^T (\Gamma^{-1} \text{Proj}(\hat{\Theta}, \Gamma Y_j) - Y_j)}_{\leq 0} \leq 0\end{aligned}\quad (11.49)$$

In addition, one can show that for all matrices $\Theta(0)$, whose columns belong to the set Ω_0 from (11.41), the corresponding trajectory $\hat{\Theta}(t)$ of the matrix differential equation,

$$\dot{\Theta} = \text{Proj}(\Theta, \Gamma Y) \quad (11.50)$$

will have its columns evolving within the set Ω_1 from (11.42), for all $t \geq 0$. This statement directly follows from Lemma 11.4.

11.4 Projection-Based MRAC Design

In Sect. 11.1, we have designed robust MRAC systems for MIMO dynamics (11.1), with matched parametric uncertainties and a bounded process noise. These designs were carried out to force time derivatives of the selected Lyapunov function (11.10), computed along the trajectories of the error dynamics (11.9), to become negative semidefinite outside of a compact set. For example, in (11.14) we had,

$$\dot{V}(e, \Delta\Theta) = -e^T Q e + 2 \operatorname{trace} \left(\Delta\Theta^T \left\{ \Gamma_\Theta^{-1} \dot{\hat{\Theta}} - \Phi e^T P B \right\} \Lambda \right) + 2 e^T P \xi(t) \quad (11.51)$$

and the design task was to choose $\dot{\hat{\Theta}}$ such that the trace term in (11.51) became non-positive, while the adaptive parameters $\hat{\Theta}(t)$ remained uniformly bounded functions of time.

In what follows, we shall investigate how to force the trace term to be seminegative via the matrix version of the Projection Operator (11.50), with its convex property (11.49), while enforcing uniform boundedness of the corresponding solutions $\hat{\Theta}(t)$. Since,

$$\begin{aligned} & \operatorname{tr} \left(\underbrace{\Delta\Theta^T}_{(\hat{\Theta}-\Theta)^T} \left[\begin{array}{cc} \Gamma_\Theta^{-1} & \underbrace{\dot{\hat{\Theta}}}_{\operatorname{Proj}(\hat{\Theta}, \Gamma_\Theta Y)} \\ & \operatorname{Proj}(\hat{\Theta}, \Gamma_\Theta Y) \end{array} \right] \Lambda \right) \\ &= \sum_{j=1}^m \underbrace{\left(\hat{\Theta} - \Theta \right)_j^T \left(\Gamma_\Theta^{-1} \operatorname{Proj}(\hat{\Theta}, \Gamma_\Theta Y_j) - Y_j \right)}_{\leq 0} \underbrace{\lambda_j}_{\geq 0} \leq 0 \end{aligned} \quad (11.52)$$

then we can define the following projection-based adaptive law,

$$\dot{\hat{\Theta}} = \operatorname{Proj}(\hat{\Theta}, \Gamma_\Theta \Phi e^T P B) \quad (11.53)$$

to guarantee uniform boundedness of the adaptive gains, column-wise (Lemma 11.4). Essentially, the Projection Operator ensures that the columns $\hat{\Theta}_j$ of the adaptive time-dependent parameter matrix $\hat{\Theta}(t)$ do not exceed their pre-specified bounds Θ_j^{\max} . At the same time and because of (11.52), it is easy to see that the operator contributes to the negative semidefiniteness of the Lyapunov function (11.51). Indeed,

$$\begin{aligned} \dot{V}(e, \Delta\Theta) &\leq -e^T Q e + 2 e^T P \xi(t) \leq -\lambda_{\min}(Q) \|e\|^2 + 2 \|e\| \lambda_{\max}(P) \xi_{\max} \\ &= -\lambda_{\min}(Q) \|e\| \left(\|e\| - 2 \frac{\lambda_{\max}(P) \xi_{\max}}{\lambda_{\min}(Q)} \right) \end{aligned} \quad (11.54)$$

and consequently, $\dot{V}(e, \Delta\Theta) < 0$ outside of the compact set,

$$\Omega = \left\{ (e, \Delta\Theta) \in R^n \times R^{N \times m} : \|e\| \leq 2 \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)} \xi_{\max} \wedge \|\Delta\Theta\|_F \leq \Delta\Theta_{\max} \right\} \quad (11.55)$$

where

$$\Delta\Theta_{\max} = 2 \underbrace{(\Theta_1^{\max} \dots \Theta_m^{\max})}_{\Theta_{\max}} = 2 \Theta_{\max} \quad (11.56)$$

and Θ_j^{\max} is the maximum allowable bound for the j th column $\hat{\Theta}_j(t)$. This formal argument proves the UUB property of all signals in the corresponding closed-loop system. In particular, we have proven that the system regulated output $y(t)$ can track any external bounded command $y_{\text{cmd}}(t)$ with bounded errors.

Next, we show how to construct the convex vector function $f = (f_1 \dots f_m)^T$ and the related m -convex sets $\left\{ \Omega_{\delta}^j \right\}_{j=1, \dots, m}$. These are the sets that define the Projection Operator domains for each column of adaptive parameters $\hat{\Theta}_j(t)$. Both the function and the convex set definitions will be constructed based on the desired column-wise upper bounds $\|\hat{\Theta}_j(t)\| \leq \Theta_j^{\max}$.

For the j th column $\hat{\Theta}_j$ of the adaptive parameter matrix $\hat{\Theta} \in R^{N \times m}$, we introduce the projection tolerance $\varepsilon_j^{\Theta} > 0$, and choose a convex function in the form

$$f_j = f(\hat{\Theta}_j) = \frac{(1 + \varepsilon_j^{\Theta}) \|\hat{\Theta}_j\|^2 - (\Theta_j^{\max})^2}{\varepsilon_j^{\Theta} (\Theta_j^{\max})^2} \quad (11.57)$$

The idea here is very similar to the one in Example 11.4. The two convex sets are defined for each $j = 1, \dots, m$.

$$\begin{aligned} \Omega_0^j &= \left\{ \hat{\Theta}_j \in R^{N \times 1} : f(\hat{\Theta}_j) \leq 0 \right\} = \left\{ \hat{\Theta}_j \in R^{N \times 1} : \|\hat{\Theta}_j\| \leq \frac{\Theta_j^{\max}}{\sqrt{1 + \varepsilon_j^{\Theta}}} \right\} \\ \Omega_1^j &= \left\{ \hat{\Theta}_j \in R^{N \times 1} : f(\hat{\Theta}_j) \leq 1 \right\} = \left\{ \hat{\Theta}_j \in R^{N \times 1} : \|\hat{\Theta}_j\| \leq \Theta_j^{\max} \right\} \end{aligned} \quad (11.58)$$

The gradient of the j th convex function (11.57) can be easily computed as

$$\nabla f_j = \frac{(1 + \varepsilon_j^{\Theta})}{\varepsilon_j^{\Theta} (\Theta_j^{\max})^2} \nabla \left[\|\hat{\Theta}_j\|^2 \right] = \frac{2(1 + \varepsilon_j^{\Theta})}{\varepsilon_j^{\Theta} (\Theta_j^{\max})^2} \hat{\Theta}_j \quad (11.59)$$

Via (11.53), the adaptive law for $\hat{\Theta}_j$ becomes

$$\dot{\hat{\Theta}}_j = \Gamma_\Theta \begin{cases} (\Phi e^T P B)_j - \frac{\nabla f_j \nabla f_j^T}{\|\nabla f\|_{\Gamma_\Theta}^2} \Gamma_\theta (\Phi e^T P B)_j f_j, \\ \quad \text{if } [f_j > 0 \wedge (\Phi e^T P B)_j^T \Gamma_\theta \nabla f_j > 0] \\ (\Phi e^T P B)_j, \quad \text{if not} \end{cases} \quad (11.60)$$

By construction, the adaptation process in (11.60) ensures uniform boundedness of the adaptive time-dependent parameter matrix $\hat{\Theta}(t)$ forward in time, that is

$$\left\{ \|\hat{\Theta}_j(0)\| \leq \frac{\Theta_j^{\max}}{\sqrt{(1 + \varepsilon_j^\Theta)}} \right\} \Rightarrow \left\{ \|\hat{\Theta}_j(t)\| \leq \Theta_j^{\max}, \quad \forall t \geq 0, \quad 1 \leq j \leq m \right\} \quad (11.61)$$

Consequently, the adaptive parameter errors $\Delta\Theta(t)$ and the state tracking error $e(t)$ enter a compact set that contains the set Ω from (11.55) in finite time. The MIMO bounded command tracking problem is solved.

Table 11.1 gives a summary of the four robustness modifications that were introduced in this chapter.

Table 11.2 presents an overview of the continuous Projection Operator, which acts on a pair of n -dimensional vectors θ and y .

The next example illustrates key design points in application of the projection-based MRAC to lateral-direction dynamics of an aircraft.

Table 11.1 MRAC design with robustness modifications

Open-loop plant	$\dot{x} = A_{\text{ref}} x + B \Lambda(u + \Theta^T \Phi(x)) + B_{\text{ref}} y_{\text{cmd}} + \xi(t)$ $y = C_{\text{ref}} x$
Reference model	$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} y_{\text{cmd}}, \quad y_{\text{ref}} = C_{\text{ref}} x_{\text{ref}}$
State tracking error	$e = x - x_{\text{ref}}$
Lyapunov equation	$P A_{\text{ref}} + A_{\text{ref}}^T P = -Q$
Total control input	$u = -\hat{\Theta}^T \Phi(x)$
MRAC with dead-zone	$\dot{\hat{\Theta}} = \Gamma_\Theta \Phi(x) \mu(\ e\) e^T P B$
MRAC with σ -mod	$\dot{\hat{\Theta}} = \Gamma_\Theta (\Phi(x) e^T P B - \sigma \hat{\Theta})$
MRAC with e -mod	$\dot{\hat{\Theta}} = \Gamma_\Theta (\Phi(x) e^T P B - \sigma \ e^T P B\ \hat{\Theta})$
MRAC with Projection Operator	$\dot{\hat{\Theta}} = \text{Proj}(\hat{\Theta}, \Gamma_\Theta \Phi e^T P B)$

Table 11.2 The Projection Operator design summary

Max parameter bounds	$\ \theta\ \leq \theta^{\max}$
Convex function	$f(\hat{\theta}) = \frac{(1+\varepsilon)\ \theta\ ^2 - (\theta^{\max})^2}{\varepsilon(\theta^{\max})^2}$
Two convex sets	$\Omega_0 = \{\theta : f(\theta) \leq 0\} = \left\{ \theta : \ \theta\ \leq \frac{\theta^{\max}}{\sqrt{1+\varepsilon}} \right\}$ $\Omega_1 = \{\theta : f(\theta) \leq 1\} = \left\{ \theta : \ \theta\ \leq \theta^{\max} \right\}$
Projection Operator	$\text{Proj}(\theta, y) = \begin{cases} y - \frac{\Gamma \nabla f(\nabla f)^T}{(\nabla f)^T \Gamma \nabla f} y f, & \text{if } [f > 0 \wedge (y^T \nabla f) > 0] \\ y, & \text{if not} \end{cases}$
Convex inequality for proof of stability	$(\theta - \theta^*)^T (\Gamma^{-1} \text{Proj}(\theta, \Gamma y) - y) \leq 0,$ $\forall \theta^* \in \Omega_0, \theta \in \Omega_1, y \in R^n$
Uniform boundedness of parameters	$\dot{\theta} = \text{Proj}(\theta, \Gamma y)$ $[\theta(0) \in \Omega_0] \Rightarrow [\theta(t) \in \Omega_1, \forall t \geq 0]$

Example 11.5 Aircraft Lateral-directional Dynamics and Control Lateral-directional motion of a conventional aircraft is controlled by vertical tail panels (rudders) and wing-mounted surfaces (ailerons). Figure 11.13 shows a sketch.

The rudder (δ_r) is the primary control device for turning the aircraft, thus regulating the vehicle yaw rate r and the sideslip angle β . Moving ailerons differentially (i.e., left aileron trailing edge down and right aileron trailing edge up, δ_a) will force the aircraft to roll (write wing down), changing (increasing) its roll rate p , and thus the bank angle φ , with some induced coupling into the yaw and sideslip dynamics.

For small angles, the aircraft lateral-directional dynamics can be approximated by a linear time-invariant system in the form,

$$\underbrace{\begin{pmatrix} \dot{\varphi} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \end{pmatrix}}_{\dot{x}_p} = \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{g}{V} & \frac{Y_\beta}{V} & \frac{Y_p}{V} & \frac{Y_r}{V} - 1 \\ 0 & L_\beta & L_p & Y_r \\ 0 & N_\beta & N_p & N_r \end{pmatrix}}_{A_p} \underbrace{\begin{pmatrix} \varphi \\ \beta \\ p \\ r \end{pmatrix}}_{x_p} + \underbrace{\begin{pmatrix} 0 & 0 \\ \frac{Y_{\delta_a}}{V} & \frac{Y_{\delta_r}}{V} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{pmatrix}}_{B_p} \underbrace{\begin{pmatrix} \delta_a \\ \delta_r \end{pmatrix}}_u$$

where $g = 32.174$ is the acceleration due to gravity (ft/s^2), V is the trimmed airspeed (positive constant, ft/s), and the system matrices (A_p, B_p) are comprised of the vehicle aerodynamic stability and control derivatives.

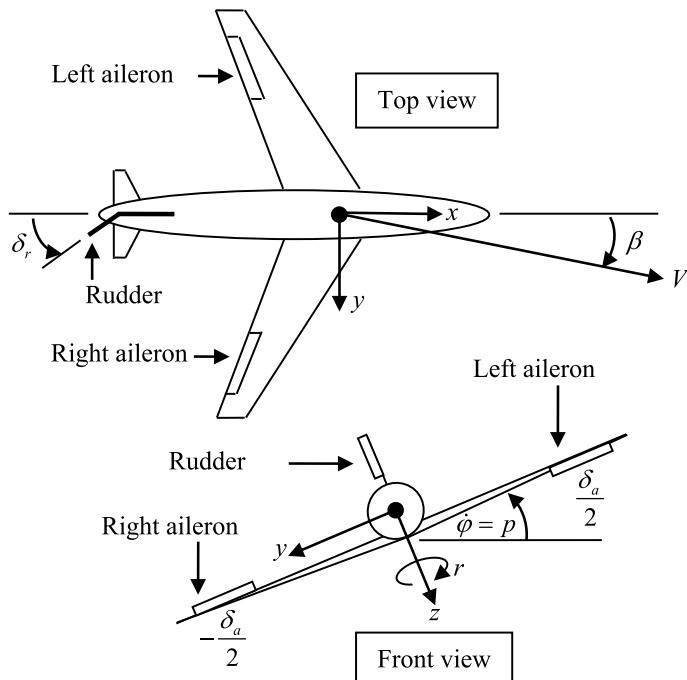


Fig. 11.13 Top and front views of a conventional aircraft in Example 11.5

For a small passenger aircraft in a cruise configuration, typical values of these parameters are [5],

$$A_p = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.0487 & -0.0829 & 0 & -1 \\ 0 & -4.546 & -1.699 & 0.1717 \\ 0 & 3.382 & -0.0654 & -0.0893 \end{pmatrix}, \quad B_p = \begin{pmatrix} 0 & 0 \\ 0 & 0.0116 \\ 27.276 & 0.5758 \\ 0.3952 & -1.362 \end{pmatrix}$$

where the units for all angles and angular rates are expressed in rad and rad/s, respectively. Also, negligible coefficients in the β -dynamics are zeroed out.

A typical (for lateral-directional dynamics) regulated output would consist of the vehicle bank and sideslip angles,

$$y = \begin{pmatrix} \varphi \\ \beta \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}}_{C_p} x_p$$

while the available control inputs are represented by the differential aileron and the rudder deflections, both expressed in radians.

$$u = (\delta_a \ \delta_r)^T$$

Table 11.3 Nominal open-loop vehicle eigenvalues in Example 11.5

Eigenvalue	Damping	Frequency (rad/s)
$-0.0464 \pm 1.88 j$	0.0247	1.88
0.00135	-1	0.00135
-1.78	1	1.78

The control task is to design u to enable independent and simultaneous tracking of bounded time-varying bank and sideslip commands, that are stored in the vector $y_{\text{cmd}} = (\varphi_{\text{cmd}} \beta_{\text{cmd}})^T$.

The nominal open-loop vehicle dynamics are unstable with the corresponding eigenvalues shown in Table 11.3.

In order to stabilize these dynamics and regulate the selected two outputs, we are going to design a baseline LQR tracking controller with Proportional + Integral (PI) action. Toward that end, we augment the system with two integrated tracking errors, and obtain the baseline/nominal extended open-loop system,

$$\underbrace{\begin{pmatrix} \dot{e}_{\varphi I} \\ \dot{e}_{\beta I} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0_{2 \times 2} & C_p \\ 0_{4 \times 2} & A_p \end{pmatrix}}_A \underbrace{\begin{pmatrix} e_{\varphi I} \\ e_{\beta I} \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} 0_{2 \times 2} \\ B_p \end{pmatrix}}_B \underbrace{\begin{pmatrix} \delta_a \\ \delta_r \end{pmatrix}}_u + \underbrace{\begin{pmatrix} -I_{2 \times 2} \\ 0_{4 \times 2} \end{pmatrix}}_B \underbrace{\begin{pmatrix} \varphi_{\text{cmd}} \\ \beta_{\text{cmd}} \end{pmatrix}}_{y_{\text{cmd}}}$$

$$y = \underbrace{\begin{pmatrix} 0_{2 \times 2} & C_p \end{pmatrix}}_C x = (\varphi \ \beta)^T$$

where

$$\dot{e}_{\varphi I} = \varphi - \varphi_{\text{cmd}}, \quad \dot{e}_{\beta I} = \beta - \beta_{\text{cmd}}$$

are the dynamics of the two integrated tracking error signals. After several design iterations, we have selected diagonal LQR weights.

$$Q = \text{diag}(1 \ 10 \ 0 \ 0 \ 0.1 \ 5), \quad R = I_{2 \times 2}$$

The first two diagonal elements of Q give adequate natural frequencies, while the last two yield desired damping ratios in both regulated output channels (Table 11.4).

Table 11.4 Nominal closed-loop vehicle eigenvalues in Example 11.5

Eigenvalue	Damping	Frequency (rad/s)
$-1.34 \pm 1.29 j$	0.72	1.86
$-1.25 \pm 1.17 j$	0.73	1.71
-1.33	1	1.33
-8.84	1	8.84

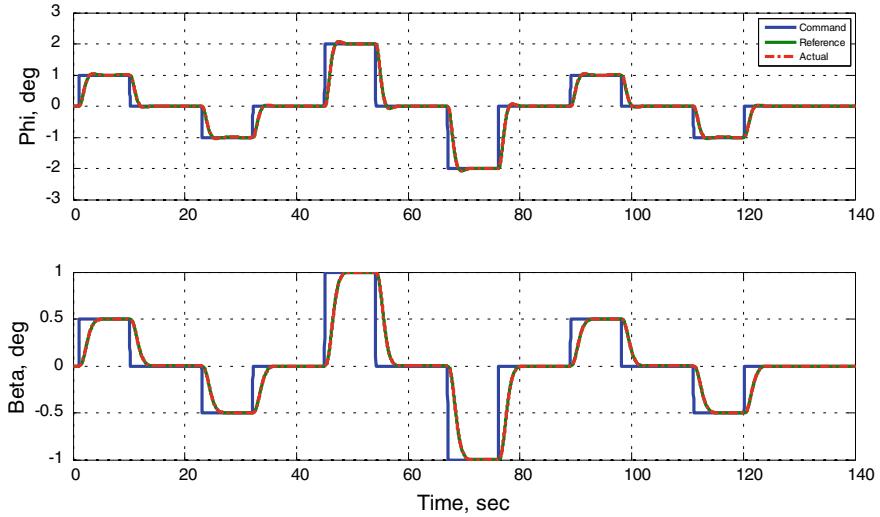


Fig. 11.14 Command tracking with baseline LQR PI controller in Example 11.5

The resulting baseline LQR PI state feedback solution is

$$u_{bl} = - \underbrace{\begin{pmatrix} 0.9987 & 0.1627 & 0.9184 & 0.0896 & 0.3529 & -0.0166 \\ -0.0514 & 3.1581 & 0.0755 & 2.2907 & 0.0487 & -2.7885 \end{pmatrix}}_{K_{xLQR}^T} x = -K_{xLQR}^T x$$

and the closed-loop simulation results are shown in Fig. 11.14, where we have tested the baseline LQR PI controller performance in tracking a series of step-input bank and sideslip commands, simultaneously.

There are three signals per plot that are shown in Fig. 11.14: (1) the command response; (2) the reference response, and (3) the actual system response. As in all our previous examples, the reference data represent the closed-loop vehicle behavior under the baseline LQR PI controller. Since there are no uncertainties in the baseline system dynamics, the reference and the actual responses are identical. The required aileron and rudder deflections (Fig. 11.15) are well behaved and definitely reside within realistic actuation limits.

Next, we introduce matched linear-in-parameters uncertainties into the system,

$$\dot{x} = A x + B \Lambda \left(u + \Theta^T \Phi(x_p) \right) + B_{\text{ref}} y_{\text{cmd}}$$

embed the baseline LQR PI solution $u_{bl} = -K_{xLQR}^T x$, and arrive at the extended open-loop dynamics.

$$\dot{x} = \underbrace{\left(A - B K_{xLQR}^T \right)}_{A_{\text{ref}}} x + B \Lambda \left(u + \Lambda^{-1} K_{xLQR}^T x + \Theta^T \Phi(x_p) \right) + B_{\text{ref}} y_{\text{cmd}}$$

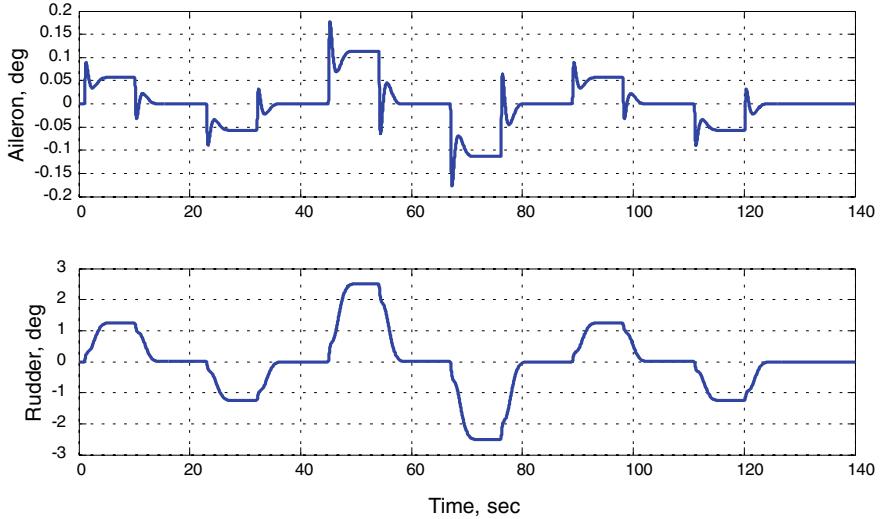


Fig. 11.15 Baseline aileron and rudder deflections in Example 11.5

Let u_{ad} denote an adaptive control augmentation signal. With the total control input,

$$u = u_{bl} + u_{ad}$$

the extended open-loop system becomes,

$$\begin{aligned}\dot{x} &= A_{\text{ref}}x + B \Lambda \left(u_{ad} + \underbrace{\left(\Lambda^{-1} - I_{2 \times 2} \right) K_x^T \text{LQR } x + \Theta^T \Phi(x_p) \over \Theta^T \bar{\Phi}(x)} \right) + B_{\text{ref}} y_{\text{cmd}} \\ &= A_{\text{ref}}x + B \Lambda \left(u_{ad} + \bar{\Theta}^T \bar{\Phi}(x) \right) + B_{\text{ref}} y_{\text{cmd}}\end{aligned}$$

which is in the same exact form as in Table 11.1.

For simulation studies, we have selected the following uncertainty-related parameters

$$\Lambda = 0.5 I_{2 \times 2}, \quad \Phi(x_p) = (\beta \ p \ r)^T$$

$$\Theta = \begin{pmatrix} 4 A_p(2, 2) & 2 A_p(2, 3) & 2 A_p(2, 4) \\ 4 A_p(2, 1) & 2 A_p(3, 3) & 2 A_p(3, 4) \end{pmatrix}^T$$

With 50% control effectiveness reduction in aileron and rudder, these parameters emulate 200% change in the aircraft sideslip coefficients, and 100% change in the vehicle roll and yaw stability derivatives. The perturbed system is open-loop unstable.

Its command tracking responses under the baseline LQR PI controller become highly oscillatory and thus inadequate. The data are shown in Fig. 11.16.

Although the baseline controller was able to stabilize the perturbed dynamics, the tracking performance is clearly unacceptable. Also, the corresponding aileron and rudder deflections exhibit the unwanted oscillations (Fig. 11.17).

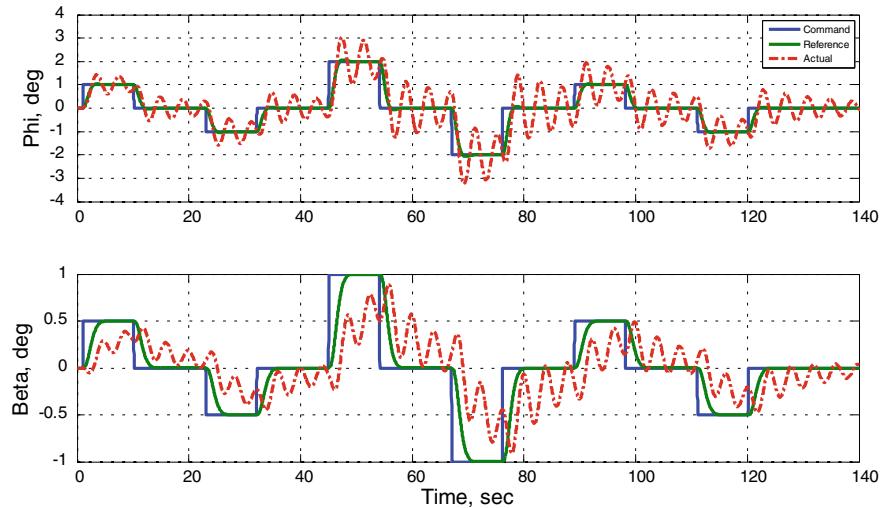


Fig. 11.16 Closed-loop response with uncertainties and LQR PI controller in Example 11.5

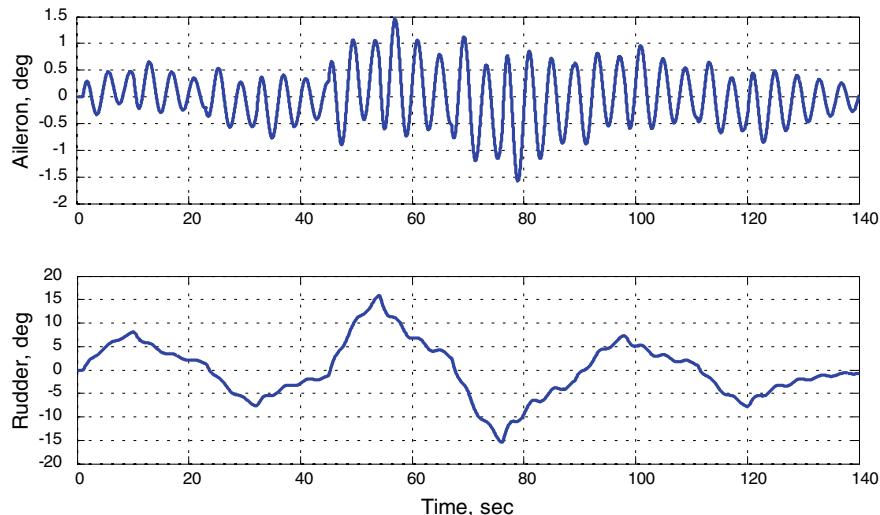


Fig. 11.17 LQR PI control inputs with uncertainties turned on in Example 11.5

In order to mitigate the system uncertainties, we add an adaptive augmentation component in the form,

$$u_{ad} = -\hat{\Theta}^T \bar{\Phi}(x)$$

with the estimated parameters $\hat{\Theta}(t)$ evolving according to the projection-based adaptive laws as shown in Table 11.2. The reference model is chosen to represent the closed-loop nominal system under the LQR PI controller and without uncertainties. The Q matrix in the Lyapunov algebraic equation is

$$Q = \text{diag}(0 \ 0 \ 0 \ 0 \ 10 \ 800)$$

and the rates of adaptation are

$$\Gamma_\Theta = \text{diag}(100 \ 100 \ 600 \ 600 \ 600 \ 600)$$

With the uncertainties turned on, the (LQR PI + Adaptive) controller recovers the desired closed-loop tracking performance (Fig. 11.18).

The required control effort is reasonable and well within the actuator capabilities of a generic aircraft such as the one considered (Fig. 11.19).

The magnitudes of the estimated parameters are shown in Fig. 11.20.

In this simulation, maximum allowable bounds for the adaptive parameters were set to 10, but the adaptive parameters never reached their bounds. So, it would be interesting to simulate a case when these bounds are reduced below their maximum

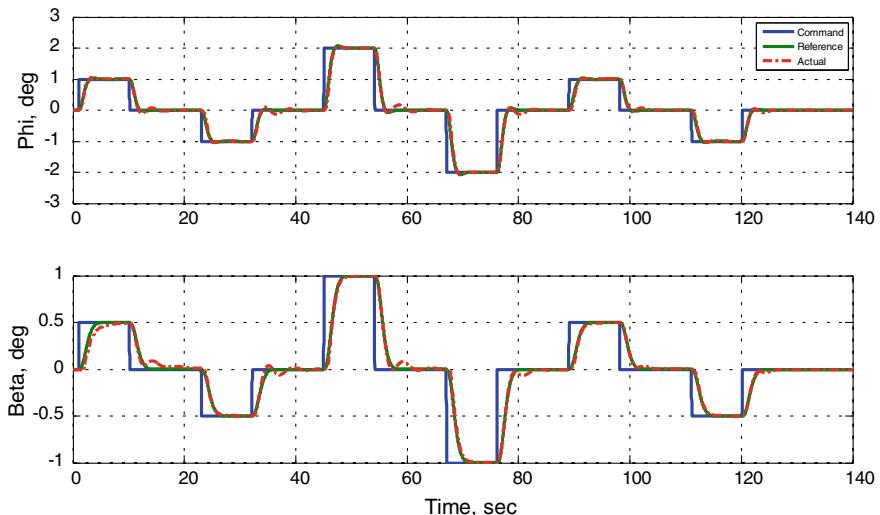


Fig. 11.18 Closed-loop tracking with (LQR PI + Adaptive) controller in Example 11.5

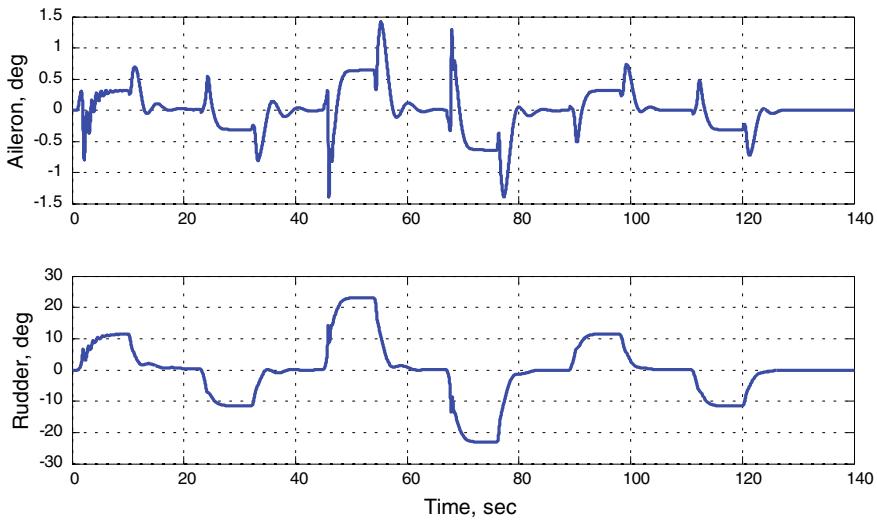


Fig. 11.19 (LQR PI + Adaptive) aileron and rudder deflections in Example 11.5

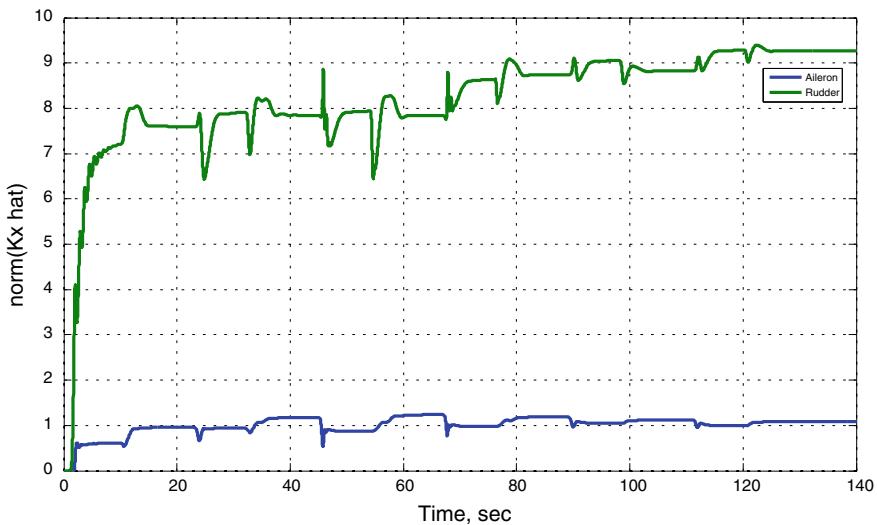


Fig. 11.20 Adaptive parameter dynamics in Example 11.5

achieved values. We set the aileron-related max bound to 0.5 and the rudder-related bound to 5. With the same uncertainties activated, Fig. 11.21 shows “graceful degradation” of the system closed-loop tracking performance.

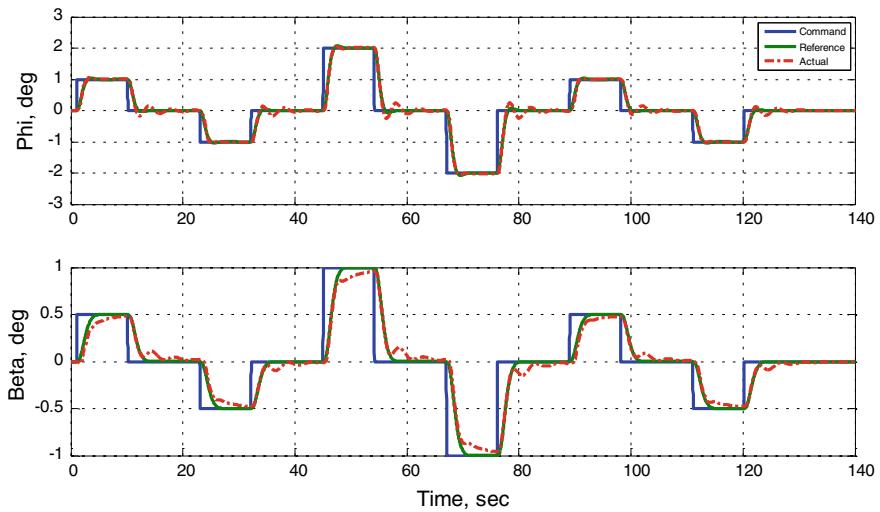


Fig. 11.21 Performance degradation with small projection limits in Example 11.5

Per design, the adaptive parameters evolve within the smaller projection bounds (Fig. 11.22), and because of that the closed-loop system performance degrades slightly.

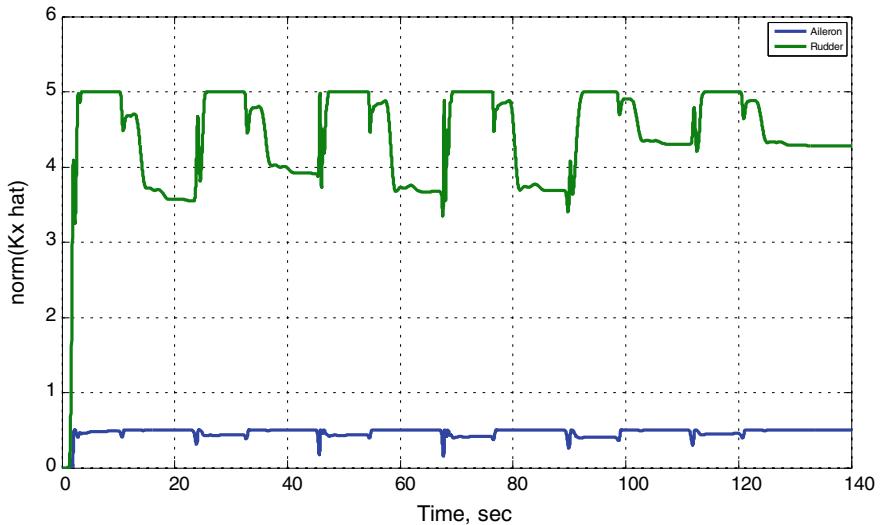


Fig. 11.22 Adaptive parameters with small projection bounds in Example 11.5

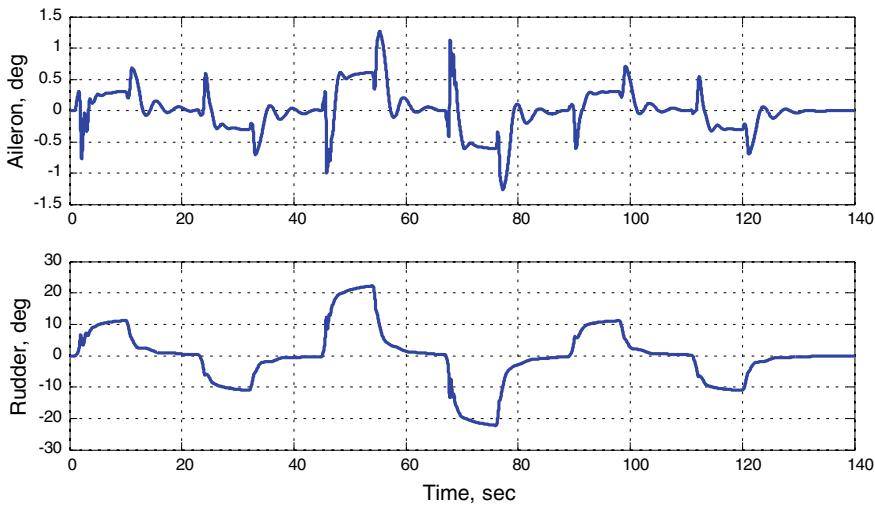


Fig. 11.23 Aileron and rudder deflections with small projection bounds in Example 11.5

However, the aileron and the rudder control activity (Fig. 11.23) remained very similar to the previous simulation case, where we had large projection bounds and thus attained a slightly better performance.

The main purpose of this simulation test is to verify that the Projection Operator has been implemented and functioned correctly. Additionally, we want to expose an iterative nature of a control design process, such as MRAC. Based on theoretical predictions, the control designer is always expected to perform a trade-off study to find the best set of tuning parameters for the selected method, while performing an assessment of simulation trials versus theoretical predictions. ■

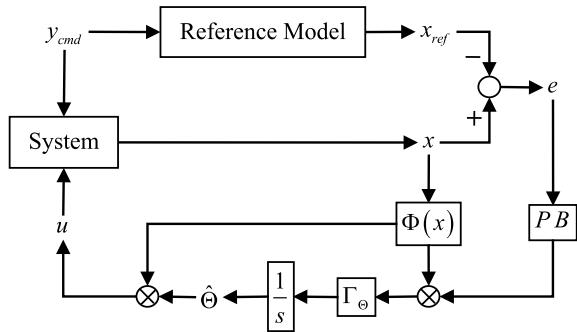
11.5 Summary and Discussion

In conclusion, we would like to offer our opinion on the choice of robustness modifications in MRAC systems. These recommendations are not “theoretical” by any means. They are merely based on the authors’ extensive experience during the design of MRAC systems for a multitude of aerospace applications.

In our view, any adaptive system must have the dead-zone modification (11.18) or its continuous version (11.19). The latter is the preferred choice since it avoids potential discontinuities in feedback connections. The “must-have” dead-zone modification will prevent adaptive parameters from drifting away.

As seen from (11.15), the adaptive law dynamics without robustness modifications are defined by integrating a nonlinear function, represented by the regressor vector $\Phi(x)$, multiplied by a linear combination of the state tracking

Fig. 11.24 Adaptive system viewed as a nonlinear integral feedback controller



errors ($e^T P B$). This product is further multiplied by a constant matrix Γ_Θ (the integral gain), and finally it is integrated to yield the adaptive parameters $\hat{\Theta}(t)$ (Fig. 11.24).

As seen from the block diagram of Fig. 11.24, there is a chain of nonlinear integrators in a feedback loop, whose output constitute the adaptive parameters. In all practical applications, feedback integrators must be “managed” in the sense that their output signals (i.e., the adaptive parameters) need to be constrained. This prevents integrators against “winding up” due to nonlinear saturation functions in the control channels, where the system achievable control limits are defined and enforced. Control techniques that prevent the integrator windup problems are called the “anti-windup” methods, and the Projection Operator is one of them. This is why we highly recommend using projection-based adaptive laws.

In summary, our suggested MRAC architecture consists of the smoothed dead-zone modification coupled with the Projection Operator,

$$\dot{\hat{\Theta}} = \text{Proj}\left(\hat{\Theta}, \Gamma_\Theta \Phi \mu(\|e\|) e^T P B\right) \quad (11.62)$$

where $\mu(\|e\|)$ is the Lipchitz-continuous modulation function from (11.19). Essentially, the dead-zone modification protects the adaptive parameters from drifting due to noise, while the Projection Operator bounds the overall adaptive process, and at the same time, it prevents MRAC integrators against the undesirable windup phenomenon.

11.6 Exercises

Exercise 11.1 Prove that the Projection Operator (11.37) is locally Lipschitz.

Exercise 11.2 Simulate the two actuator models from Example 11.4. Select commands to violate position and rate constraints. Compare and discuss your results.

Exercise 11.3 [10] Consider a convex hypercube in R^n ,

$$\Omega = \left\{ \theta \in R^n : (\theta_i^{\min} \leq \theta_i \leq \theta_i^{\max})_{i=1, 2, \dots, n} \right\}$$

where $(\theta_i^{\min}, \theta_i^{\max})$ represent the minimum and maximum bounds for the i th component of the n -dimensional parameter vector θ . Choose a sufficiently small positive constant δ , and define another hypercube,

$$\Omega_\delta = \left\{ \theta \in R^n : (\theta_i^{\min} + \delta \leq \theta_i \leq \theta_i^{\max} - \delta)_{i=1, 2, \dots, n} \right\}$$

such that $\Omega_\delta \subset \Omega$.

For two n -dimensional vectors (θ, y) , a rectangular version of the Projection Operator is defined component wise as,

$$\text{Proj}_i(\theta, y) = \begin{cases} \left(\frac{\theta_i^{\max} - \theta_i}{\delta} \right) y_i, & [(\theta_i > \theta_i^{\max} - \delta) \wedge (y_i > 0)] \\ \left(\frac{\theta_i - \theta_i^{\min}}{\delta} \right) y_i, & [(\theta_i < \theta_i^{\min} + \delta) \wedge (y_i < 0)] \\ y_i, & \text{otherwise} \end{cases}$$

Suppose that $\theta^* \in \Omega_\delta$ is a constant vector. Prove that for any $\theta \in \Omega$ and for any $y \in R^n$, the following inequality takes place

$$(\theta - \theta^*)^T (\text{Proj}(\theta, y) - y) \leq 0$$

Let Γ be a positive definite diagonal matrix. For the system dynamics (11.1), using the above inequality and adaptive laws (11.53) with the rectangular version of the Projection Operator, carry out stability proofs starting from (11.51), arriving at a UUB-type argument about the closed-loop system tracking performance.

Exercise 11.4 A second-order actuator model (transfer function) is given in the form,

$$\delta = \left(\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \right) \delta_{\text{cmd}}$$

where $(\delta, \delta_{\text{cmd}})$ are the actual and commanded actuator positions (rad), while (ξ, ω_n) are the actuator model damping ratio and its natural frequency, respectively. Assume $\xi = 0.7$ and $\omega_n = 1$. Simulate the system response to a sinusoidal command. Introduce actuator position and rate limits. Use Projection Operator (11.37) to create an actuator model with position and rate constraints. Create another model using the rectangular version of the Projection Operator from Exercise 11.3. Select actuator commands to violate the actuator position constraints. Simulate both models and compare their performance.

Exercise 11.5 Implement the aircraft lateral-directional data from Example 11.5. Design an (LQR PI + Adaptive) controller using (11.62). Repeat simulation tests from Example 11.5. Compare and discuss your results.

References

1. Peterson BB, Narendra KS (1982) Bounded error adaptive control. *IEEE Trans Autom Control* 27:1161–1168
2. Slotine JJE, Coetsee JA (1986) Adaptive sliding controller synthesis for nonlinear systems. *Int J Control*
3. Ioannou P, Kokotovic P (1983) Adaptive systems with reduced models. Springer, New York
4. Narendra KS, Annaswamy AM (1987) A new adaptive law for robust adaptive control without persistency of excitation. *IEEE Trans Autom Control* 32:134–145
5. McRuer D, Ashkenas I, Graham D (1990) Aircraft dynamics and automatic control. Princeton University Press, Princeton, NJ
6. Khalil H (1996) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, NJ
7. Pomet JB, Praly L (1992) Adaptive nonlinear regulation: estimation from the Lyapunov equation. *IEEE Trans Autom Control* 37:729–740
8. Slotine JJE, Li W (1995) Applied nonlinear control. Prentice Hall, New Jersey
9. Kreisselmeier G, Narendra KS (1982) Stable model reference adaptive control in the presence of bounded disturbances. *IEEE Trans Autom Control* 27:1169–1175
10. Khalil HK (1996) Adaptive output feedback control of nonlinear systems represented by input-output models. *IEEE Trans Autom Control* 41:177–188
11. Ioannou P, Fidan P (2006) Adaptive control tutorial. Advances in design and control. SIAM, PA



Approximation-Based Adaptive Control

12

This chapter is focused on the design and analysis of adaptive controllers for dynamical systems operating in the presence of nonparametric unknown nonlinear functions and bounded time-varying disturbances. In order to counter these types of uncertainties, we will employ direct adaptive model reference controllers equipped with online function approximation architectures, such as artificial neural networks (NNs). We begin with an introductory review of theoretical results related to function approximation by NNs (Sects. 12.1–12.3). As for any other function representation constructs, NN-based approximations are valid only on bounded sets. So, a suitable control design must account for a set of state limiting constraints imposed by the chosen function approximation method. For our proposed adaptive control design (Sect. 12.4), we will utilize online tunable artificial NNs to represent unstructured uncertainties in the system dynamics of interest. In addition, we will add a state limiting design modification to keep the system trajectories within predefined NN-induced state limiting constraints. We end this chapter with a comprehensive step-by-step design example of an automatic landing system for a medium-size transport aircraft.

12.1 Motivation

A typical control design starts with modeling, which is basically a procedure of constructing a mathematical description (such as a set of ordinary differential equations) for the physical system to be controlled. This selected model needs to reflect main features of the physical process. Accurate models are not always better. They may require unnecessarily complex control design and demand excessive computations. From a control point of view, the key in modeling is to capture

the essential effects in the system dynamics within an operating range of interest. In addition, a good model should also provide some characterization of the system uncertainties—the so-called unknown unknowns in the physical process. Such a characterization can later be used to perform robust and/or adaptive design, or to run Monte Carlo-based analysis, eventually leading to quantification and assessment of the closed-loop system stability, performance, and robustness.

In essence, model uncertainties symbolize the differences between the model and the real physical process. Uncertainties in the system-specific parameters are called “parametric”, while all other uncertainties are “nonparametric”.

Example 12.1 Point-Mass Dynamics with Parametric Uncertainties For the model of a controlled mass $m \ddot{x} = u$, the uncertainty in m is parametric, while the neglected motor dynamics, measurement noise, and sensor dynamics represent the nonparametric uncertainties. ■

Example 12.2 Scalar Dynamics with Nonparametric Uncertainties Consider a scalar model with uncertain dynamics, such as $\dot{x} = f(x) + u + \xi(t)$, where x is the system state, u is the control input, $\xi(t)$ is the process noise, and the function $f(x)$ is unknown. Suppose that,

$$f(x) = \sum_{i=1}^N \theta_i \varphi_i(x) + \varepsilon(x) = \underbrace{\theta^T \Phi(x)}_{\text{Parametric}} + \underbrace{\varepsilon(x)}_{\text{Nonparametric}}$$

In other words, we assume that the unknown function $f(x)$ can be approximated by a finite linear combination of known basis functions $\varphi_i(x)$ and unknown constant parameters θ_i . In this case, the state-dependent function approximation error $\varepsilon(x)$ and the process noise $\xi(t)$ represent the nonparametric uncertainties, while the unknown constant parameters θ constitute the parametric uncertainty in the system dynamics. In order to characterize the latter, one needs to be able to find a good set of basis functions $\Phi(x)$, such that the approximation error $\varepsilon(x)$ becomes small on a compact (closed and bounded) set. Polynomials, Fourier series expansions, splines, and artificial feedforward neural networks can be used to represent and approximate functions on compact sets. ■

In Sect. 12.4, we will design MRAC systems that can cope with both parametric and nonparametric uncertainties. In order to justify our design approach, we begin with a concise background material and an overview of important facts related to function approximation using artificial NNs to represent large classes of functions on given compact sets and within pre-specified approximation tolerances.

12.2 Basic Definitions

An artificial feedforward NN is a multi-input–multi-output static map composed of many interconnected nonlinear processing elements (neurons) operating in parallel. Figures 12.1 and 12.2 show sketches of two feedforward NNs.

An artificial feedforward NN consists of basic units called the “neurons” and their connections. A block diagram of a single artificial neuron is shown in Fig. 12.3.

Neurons, the basic processing elements of NNs, have two main components: (a) a weighted summer and (b) a nonlinear activation function. The activation

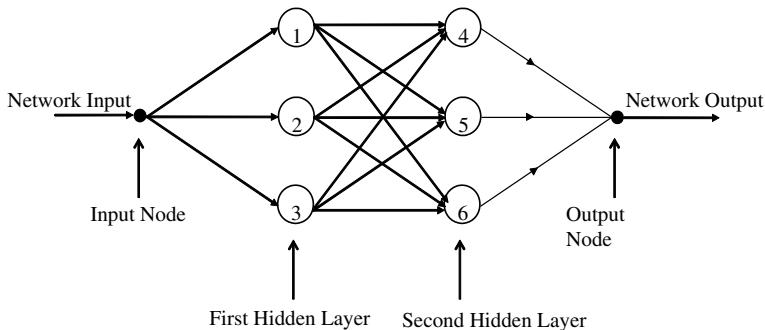


Fig. 12.1 Feedforward neural network with 2 hidden layers and 6 neurons

Fig. 12.2 Feedforward neural network with 1 hidden layer and 5 neurons

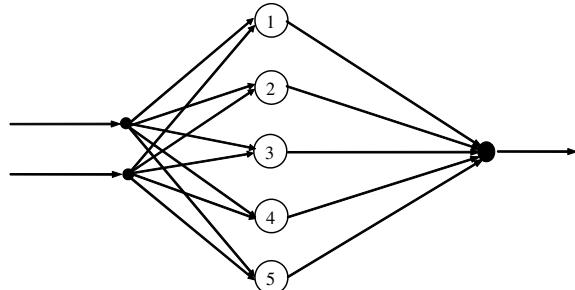
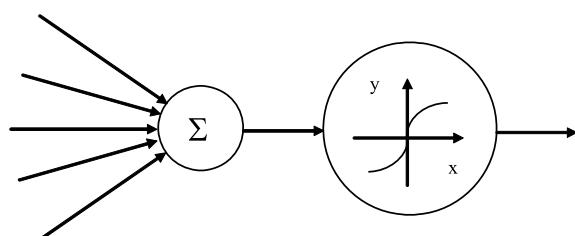


Fig. 12.3 Artificial neuron block diagram



functions of interest to us are the radial basis functions (RBFs) and the ridge functions, also called the “sigmoids”.

Definition 12.1 (*Radial Basis Functions (RBFs)*) An RBF is a Gaussian in the form

$$\varphi(x, x_c) = e^{-(x-x_c)^T W (x-x_c)} = e^{-\|x-x_c\|_W^2} \quad (12.1)$$

In (12.1), $x \in R^n$ is the input, $x_c \in R^n$ is the center, and $W = W^T > 0$ is a positive-definite symmetric matrix of weights. Most often, we will write $\varphi(x, x_i) = \varphi_i(x)$ to abbreviate and to denote an RBF which is centered at the i th center x_i .

Other definitions of RBFs are available in the literature [1, 2]. A generic RBF can be defined as $\varphi = \varphi(\|x - x_c\|_W)$, where $\|x\|_W = \sqrt{x^T W x}$ denotes the weighted Euclidean norm of a vector x . In addition, it is required that $\varphi(x)$ be integrable on R^n and $\int_{R^n} \varphi(x) dx \neq 0$. This activation function depends only on the weighted distance $r = \|x - x_c\|_W$ between its current input x and the center x_c . The Gaussian RBF in (12.1) is an example of this type of activation function. Others include: (a) multi-quadrics, $\varphi(r) = \sqrt{(r^2 + c^2)}$, $c > 0$, and (b) inverse multi-quadrics, $\varphi(r) = \frac{1}{\sqrt{(r^2 + c^2)}}$, $c > 0$.

Definition 12.2 (*Ridge Functions*) A ridge function or a sigmoid is a nonlinear scalar map $\sigma: R \rightarrow R$ of the form,

$$\sigma = \sigma(w^T x + b) \quad (12.2)$$

where $w \in R^n$ denotes the vector of weights, b is a scalar threshold, and $\sigma(\cdot)$ is a scalar nonlinear function (not necessarily continuous) on R , with the following property:

$$\lim_{v \rightarrow \pm\infty} \sigma(v) < \infty \quad (12.3)$$

The two most common examples of a ridge function are: (a) the logistic sigmoid, $\sigma(v) = \frac{1}{1+e^{-v}}$, and (b) the hyperbolic tangent, $\sigma(s) = \frac{1-e^{-s}}{1+e^{-s}}$.

A feedforward NN with N neurons in its hidden layer is shown in Fig. 12.4.

Formally speaking, a feedforward NN is a map from R^n to R^m , that is,

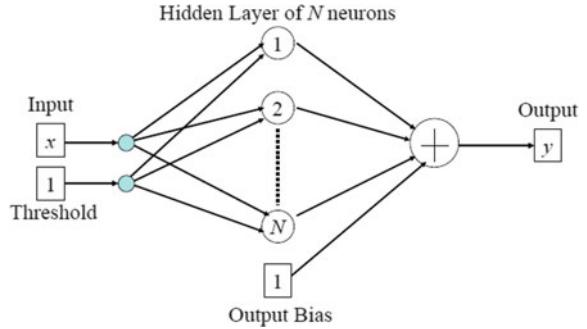
$$y = \text{NN}(x): R^n \rightarrow R^m \quad (12.4)$$

Definition 12.3 (*Sigmoidal Feedforward NNs*) A sigmoidal feedforward NN with N neurons is a map from R^n to R^m in the form,

$$\text{NN}(x) = W^T \vec{\sigma}(V^T x + \theta) + b \quad (12.5)$$

Fig. 12.4

Single-hidden-layer feedforward NN with N neurons



where $W \in R^{N \times m}$ is the matrix of the outer-layer weights,

$$\vec{\sigma}(x) = (\sigma(V_1^T x + \theta_1) \dots \sigma(V_N^T x + \theta_N))^T \in R^N$$

is the vector of N sigmoids, $V \in R^{n \times N}$ is the matrix of the inner-layer synaptic weights, with its i th column denoted by $V_i \in R^n$, $\theta \in R^N$ is the vector of thresholds, and $b \in R^m$ denotes the NN bias vector.

Definition 12.4 (Feedforward RBF NNs) A feedforward RBF NN is a map from R^n to R^m in the form,

$$\text{NN}(x) = \theta^T \begin{pmatrix} \varphi(\|x - C_1\|_{W_1}) \\ \vdots \\ \varphi(\|x - C_N\|_{W_N}) \end{pmatrix} + b = \underbrace{(\theta^T \ b)}_{\Theta^T} \underbrace{\begin{pmatrix} \varphi_1(x) \\ \vdots \\ \varphi_N(x) \\ 1 \end{pmatrix}}_{\Phi(x)} = \Theta^T \Phi(x) \quad (12.6)$$

where $\Theta = (\theta^T \ b)^T \in R^{(N+1) \times m}$ is the vector of weights, $C_i \in R^n$ is the center of the i th receptive field, $W_i = W_i^T > 0$ is the norm weighting matrix, $b \in R^m$ is the NN bias, and $\Phi(x) = (\varphi_1(x) \dots \varphi_N(x) \ 1)^T \in R^{N+1}$ is the regressor vector, whose components are the basis (activation) functions $\varphi_i(x) = \varphi(\|x - C_i\|_{W_i})$ and the unit function. ■

Often in practical applications, the symmetric positive-definite matrix W in (12.6) is chosen to be diagonal and in the form

$$W_i = \frac{1}{2\sigma_i^2}, \quad (i = 1, \dots, N)$$

where σ_i represents the width of the i th Gaussian function. In this case,

$$\varphi_i(x) = e^{-\frac{\|x-C_i\|^2}{2\sigma_i^2}}$$

becomes the i th component of the regressor vector $\Phi(x)$ in (12.6). Also, components of the regressor can be constructed using the Gaussian,

$$\varphi_i(x) = e^{-\left(\frac{N}{d_{\max}^2}\right)\|x-C_i\|^2}$$

whose standard deviation (width) σ is fixed according to the spread of the centers C_i , N is the number of centers, and d_{\max} is the maximum distance between the chosen centers. Here, the standard deviation σ of all the isotropic Gaussian RBF components is fixed at

$$\sigma = \frac{d_{\max}}{\sqrt{2N}}$$

This formula ensures that the individual RBFs are not too peaked or too flat. Both of these two extreme conditions should be avoided.

12.3 Approximation Properties of Feedforward Neural Networks

Feedforward NNs have been shown to be capable of approximating generic classes of functions, on compact sets and to within any pre-specified tolerance. This property of feedforward NNs is often referred to as the universal approximation, while the NNs themselves are often called the universal approximators. Related theorems are stated below without proofs.

Theorem 12.1 (Micchelli's Theorem [3]) *Let $\varphi = \varphi(r)$ be the Gaussian, the multi-quadratics, or the inverse multi-quadratics function. Let $\{x_i\}_{i=1}^N$ be a set of distinct points in R^n . Then the $(N \times N)$ interpolation matrix Φ , whose $(i, j)^{\text{th}}$ element is $\varphi_{ij} = \varphi(\|x_i - x_j\|)$, is non-singular. ■*

There is a large class of RBFs that is covered by Micchelli's theorem. In fact, this theorem provides a theoretical basis for RBF-based function approximation and regression techniques. Specifically, using an RBF $\varphi = \varphi(r)$ and a finite set of N points $\{x_i\}_{i=1}^N$ in R^n , the above theorem assures that it is always possible to approximate functions $f(x)$ on a grid of points, using a linear combination of RBFs in the form $\hat{f}(x) = \sum_i^N \theta_i \varphi(x - x_i)$, such that $f(x_i) = \hat{f}(x_i)$, for all $\{x_i\}_{i=1}^N$.

Theorem 12.2 (Universal Approximation Theorem for Sigmoidal NNs [4]) Any continuous function $f(x): R^n \rightarrow R$ can be uniformly approximated by a single-hidden-layer NN,

$$\forall \varepsilon > 0, \exists N, W, b, V, \theta, \forall x \in X \subset R^n: \left\| \underbrace{W^T \tilde{\sigma}(V^T x + \theta) + b}_{\text{NN}(x)} - f(x) \right\|_{\infty} \leq \varepsilon \quad (12.7)$$

with a bounded monotone-increasing continuous activation vector function $\tilde{\sigma}(\cdot)$, on a compact domain $X \subset R^n$. \blacksquare

The universal approximation theorem extends to the class of L_1 functions defined on compact sets. In that case, it is assumed that the selected activation function is a bounded measurable sigmoid and the approximation is understood in terms of the L_1 functional norm.

Theorem 12.3 (Rates of Approximation Theorem for Sigmoidal NNs [5]) Consider a class of functions $f(x)$ on R^n for which there is a Fourier representation of the form

$$f(x) = \int_{R^n} e^{i \omega x} \tilde{f}(\omega) d\omega$$

for some complex-valued function $\tilde{f}(\omega)$ for which $\omega \tilde{f}(\omega)$ is integrable, and define

$$C_f = \int_{R^n} \|\omega\| \left\| \tilde{f}(\omega) \right\| d\omega < \infty.$$

Then for every function $f(x)$ with C_f finite, and every $N \geq 1$, there exists a sigmoidal NN of the form (12.5), such that

$$\|f(x) - \text{NN}(x)\|_{L_2}^2 = \int_{\|x\| \leq r} (f(x) - \text{NN}(x))^2 dx \leq \frac{(2r C_f)^2}{N} \quad \blacksquare$$

Functions with C_f finite are continuously differentiable on R^d . Moreover, the NN approximation error is measured by the L_2 -norm, on the ball of radius r .

Theorem 12.4 (Universal Approximation Theorem for RBF NNs [6]) Let $\varphi(x): R^n \rightarrow R$ be an integrable bounded continuous function and assume that

$$\int_{R^n} \varphi(x) dx \neq 0$$

Then for any continuous function $f(x)$ and any $\varepsilon > 0$ there is an RBF NN with N neurons, a set of centers $\{C_i\}_{i=1}^N$, and a common width $\sigma > 0$,

$$\hat{f}(x) = \sum_{i=1}^N \theta_i \underbrace{\varphi\left(\frac{x - C_i}{\sigma}\right)}_{\varphi_i(x)} = \Theta^T \Phi(x)$$

such that

$$\|f(x) - \text{NN}(x)\|_{L_2}^2 = \int_{\|x\| \leq r} (f(x) - \text{NN}(x))^2 dx \leq \varepsilon = O(N^{-\frac{1}{n}})$$

■

In conclusion, we present a comparison of key features and properties possessed by the sigmoidal NNs and by the RBF NNs.

- Both RBF and sigmoidal NNs are universal approximators.
- A RBF NN depends on the Euclidean distances between the input vector x and the centers C_i . On the other hand, a sigmoidal NN depends on the sum of the inner product of the input vector x with its synaptic weight vectors V_i and a bias θ .
- Sigmoidal NNs provide $O(N^{-\frac{1}{2}})$ rate of approximation which does not explicitly depend on the dimension of x . The rate of approximation for the RBF NNs is of order $O(N^{-\frac{1}{2n}})$, and consequently it decreases exponentially as the dimension of the input vector x increases. This phenomenon is called the “Curse of Dimensionality” (due to R. E. Bellman).
- A RBF has a local support, while a sigmoid does not. The local support implies learning and adaptation ability of RBF NNs. Sigmoidal NNs adapt but do not learn.

With the specific reference to artificial NNs in control, it is their ability to represent inherently nonlinear mappings and hence to model nonlinear dynamical systems, which is the feature to be most readily exploited in the synthesis of nonlinear controllers. This is the topic that we shall begin to address in the next section.

12.4 Adaptive Control with State Limiting Constraints

We are interested in the design of adaptive command tracking controllers for affine-in-control multi-input–multi-output (MIMO) dynamical systems in the form,

$$\dot{x} = Ax + B \Lambda(u + f(x)) + \xi(t) \quad (12.8)$$

where $x \in R^n$ is the system state vector, $u \in R^m$ is the control input, $B \in R^{n \times m}$ is a known constant matrix, $A \in R^{n \times n}$ and $\Lambda \in R^{m \times m}$ (a diagonal matrix with positive elements) are unknown constant matrices, $f(x): R^n \rightarrow R^m$ is a state-dependent (possibly nonlinear) uncertainty, and $\xi(t) \in R^n$ is a bounded time-varying unknown disturbance, whose upper bound

$$\|\xi(t)\| \leq \xi_{\max} \quad (12.9)$$

is known.

In the previous chapters, we have developed model reference adaptive control (MRAC) command tracking design methods, assuming that the matched nonlinear uncertainty admits an exact parameterization in the form $f(x) = \Theta^T \Phi(x)$, with constant unknown coefficients $\Theta \in R^{N \times m}$ and with a preselected known locally Lipschitz-continuous regressor vector $\Phi(x) \in R^N$.

In this section, we shall extend our design to nonlinear-in-parameters functions. Our main assumption here is that these uncertainties can be parameterized (i.e., approximated on a bounded closed set to within a small tolerance) using artificial NNs, whose fixed basis functions are known, (such as sigmoids with fixed inner-layer weights and thresholds or Gaussians with fixed centers).

In particular, using the universal approximation properties of artificial NNs, we shall assume that the unknown mapping $f(x): R^n \rightarrow R^m$ can be approximated/represented on a known compact set $X \subset R^n$ by an NN with N fixed neurons $\varphi_i(x)$ and using unknown ideal constant connection weights that are stored in a matrix $\Theta \in R^{N \times m}$,

$$f(x) = \Theta^T \Phi(x) + \varepsilon(x) \quad (12.10)$$

Without a loss of generality, we define the approximation set

$$X = X_R = \{x \in R^n: \|x\| \leq R\} \quad (12.11)$$

to represent a sphere of a finite and known radius R . We shall also assume that inside the sphere, the ideal (unknown to the designer) approximation can be achieved within a known approximation tolerance $\varepsilon_0 > 0$,

$$\|\varepsilon(x)\| \leq \varepsilon_0, \quad \forall x \in X_R \quad (12.12)$$

Outside of X_R , we postulate that the approximation error can be upper-bounded (norm-wise) by a known possibly unbounded positive scalar function $\varepsilon_{\max}(x)$.

$$\|\varepsilon(x)\| \leq \varepsilon_{\max}(x), \quad \forall x \notin X_R \quad (12.13)$$

The control objective is to design a state feedback MRAC system, which guarantees boundedness of all signals in the corresponding closed-loop dynamics,

while forcing the system state $x(t) \in R^n$ follows the state $x_{\text{ref}}(t) \in R^n$ of the desired exponentially stable reference model,

$$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} r(t) \quad (12.14)$$

driven by a known bounded time-varying reference command signal $r(t) \in R^m$,

$$\|r(t)\| \leq r_{\max}, \quad \forall t \geq 0 \quad (12.15)$$

whose maximum bound r_{\max} is known.

We are going to construct an adaptive command tracking controller, capable of operating in the presence of the system structured and unstructured uncertainties, where the latter are represented by: (a) the state-dependent function approximation error $\varepsilon(x) \in R^m$ in (12.12) and (b) the bounded disturbance $\xi(t) \in R^n$ in (12.9).

Let us immediately note that while the disturbance term $\xi(t)$ is uniformly bounded, the approximation error $\varepsilon(x)$ becomes bounded only if the system state $x(t)$ is located inside the sphere X_R . So, in addition to command tracking, we need a state limiter logic that would keep the system state within the approximation set X_R , or it would bring it back to X_R , if the state happens to be outside of the approximation set. This observation suggests a control law in the form,

$$\begin{aligned} u &= \underbrace{\hat{K}_x^T x - \hat{\Theta}^T \Phi(x)}_{u_x} + (1 - \mu(x)) \underbrace{\hat{K}_r^T r}_{u_r} + \mu(x) u_{\text{sl}} \\ &= u_x + (1 - \mu(x)) u_r + \mu(x) u_{\text{sl}} \end{aligned} \quad (12.16)$$

where

$$u_x = \hat{K}_x^T x - \underbrace{\hat{\Theta}^T \Phi(x)}_{\hat{f}} \quad (12.17)$$

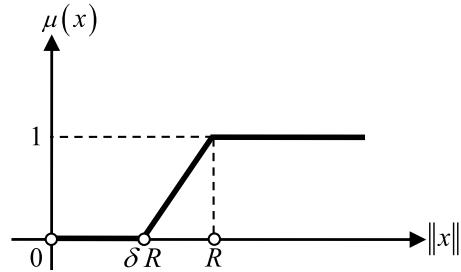
is the adaptive stabilizing term with adaptive gains $\hat{K}_x \in R^{n \times m}$ and $\hat{\Theta} \in R^{N \times m}$,

$$u_r = \hat{K}_r^T r \quad (12.18)$$

is the adaptive command tracking component with an adaptive feedforward command gain $\hat{K}_r \in R^{m \times m}$, $\mu(x)$ is the state modulation function, and u_{sl} is the state limiter.

This controller will be designed to operate as follows. The adaptive stabilizing term u_x will provide closed-loop stability for all $x \in X_R$. At the same time, the adaptive command tracking component u_r will force the system to follow commanded trajectories of the desired reference model. If the system state $x(t)$ starts outside of X_R or if the system disturbance $\xi(t)$ pushes it outside of the approximation set, then command tracking will subside and the state limiter u_{sl} will be responsible to bring x back into X_R (in finite time), where command tracking would resume.

Fig. 12.5 State modulation function



These two modes of operation, tracking and state limiting, are governed by the state limiter modulation function $\mu(x)$, which essentially “gain schedules” the controller (12.16) to smoothly transition between the adaptive tracking and the state limiting tasks.

We define the state limiter function as

$$\mu(x) = \max\left(0, \min\left(1, \frac{\|x\| - \delta R}{(1 - \delta)R}\right)\right) \quad (12.19)$$

where $0 < \delta < 1$ is a constant. A sketch of this function is shown in Fig. 12.5.

By definition,

$$\mu(x) = \begin{cases} 0, & x \in X_{\delta R} \\ 1, & x \notin X_R \end{cases} \quad (12.20)$$

and the positive constant δ defines the width of an annulus inside X_R where $0 \leq \mu(x) \leq 1$. According to (12.16) and (12.19)–(12.20), the state limiter will turn the adaptive tracking on for all $x \in X_{\delta R}$, or it will turn the state limiter on for $x \notin X_R$. In the annulus set $\delta R \leq \|x\| \leq R$, both tasks are active, with one of them fading out and the other fading in, linearly in $\|x\|$.

Our choice of the modulation function in (12.19) is by no means unique. Other definitions can easily be constructed to accomplish the gain-scheduling feature of the adaptive controller (12.16).

In order for such a control solution to exist, the model matching conditions must hold

$$\begin{aligned} A + B \Lambda K_x^T &= A_{\text{ref}} \\ B \Lambda K_r^T &= B_{\text{ref}} \end{aligned} \quad (12.21)$$

where K_x, K_r denote the ideal unknown constant feedback and feedforward gain matrices, respectively. Only existence of the ideal gains is assumed, whereas their knowledge will not be required to perform the design.

In (12.17), $\hat{f}(x) = \hat{\Theta}^T \Phi(x)$ is the function approximator. It is easy to see that the related function approximation error,

$$\Delta f(x) = \hat{f}(x) - f(x) \quad (12.22)$$

depends linearly on the parameter estimation error $\Delta\Theta = \hat{\Theta} - \Theta$.

$$\Delta f(x) = \underbrace{\left(\hat{\Theta} - \Theta \right)^T}_{\Delta\Theta} \Phi(x) - \varepsilon(x) = \Delta\Theta^T \Phi(x) - \varepsilon(x) \quad (12.23)$$

Using the model matching conditions (12.21), the open-loop system dynamics (12.8) can be written as

$$\dot{x} = A_{\text{ref}} x + B_{\text{ref}} r + B \Lambda \left(u - K_x^T x - K_r^T r + f(x) \right) + \xi(t) \quad (12.24)$$

Substituting (12.16) into (12.24), yields

$$\begin{aligned} \dot{x} &= A_{\text{ref}} x + B_{\text{ref}} r + B \Lambda \left(u_x + (1 - \mu) u_r + \mu u_{\text{sl}} - K_x^T x - K_r^T r + f \right) + \xi \\ &= A_{\text{ref}} x + B_{\text{ref}} r \\ &\quad + B \Lambda \left(u_x - K_x^T x + f + (1 - \mu) \left(u_r - K_r^T r \right) + \mu \left(u_{\text{sl}} - K_r^T r \right) \right) + \xi \end{aligned} \quad (12.25)$$

With (12.17) and (12.18), we get

$$\begin{aligned} \dot{x} &= A_{\text{ref}} x + B_{\text{ref}} r + \xi \\ &\quad + B \Lambda \left(\underbrace{\left(\hat{K}_x^T - K_x^T \right)}_{\Delta K_x} x - \underbrace{\left(\hat{\Theta}^T - \Theta^T \right)}_{\Delta\Theta} \Phi + \varepsilon + (1 - \mu) \underbrace{\left(\hat{K}_r^T - K_r^T \right)}_{\Delta K_r} r \right) \\ &\quad + B \Lambda \mu \left(u_{\text{sl}} - K_r^T r \right) \end{aligned} \quad (12.26)$$

or, equivalently

$$\begin{aligned} \dot{x} &= A_{\text{ref}} x + B_{\text{ref}} r + \xi \\ &\quad + B \Lambda \left(\Delta K_x^T x - \Delta\Theta^T \Phi + \varepsilon + (1 - \mu) \Delta K_r^T r + \mu \left(u_{\text{sl}} - K_r^T r \right) \right) \end{aligned} \quad (12.27)$$

where

$$\Delta K_x = \hat{K}_x - K_x, \quad \Delta K_r = \hat{K}_r - K_r, \quad \Delta\Theta = \hat{\Theta} - \Theta \quad (12.28)$$

are the parameter estimation errors. Let,

$$e = x - x_{\text{ref}} \quad (12.29)$$

denote the state tracking error. Subtracting (12.14) from (12.27), gives the state tracking error dynamics,

$$\begin{aligned} \dot{e} &= A_{\text{ref}} e + \xi \\ &+ B \Lambda \left(\Delta K_x^T x - \Delta \Theta^T \Phi + \varepsilon + (1 - \mu) \Delta K_r^T r + \mu (u_{\text{sl}} - K_r^T r) \right) \end{aligned} \quad (12.30)$$

We introduce matrix $P \in R^{n \times n}$ to represent the unique positive-definite symmetric solution of the algebraic Lyapunov equation,

$$P A_{\text{ref}} + A_{\text{ref}}^T P = -Q, \quad Q = Q^T > 0 \quad (12.31)$$

and consider a quadratic radially unbounded Lyapunov function candidate in the form,

$$\begin{aligned} V(e, \Delta K_x, \Delta K_r, \Delta \Theta) &= e^T P e \\ &+ \text{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \Delta K_x + \Delta K_r^T \Gamma_r^{-1} \Delta K_r + \Delta \Theta^T \Gamma_\Theta^{-1} \Delta \Theta \right] \Lambda \right) \end{aligned} \quad (12.32)$$

where $\Gamma_x = \Gamma_x^T > 0$, $\Gamma_r = \Gamma_r^T > 0$, $\Gamma_\Theta = \Gamma_\Theta^T > 0$ are the rates of adaptation. The time derivative of V , along the trajectories of the error dynamics (12.30), is given by

$$\begin{aligned} \dot{V} &= \dot{e}^T P e + e^T P \dot{e} \\ &+ 2 \text{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \right] \Lambda \right) \\ &= -e^T Q_{\text{ref}} e + 2e^T P \xi \\ &+ 2e^T P B \Lambda \left(\Delta K_x^T x - \Delta \Theta^T \Phi + \varepsilon + (1 - \mu) \Delta K_r^T r + \mu (u_{\text{sl}} - K_r^T r) \right) \\ &+ 2 \text{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \right] \Lambda \right) \end{aligned} \quad (12.33)$$

Regrouping terms further yields

$$\begin{aligned} \dot{V} &= -e^T Q e + 2e^T P \xi + 2e^T P B \Lambda \left(\mu (u_{\text{sl}} - K_r^T r) + \varepsilon \right) \\ &+ 2 \left[e^T P B \Lambda \Delta K_x^T x + \text{tr} \left(\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x \Lambda \right) \right] \\ &+ 2 \left[(1 - \mu) e^T P B \Lambda \Delta K_r^T r + \text{tr} \left(\Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r \Lambda \right) \right] \\ &+ 2 \left[-e^T P B \Lambda \Delta \Theta^T \Phi + 2 \text{tr} \left(\Delta \Theta^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \Lambda \right) \right] \end{aligned} \quad (12.34)$$

Via the vector trace identity $a^T b = \text{tr}(b a^T)$, which is valid for any two column vectors a and b , we obtain

$$\begin{aligned} \underbrace{e^T P B \Lambda}_{a^T} \underbrace{\Delta K_x^T x}_{b} &= \text{tr} \left(\underbrace{\Delta K_x^T x}_{b} \underbrace{e^T P B \Lambda}_{a^T} \right) \\ \underbrace{e^T P B \Lambda}_{a^T} \underbrace{\Delta K_r^T r}_{b} &= \text{tr} \left(\underbrace{\Delta K_r^T r}_{b} \underbrace{e^T P B \Lambda}_{a^T} \right) \\ \underbrace{e^T P B \Lambda}_{a^T} \underbrace{\Delta \Theta^T \Phi(x)}_{b} &= \text{tr} \left(\underbrace{\Delta \Theta^T \Phi}_{b} \underbrace{e^T P B \Lambda}_{a^T} \right) \end{aligned} \quad (12.35)$$

Substituting (12.35) into (12.34), results in

$$\begin{aligned} \dot{V} &= -e^T Q e + 2e^T P \xi + 2e^T P B \Lambda \left(\mu \left(u_{\text{sl}} - K_r^T r \right) + \varepsilon \right) \\ &\quad + 2 \text{tr} \left(\Delta K_x^T \left[\Gamma_x^{-1} \dot{K}_x + x e^T P B \right] \Lambda \right) \\ &\quad + 2 \text{tr} \left(\Delta K_r^T \left[\Gamma_r^{-1} \dot{K}_r + (1 - \mu) r e^T P B \right] \Lambda \right) \\ &\quad + 2 \text{tr} \left(\Delta \Theta^T \left[\Gamma_\Theta^{-1} \dot{\Theta} - \Phi e^T P B \right] \Lambda \right) \end{aligned} \quad (12.36)$$

In order to keep the adaptive gains $\hat{K}_x, \hat{K}_r, \hat{\Theta}$ uniformly bounded, we shall employ projection-based adaptive laws (Sects. 11.3–11.4) in the form,

$$\begin{aligned} \dot{\hat{K}}_x &= \text{Proj} \left(\hat{K}_x, -\Gamma_x x e^T P B \right) \\ \dot{\hat{K}}_r &= \text{Proj} \left(\hat{K}_r, -(1 - \mu) \Gamma_r r e^T P B \right) \\ \dot{\hat{\Theta}} &= \text{Proj} \left(\hat{\Theta}, \Gamma_\Theta \Phi e^T P B \right) \end{aligned} \quad (12.37)$$

where $\text{Proj}(\Theta, Y)$ is the Projection Operator, which maps two $(n \times N)$ matrices, $\Omega = [\vec{\theta}_1 \dots \vec{\theta}_N] \in R^{n \times N}$ and $Y = [\vec{y}_1 \dots \vec{y}_N] \in R^{n \times N}$, into a $(n \times N)$ matrix, denoted by $\text{Proj}(\Omega, Y)$. The operator is defined column-wise,

$$\text{Proj}(\Omega, Y) = \left(\text{Proj} \left(\vec{\theta}_1, \vec{y}_1 \right) \dots \text{Proj} \left(\vec{\theta}_N, \vec{y}_N \right) \right) \quad (12.38)$$

and its vector column components are

$$\text{Proj} \left(\vec{\theta}_j, \vec{y}_j \right) = \begin{cases} y - \frac{\Gamma \nabla f_j (\nabla f_j)^T}{(\nabla f_j)^T \Gamma \nabla f_j} \vec{y}_j f_j, & \text{if } [f_j > 0 \wedge (\vec{y}_j^T \nabla f_j) > 0] \\ y, & \text{if not} \end{cases} \quad (12.39)$$

where $f(\vec{\theta}_j) : R^n \rightarrow R$ is a convex function that defines the desired parameter domain. Given θ_j^{\max} —the maximum allowable magnitude of the column vector $\vec{\theta}_j$, and a small constant $\varepsilon_j > 0$, the convex function is

$$f(\theta_j) = \frac{(1 + \varepsilon_j) \|\theta_j\|^2 - (\theta_j^{\max})^2}{\varepsilon_j (\theta_j^{\max})^2} \quad (12.40)$$

With the adaptive laws (12.37) and because of the previously established convex properties of the Projection Operator, one can show that the derivative of the Lyapunov function (12.36) satisfies the following inequality:

$$\dot{V} \leq -e^T Q e + 2 e^T P \xi + 2 e^T P B \Lambda \left(\mu \left(u_{\text{sl}} - K_r^T r \right) + \varepsilon \right) \quad (12.41)$$

In order to eventually prove stability and bounded command tracking, we need to analyze if \dot{V} can be made non-positive outside of a compact set. Toward that end, let us suppose that $x \notin X_R$. Then $\mu(x) = 1$ and (12.41) becomes

$$\begin{aligned} \dot{V} &\leq -e^T Q e + 2 e^T P \xi + 2 e^T P B \Lambda \left(u_{\text{sl}} - K_r^T r + \varepsilon \right) \\ &\leq -\lambda_{\min}(Q) \|e\|^2 + 2 \|e\| \lambda_{\max}(P) \xi_{\max} + 2 e^T P B \Lambda \left(u_{\text{sl}} - K_r^T r + \varepsilon \right) \end{aligned} \quad (12.42)$$

In order to make the right-hand side of (12.42) non-positive, we choose the state limiting control u_{sl} in the form

$$u_{\text{sl}} = -k_{\text{sl}}(x) \operatorname{sgn}(B^T P e) \quad (12.43)$$

where $k_{\text{sl}}(x) > 0$ represents the state limiter gain, and the sign function is understood component-wise. Then,

$$e^T P B \Lambda u_{\text{sl}} = \sum_{i=1}^m \left(e^T P B \right)_i \lambda_i u_{\text{sl}i} = -k_{\text{sl}}(x) \sum_{i=1}^m \left| e^T P B \right|_i \lambda_i \quad (12.44)$$

and with (12.43) inserted into (12.42), we get

$$\begin{aligned}
\dot{V} &\leq -\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\lambda_{\max}(P)\xi_{\max} \\
&\quad - 2k_{\text{sl}}(x) \sum_{i=1}^m \left| e^T P B \right|_i \lambda_i + e^T P B \Lambda \left(-K_r^T r + \varepsilon \right) \\
&= -\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\lambda_{\max}(P)\xi_{\max} \\
&\quad - 2 \sum_{i=1}^m \left| e^T P B \right|_i \lambda_i \left(k_{\text{sl}}(x) - \text{sgn} \left(\left(e^T P B \right)_i \right) \left(K_r^T r - \varepsilon \right) \right)
\end{aligned} \tag{12.45}$$

If we now choose the state limiter gain to be large enough,

$$k_{\text{sl}}(x) = K_r \max r_{\max} + \varepsilon_{\max}(x) \tag{12.46}$$

where $K_r \max \geq \|K_r\|$ and $r_{\max} = \max_{t \geq 0} \|r(t)\|$, then

$$\begin{aligned}
\dot{V} &\leq -\lambda_{\min}(Q)\|e\|^2 + 2\|e\|\lambda_{\max}(P)\xi_{\max} \\
&= -\lambda_{\min}(Q)\|e\| \left(\|e\| - 2 \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)} \xi_{\max} \right) < 0
\end{aligned} \tag{12.47}$$

outside of the compact set

$$E_0 = \left\{ e \in R^n : \|e\| \leq 2 \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)} \xi_{\max} = e_0 \right\} \tag{12.48}$$

Therefore, $e(t)$ enters a larger compact set $\tilde{E}_0 \supset E_0$, in finite time T , [7–9]. Moreover, for all $t \geq T$, there must exist a positive constant \bar{e}_0 , such that

$$\bar{e}_0 \geq \|e(t)\| = \|x(t) - x_{\text{ref}}(t)\| \geq \|x(t)\| - \|x_{\text{ref}}(t)\| \tag{12.49}$$

Hence,

$$\|x(t)\| \leq \bar{e}_0 + \|x_{\text{ref}}(t)\| \leq \bar{e}_0 + x_{\text{ref max}} \tag{12.50}$$

where the upper bound $x_{\text{ref max}}$ can be explicitly computed based on r_{\max} and the properties of the reference model (12.14). So, if we choose the approximation set X_R to be large enough,

$$R > \bar{e}_0 + x_{\text{ref max}} \tag{12.51}$$

then for all $t \geq T$,

$$\|x(t)\| \leq R \tag{12.52}$$

that is, the system state $x(t)$ enters X_R in finite time T and remains there afterward.

Inside the set X_R , the state modulation function is zero and the approximation error $\varepsilon(x)$ becomes small. Hence, the Lyapunov function time derivative from (12.41) can be upper-bounded as

$$\begin{aligned}\dot{V} &\leq -e^T Q e + 2e^T P \xi + 2e^T P B \Lambda \varepsilon \leq -\lambda_{\min}(Q) \|e\|^2 \\ &+ 2\|e\|\lambda_{\max}(P)(\xi_{\max} + \|B\|\Lambda_{\max}\varepsilon_0) \\ &= -\lambda_{\min}(Q)\|e\|\left(\|e\| - 2\frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)}(\xi_{\max} + \|B\|\Lambda_{\max}\varepsilon_0)\right)\end{aligned}\quad (12.53)$$

Consequently, $\dot{V} < 0$ outside of the compact set

$$E_1 = \left\{ e \in R^n : \|e\| \leq 2 \frac{\lambda_{\max}(P)}{\lambda_{\min}(Q)} (\xi_{\max} + \|B\|\Lambda_{\max}\varepsilon_0) = \bar{e}_1 \right\} \quad (12.54)$$

Hence, $e(t)$ enters a compact set $\tilde{E}_1 \supset E_1$ in finite time T_1 , [7–9], where it will remain afterward. Similarly to (12.49) and (12.50), we get an upper bound,

$$\|x(t)\| \leq \bar{e}_1 + x_{\text{ref max}} \quad (12.55)$$

for some positive constant $\bar{e}_1 \geq \bar{e}_0$. In order to ensure that $x(t)$ remains inside X_R , it is sufficient to strengthen the inequality (12.51) and assume

$$R > \bar{e}_1 + x_{\text{ref max}} \quad (12.56)$$

As we have already mentioned, the adaptive parameters will remain uniformly ultimately bounded (UUB). This property is due to the convexity of the Projection Operator (Sect. 11.3, Lemmas 11.3, 11.4). Consequently, all trajectories of the closed-loop system (12.8), (12.16) and (12.37) are UUB. Moreover, the tracking error $e = x - x_{\text{ref}}$ enters a neighborhood of the origin, in finite time. The radius of this neighborhood (i.e., the tracking error ultimate bound) is determined by the minimum level set of the Lyapunov function V , which contains the set

$$\begin{aligned}E = \{e \in R^n : \|e\| \leq \bar{e}_1\} \\ \times \left\{ \hat{K}_x \in R^{n \times m} : \left\| \left(\hat{K}_x \right)_j \right\| \leq \left(\hat{K}_x^{\max} \right)_j, \quad 1 \leq j \leq m \right\} \\ \times \left\{ \hat{K}_r \in R^{m \times m} : \left\| \left(\hat{K}_r \right)_j \right\| \leq \left(\hat{K}_r^{\max} \right)_j, \quad 1 \leq j \leq m \right\} \\ \times \left\{ \hat{\theta} \in R^{N \times m} : \left\| \left(\hat{\theta} \right)_j \right\| \leq \left(\hat{\theta}^{\max} \right)_j, \quad 1 \leq j \leq m \right\}\end{aligned}\quad (12.57)$$

outside of which $\dot{V} \leq 0$.

This argument completes the design and analysis of the MRAC controller with state limiting constraints for MIMO dynamics with both structured and unstructured uncertainties. We summarize the derived design equations in Table 12.1.

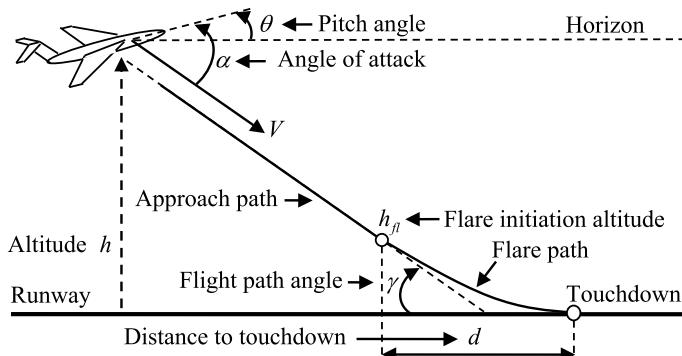
Next, as an illustrative example, we are going to design an adaptive automatic landing system for a generic medium-size transport aircraft.

Table 12.1 Projection-based MRAC design with state limiter constraints

Open-loop plant	$\dot{x} = Ax + B \Lambda(u + f(x)) + \xi(t)$
Reference model	$\dot{x}_{\text{ref}} = A_{\text{ref}}x_{\text{ref}} + B_{\text{ref}}r(t)$
State tracking error	$e = x - x_{\text{ref}}$
Lyapunov equation for adaptive laws	$P A_{\text{ref}} + A_{\text{ref}}^T P = -Q$
Total control input	$u = u_x + (1 - \mu(x))u_r + \mu(x)u_{\text{sl}}$
State modulation function	$\mu(x) = \max(0, \min(1, \frac{\ x\ - \delta R}{(1-\delta)R}))$
Adaptive stabilizing term	$u_x = \hat{K}_x^T x - \hat{\Theta}^T \Phi(x)$
Adaptive tracking term	$u_r = \hat{K}_r^T r$
State limiter	$u_{\text{sl}} = -k_{\text{sl}}(x) \text{sgn}(B^T P e)$
State limiter gain	$k_{\text{sl}}(x) = K_r \max(r_{\text{max}} + \varepsilon_{\text{max}}(x))$
MRAC laws with Projection operator	$\dot{\hat{K}}_x = \text{Proj}\left(\hat{K}_x, -\Gamma_x x e^T P B\right)$ $\dot{\hat{K}}_r = \text{Proj}\left(\hat{K}_r, (\mu(x) - 1)\Gamma_r r e^T P B\right)$ $\dot{\hat{\Theta}} = \text{Proj}\left(\hat{\Theta}, \Gamma_\Theta \Phi e^T P B\right)$

Example 12.3 Automatic Landing System for a Medium-Size Transport Aircraft
 Modern transport aircraft is equipped with automatic landing systems whose sole purpose is to fly the vehicle along the desired trajectory, all the way until a predetermined touchdown point on a runway (Fig. 12.6).

On final approach to a landing, an aircraft would extend its wing leading edges (slats), move wing trailing edges (flaps) down, and deploy its landing gear. As a result, the vehicle aerodynamic drag increases and the airspeed decreases. With the flaps and slats extended, the aircraft wing would be optimized (per design) to produce

**Fig. 12.6** Aircraft on final approach to landing

a sufficient lift force and to enable a low-speed landing, with a gentle touchdown at the designated runway touchdown point.

For clarity, we assume that the runway is parallel to the horizon and that the aircraft undergoes vertical and longitudinal motion only, that is the vehicle can change its vertical and forward velocity components, and it can also pitch up or down. In this case, the vehicle primary control inputs are engine thrust (δ_{th} , %) and elevators (collectively movable tail surfaces, δ_e , °). The regulated outputs are represented by the true airspeed V (ft/s) and altitude above the runway (h , ft). From Fig. 12.6, it is not difficult to see that the aircraft angle of attack α (rad), the pitch angle (θ , °), and the flight path angle γ (rad) satisfy the following equality:

$$\alpha = \theta - \gamma$$

Another important relation exists between the vehicle rate of climb \dot{h} (ft/s), the runway velocity \dot{d} , the airspeed V , and the flight path angle γ . From Fig. 12.6 (for a small flight path angle), we get

$$\begin{aligned}\dot{h} &= V \sin \gamma \approx V \gamma \\ \dot{d} &= V \cos \gamma \approx V\end{aligned}$$

The desired trajectory for the aircraft to follow consists of two segments: (a) straight line approach and (b) flare. Typical approach angles range between negative 2 and 3° of γ , with the desired airspeed of 140–160 knots.

During the approach phase, the aircraft is commanded to fly a constant airspeed and a constant flight path angle. On the other hand, the main purpose of the flare is to slow down the aircraft rate of descent (called the “sink rate”), and to make a smooth transition from the selected approach glide slope (i.e., the flight path angle γ) to a shallow angle, at an altitude of approximately 50–65 ft above the runway.

A moderate flare can be described by a linear first-order differential equation such as,

$$\dot{h} = -\frac{1}{\tau_h} h, \quad h(t_{fl}) = h_{fl}$$

where (t_{fl}, h_{fl}) are the flare initiation time and altitude, respectively. Also, τ_h is a positive time constant. The flare initiation altitude h_{fl} and the time constant τ_h can be chosen such that the vehicle would make a smooth transition from approach to flare, and it would touch down within a predetermined distance along the runway. Let us formally define these two requirements.

A smooth transition from the approach phase ($V_{cmd} = V_0$, $\gamma_{cmd} = \gamma_0$) to flare implies that at $t = t_{fl}$ the following relation must take place:

$$\dot{h}(t_{fl}) = V\gamma(t_{fl}) \approx V_{cmd} \gamma_{cmd} = -\frac{1}{\tau_h} h_{fl}$$

In addition, we impose a restriction on the runway distance traveled in $4 \tau_h$ seconds from the start of the flare maneuver,

$$\int_{t_f}^{t_f+4\tau_h} V(t) dt = d$$

where d is the desired distance to touchdown (Fig. 12.6). Assuming constant airspeed throughout the entire maneuver, $V(t) \approx V_{cmd}$, gives

$$\int_{t_f}^{t_f+4\tau_h} V(t) dt \approx V_{cmd} 4 \tau_h = d$$

Then, the flare time constant is

$$\tau_h = \frac{d}{4 V_{cmd}}$$

and the flare initiation altitude can be computed as

$$h_f = -\tau_h V_{cmd} \gamma_{cmd} = -\frac{d}{4} \gamma_{cmd}$$

For simulation purposes, we consider a generic mid-size transport aircraft flying wings-level at an altitude of $h_0 = 300$ ft above ground, with its landing gear down and with flaps/slats extended. The vehicle true airspeed is $V_0 = 250$ ft/s. The corresponding longitudinal linear (nominal) dynamics are of the form

$$\underbrace{\begin{pmatrix} \dot{V} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} -0.038 & 18.984 & 0 & -32.174 & 0 \\ -0.001 & -0.632 & 1 & 0 & 0 \\ 0 & -0.759 & -0.518 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -250 & 0 & 250 & 0 \end{pmatrix}}_A \underbrace{\begin{pmatrix} V \\ \alpha \\ q \\ \theta \\ h \end{pmatrix}}_x + \underbrace{\begin{pmatrix} 10.1 & 0 \\ 0 & 0.00044 \\ 0.025 & -0.011 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}}_B \underbrace{\begin{pmatrix} \delta_{th} \\ \delta_e \end{pmatrix}}_u \Leftrightarrow \boxed{\dot{x} = Ax + Bu}$$

We wish to emphasize that our model represents a generic mid-size aircraft, and that the linear data are selected for the purposes of design, analysis, and simulation [10, p. 300].

As the aircraft approaches the runway, it will experience a significant increase in its aerodynamic lift force and the pitching moment. This phenomenon is called the “ground effect”. Flying in close proximity to the ground drastically changes the airflow beneath and past the airplane. As a result, the ground effect tends to make the vehicle float along the runway.

In order to properly account for the ground effect, we need to modify the aircraft linear dynamics. The vehicle aerodynamic forces and moments depend on the relative motion of the aircraft with respect to the atmosphere. In our example, dynamics of these forces and moments are defined by the first three equations. The ground effect induces a change in the vertical (updraft) linear displacement of the air mass and so, the aircraft aerodynamic forces and moments depend on the difference $\alpha - \alpha_g(h)$ between the aircraft angle of attack α and the angle of attack induced by the vertical updraft $\alpha_g(h)$, which in turn represents a uniformly bounded function of the ground proximity (i.e., altitude) h . So, the ground effect phenomenon can be embedded into the linear model as follows:

$$\underbrace{\begin{pmatrix} \dot{V} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} -0.038 & 18.984 & 0 & -32.174 & 0 \\ -0.001 & -0.632 & 1 & 0 & 0 \\ 0 & -0.759 & -0.518 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -250 & 0 & 250 & 0 \end{pmatrix}}_A \underbrace{\begin{pmatrix} V \\ \alpha \\ q \\ \theta \\ h \end{pmatrix}}_x + \underbrace{\begin{pmatrix} 10.1 & 0 \\ 0 & 0.00044 \\ 0.025 & -0.011 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}}_B \underbrace{\begin{pmatrix} \delta_{th} \\ \delta_e \end{pmatrix}}_u + \underbrace{\begin{pmatrix} -18.984 \\ 0.632 \\ 0.759 \\ 0 \\ 0 \end{pmatrix}}_{B_g} \alpha_g(h)$$

or, equivalently

$$\dot{x} = Ax + Bu + B_g \alpha_g(h)$$

where we have added an extra term $B_g \alpha_g(h)$, with a constant vector B_g , whose first three components are equal to the opposite of the corresponding values in the second column of A . This modification reflects our observation that the first three equations in the aircraft dynamics depend on the relative (with respect to the air mass) angle of attack $\alpha - \alpha_g(h)$.

It is not difficult to see that B_g can be reconstructed as a linear combination of the columns in B .

$$B \underbrace{\begin{pmatrix} -1.8796 \\ -73.2718 \end{pmatrix}}_{\theta_g} = B_g$$

In other words, the ground effect represents a matched uncertainty, and the resulting model takes the form of (12.8).

$$\dot{x} = A x + B (u + \theta_g \alpha_g(h))$$

Let us make a quick remark about the ground effect matching condition: It is not a requirement for our design. Since $\alpha_g(h)$ is a uniformly bounded function of h , it can be treated similar to the bounded disturbance $\xi(t)$ in (12.8), as long as we can ensure that h is bounded. So, the ground effect unmatched effects on the aircraft dynamics can also be mitigated (see Exercise 12.3).

Continuing on, we define the system regulated output to consist of the aircraft true airspeed and altitude (same as the aircraft height above the runway).

$$y = \begin{pmatrix} V \\ h \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_C x = C x$$

Accurate aerodynamic data that describe the ground effect are often not available or highly uncertain, yet their undesirable influences on the aircraft landing performance must be taken into account. In order to mitigate these uncertainties, we shall design a (Robust + Adaptive) automatic landing flight controller, with predictable and quantifiable landing performance characteristics. Specifically, we are going to design a robust adaptive controller to simultaneously track commanded airspeed V_{cmd} and commanded altitude h_{cmd} . These two external commands are grouped into the external vector signal,

$$r = (V_{\text{cmd}} \ h_{\text{cmd}})^T$$

Our selection of these two specific commands will enable automatic steering of the aircraft along a given flight path, all the way to a designated touchdown point on the runway. So, our main control goal is to design u to force y to follow the external vector signal r , in the presence of unknown ground effects.

Beginning with the design of a baseline controller for automatic landing, we use the aircraft model without the ground effect uncertainty and employ the familiar LQR method. The baseline control input is

$$u_{\text{bl}} = K_x^T x + K_r^T r$$

where $K_x \in R^{5 \times 2}$ and $K_r \in R^{2 \times 2}$ are the baseline feedback and feedforward gain matrices, respectively. These gains can be calculated as follows. We choose,

$$Q_{\text{lqr}} = \text{diag}(0.2 \ 0 \ 0 \ 0 \ 1), \quad R_{\text{lqr}} = \text{diag}(10 \ 10)$$

compute K_x using the LQR method,

$$K_x^T = - \begin{pmatrix} 0.1173 & -89.1740 & 42.8761 & 140.0007 & 0.2340 \\ 0.0186 & -40.6065 & 4.3798 & 58.6016 & 0.2127 \end{pmatrix}$$

form the reference (nominal closed-loop) matrix,

$$A_{\text{ref}} = A + B K_x^T$$

and then, determine K_r such that the closed-loop baseline system DC gain, from the commanded input r to the regulated output y , is the (2×2) identity matrix.

$$\left[\text{DC Gain} = -C A_{\text{ref}}^{-1} B K_r^T = I_{2 \times 2} \right] \Rightarrow \left[K_r^T = - \left(C A_{\text{ref}}^{-1} B \right)^{-1} \right]$$

This gives,

$$K_r^T = \begin{pmatrix} -0.7753 & 0.8531 \\ 0.2340 & 0.2127 \end{pmatrix}$$

and the reference model dynamics, as in (12.14),

$$\dot{x}_{\text{ref}} = \underbrace{\left(A + B K_x^T \right)}_{A_{\text{ref}}} x_{\text{ref}} + \underbrace{\left(B K_r^T \right)}_{B_{\text{ref}}} r$$

whose eigenvalues are shown in Table 12.2.

We now turn our attention to the definition of the desired altitude profile. Given the approach airspeed $V_0 = 250$ (ft/s) and the target glide slope $\gamma_0 = -2.5$ ($^\circ$), we set the runway distance at $d = 3000$ (ft), compute the corresponding flare time constant,

$$\tau_h = \frac{d}{4 V_0} = 3.0 \text{ (s)}$$

and define the flare initiation altitude.

$$h_{fl} = -\frac{d}{4} \gamma_0 = 32.7249 \text{ (ft)}$$

Table 12.2 Reference model eigenvalues in Example 12.3

Eigenvalue	Damping, n/d	Frequency, rad/s
$-0.647 \pm 1.03 j$	0.531	1.22
$-0.529 \pm 0.158 j$	0.958	0.552
-1.39	1.0	1.39

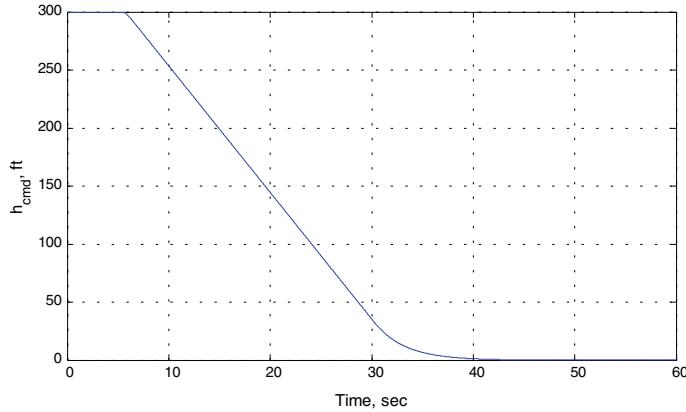


Fig. 12.7 Approach, flare, and landing altitude profile in Example 12.3

Then, we compute the commanded altitude trajectory, starting from the initial altitude $h_0 = 300$ (ft) and continuing all the way down to the runway touchdown point.

$$h_{\text{cmd}}(t) = \begin{cases} h_0, & \text{if } 0 \leq t \leq 1 \\ h_0 + V_0 \gamma_0 (t - 1), & \text{if } h_{\text{cmd}} > h_{\text{fl}}, t > 1 \\ e^{-\frac{1}{\tau_{\text{fl}}}(t-1-t_{\text{fl}})} h_{\text{fl}}, & \text{if } h_{\text{cmd}} \leq h_{\text{fl}}, t > 1 \end{cases}$$

The resulting altitude command profile is shown in Fig. 12.7.

Note that during the first one second of flight, we set the altitude command constant. This will enable a smooth initiation of the landing sequence.

With the baseline controller turned on and without the ground effect, the baseline closed-loop system tracking performance is satisfactory (Fig. 12.8).

Required for the baseline landing maneuver, the elevator and thrust values are very benign (Fig. 12.9).

For pure academic purposes, we shall use the following equation to emulate the ground effect:

$$\alpha_g(h) = -0.0698 (1 - \tanh(0.1(h - 60)))$$

The ground effect equation is plotted in Fig. 12.10.

As seen from the plot, the ground effect contributes to as much as 8° of angle-of-attack change (negative), as the aircraft approaches the runway.

Turning the ground effect on, while using only the baseline controller, results in a significant degradation of the aircraft landing performance (Fig. 12.11).

The simulation data show that while operating under the baseline controller only and in the presence of the ground effect, the vehicle floats along the runway, while its airspeed increases and deviates from its commanded value.

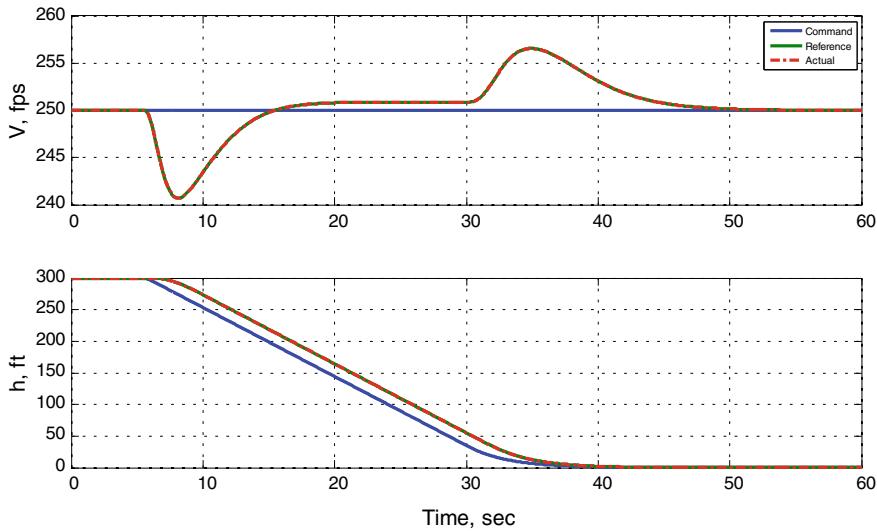


Fig. 12.8 Baseline closed-loop system performance (no ground effect) in Example 12.3

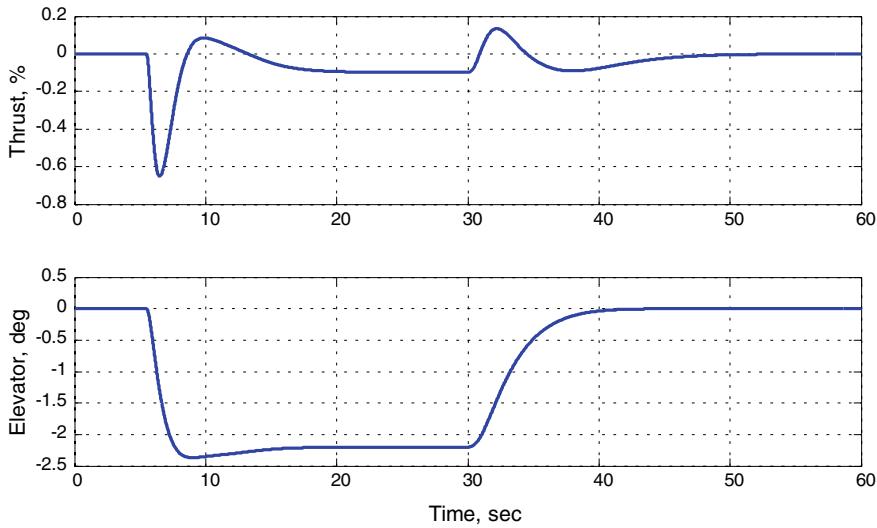


Fig. 12.9 Elevator and thrust data during baseline landing (no ground effect) in Example 12.3

The corresponding elevator and thrust control inputs remain within reasonable limits (Fig. 12.12).

So, the baseline system attempts to counteract the unknown ground effect by reducing thrust to keep the speed down and by moving the elevator trailing edge up

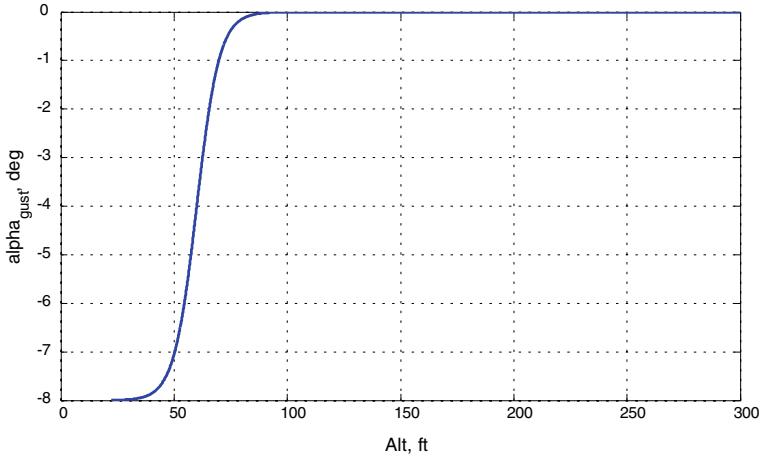


Fig. 12.10 Incremental angle-of-attack data due to ground effect in Example 12.3

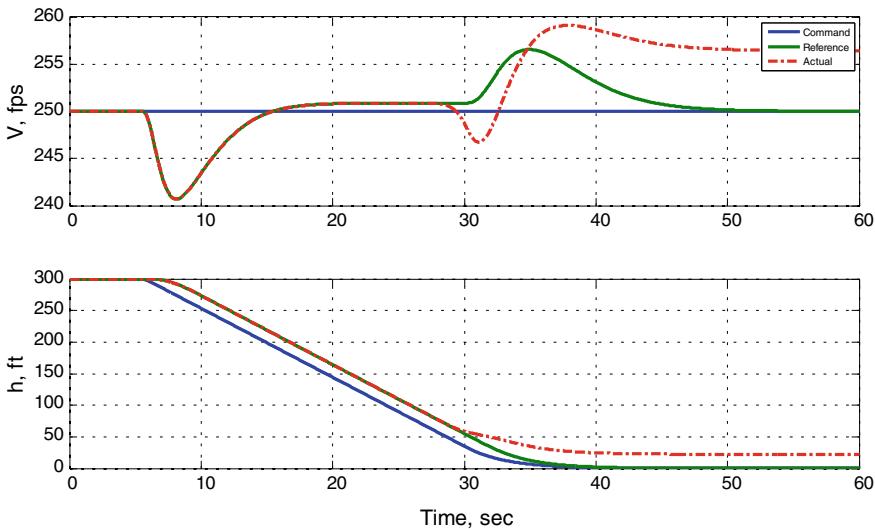


Fig. 12.11 Baseline closed-loop performance during landing with ground effect in Example 12.3

to stabilize the aircraft pitching motion. Nevertheless, the baseline controller fails in the sense that overall, the vehicle landing performance is clearly unacceptable.

Next, we design an adaptive augmentation to help the baseline system cope with the ground effect-induced unknown effects. First, we choose the regressor vector with five altitude-dependent RBFs and with a single constant bias.

$$\Phi = (\Phi_1(h) \ \Phi_2(h) \ \Phi_3(h) \ \Phi_4(h) \ \Phi_5(h) \ 1)^T$$

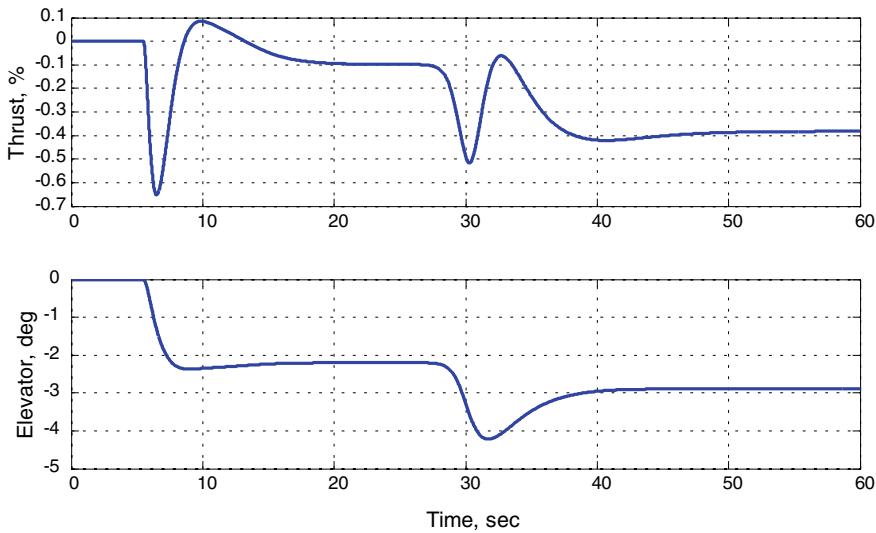


Fig. 12.12 Elevator and thrust data during baseline landing with ground effect in Example 12.3

The selected RBFs are uniformly distributed on the altitude interval $[-20, 60]$, with 20 ft separation from each other. All five RBFs have the same input scaling

$$\Phi_i(h) = \exp(-0.0056(h - h_i)^2), \quad i = 1, \dots, 5$$

When plotted versus altitude, these functions give a homogeneous coverage of the altitude range, where the ground effect is prevalent (Fig. 12.13).

Using the selected regressor vector Φ , it is possible to closely approximate the ground effect-induced angle-of-attack function $\alpha_g(h)$ on the interval of interest (Exercise 12.3).

To design an adaptive augmentation, we choose

$$Q = \text{diag}(1 \ 0 \ 1 \ 0 \ 0)$$

to solve the algebraic Lyapunov equation (12.31) for P , select rates of adaptation,

$$\Gamma_x = \Gamma_r = 0, \quad \Gamma_\Theta = 20 I_{6 \times 6}$$

and form the adaptive laws per Table 12.1.

After several design trials, we have decided to set the modulation function $\mu(x)$ to zero, since its contribution to improving landing performance is negligible (in this case).

Total control is defined as an adaptive augmentation of the baseline LQR system,

$$u = u_{\text{bl}} + u_{\text{ad}}$$

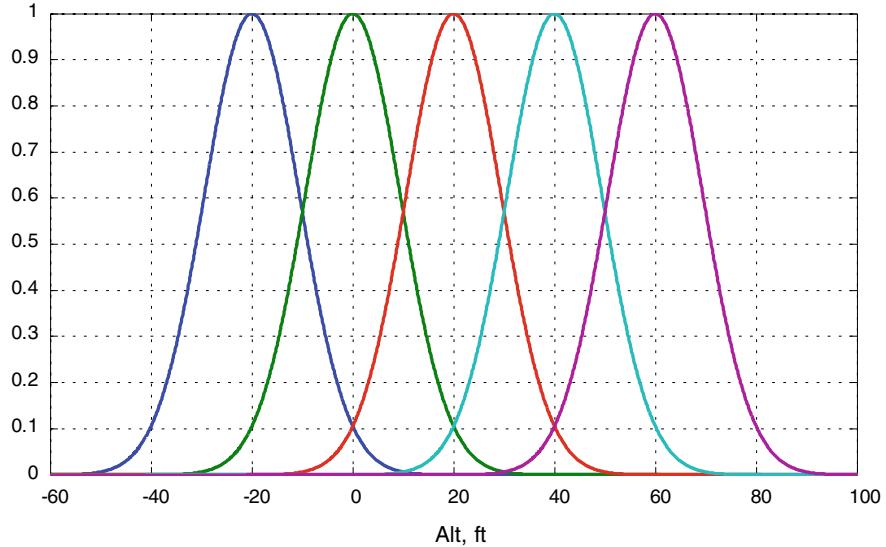


Fig. 12.13 RBF selection for adaptive control design in Example 12.3

where

$$u_{\text{ad}} = -\hat{\Theta}^T \Phi(h)$$

represents the adaptive component. This is a slight deviation from the design equations in Table 12.1, where we have an adaptive controller without a baseline system (Exercise 12.2).

With the (Baseline + Adaptive) controller turned on and in the presence of the unknown ground effect, the system closed-loop performance is well recovered to that of the desired baseline (Fig. 12.14).

In fact, the data are almost indistinguishable from the baseline tracking (Fig. 12.8). However, once the aircraft descents below 60 ft, where the ground effect is active, the required control inputs (Fig. 12.15) differ from the baseline data (Fig. 12.9).

Yet, all controls remain smooth and reside within practical limits. In addition, the adaptive augmentation provides a sufficiently close estimate of the ground effect. This “bonus” outcome can be attributed to the fact that the ground effect persistently excites the vehicle dynamics, and as a result, the adaptive parameters converge to their constant unknown values (Exercise 12.3). ■

In conclusion, we note that the adaptive feedback/feedforward design method from Table 12.1 can be modified to incorporate a robust baseline controller with proportional and integral feedback (see Exercise 12.4). This would eliminate feed-forward connections, which in its own right may become a desirable feature or even a requirement in some applications.

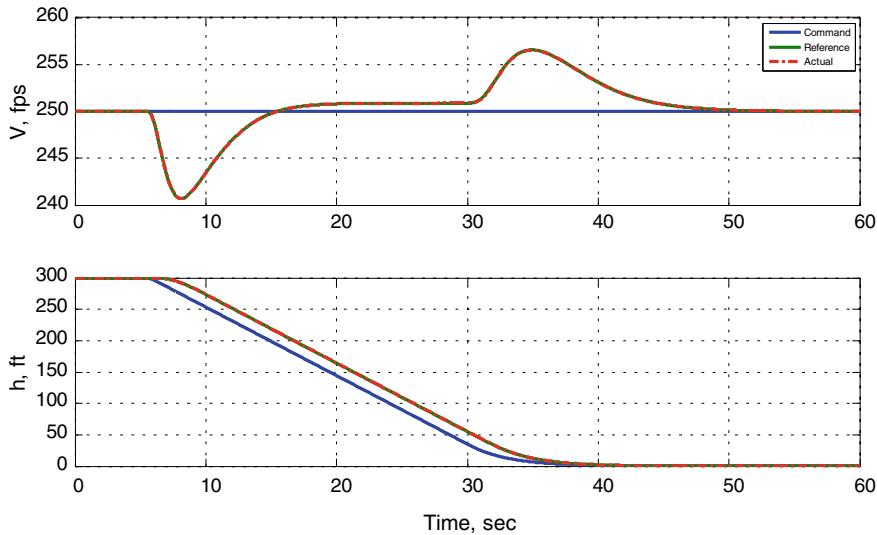


Fig. 12.14 (Baseline + Adaptive) closed-loop performance with ground effect in Example 12.3

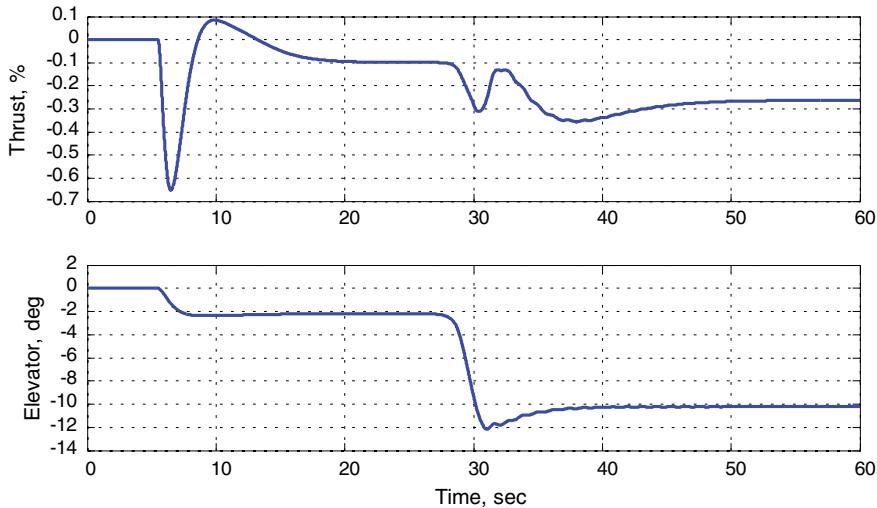


Fig. 12.15 Total control during landing with ground effect in Example 12.3

12.5 Summary

We have developed an adaptive design method to control MIMO dynamics in the presence of unstructured uncertainties, such as nonlinear state-dependent functions and bounded time-varying process noise. The resulting MRAC system represents

an extension of the previously derived adaptive controllers for linear-in-parameter matched uncertainties.

Our current adaptive design includes a state limiter and a state modulation function. The state limiting logic was originally proposed in [11]. The state limiter keeps the system trajectories within predefined boundaries that define an operational envelope for the system. This is the set where we can represent the state-dependent uncertainties by linear-in-parameters RBF NNs. The state limiter is also capable of bringing the system state back into the operational envelope in finite time.

The state limiting mode is turned on or off by the state modulation function, which in turn provides a gain-scheduling feature between the command tracking and the state limiting modes of operation. In other words, the state limiter can seamlessly fade in and/or out the command tracking or the state limiting tasks, depending if the system state is located inside or outside of the operational envelope, respectively.

We have also presented a concise overview of function approximation properties using artificial NNs. This material justifies our model formulation and control design approaches for representation and attenuation of the system nonlinear-in-parameters uncertainties.

In summary, we have employed Lyapunov-based arguments and artificial NNs to attain UUB tracking performance for MIMO dynamics with both structured and unstructured uncertainties.

12.6 Exercises

Exercise 12.1 Select a scalar non-monotonic function. Use an offline regression to approximate the selected function with sigmoidal and RBF neural networks. Increase the number of neurons and record the corresponding function errors. Plot the following data: (a) the function and the approximating NN and (b) the function approximation error versus the number of neurons. Repeat these tasks for a function of two independent variables. Comment on your results.

Exercise 12.2 Modify design equations in Table 12.1 to justify an adaptive augmentation-based design. Prove closed-loop system stability, show boundedness of all signals, and quantify tracking performance.

Exercise 12.3 Repeat the control design and all simulation steps from Example 12.3. Using the selected regressor Φ , perform offline approximation of the ground effect-induced angle-of-attack function $\alpha_g(h)$. Compute the corresponding online approximation of the same function. Compare and discuss the offline versus online approximation data. Modify the aircraft open-loop matrix A such that the ground effect matrix B_g is no longer matched. Introduce a control uncertainty $\Lambda \neq I_{m \times m}$, and add a uniformly bounded process noise $\xi(t)$. Redesign the controller (if needed),

and rerun all simulation tests. Discuss robustness properties of the controller with respect to the unmatched ground effect uncertainties and the process noise.

Exercise 12.4 Similarly to (12.37), derive an adaptive augmentation of a baseline proportional–integral (PI) controller, (Baseline PI + Adaptive), with a state limiter modification similar to (12.16), for the extended open-loop system dynamics,

$$\dot{x} = Ax + B \Lambda(u + f(x_p)) + B_{\text{ref}} y_{\text{cmd}} + \xi(t)$$

with m inputs u , m regulated outputs $y = Cx$, m commands $y_{\text{cmd}} \in R^m$, and n uniformly bounded noise components $\xi(t) \in R^n$, where (A, B) is a controllable pair of unknown matrices, $\Lambda \in R^m$ is an unknown positive-definite diagonal matrix, and $f(x_p)$ denotes an unstructured matched state-dependent uncertainty. It is assumed that the first m components of the state vector x represent the integrated output tracking error, whose dynamics are: $\dot{e}_y I = y - y_{\text{cmd}}$.

Exercise 12.5 Using the design equations from Exercise 12.4 and the aircraft data from Example 12.3, design and simulate a (Baseline PI + Adaptive) automatic landing system. Are there any advantages for using a PI baseline controller versus a feedback/feedforward system from Example 12.3? Compare and discuss your results.

References

1. Scarselli F, Tsoi C (1998) Universal approximation using feedforward neural networks: a survey of some existing methods, and some new results. *Neural Netw* 11(1):15–37
2. Hunt KJ, Sbarbaro D, Zbikowski R, Gawthrop PJ (1992) Neural networks for control systems—a survey. *Automatica* 28(6):1083–1112
3. Micchelli C (1986) Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constr Approx* 2:11–12
4. Cybenko G (1983) Approximation by superposition of a sigmoidal function. *Math Control Signals Syst* 2:303–314
5. Barron A (1993) Universal approximation bounds for superposition of a sigmoidal function. *IEEE Trans Inf Theory* 3:930–945
6. Park J, Sandberg IW (1991) Universal approximation using radial-basis-function networks. *Neural Comput* 3(2):246–257
7. Narendra KS, Annaswamy AM (2005) Stable adaptive control. Dover, New York
8. Ioannou P, Fidan P (2006) Adaptive control tutorial, SIAM, advances in design and control. SIAM, PA
9. Khalil H (1996) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, NJ, p 07458
10. Stevens BL, Lewis FL (1992) Aircraft control and simulation. Wiley
11. Sanner MR, Slotine J-JE (1992) Gaussian networks for direct adaptive control. *IEEE Trans Neural Netw* 3(6):837–863



Adaptive Control with Improved Transient Dynamics

13

In this chapter, we revisit the original formulation of the model reference adaptive control (MRAC), examine its transient performance, and then propose a modification to improve the latter. Often in practice, it is the transient of a tracking controller that defines feasibility of the selected system. Tracking performance during the first few seconds of operation is often more important than its asymptotic behavior. For linear systems, we can quantify transients by invoking the classical notions, such as damping ratio, natural frequency, overshoot, and undershoot. As for MRAC, uniform transient performance characterization is not at all straightforward. To remedy this situation, we offer a design modification to enforce and quantify transients for a class of MRAC-controlled nonlinear dynamical systems with uncertainties. We begin by drawing a parallel between reference models in adaptive control and Luenberger asymptotic state observers [1]. Based on this comparison, we reformulate the MRAC reference model structure. Our design change consists of adding an observer-like tracking error feedback/mismatch term to the reference model dynamics. This term is also known as the “innovation process” in Kalman filtering problems. Similar to the observer design, we will show that such a modification allows the enforcing of sufficiently fast, smooth and thus improved, tracking error dynamics in MRAC systems. We substantiate our method with Lyapunov-based arguments to show that it is capable of solving servomechanism tracking problems for a selected class of multi-input–multi-output dynamical systems with a wide class of uncertainties.

13.1 Motivation

Let us return to the original concept of the model reference adaptive control (MRAC), as it was first proposed in 1958 by Whitaker et al., at MIT [2]. The main idea was to specify the desired command-to-output performance of a servotacking system using a reference model that would define the ideal response of the system due to external commands. A generic block diagram of the MRAC system is shown in Fig. 13.1.

As seen from the diagram, the controller parameter adjustments (the adaptive law) are made based on the tracking error (the difference between the system actual response and its target specified by the Reference Model output), an output feedback from the process, and the external command.

For clarity and in order to motivate further discussions, we consider MRAC design equations for a scalar system shown below:

$$\begin{aligned}
 \text{Process : } & \dot{x} = a x + b u \\
 \text{Ref. Model : } & \dot{x}_{\text{ref}} = a_{\text{ref}} x_{\text{ref}} + b_{\text{ref}} r \\
 \text{Controller : } & u = \hat{k}_x x + \hat{k}_r r \\
 \text{Adaptive Law : } & \begin{cases} \dot{\hat{k}}_x = -\gamma_x \dot{x}(x - x_{\text{ref}}) \\ \dot{\hat{k}}_r = -\gamma_r r(x - x_{\text{ref}}) \end{cases}
 \end{aligned} \tag{13.1}$$

where a and b are unknown constant parameters in the process dynamics with the known $\text{sgn } b > 0$. The control input u is selected such that the system state x follows the reference model state x_{ref} , driven by any bounded external command $r = r(t)$. Also in (13.1), the reference model data $a_{\text{ref}} < 0$ and b_{ref} are chosen to yield the desired speed of response and a DC gain (unity in most applications)

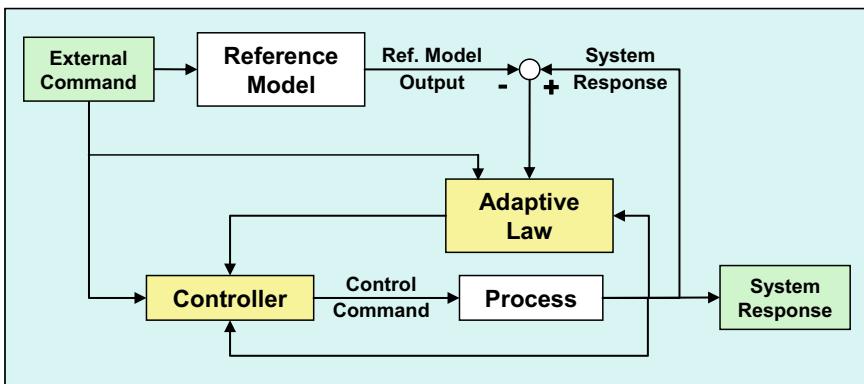


Fig. 13.1 MRAC block diagram

from the reference model output $y_{\text{ref}} = x_{\text{ref}}$ to the regulated output $y_{\text{reg}} = x$, represented by the system state.

In this case, closed-loop system stability and global asymptotic tracking are achieved via a specific choice of the adaptive law in (13.1), with the adaptive gains (\hat{k}_x, \hat{k}_r) , whose dynamics are influenced by two positive constant rates of adaptation (γ_x, γ_r) . As seen from (13.1), the state tracking error

$$e = x - x_{\text{ref}} \quad (13.2)$$

drives the adaptive laws. Existence of a servocontrol solution for these particular scalar dynamics is provided by the matching conditions,

$$\begin{aligned} a_{\text{ref}} &= a + b k_x \\ a_{\text{ref}} &= b k_r \end{aligned} \quad (13.3)$$

where k_x and k_r denote the ideal unknown constant parameters (gains of the ideal controller). For scalar dynamics, such as the process in (13.1), it is clear that the matching relations (13.3) always have a solution.

Let,

$$\Delta k_x = \hat{k}_x - k_x, \quad \Delta k_r = \hat{k}_r - k_r \quad (13.4)$$

represent the parameter estimation errors. Substituting the matching conditions (13.3) into (13.1), one can derive the tracking error dynamics,

$$\dot{e} = a_{\text{ref}} e + b(\Delta k_x x + \Delta k_r r) \quad (13.5)$$

which indeed define transients in the corresponding closed-loop system.

We emphasize that both the tracking error dynamics and the transient dynamics are indistinguishable. In other words, if and when e becomes small, the system-regulated output tracks the reference model with diminishing errors. On the other hand, the transient dynamics define what happens between the start of a maneuver and the time when the tracking error gets small. We shall address this question in details.

Returning to (13.5), we can employ Lyapunov arguments to prove global asymptotic stability of the tracking error dynamics. In fact, using a radially unbounded quadratic Lyapunov function candidate in the form,

$$V(e, \Delta k_x, \Delta k_r) = e^2 + b \left(\frac{\Delta k_x^2}{\gamma_x} + \frac{\Delta k_r^2}{\gamma_r} \right) \quad (13.6)$$

it is not difficult to show that with the adaptive law (13.1), the time derivative of V , evaluated along the trajectories of the error dynamics (13.5), becomes non-positive. Invoking the Barbalat's lemma [3], this argument constitutes the inverse

Lyapunov-based design. It provides: (a) the adaptive law; and (b) the required proof of closed-loop global asymptotic stability. As a result, we can formally show that for any initial condition, any bounded time-varying external command, and any positive rates of adaptation, the tracking error dynamics (13.5) are globally asymptotically stable,

$$\lim_{t \rightarrow \infty} |e(t)| = \lim_{t \rightarrow \infty} |x(t) - x_{\text{ref}}(t)| = 0 \quad (13.7)$$

and all signals in the corresponding closed-loop dynamics remain uniformly bounded, forward in time.

We immediately note that this adaptive controller solves the servotracking problem asymptotically in time, as $t \rightarrow \infty$, while it provides no uniformly guaranteed bounds on how large the transients might become prior to acquiring the command.

In the previous chapters, we have shown that in order to yield fast tracking and thus shorten transient times, one needs to increase the rates of adaptation (γ_x , γ_r). However, experience shows that if these rates grow large then unwanted transient oscillations will start to occur during the initial few seconds (the transient time) of the closed-loop system operation. The balance between achieving fast tracking and avoiding undesired transients constitutes the MRAC design tradeoff phenomenon. In essence, the rates of adaptation must be chosen large enough but not too large.

What also complicates the MRAC design tuning process is the direct dependence of the transient dynamics (13.5) on: (a) the external command; and (b) the initial conditions for the system and the adaptive controller. These parameters may too lead to undesirable transients.

Let us take a step back and look again at the error dynamics (13.5). We know that the time-varying signal

$$\varphi(t) = b(\Delta k_x(t)x(t) + \Delta k_r(t)r(t)) \quad (13.8)$$

is uniformly bounded and that the tracking error $e(t)$ globally asymptotically tends to zero, as shown in (13.7). Still, the time constant of the transient dynamics (13.5) $\tau_e = \frac{1}{|a_{\text{ref}}|}$ is exactly the same as the one for the reference model in (13.1).

Even though having the same time constant in both systems is theoretically correct, any controls practitioner would want to have the transient dynamics (13.5) evolve faster than the desired reference model. In other words, we want the transients to die out quickly, relative to the dynamics of the reference model trajectories. This design requirement is identical to the one that takes place during the construction of asymptotic state observers, originally developed by Luenberger in his Ph.D. thesis at The Stanford University (1963) and later published in [1]. Per Luenberger, the reference model in (13.1) represents an open-loop observer. So, just like in the closed-loop observer dynamics, we can add an error feedback term to the reference model and arrive at the observer-like reference model,

$$\dot{x}_{\text{ref}} = a_{\text{ref}} x_{\text{ref}} + b_{\text{ref}} r + \boxed{k_e(x - x_{\text{ref}})} \quad (13.9)$$

Error Feedback Term

where $k_e > 0$ is the reference model feedback gain. To draw a parallel between the original “open-loop” reference model in (13.1) and its modified version (13.9), we will often refer to the latter as the “Closed-Loop Reference Model”, or in short CRM.

The newly introduced error feedback term in (13.9) is equivalent to the output innovation feedback in a state observer. It is easy to see that in this case, the corresponding error dynamics become faster than the open-loop reference model from (13.1).

$$\dot{e} = (a_{\text{ref}} - k_e)e + b(\Delta k_x x + \Delta k_r r) \quad (13.10)$$

Once again, Lyapunov-based arguments coupled with Barbalat’s lemma can be easily repeated to prove: (a) global asymptotic stability of the modified error dynamics (13.9) and (b) uniform boundedness of all signals in the related closed-loop system. For those readers who are familiar with the MRAC stability proof concept, we briefly note that using the same Lyapunov function candidate (13.6), one needs to compute its time derivative along the trajectories of (13.10), substitute the adaptive law from (13.1), and then show that the resulting time derivative is globally non-positive. This will prove uniform boundedness of the tracking error e and of the parameter estimation errors (13.4). Furthermore, since in the observer-like reference model (13.9), $a_{\text{ref}} < 0$ and the error feedback term is bounded, then the model state x_{ref} is bounded as well. The rest of the proof follows standard (in MRAC) stability arguments, finally arriving at (13.7).

Revised block diagram with the observer-like reference model (13.9) is shown in Fig. 13.2.

Example 13.1 Before proceeding any further, we would like to briefly present and discuss simulation comparison data for the CRM modification (13.9), while using

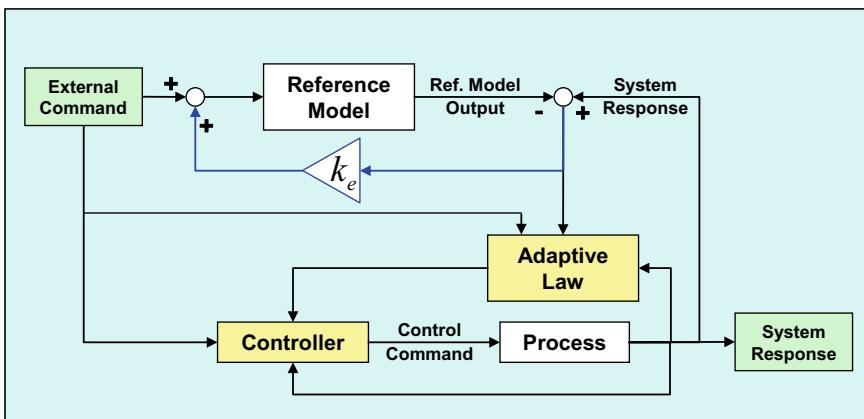


Fig. 13.2 MRAC block diagram with CRM modification

the scalar process dynamics from (13.1) and the simulation parameters as indicated below:

$$\text{Process : } \dot{x} = x + 3u$$

$$\text{Ref. Model : } \dot{x}_{\text{ref}} = -10x_{\text{ref}} + 10r + k_e(x - x_{\text{ref}})$$

$$\text{Controller : } u = \hat{k}_x x + \hat{k}_r r$$

$$\text{Adaptive Law : } \begin{cases} \dot{\hat{k}}_x = -10x(x - x_{\text{ref}}) \\ \dot{\hat{k}}_r = -10r(x - x_{\text{ref}}) \end{cases}$$

(13.11)

In order to assess transient improvements, we perform three distinct simulation scenarios, where the error feedback gain k_e is set to 0 (standard MRAC case), 10, and 80. Figure 13.3 shows step-input response data for the three cases.

The original MRAC transient dynamics are quite oscillatory. As the reference model feedback gain k_e is increased, the transient dynamics become faster and the unwanted oscillations subside. Figure 13.4 presents simulation data comparison between the first and the third cases. Both the system state x and the control input u are shown. These responses were computed for a series of commanded step-inputs of increased magnitude.

As seen from the simulation data, the use of the observer-like reference model (13.9) gives a predictable, scalable, and non-oscillatory (in transient) tracking performance (data shown in green). ■

Now, we shall pose the following question—can the simulated transient improvements of the observer-like reference model be formally explained? We

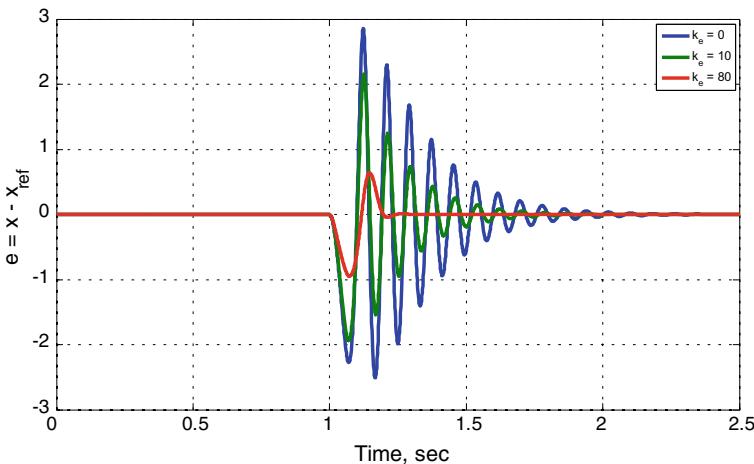


Fig. 13.3 MRAC transient dynamics due to step-input command in Example 13.1

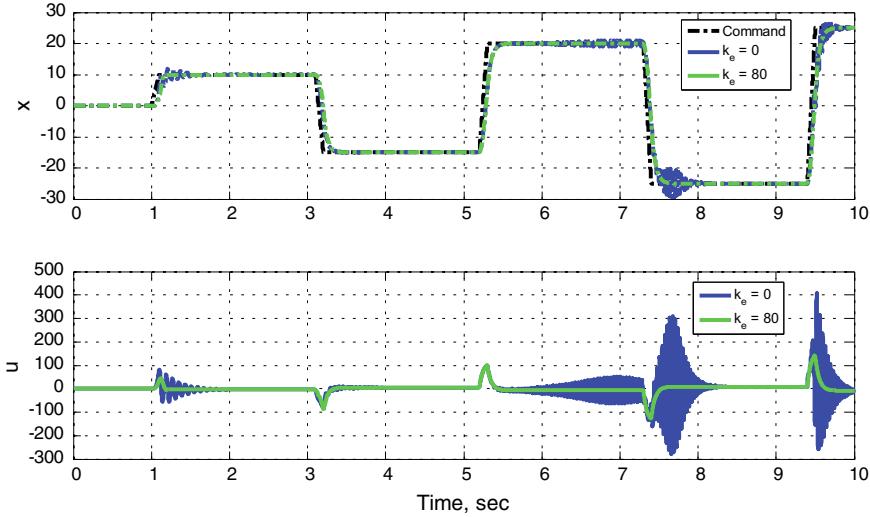


Fig. 13.4 MRAC tracking performance due to a sequence of step-inputs in Example 13.1

claim that as the reference model error feedback gain k_e is increased, the system transient dynamics become less oscillatory.

In order to gain further insights into the transient behavior, we choose $k_0 > 0$, a small positive parameter ε , and redefine the reference model feedback gain,

$$k_e = \frac{k_0}{\varepsilon} \quad (13.12)$$

This allows to rewrite the modified error dynamics (13.10) in the form:

$$\varepsilon \dot{e} = (\varepsilon a_{\text{ref}} - k_0)e + \underbrace{\varepsilon [b(\Delta k_x x + \Delta k_r r)]}_{\varphi(t)} \quad (13.13)$$

Since all signals in the corresponding closed-loop system are uniformly bounded, it is not difficult to find sufficient conditions so that there exists a strictly positive finite constant $0 < \varphi_{\max} < \infty$, such that for any $\varepsilon > 0$, the upper bound $|\varphi(t)| \leq \varphi_{\max}$ holds uniformly in time and ε . Furthermore, starting from an initial condition $e(0) = e_0$, the solution of (13.13) can be written explicitly.

$$e(t) = e^{\left(a_{\text{ref}} - \frac{k_0}{\varepsilon}\right)t} e(0) + \int_0^t e^{\left(a_{\text{ref}} - \frac{k_0}{\varepsilon}\right)(t-\tau)} \varphi(\tau) d\tau \quad (13.14)$$

We can compute an upper bound for this signal.

$$|e(t)| \leq e^{-k_0 \frac{t}{\varepsilon}} |e_0| + \frac{\varphi_{\max}}{k_0} \varepsilon \quad (13.15)$$

This relation is valid for any fixed $\varepsilon > 0$, uniformly in time. So, the system state $x(t)$ converges to within $(\pm \frac{\varphi_{\max}}{k_0} \varepsilon)$ of the reference model state $x_{\text{ref}}(t)$ exponentially fast, and at the rate which is no slower than $e^{-k_0 \frac{t}{\varepsilon}}$. This term gives an upper bound quantification for the decay rate of the MRAC transient dynamics, due to initial conditions mismatch, $x(0) \neq x_{\text{ref}}(0)$. Otherwise, the system transients would remain within ε -dependent bounds $(\pm \frac{\varphi_{\max}}{k_0} \varepsilon)$. Consequently, we can reduce the system transients by decreasing ε , which according to (13.12) corresponds to increasing the reference model feedback gain k_e . Being able to influence and shape the MRAC transient dynamics constitutes the essential benefit of the Luenberger-like reference model modification (13.9)–(13.12).

Let us give an alternative explanation for the noted transient improvements in scalar MRAC systems with the CRM modification. The transient dynamics (13.13) can be analyzed using the singular perturbation methods [3]. Setting $\varepsilon = 0$, gives the so-called slow component,

$$e = 0 \quad (13.16)$$

or, equivalently

$$x = x_{\text{ref}} \quad (13.17)$$

Asymptotic stability of the slow component has already been established during Lyapunov-based proofs. Therefore as $t \rightarrow \infty$,

$$\dot{x} = a_{\text{ref}} x + b_{\text{ref}} r + o(1) \quad (13.18)$$

where the Small o-symbol $o(1)$ denotes a function of time that asymptotically tends zero as $t \rightarrow \infty$. According to (13.18), the system state x asymptotically tracks the state of the observer-like reference model x_{ref} , with the latter asymptotically approaching the state of the original reference model from (13.1). We compute the “fast” dynamics by “stretching” time,

$$\tau = \frac{t}{\varepsilon} \quad (13.19)$$

rewrite (13.13) in the “fast” τ -time scale, set $\varepsilon = 0$, and arrive at the exponentially stable fast dynamics

$$\frac{d e(\tau)}{d \tau} = -k_0 e(\tau) \quad (13.20)$$

It follows from (13.20) that during a finite transient time interval, the error dynamics (13.20) behave like a first-order exponentially stable system. This observation confirms our claim that for a sufficiently small ε , i.e., for a large enough gain

k_e , the resulting transient dynamics become smooth and approach the response of the scalar system (13.20), where k_0 defines the inverse time constant of the transient system.

This result can be formally summarized as follows: For a sufficiently small $\varepsilon > 0$, the state of the original system in (13.1) permits the following asymptotic expansion,

$$x(t) = x_{\text{ref}}(t) + C e^{-\frac{k_0}{\varepsilon} t} + o(1) \quad (13.21)$$

or equivalently,

$$x(t) = x_{\text{ref}}(t) + C e^{-k_e t} + o(1) \quad (13.22)$$

where $C > 0$ is a constant independent of $k_e = \frac{k_0}{\varepsilon}$. The second term in (13.22) defines the transient (i.e., “fast”) dynamics due to initial conditions. Consequently, with a large enough feedback gain k_e , MRAC transient dynamics can be quantified and forced to decay as fast as needed. We should immediately point out that since k_e is inversely proportional to ε , then the obvious “trade off” in the modified design would be to avoid high gain effects in the reference model.

In the sections that follow, we will further exploit methods to analyze and enforce desired transient dynamics in adaptive control systems. But first, we present an overview of the mathematical preliminaries related to asymptotic expansions and their analysis. After that, we will generalize the observer-like reference model idea to a class of multi-input–multi-output (MIMO) dynamical systems with matched linear-in-parameters uncertainties. We shall conclude this chapter with practical observations, a summary of the derived results, and a numerical example.

13.2 Asymptotic Orders and Singular Perturbations

Basic notions in asymptotic orders and singular perturbations are introduced and discussed in Chap. 6. For the sake of completeness, we shall repeat some of these concepts here.

Let R^n represent the Euclidean n -dimensional space, R^+ be the set of all positive real numbers, and let $R^{n \times m}$ denote the space of all n -by- m matrices, with integers n and m . For any $x \in R^n$, we write $\|x\|$ for a Euclidean vector norm of x , and $\|A\|$ to be the corresponding induced matrix norm, for $A \in R^{n \times m}$.

We shall use the Bachmann–Landau asymptotic-order notation, denoted by the “Big O” and the “Small o” symbols [4, 5]. Given any two parameter-dependent functions (maps) $f(x; \varepsilon)$ and $g(x; \varepsilon)$, from a domain $X \subset R^n$ to another domain $Y \subset R^m$, with a scalar parameter $\varepsilon \in E \subset R$ from an interval E , we say that

$$f(x; \varepsilon) = O(g(x; \varepsilon)) \quad (13.23)$$

if for each $x \in X$, there exists a positive scalar $k(x)$ such that,

$$\|f(x; \varepsilon)\| \leq k(x)\|g(x; \varepsilon)\| \quad (13.24)$$

for all $\varepsilon \in E$. Choose $\varepsilon_0 \in E$ and suppose that the two limits

$$\begin{aligned} \lim_{\varepsilon \rightarrow \varepsilon_0} \|f(x; \varepsilon)\| &= \|f(x; \varepsilon_0)\| \\ \lim_{\varepsilon \rightarrow \varepsilon_0} \|g(x; \varepsilon)\| &= \|g(x; \varepsilon_0)\| \end{aligned} \quad (13.25)$$

exist. We write

$$f(x; \varepsilon) = O(g(x; \varepsilon)), \quad \text{as } \varepsilon \rightarrow \varepsilon_0 \quad (13.26)$$

if for each $x \in X$, there exists a positive scalar $k(x)$ and a neighborhood/interval $N(x, \varepsilon_0)$ of $\varepsilon = \varepsilon_0$, such that (13.24) holds for all $\varepsilon \in N(x, \varepsilon_0)$. Without a loss of generality, we assume that $\varepsilon_0 = 0$. In this case, the asymptotic-order relation (13.26) defines the convergence rate of $\|f(x; \varepsilon)\|$ to $\|f(x; 0)\|$, as $\varepsilon \rightarrow 0$, while holding x fixed. Specifically, for every fixed $x \in X$, $\|f(x; \varepsilon)\|$ converges to its limit $\|f(x; 0)\|$ no slower than $\|g(x; \varepsilon)\|$ converges to $\|g(x; 0)\|$, as $\varepsilon \rightarrow 0$. This convergence may hold uniformly in X , yet it could completely fail outside of X . The statement (13.23) is said to be uniformly valid in X if $k(x)$ is a finite constant independent of x . In addition, if the set $N(x, \varepsilon_0) = N(\varepsilon_0)$ is independent of x then (13.26) is said to be uniformly valid in x . The relations (13.23) and (13.26) define the Big O-symbol.

The Small o-symbol is defined as follows. For a given domain $X \subset R^n$, the statement

$$f(x; \varepsilon) = o(g(x; \varepsilon)), \quad \text{as } \varepsilon \rightarrow 0 \quad (13.27)$$

means that for each $x \in X$ and any given $\delta > 0$, there exists an ε -interval $N(x, \delta) = \{\varepsilon : 0 < \varepsilon \leq \varepsilon_1(x, \delta)\}$ such that

$$\|f(x; \varepsilon)\| \leq \delta \|g(x; \varepsilon)\| \quad (13.28)$$

for all $\varepsilon \in N(x, \delta)$. We say that (13.27) is uniformly valid in X if $\varepsilon_1(x, \delta) = \varepsilon_1(\delta)$ is independent of x . Often, the notation $f \ll g$ is used to indicate (13.27).

The O- and o-symbols can be easily extended to parameter-dependent matrices $A(x; \varepsilon) \in R^{n \times n}$, using vector-induced matrix norms. For example, given a matrix $A_\varepsilon(x) = A(x; \varepsilon) \in R^{n \times n}$, the matrix asymptotic expansion

$$A_\varepsilon(x) = A_0(x) + A_1(x)\varepsilon + O(\varepsilon^2), \quad \text{as } \varepsilon \rightarrow 0 \quad (13.29)$$

means that for every x from a domain $X \subset R^n$,

$$\lim_{\varepsilon \rightarrow 0} \|A_\varepsilon(x) - A_0(x) - A_1(x)\varepsilon\| = \lim_{\varepsilon \rightarrow 0} \|O(\varepsilon^2)\| = 0 \quad (13.30)$$

and the convergence rate in (13.30) is no slower than ε^2 , for every fixed x . We immediately note that there is a difference between the asymptotic expansion (13.29) and, for example, the Taylor series expansion of a state-parameter-dependent matrix $A_\varepsilon(x)$. In fact, the Taylor series expansion may not even exist since differentiability of $A_\varepsilon(x)$ with respect to ε is not assumed.

In our forthcoming derivations, we will encounter singular perturbation models [3–5]. These are dynamical systems with a small positive scale factor ε on some of the system state derivatives. For example, the transient dynamics (13.13) represent a singular perturbation model.

Generalizing (13.13), we get a singular perturbation model in the form,

$$\varepsilon \dot{z} = \underbrace{(A_0 + O(\varepsilon))}_{{A_\varepsilon}} z + \varepsilon f(z, t, \varepsilon) \quad (13.31)$$

where $z \in R^n$ is the system state, $\varepsilon > 0$ is a constant parameter, $A_0 \in R^{n \times n}$ is Hurwitz, and

$$A_\varepsilon = (A_0 + O(\varepsilon)) \in R^{n \times n} \quad (13.32)$$

is Hurwitz, uniformly in ε . We also suppose that $f(z, t, \varepsilon) : R^n \times R^+ \times R^+ \rightarrow R^n$ is a uniformly bounded function of its arguments,

$$f(z, t, \varepsilon) = O(1) \quad (13.33)$$

or, equivalently

$$\|f(z, t, \varepsilon)\| \leq f_{\max} < \infty \quad (13.34)$$

uniformly in (z, t, ε) , where f_{\max} is a constant finite upper bound of the norm of f . In addition, we assume that $f(z, t, \varepsilon)$ is Lipschitz-continuous in z and piecewise continuous in (t, ε) . It is not difficult to show that all of the above-stated assumptions assure existence and uniqueness of the system solutions, starting at any set of initial conditions $z_0 = z(0)$.

It is possible to show that for a sufficiently small ε , all trajectories of (13.31) converge to an $O(\varepsilon)$ neighborhood of the origin, exponentially fast. This fact is stated next.

Theorem 13.1 Consider the singularly perturbed n -dimensional dynamics (13.31),

$$\varepsilon \dot{z} = \underbrace{(A_0 + O(\varepsilon))}_{{A_\varepsilon}} z + \varepsilon f(z, t, \varepsilon)$$

where $\varepsilon > 0$ is a constant, $A_0 \in R^{n \times n}$ is a constant Hurwitz matrix, and $A_\varepsilon \in R^{n \times n}$ is a Hurwitz (uniformly in ε) matrix. Suppose that $f(z, t, \varepsilon) \in R^n$ is a uniformly

bounded vector-function, Lipschitz-continuous in z , and piece-wise continuous in t and ε . Then there exists a strictly positive constant $\gamma > 0$, independent of ε , such that the asymptotic relation,

$$z(t) = O\left(e^{-\gamma \frac{t}{\varepsilon}}\right) + O(\varepsilon), \quad (\varepsilon \rightarrow 0) \quad (13.35)$$

holds for all $t \geq 0$. If in addition to being uniformly bounded, the function $f(z, t, \varepsilon)$ asymptotically decays to zero in time (uniformly in z), then

$$z(t) = O\left(e^{-\gamma \frac{t}{\varepsilon}}\right) + O(\varepsilon)o(1), \quad (\varepsilon \rightarrow 0) \quad (13.36)$$

for all $t \geq 0$, where $o(1) \xrightarrow[t \rightarrow \infty]{} 0$ is an asymptotically decaying time function. ■

Proof of Theorem 13.1 Since A_ε is Hurwitz uniformly in ε , and

$$A_\varepsilon \frac{t}{\varepsilon} = (A_0 + O(\varepsilon)) \frac{t}{\varepsilon} = A_0 \frac{t}{\varepsilon} + O(1) \quad (13.37)$$

then following the proof arguments from [3, Lemma 9.9, pp. 369–371], we can claim existence of two strictly positive constants, k and γ , such that for a sufficiently small $\varepsilon > 0$, the induced 2-norm of the exponential matrix $e^{A_\varepsilon \frac{t}{\varepsilon}}$ satisfies,

$$\left\| e^{A_\varepsilon \frac{t}{\varepsilon}} \right\| = \left\| e^{(A_0 + O(\varepsilon)) \frac{t}{\varepsilon}} \right\| \leq k e^{-\gamma \frac{t}{\varepsilon}} \quad (13.38)$$

where $(k, \gamma) > 0$ are independent of ε . This fact merely states that the 2-norm of a parameter-dependent Hurwitz matrix exponentially decays to zero, if the parameter is selected small enough. In terms of the asymptotic-order notation, (13.38) implies,

$$\left\| e^{A_\varepsilon \frac{t}{\varepsilon}} \right\| = \left\| e^{(A_0 + O(\varepsilon)) \frac{t}{\varepsilon}} \right\| = O\left(e^{-\gamma \frac{t}{\varepsilon}}\right), \quad (\varepsilon \rightarrow 0) \quad (13.39)$$

for all $t \geq 0$.

Because of its specific form, the singular perturbation dynamics (13.31) can be analyzed directly by explicitly writing the system solution.

$$z(t) = e^{A_\varepsilon \frac{t}{\varepsilon}} z_0 + \int_0^t e^{A_\varepsilon \frac{(t-\tau)}{\varepsilon}} f(z(\tau), \tau, \varepsilon) d\tau \quad (13.40)$$

With the help of (13.39), we can easily derive an upper bound for the norm of the system solution.

$$\|z(t)\| \leq \left\| e^{A_\varepsilon \frac{t}{\varepsilon}} \right\| \|z_0\| + f_{\max} \int_0^t \left\| e^{A_\varepsilon \frac{(t-\tau)}{\varepsilon}} \right\| d\tau \leq k \|z_0\| e^{-\gamma \frac{t}{\varepsilon}} + \left(\frac{k f_{\max}}{\gamma} \right) \varepsilon \quad (13.41)$$

Thus, we have proven the asymptotics (13.35),

$$z(t) = O\left(e^{-\gamma \frac{t}{\varepsilon}}\right) + O(\varepsilon), \quad (\varepsilon \rightarrow 0)$$

for the system solutions (13.40), evolving on an infinite time interval.

In the context of singular perturbations, this relation implies that for a sufficiently small fixed $\varepsilon > 0$, all solutions of (13.31) converge to an ε -neighborhood of the origin exponentially fast, at the rate of no slower than $e^{-\gamma \frac{t}{\varepsilon}}$. The first term in (13.35) describes the “fast” (transient) dynamics of the system solutions, as they approach an ε -neighborhood of the origin. Also, in this case, $\frac{t}{\varepsilon}$ can be interpreted as the “stretched” time, which allows us to look at the system transients through a “magnifying time-glass”, so to speak.

Suppose that for a constant $\varepsilon > 0$,

$$f(z, t, \varepsilon) = o(1) \xrightarrow[t \rightarrow \infty]{} 0 \quad (13.42)$$

uniformly in z . Repeating the previous arguments that have led us to (13.35), one can derive (13.36) (see Exercise 13.2),

$$z(t) = O\left(e^{-\gamma \frac{t}{\varepsilon}}\right) + O(\varepsilon)o(1), \quad (\varepsilon \rightarrow 0)$$

for all $t \geq 0$. So in this case, the system trajectories converge to a neighborhood of the origin exponentially fast, and after that, the solutions continue to asymptotically converge to the origin, but at perhaps a much slower rate. The theorem proof is complete. ■

Observe that setting $\varepsilon = 0$ reduces the differential equation (13.31) to an algebraic relation $z = 0$. This is the singularity phenomenon, whereby the origin becomes the “slow” manifold of the system. Overall, we have decomposed the system trajectories into “fast” and “slow” components, with the former describing the rate of convergence to the latter.

Let us mention that the asymptotic behavior (13.35) could have also been derived using the singular perturbation methods [3–5] for trajectory analysis of ordinary differential equations, such as the one in (13.31). We have decided to perform a direct analysis of the system trajectories (instead of using the singular

perturbation techniques), only because for the system at hand we could explicitly write solutions and estimate their norm upper bounds.

We are going to utilize the two asymptotic relations (13.35)–(13.36) to aid in the design of adaptive output feedback controllers.

13.3 Asymptotic Properties of the Algebraic Riccati Equation

In our forthcoming design and analysis of MRAC transient dynamics, we will encounter parameter-dependent n -dimensional Algebraic Riccati Equations (ARE) in the form,

$$P_v A + A^T P_v - P_v B R_v^{-1} B^T P_v + Q_v = 0 \quad (13.43)$$

where $v > 0$ is a constant parameter, (A, B) and (A, C) are controllable and observable pairs of matrices, with $A \in R^{n \times n}$, $B \in R^{n \times m}$, $C \in R^{m \times n}$, and $m \leq n$. The ARE weight matrices are defined as

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) C^T C, \quad R_v = \left(\frac{v}{v+1} \right) R_0 \quad (13.44)$$

where $Q_0 \in R^{n \times n}$ and $R_0 \in R^{m \times m}$ are both symmetric and strictly positive definite. This formulation appears in [6].

The well-known fact from optimal control of linear systems (with quadratic cost index) states that for any $v > 0$, the ARE (13.43) has the unique symmetric positive-definite solution $P_v > 0$. This equation arises in the optimal linear quadratic regulator (LQR) control problems for linear time-invariant dynamics,

$$\dot{x} = A x + B u, \quad y = C x \quad (13.45)$$

with a quadratic minimization criterion in the form,

$$\begin{aligned} J_v &= \int_0^\infty (x^T Q_v x + u^T R_v u) dt = \int_0^\infty \left(x^T \left(Q_0 + \frac{1}{\rho_v} C^T C \right) x + \rho_v u^T R_0 u \right) dt \\ &= \int_0^\infty \left(x^T Q_0 x + \frac{1}{\rho_v} y^T y + \rho_v u^T R_0 u \right) dt \end{aligned} \quad (13.46)$$

where

$$\rho_v = \frac{v}{v+1} \quad (13.47)$$

is a positive constant.

Let us remark that the main difference of (13.46), from a typical cost function considered in classical textbooks on optimal control, is the presence of the second term, which is inversely proportional to ρ_v . This expression “punishes” the system output, as the system input is allowed to become large, with ρ_v getting small.

Substituting (13.44) into (13.43) gives

$$P_v A + A^T P_v - \left(1 + \frac{1}{v}\right) P_v B R_0^{-1} B^T P_v + Q_0 + \left(1 + \frac{1}{v}\right) C^T C = 0 \quad (13.48)$$

or, equivalently

$$P_v A + A^T P_v - P_v B R_0^{-1} B^T P_v + Q_0 + C^T C + \frac{1}{v} [C^T C - P_v B R_0^{-1} B^T P_v] = 0 \quad (13.49)$$

We are interested in analyzing asymptotic properties of the ARE solution P_v , as $v \rightarrow 0$. Hence, let us consider the following asymptotic expansion:

$$P_v = P_0 + P_1 v + O(v^2), \quad \text{as } v \rightarrow 0 \quad (13.50)$$

Similar to (13.29) and (13.30), the big O-symbol $O(v^2)$ in (13.50) denotes a v -dependent $(n \times n)$ -matrix, whose induced norm tends to zero no slower than v^2 , as $v \rightarrow 0$, that is:

$$\lim_{v \rightarrow 0} \|P_v - P_0 - P_1 v\| = \lim_{v \rightarrow 0} \|O(v^2)\| = 0 \quad (13.51)$$

For matrices satisfying (13.50), we can also write

$$P_0 = \lim_{v \rightarrow 0} P_v$$

which means $\lim_{v \rightarrow 0} \|P_v - P_0\| = 0$, that is limits of parameter-dependent matrices are understood in terms of their induced norms. Before proceeding any further, we need to introduce a square root of a matrix according to [7, p. 245].

Definition 13.1 An $(n \times n)$ matrix $S = P^{\frac{1}{2}} = \sqrt{P}$ is called a square root of a symmetric positive-definite $(n \times n)$ matrix P , if $P = S^T S$.

It is not so difficult to see that matrix square roots are by no means unique. However, we can define the unique square root by taking S to be symmetric.

Let us now state and prove several interesting asymptotic properties of a parameter-dependent ARE in the form of (13.49).

Theorem 13.2 Consider the ARE (13.43) with any two controllable and observable matrix pairs, (A, B) and (A, C) , and with the two symmetric positive-definite matrices Q_v and R_v from (13.44). Let $A \in R^{n \times n}$, $B \in R^{m \times n}$, $C \in R^{p \times n}$, $Q_v \in R^{n \times n}$, and $R_v \in R^{m \times m}$, where n , p , and m are integers. Then the ARE has the unique symmetric positive-definite solution P_v .

Moreover, if $p = m$, $\det(CB) \neq 0$, and the transfer function $G(s) = C(sI_{n \times n} - A)^{-1}B$ is minimum phase then the ARE solution P_v can be represented by the asymptotic expansion (13.50), while the following statements hold true:

1. P_0 and P_1 are symmetric.
2. P_0 is the unique symmetric strictly positive-definite solution of the following algebraic Lyapunov equation:

$$P_0 \left(A - B R_0^{-1} B^T P_1 \right) + \left(A - B R_0^{-1} B^T P_1 \right)^T P_0 + Q_0 = 0 \quad (13.52)$$

3. There exists a unitary matrix $W \in R^{m \times m}$ such that:

$$P_0 B = C^T W^T \sqrt{R_0} \quad (13.53)$$

4. The unitary matrix W in (13.53) can be chosen as:

$$W = (U V)^T \quad (13.54)$$

where U and V are two unitary matrices, defined by the singular value decomposition,

$$C B R_0^{-\frac{1}{2}} = U \Lambda V \quad (13.55)$$

and Λ represents the diagonal matrix of the corresponding singular values.

5. P_v is invertible for any $v \geq 0$, and for any unit vector $x \in R^n$,

$$\lim_{v \rightarrow 0} x^T P_v x \geq \lambda_{\min}(P_0) > 0 \quad (13.56)$$

where $\lambda_{\min}(P_0)$ denotes the minimum eigenvalue of P_0 .

6. The following asymptotic relation holds:

$$P_v B = C^T W^T \sqrt{R_0} + O(v), \quad \text{as } v \rightarrow 0 \quad (13.57)$$

Before proving the theorem, an immediate remark is in order. Relations (13.52) and (13.53) imply that the transfer function

$$\begin{aligned} G_0(s) &= B^T P_0 \left(s I_{n \times n} - A + B R_0^{-1} B^T P_1 \right)^{-1} B \\ &= \sqrt{R_0} W C \left(s I_{n \times n} - A + B R_0^{-1} B^T P_1 \right)^{-1} B \end{aligned}$$

becomes strictly positive real (SPR) [3] via feedback $u = -R_0^{-1} B^T P_1 x$, when the latter is applied to the linear dynamics (13.45). At the same time, the asymptotic expansions (13.50) and (13.57) mean that the transfer function,

$$G_v(s) = B^T P_v \left(s I_{n \times n} - A + B R_0^{-1} B^T P_v \right)^{-1} B \quad (13.58)$$

which is SPR by the design, approaches the transfer function,

$$G_y(s) = \sqrt{R_0} W C \left(s I_{n \times n} - A + B R_0^{-1} B^T P_0 \right)^{-1} B \quad (13.59)$$

that is

$$G_v(s) = G_y(s) + O(v), \quad \text{as } v \rightarrow 0 \quad (13.60)$$

uniformly in s .

Proof of Theorem 13.2 Existence and uniqueness of P_v is a well-known fact. We proceed by showing that matrices P_0 and P_1 in (13.50) are symmetric. Using (13.50), gives:

$$P_0 = \lim_{v \rightarrow 0} P_v = \lim_{v \rightarrow 0} P_v^T = P_0^T \quad (13.61)$$

Consequently,

$$P_1 = \lim_{v \rightarrow 0} \frac{1}{v} (P_v - P_0) = \lim_{v \rightarrow 0} \frac{1}{v} (P_v^T - P_0^T) = P_1^T \quad (13.62)$$

Next, we substitute (13.50) into (13.49).

$$\begin{aligned} v &\left[(P_0 + P_1 v + O(v^2)) A + A^T (P_0 + P_1 v + O(v^2)) \right] \\ &- v \left[(P_0 + P_1 v + O(v^2)) B R_0^{-1} B^T (P_0 + P_1 v + O(v^2)) + Q_0 + C^T C \right] \\ &+ C^T C - (P_0 + P_1 v + O(v^2)) B R_0^{-1} B^T (P_0 + P_1 v + O(v^2)) = 0 \end{aligned} \quad (13.63)$$

Collecting the zero-order terms in v , gives

$$C^T C - P_0 B R_0^{-1} B^T P_0 = 0 \quad (13.64)$$

The matrix solution $(P_0 B)$ of (13.64) may be expressed as in (13.53), whose validity can be verified by its direct substitution into (13.64).

Collecting the first-order terms in v from (13.63) gives

$$\underbrace{P_0 \left(A - B R_0^{-1} B^T P_1 \right)}_{\tilde{A}} + \left(A - B R_0^{-1} B^T P_1 \right)^T P_0 - P_0 B R_0^{-1} B^T P_0 + \underbrace{\left(Q_0 + C^T C \right)}_{\tilde{Q}} = 0 \quad (13.65)$$

or, equivalently:

$$P_0 \tilde{A} + \tilde{A}^T P_0 - P_0 B R_0^{-1} B^T P_0 + \tilde{Q} = 0 \quad (13.66)$$

Since a feedback connection, such as $u = -R_0^{-1} B^T P_1 x$, does not change controllability of (A, B) then (\tilde{A}, B) is also controllable. Moreover, since $\tilde{Q} = \tilde{Q}^T > 0$, then the ARE has the unique symmetric positive-definite solution $P_0 = P_0^T > 0$. Finally, using (13.64) in (13.65) gives (13.52) and thus proves the second claim of the theorem.

Choosing the unitary matrix W as in (13.54), while using (13.55), results in

$$\begin{aligned} B^T P_0 B &= B^T C^T W^T \sqrt{R_0} = \sqrt{R_0} \left(R^{-\frac{1}{2}} B^T C^T \right) W^T \sqrt{R_0} \\ &= \sqrt{R_0} V^T \Lambda \underbrace{U^T U}_{I_{m \times m}} V \sqrt{R_0} = \sqrt{R_0} (V^T \Lambda V) \sqrt{R_0} > 0 \end{aligned} \quad (13.67)$$

Note that this particular choice of W supports the established positive-definiteness property of P_0 .

Let us select a unit vector $x \in R^n$. Then,

$$\lim_{v \rightarrow 0} x^T P_v x = \lim_{v \rightarrow 0} x^T [P_0 + O(v)]x = x^T P_0 x \geq \lambda_{\min}(P_0) > 0 \quad (13.68)$$

We know that the ARE solution P_v is invertible for any fixed $v > 0$. Also, from (13.68) it follows that for a sufficiently small $v \geq 0$, the eigenvalues of P_v are bounded away from zero. Therefore, P_v is invertible globally and for any $v \geq 0$. Finally, we note that (13.57) is a direct consequence of (13.50) and (13.53). The proof of the theorem is complete. ■

Let us now make the following substitutions into the ARE (13.43),

$$A := A^T, \quad B := C^T \quad (13.69)$$

The resulting equation becomes

$$P_v A^T + A P_v - P_v C^T R_v^{-1} C P_v + Q_v = 0 \quad (13.70)$$

where according to (13.44)

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) B B^T, \quad R_v = \frac{v}{v+1} R_0 \quad (13.71)$$

The reader may have noticed that such an ARE appears in the design of Kalman filters and Luenberger observers. Substituting (13.71) into (13.70) gives

$$P_v A^T + A P_v - \left(1 + \frac{1}{v} \right) P_v C^T R_0^{-1} C P_v + Q_0 + \left(1 + \frac{1}{v} \right) B B^T = 0 \quad (13.72)$$

or, equivalently

$$P_v A^T + A P_v - P_v C^T R_0^{-1} C P_v + Q_0 + B B^T + \frac{1}{v} [B B^T - P_v C^T R_0^{-1} C P_v] = 0 \quad (13.73)$$

For the parameter-dependent ARE in (13.73), all statements from the Theorem 13.2 can be easily reformulated. These claims are summarized (without proofs) below.

Corollary 13.1 Suppose that all assumptions from Theorem 13.2 hold. Then the unique positive-definite solution P_v of the ARE (13.70), with the weight matrices Q_v and R_v from (13.71), can be represented by the asymptotic expansion (13.50). Moreover, the following statements hold:

1. P_0 and P_1 are symmetric.
2. P_0 is the unique symmetric strictly positive-definite solution of the following algebraic Lyapunov equation:

$$P_0 \left(A - C^T R_0^{-1} C P_1 \right)^T + \left(A - C^T R_0^{-1} C P_1 \right) P_0 + Q_0 = 0 \quad (13.74)$$

3. There exists a unitary matrix $W \in R^{m \times m}$ such that

$$P_0 C^T = B W^T \sqrt{R_0} \quad (13.75)$$

4. The unitary matrix W in (13.53) can be chosen as

$$W = (U V)^T \quad (13.76)$$

where U and V are two unitary matrices, defined by the singular value decomposition,

$$B^T C^T R_0^{-\frac{1}{2}} = U \Lambda V \quad (13.77)$$

and Λ represents the diagonal matrix of the corresponding singular values.

5. P_v is invertible for any $v \geq 0$, and

$$\lim_{v \rightarrow 0} x^T P_v x \geq \lambda_{\min}(P_0) > 0 \quad (13.78)$$

where $\lambda_{\min}(P_0)$ denotes the minimum eigenvalue of P_0 .

6. The following asymptotic relation holds:

$$P_v C^T = B W^T \sqrt{R_0} + O(v), \quad \text{as } v \rightarrow 0 \quad (13.79)$$

Soon in this chapter, we shall use the above statements in our design of MRAC controllers with smooth transient dynamics, but at this moment let us make the following remark: Since P_v is invertible for any $v \geq 0$, one can define the matrix inverse,

$$\tilde{P}_v = P_v^{-1} \quad (13.80)$$

and analyze its property using an asymptotic expansion in the form:

$$\tilde{P}_v = \tilde{P}_0 + O(v), \quad \text{as } v \rightarrow 0 \quad (13.81)$$

Substituting (13.81) into $\tilde{P}_v P_v = I_{n \times n}$ gives

$$I_{n \times n} = \tilde{P}_v P_v = (\tilde{P}_0 + O(v))(P_0 + O(v)) = \tilde{P}_0 P_0 + O(v), \quad \text{as } v \rightarrow 0 \quad (13.82)$$

Consequently,

$$I_{n \times n} = \lim_{v \rightarrow 0} \tilde{P}_v P_v = \tilde{P}_0 P_0 \quad (13.83)$$

and therefore

$$\left\{ \left[\tilde{P}_0 = P_0^{-1} \right] \Rightarrow \left[P_v^{-1} = P_0^{-1} + O(v) \right] \right\}, \quad \text{as } v \rightarrow 0 \quad (13.84)$$

Using (13.84) and (13.79) yields

$$\begin{aligned} C^T &= \tilde{P}_v (B W^T \sqrt{R_0} + O(v)) = \tilde{P}_v B W^T \sqrt{R_0} + (P_0^{-1} + O(v)) O(v) \\ &= \tilde{P}_v B W^T \sqrt{R_0} + O(v) \end{aligned} \quad (13.85)$$

and as a result, we obtain the asymptotic relation,

$$\tilde{P}_v B = C^T R_0^{-\frac{1}{2}} W + O(v) \quad (13.86)$$

which we shall employ in the design of adaptive output feedback controllers in Chap. 14. This concludes our asymptotic analysis of parameter-dependent ARE solutions.

13.4 System Dynamics and Control Problem Formulation

We are going to design an MRAC controller, with an observer-like reference dynamics, for a class of nonlinear MIMO uncertain dynamical systems in the form:

$$\begin{aligned} \underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} &= \underbrace{\begin{pmatrix} 0_{m \times m} & C_p \\ 0_{n_p \times m} & A_p \end{pmatrix}}_A \underbrace{\begin{pmatrix} e_{yI} \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} 0_{m \times m} \\ B_p \end{pmatrix}}_B \Lambda \underbrace{\left(u + \overbrace{\Theta_d^T \Phi_d(x_p)}^{d(x_p)} \right)}_{u + \overbrace{\Theta_d^T \Phi_d(x_p)}^{d(x_p)}} \\ &\quad + \underbrace{\begin{pmatrix} -I_{m \times m} \\ 0_{n_p \times m} \end{pmatrix}}_{B_{\text{ref}}} y_{\text{cmd}} \\ y_{\text{reg}} &= \underbrace{\begin{pmatrix} 0_{m \times m} & C_p \end{pmatrix}}_C x \end{aligned} \quad (13.87)$$

The dynamics (13.87) include an n_p -dimensional open-loop system with m control inputs u and m regulated outputs y_{reg} . This is the original plant, whose state is $x_p \in R^{n_p}$. The plant is augmented by the m -dimensional integrated output tracking error dynamics, $\dot{e}_{yI} = y_{\text{reg}} - y_{\text{cmd}} = C_p x_p - y_{\text{cmd}}$, where $C_p \in R^{m \times n_p}$ is a known constant matrix. The order of the complete system (13.87) is $n = n_p + m$. In addition, $x \in R^n$ is the system state vector, $u \in R^m$ is the control input, $y_{\text{reg}} \in R^m$ is the regulated output, $y_{\text{cmd}} \in R^m$ is the commanded signal for y_{reg} to follow, $d(x_p) = \Theta_d^T \Phi_d(x_p) \in R^m$ is a nonlinear state-dependent matched parametric uncertainty, $\Theta_d \in R^{N \times m}$ is the matrix of unknown constant “true” parameters, and $\Phi_d(x_p) \in R^N$ is the known N -dimensional regressor vector, whose components are locally Lipschitz-continuous in x , i.e., there exists a finite positive known constant $0 < L_{\Phi_d} < \infty$, such that for any $(x_1, x_2) \in R^{n_p}$ from a bounded neighborhood of the origin, the following inequality holds:

$$\|\Phi_d(x_1) - \Phi_d(x_2)\| \leq L_{\Phi_d} \|x_1 - x_2\| \quad (13.88)$$

Also in (13.87), $A \in R^{n \times n}$, $B \in R^{n \times m}$, $B_{\text{ref}} \in R^{n \times m}$, and $C \in R^{m \times n}$ are constant known matrices, while $\Lambda \in R^{m \times m}$ is a constant diagonal unknown matrix with strictly positive diagonal elements.

Our choice of the process dynamics (13.87) is largely motivated by aerospace applications, where x_p models six-degrees-of-freedom of an airborne platform, and $d(x_p)$ represents uncertainties in the vehicle aerodynamic moments. By definition, the moment uncertainties appear together with the system control inputs, thus enforcing the matching conditions needed to justify mere existence of a control solution. Moreover, control actuator uncertainties, control effectiveness reduction, and other control failures are modeled by an unknown constant matrix Λ . Finally, inclusion of the integrated output tracking error $\dot{e}_{yI} = C_p x_p - y_{\text{cmd}}$ into the open-loop system leads to the extended system formulation (13.87). This inclusion is

optional, yet it allows the designer to explicitly account for baseline controllers with integral feedback, and it also allows to avoid feedforward terms in a control solution. Other dynamics, such as structural notch filters, sensors, and actuators, can also be added in the formulation of the extended open-loop system.

In order to control a dynamical system such as (13.87), we need the nominal system (no uncertainties) to be controllable.

Assumption 13.1 The nominal system matrix pair (A_p, B_p) is controllable.

It is well known that controllability of (A_p, B_p) , coupled with the rank condition,

$$\text{rank} \begin{pmatrix} A_p & B_p \\ C_p & 0_{p \times m} \end{pmatrix} = n_p + m = n \quad (13.89)$$

ensures controllability of the extended pair (A, B) .

Disregarding the system uncertainties, we form the ideal reference model dynamics,

$$\dot{x}_{\text{ref ideal}} = A_{\text{ref}} x_{\text{ref ideal}} + B_{\text{ref}} y_{\text{cmd}} \quad (13.90)$$

where

$$A_{\text{ref}} = A - B \underbrace{\left(R_{\text{ref}}^{-1} B^T P_{\text{ref}} \right)}_{K_{\text{lqr}}^T} \quad (13.91)$$

is Hurwitz, K_{lqr}^T is the baseline linear quadratic regulator (LQR) feedback gain, P_{ref} is the unique symmetric positive-definite solution of the ARE,

$$P_{\text{ref}} A + A^T P_{\text{ref}} - P_{\text{ref}} B R_{\text{ref}}^{-1} B^T P_{\text{ref}} + Q_{\text{ref}} = 0 \quad (13.92)$$

and $(Q_{\text{ref}}, R_{\text{ref}})$ are some appropriately chosen symmetric positive-definite matrices. Using the LQR design is not a requirement here. This is simply our preferred way to formulate ideal reference models and embed basic performance specifications into the system. Due to the inclusion of the integrated tracking error in (13.87), the DC gain of the reference model (13.90) is unity. Consequently, if $\Lambda = I_{m \times m}$ and $d(x) = 0_{m \times 1}$, then the LQR linear state feedback control $u_{\text{lqr}} = -K_{\text{lqr}}^T x$ enforces global exponential stability of the ideal reference model (13.90), and makes the regulated output $y_{\text{reg}}(t)$ track any bounded command $y_{\text{cmd}}(t)$ with bounded errors. Note that for a step-input command, the LQR controller provides global exponential tracking with zero-steady-state errors. Also, it is easy to see that such a choice of the reference model enforces the Model Matching Conditions stated below.

Assumption 13.2 (Model Matching Conditions) Given a Hurwitz matrix A_{ref} and an unknown constant positive-definite diagonal matrix Λ , there exists a constant possibly unknown gain matrix K_x such that

$$A_{\text{ref}} = A - B \Lambda K_x^T \quad (13.93)$$

We shall note that existence of K_x is guaranteed for any controllable pair (A, B) and any non-singular matrix Λ . In particular, relations (13.91) and (13.93) imply:

$$K_x = K_{\text{lqr}} \Lambda^{-1} \quad (13.94)$$

Using (13.93), we rewrite the system dynamics (13.87) in the form,

$$\dot{x} = A_{\text{ref}} x + B \Lambda \left(u + \underbrace{\left[K_x^T x + \Theta_d^T \Phi_d(x_p) \right]}_{\Theta^T} \right) + B_{\text{ref}} y_{\text{cmd}} \quad (13.95)$$

$$\quad \quad \quad \left(\underbrace{\left(K_x^T \Theta_d^T \right)}_{\Theta^T} \underbrace{\left(\begin{array}{c} x \\ \Phi_d(x_p) \end{array} \right)}_{\Phi(x)} \right)$$

and get:

$$\dot{x} = A_{\text{ref}} x + B \Lambda (u + \Theta^T \Phi(x)) + B_{\text{ref}} y_{\text{cmd}} \quad (13.96)$$

The control goal of interest is bounded tracking of y_{cmd} in the presence of the system parametric uncertainties $\{\Lambda, \Theta\}$. Specifically, we need to find a control input u such that the regulated output $y_{\text{reg}} = C x \in R^m$ tracks any bounded time-varying command $y_{\text{cmd}}(t) \in R^m$ with bounded errors, while the rest of the signals in the corresponding closed-loop system remain bounded. In addition, we shall require smooth and quantifiable transient characteristics in the closed-loop dynamics.

13.5 Observer-Like Model Reference Adaptive Control

Similar to (13.9) and for the system dynamics (13.96), we consider a Luenberger-like reference model in the form:

$$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + \boxed{L_v(x - x_{\text{ref}})}_{\text{Error Feedback Term}} + B_{\text{ref}} y_{\text{cmd}} \quad (13.97)$$

where $x_{\text{ref}} \in R^n$ is the reference model state and $L_v \in R^{n \times n}$ is the error feedback gain, parameterized by a positive scalar $v > 0$, (to be defined). This is the CRM modification of the original ideal reference model (13.90).

The system control input u is selected as a nonlinear state feedback in the form:

$$u = -\hat{\Theta}^T \Phi(x) \quad (13.98)$$

Substituting (13.98) into the system dynamics (13.96) gives

$$\dot{x} = A_{\text{ref}} x - B \Lambda \underbrace{(\hat{\Theta} - \Theta)}_{\Delta\Theta}^T \Phi(x) + B_{\text{ref}} y_{\text{cmd}} \quad (13.99)$$

where $\Delta\Theta \in R^{N \times m}$ denotes the matrix of parameter estimation errors.

In what follows, we are going to select $(L_v, \hat{\Theta})$ such that the system state x globally asymptotically tracks x_{ref} -the state of the CRM (13.97), and so $y_{\text{reg}} \xrightarrow{t \rightarrow \infty} y_{\text{ref}}$. Also, we will show that x_{ref} tracks $x_{\text{ref ideal}}$, which in turn implies that $y_{\text{ref}} \xrightarrow{t \rightarrow \infty} y_{\text{ref ideal}}$. Furthermore, since the output of the ideal reference model follows its command, $y_{\text{ref ideal}} \rightarrow y_{\text{cmd}}$, with bounded errors, and $y_{\text{reg}} \xrightarrow{t \rightarrow \infty} y_{\text{ref}} \xrightarrow{t \rightarrow \infty} y_{\text{ref ideal}}$, then the system-regulated output y_{reg} will also track y_{cmd} with bounded errors. This argument constitutes our control design strategy.

We begin by choosing adaptive laws for $\hat{\Theta}$, so that x globally asymptotically tracks x_{ref} , in the presence of the system uncertainties. Let,

$$e = x - x_{\text{ref}} \quad (13.100)$$

denote the state tracking error. Subtracting (13.97) from (13.99), gives the system transient dynamics:

$$\dot{e} = (A_{\text{ref}} - L_v)e - B \Lambda \Delta\Theta^T \Phi(x) \quad (13.101)$$

We choose the error feedback gain L_v as,

$$L_v = P_v R_v^{-1} \quad (13.102)$$

where $P_v = P_v^T > 0$ is the unique solution of the following ARE,

$$P_v A_{\text{ref}}^T + A_{\text{ref}} P_v - P_v R_v^{-1} P_v + Q_v = 0 \quad (13.103)$$

with the ARE weight matrices (Q_v, R_v) can be selected as in (13.71),

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) B B^T, \quad R_v = \frac{v}{v+1} R_0 \quad (13.104)$$

using a constant parameter $v > 0$ and symmetric positive-definite matrices, Q_0 and R_0 , of the corresponding dimensions. It is possible to define the weight matrices in a simpler form.

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) I_{n \times n}, \quad R_v = \frac{v}{v+1} I_{n \times n} \quad (13.105)$$

We can also use the squaring-up method [8] from Chap. 6. In this case, there are no finite transmission zeros in the original nor in the squared-up system. The absence of the finite transmission zeros is due to state feedback. Therefore, none can be assigned via the squaring-up method. However, the use of the squared-up \bar{B} matrix in (13.104) has numerical benefits. This is our preferred method and so, we change (13.104) accordingly,

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) \bar{B} \bar{B}^T, \quad R_v = \frac{v}{v+1} R_0 \quad (13.106)$$

where a $B_2 \in R^{n \times (n-m)}$ matrix is computed such that the resulting squared-up matrix $\bar{B} \in R^{n \times n}$ is non-singular.

$$\bar{B} = (B \ B_2), \quad \det(\bar{B}) \neq 0 \quad (13.107)$$

As previously noted, in this case the corresponding squared-up system has no finite transmission zeros.

$m\text{-Inputs}$ \downarrow $n\text{-Outputs} \Rightarrow \begin{pmatrix} A & B \\ I_{n \times m} & 0_{n \times m} \end{pmatrix} \in R^{2n \times (n+m)}$ Original System	$n\text{-Inputs}$ \downarrow $n\text{-Outputs} \Rightarrow \begin{pmatrix} A & \bar{B} \\ I_{n \times m} & 0_{n \times m} \end{pmatrix} \in R^{2n \times 2n}$ Squared-up Systems
---	---

(13.108)

In (13.106), the constant positive parameter v will eventually become our design “tuning knob”, in the sense that small values of v will yield better MRAC transients. However, the corresponding feedback gain L_v will increase at the rate of $\frac{1}{v}$. In fact, we will show that as v tends to zero, the error feedback gain tends to infinity,

$$L_v = \left(1 + \frac{1}{v} \right) P_v = O\left(\frac{1}{v}\right) \quad (13.109)$$

while the solution P_v of the ARE (13.103) tends to a constant positive-definite symmetric matrix P_0 . It is easy to verify that the ARE (13.103) possesses the unique symmetric positive-definite solution P_v . Furthermore, because of (13.103), the observer closed-loop matrix,

$$A_v = A_{\text{ref}} - L_v = A_{\text{ref}} - P_v R_v^{-1} = A_{\text{ref}} - P_v \left(1 + \frac{1}{v} \right) \quad (13.110)$$

satisfies,

$$P_v \underbrace{\left(A_{\text{ref}} - \underbrace{P_v R_v^{-1}}_{L_v} \right)^T}_{A_v} + \underbrace{\left(A_{\text{ref}} - \underbrace{P_v R_v^{-1}}_{L_v} \right)}_{A_v} P_v + P_v R_v^{-1} P_v + Q_v = 0 \quad (13.111)$$

or, equivalently

$$P_v A_v^T + A_v P_v = -P_v R_v^{-1} P_v - Q_v < 0 \quad (13.112)$$

and therefore, A_v is Hurwitz for any $v > 0$.

Since P_v is the unique symmetric positive-definite solution of the ARE (13.103), then the matrix inverse $\tilde{P}_v = P_v^{-1}$ exists for any $v \geq 0$ and the following relation holds:

$$A_v^T \tilde{P}_v + \tilde{P}_v A_v = -R_v^{-1} - \tilde{P}_v Q_v \tilde{P}_v < 0 \quad (13.113)$$

The design task is to choose adaptive laws for $\dot{\Theta}$ so that the tracking error e globally asymptotically tends to the origin. We consider the following Lyapunov function candidate:

$$V(e, \Delta\Theta) = e^T \tilde{P}_v e + \text{trace}\left(\Lambda \Delta\Theta^T \Gamma_\Theta^{-1} \Delta\Theta\right) \quad (13.114)$$

where $\Gamma_\Theta = \Gamma_\Theta^T > 0$ is the adaptation rate. The time derivative of V , along the trajectories of the error dynamics (13.101), can be computed as:

$$\begin{aligned} \dot{V}(e, \Delta\Theta) &= e^T \tilde{P}_v \dot{e} + \dot{e}^T \tilde{P}_v e + 2 \text{trace}\left(\Lambda \Delta\Theta^T \Gamma_\Theta^{-1} \dot{\Theta}\right) \\ &= e^T \tilde{P}_v \left(A_v e - B \Lambda \Delta\Theta^T \Phi(x) \right) + \left(A_v e - B \Lambda \Delta\Theta^T \Phi(x) \right)^T \tilde{P}_v e \\ &\quad + 2 \text{trace}\left(\Lambda \Delta\Theta^T \Gamma_\Theta^{-1} \dot{\Theta}\right) \\ &= e^T \left(\tilde{P}_v A_v + A_v^T \tilde{P}_v \right) e - 2 e^T \tilde{P}_v B \Lambda \Delta\Theta^T \Phi(x) + 2 \text{trace}\left(\Lambda \Delta\Theta^T \Gamma_\Theta^{-1} \dot{\Theta}\right) \end{aligned} \quad (13.115)$$

Because of (13.112) and using the properties of the matrix trace operator, we get:

$$\dot{V}(e, \Delta\Theta) = -e^T \left(R_v^{-1} + \tilde{P}_v Q_v \tilde{P}_v \right) e + 2 \text{trace}\left(\Lambda \Delta\Theta^T \left(\Gamma_\Theta^{-1} \dot{\Theta} - \Phi(x) e^T \tilde{P}_v B \right)\right) \quad (13.116)$$

If the adaptive laws are chosen as

$$\dot{\Theta} = \Gamma_\Theta \Phi(x) e^T \tilde{P}_v B \quad (13.117)$$

then

$$\dot{V}(e, \Delta\Theta) = -e^T \left(R_v^{-1} + \tilde{P}_v Q_v \tilde{P}_v \right) e \leq 0 \quad (13.118)$$

and hence, $V(e, \Delta\Theta)$ is the Lyapunov function for the error dynamics (13.101). For this reason, the tracking error signal e , as well as the parameter error matrix

$\Delta\Theta$, is uniformly bounded in time, that is $(e, \Delta\Theta) \in L_\infty$. Since A_{ref} in (13.97) is Hurwitz by design and $(e, y_{\text{cmd}}) \in L_\infty$ then $(x_{\text{ref}}, \dot{x}_{\text{ref}}) \in L_\infty$, and consequently $x \in L_\infty$. Since the unknown parameters Θ are constant and $\Delta\Theta \in L_\infty$ then $\hat{\Theta} \in L_\infty$. We assumed that the regressor vector $\Phi(x_p)$ is Lipschitz continuous, and we have shown that $(x, \hat{\Theta}) \in L_\infty$. Therefore, from the definition (13.98) it follows that $u \in L_\infty$ and consequently $\dot{x} \in L_\infty$. Also, since $\dot{x}_{\text{ref}} \in L_\infty$ then $\dot{e} \in L_\infty$. Using (13.118) yields

$$\ddot{V}(e, \Delta\Theta) = -2e^T \left(R_v^{-1} + \tilde{P}_v Q_v \tilde{P}_v \right) \dot{e} \in L_\infty \quad (13.119)$$

The function V from (13.114) is lower bounded and has a non-increasing time derivative as in (13.118). Thus, V tends to a limit, as $t \rightarrow \infty$. Also, the function second time derivative is uniformly bounded. Therefore, \dot{V} is a uniformly continuous function of time. Using Barbalat's lemma, we immediately conclude that \dot{V} tends to zero, as $t \rightarrow \infty$. Due to (13.118), we finally arrive at,

$$\lim_{t \rightarrow \infty} \|e(t)\| = 0 \quad (13.120)$$

which proves global asymptotic stability of the tracking error, attained by the adaptive controller (13.98), the adaptive laws (13.117), and the observer-like reference model (13.97).

In order to show that x_{ref} asymptotically tracks $x_{\text{ref ideal}}$, it is sufficient to subtract (13.90) from (13.97), and write the dynamics of the reference model error $e_{\text{ref}} = x_{\text{ref}} - x_{\text{ref ideal}}$.

$$\dot{e}_{\text{ref}} = A_{\text{ref}} e_{\text{ref}} + L_v \underbrace{e(t)}_{o(1)} \quad (13.121)$$

Then (see Exercise 13.2),

$$e_{\text{ref}}(t) = \exp(A_{\text{ref}} t) e_{\text{ref}}(0) + \int_0^t \exp(A_{\text{ref}}(t-\tau)) L_v \underbrace{e(\tau)}_{o(1)} d\tau = o(1) \xrightarrow[t \rightarrow \infty]{} 0 \quad (13.122)$$

We have proven that: $x \xrightarrow[t \rightarrow \infty]{} x_{\text{ref}} \xrightarrow[t \rightarrow \infty]{} x_{\text{ref ideal}}$, and so

$$(y_{\text{reg}}(t) = C x(t)) \xrightarrow[t \rightarrow \infty]{} (y_{\text{ref}}(t) = C x_{\text{ref}}(t)) \xrightarrow[t \rightarrow \infty]{} (y_{\text{ref ideal}}(t) = C x_{\text{ref ideal}}(t)) \rightarrow y_{\text{cmd}}(t) \quad (13.123)$$

In other words, the system-regulated output y_{reg} asymptotically tracks its ideal reference command $y_{\text{ref ideal}}$, and at the same time, y_{reg} also tracks its original command y_{cmd} with bounded errors.

13.6 Transient Dynamics Analysis

Let us now analyze the transient dynamics (13.101). To do that, we shall employ the results from Theorem 13.1 and singular perturbation techniques from Sects. 13.1 and 13.2.

Substituting (13.102) into (13.101), the transient error dynamics can be written as

$$\dot{e} = \underbrace{\left(A_{\text{ref}} - P_v R_v^{-1} \right) e}_{\text{Hurwitz Matrix}} - \underbrace{B \Lambda \Delta \Theta(t)^T \Phi(x(t))}_{\varphi(t) = \text{Uniformly Bounded Function of Time}} \quad (13.124)$$

Using 13.104 gives

$$\dot{e} = \left(A_{\text{ref}} - \left(1 + \frac{1}{v} \right) P_v \right) e - \varphi(t) \quad (13.125)$$

In Sect. 13.3, we have shown that the asymptotic relation

$$P_v = P_0 + O(v), \quad \text{as } v \rightarrow 0 \quad (13.126)$$

holds with a constant positive-definite symmetric matrix P_0 . Then,

$$\dot{e} = \left(A_{\text{ref}} - \left(1 + \frac{1}{v} \right) (P_0 + O(v)) \right) e - \varphi(t) \quad (13.127)$$

or, equivalently

$$v \dot{e} = (v A_{\text{ref}} - (v + 1)(P_0 + O(v)))e - v \varphi(t) \quad (13.128)$$

We can rewrite (13.128) as,

$$\begin{aligned} v \dot{e} &= (v A_{\text{ref}} - (v + 1)(P_0 + O(v)))e - v \varphi(t) \\ &= \left(-P_0 + \underbrace{(v A_{\text{ref}} - v(P_0 + O(v)) - O(v))}_{O(v)} \right) e + v \varphi(t) \\ &= (-P_0 + O(v))e + v \varphi(t) \end{aligned} \quad (13.129)$$

and then compare it to (13.31). Then according to Theorem 13.1, the trajectories of (13.127), satisfy the following asymptotics,

$$e(t) = O\left(e^{-\gamma \frac{t}{v}}\right) + O(v), \quad (v \rightarrow 0) \quad (13.130)$$

uniformly in time, with a positive constant γ , and for all sufficiently small $v > 0$. So, the transient dynamics exponentially decays to a neighborhood of the origin,

at the decay rate no slower than $O\left(e^{-\gamma \frac{t}{v}}\right)$. Moreover, the “diameter” of the convergence set can be made smaller, by choosing sufficiently small v . This argument formally proves our claim about the transient dynamics improvement in MIMO MRAC systems with observer-like reference models.

Similar to the arguments from Sect. 13.2, we can offer an alternative way to analyze the transient dynamics in (13.128). This is a singularly perturbed system, and its dynamics are in the form of (13.31), where v (instead of ε) is the small parameter. So, in order to understand the intricacies of the system behavior, we can employ the singular perturbation arguments yet again. Setting $v = 0$ gives the isolated root $e = 0$ for the corresponding reduced system, which describes asymptotic behavior as $t \rightarrow \infty$, that is for a sufficiently small $v > 0$, the error trajectories converge to a small neighborhood of the manifold $e \equiv 0$, and will evolve near this manifold thereafter.

In order to quantify and characterize the transient dynamics, we need to form the boundary-layer system. These dynamics are formed by “stretching” the time,

$$\tau = \frac{t}{v} \quad (13.131)$$

rewriting (13.128) in the “fast” time scale τ , and then setting $v = 0$. The resulting boundary-layer dynamics

$$\frac{de}{d\tau} = -P_0 e \quad (13.132)$$

are globally exponentially stable, since P_0 is symmetric and positive definite. According to Theorem 13.2, we claim that for a sufficiently small $v > 0$, while starting from an initial time $t_0 \geq 0$, the singularly perturbed system (13.128) has a unique solution $e(t, v)$, defined on $[0, \infty)$, and the asymptotic relation

$$e(t, v) = \bar{e}\left(\frac{t}{v}\right) + O(v) \quad (13.133)$$

holds uniformly on $[0, \infty)$, where $\bar{e}\left(\frac{t}{v}\right)$ is the solution of the boundary-layer system. Since,

$$\bar{e}\left(\frac{t}{v}\right) = \exp(-P_0(t))\bar{e}(0) \quad (13.134)$$

then substituting (13.134) into (13.133), results in:

$$e(t, v) = \exp\left(-P_0\left(\frac{t}{v}\right)\right)(x(0) - x_{\text{ref}}(0)) + O(v) \quad (13.135)$$

This asymptotic relation is conservative. In fact, we have proven that the tracking error $e(t, v)$ asymptotically converges to the origin, starting from any initial condition. Consequently (see Exercise 13.3),

$$\varphi(t) = B \underbrace{\Lambda \left[\Delta \Theta(t)^T \Phi(x(t)) \right]}_{o(1)} = o(1), \quad (t \rightarrow \infty) \quad (13.136)$$

and so, we can rewrite (13.135) as

$$x(t, v) = \boxed{\exp\left(-P_0\left(\frac{t}{v}\right)\right)(x(0) - x_{\text{ref}}(0))} + \boxed{x_{\text{ref}}(t) + O(v)o(1)} \quad (13.137)$$

Transient Dynamics Global Asymptotic Stability

where $o(1)$ is a function of time, defined such that $\lim_{t \rightarrow \infty} o(1) = 0$, while $O(v)$ is a function of v only, and it decays to zero no slower than v .

Let us emphasize again that the asymptotic expansion (13.137) quantifies the transient dynamics due to the adaptive controller (13.98) and (13.117). Indeed, for a sufficiently small $v > 0$, the transients in the error dynamics are described by the linear time-invariant globally exponentially stable system (13.132), whose solution is given by (13.134) and (13.137). The second term in (13.137) defines asymptotic behavior of the tracking error, as $t \rightarrow \infty$. This fact constitutes the main benefit of the error feedback term in the observer-like reference model (13.97). Essentially, using a sufficiently small parameter $v > 0$, ensures quantifiable transient characteristics of the corresponding closed-loop tracking performance, and these transients are given by the first term in (13.137). A summary of the observer-like/CRM design for a state feedback MRAC system is given in Table 13.1.

The system dynamics (13.87) and the corresponding control problem formulations can be modified to include nonparametric uncertainties, such as matched uncertainty approximation errors and bounded possibly non-matched

Table 13.1 Observer-like/CRM state feedback MRAC design summary

Open-loop plant	$\dot{x} = A_{\text{ref}} x + B \Lambda(u + \Theta^T \Phi(x)) + B_{\text{ref}} y_{\text{cmd}}$
Observer-like reference model (aka CRM)	$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + L_v(x - x_{\text{ref}}) + B_{\text{ref}} y_{\text{cmd}}$
State tracking error	$e = x - x_{\text{ref}}$
ARE weights	$Q_v = Q_0 + \left(\frac{v+1}{v}\right) \bar{B} \bar{B}^T, \quad R_v = \frac{v}{v+1} R_0$
Squared-up B matrix	$\bar{B} = \begin{pmatrix} B & B_2 \end{pmatrix}, \quad B_2 \in R^{n \times (n-m)}, \quad \det(\bar{B}) \neq 0$
ARE for MRAC laws	$P_v A_{\text{ref}}^T + A_{\text{ref}} P_v - P_v R_v^{-1} P_v + Q_v = 0$
CRM observer gain	$L_v = P_v R_v^{-1}$
Total control input	$u = -\hat{\Theta}^T \Phi(x)$
MRAC laws	$\dot{\hat{\Theta}} = \Gamma_{\Theta} \Phi(x) e^T P_v^{-1} B$

Table 13.2 Aircraft wings-level trimmed flight conditions in Example 13.2

Alt, ft	Mach	Alpha, degree	Dynamic pressure, psf	True airspeed, fps	Elevator, degree
20,000	0.7	4.5637	278.07	711.17	-4.7153

process noise. In that case, one can use known robustification techniques (i.e., e -modification, σ -modification, and Projection Operator) to (13.135) prove bounded tracking performance and then establish similar to transient characteristics.

Example 13.2 Aircraft Longitudinal Flight Control System with the Observer-Like /CRM Modification We will use longitudinal (pitch) dynamics data extracted from aFltSim high-fidelity aircraft simulation environment, as described in Chap. 1 and Appendix A. The flight condition of interest is defined by Alt = 20,000 ft and Mach = 0.7. The aircraft dynamics are trimmed to wings-level to perform steady-state flight and the corresponding open-loop linear models are generated at the noted flight condition. Table 13.2 shows the trim data.

We are interested in regulating the aircraft vertical acceleration N_z (measured in g-s) using the elevator surface position δ_e in radians, as the system control input. Within the aFltSim environment, open-loop linear models are computed numerically. For this example, the model states consist of the aircraft angle of attack α in radians and the body pitch rate q_b , in radians per second. The aircraft short-period model data are shown below.

$$\text{Short-Period Aircraft Dynamics : } \underbrace{\begin{pmatrix} \dot{\alpha} \\ \dot{q}_b \end{pmatrix}}_{\dot{x}_p} = \underbrace{\begin{pmatrix} -0.492 & 0.982 \\ -4.896 & -0.7284 \end{pmatrix}}_{A_p} \underbrace{\begin{pmatrix} \alpha \\ q_b \end{pmatrix}}_{x_p} + \underbrace{\begin{pmatrix} -0.04528 \\ -7.796 \end{pmatrix}}_{B_p} \underbrace{\delta_e}_{u}$$

$$\text{Measurement Output (States) : } y_{\text{meas}} = x_p$$

$$\text{Regulated Output : } y_{\text{reg}} = N_z = \underbrace{\begin{pmatrix} 10.89 & 0.3988 \end{pmatrix}}_{C_p \text{ reg}} x_p + \underbrace{1.001 u}_{D_p \text{ reg}}$$

The first step is to design a proportional–integral (PI) state feedback servo-controller to regulate N_z . We form the extended system (13.87) without uncertainties and iterate on the LQR weights Q_{ref} and R_{ref} in the ARE equation (13.92) until the resulting closed-loop dynamics yields sensible design targets, such as small undershoot/overshoot values and a reasonably fast rise time, with a loop gain-crossover frequency not degrading stability margins in the presence of a second-order actuator model.

$$\frac{\delta_e}{\delta_{e \text{ cmd}}} = \underbrace{\left(\frac{\omega_{act}^2}{s^2 + 2\xi_{act}\omega_{act}s + \omega_{act}^2} \right)}_{\text{2nd Order Actuator Transfer Function}}$$

The actuator model defines commanded-to-achieved elevator position actuation dynamics. In this example, the actuator natural frequency ω_{act} (rad/s) and the damping ratio ξ_{act} values are set to aircraft representative values.

$$\omega_{act} = 80, \quad \xi_{act} = 0.8$$

We use this model to represent practical limitations due to control actuation dynamics that arise in all flight control applications. So, we design servoloop feedback gains using the short-period aircraft dynamics without the actuator model but analyze the system stability and robustness properties with the model accounted for. The analysis process can include other subsystems such as input/output time delays, structural mode filters and other signal modifications. On a typical aircraft development program, all known subsystems would have to be accounted for during analysis and software testing phases.

After several design iterations, we selected the following LQR weight matrices,

$$Q_{ref} = \begin{pmatrix} 0.0091954 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.50334 \end{pmatrix}, \quad R_{ref} = 1$$

solved ARE (13.92) and computed LQR state feedback gains as defined in (13.91).

$$K_{lqr} = (-0.95893 \ -0.49003 \ -0.71584)$$

Figure 13.5 shows the open-loop (o) and the closed-loop (x) system eigenvalues.

The integrator pole is the fastest mode, while the closed-loop short-period natural frequency and the damping ratio are both increased to provide an agile and damped command tracking performance.

The closed-loop system vertical acceleration response due to step-input command is shown in Fig. 13.6.

The command tracking rise time is about 1.7 s and is sufficiently fast for this application. Figure 13.7 shows the elevator loop gain computed with and without the actuator dynamics.

The elevator loop gain-crossover frequency is about 1 Hz and the corresponding stability margins well exceed the typical gain and phase requirements of 6 dB and 45° , respectively. As seen from the Bode data, at high frequencies above $\omega_{act} = 80$ the actuator dynamics contribute an extra 40 dB/decade gain roll-off and an additional 180 degree of phase loss. This is as expected and does not influence the system relative stability properties. The system MIMO margins are also well above the minimum

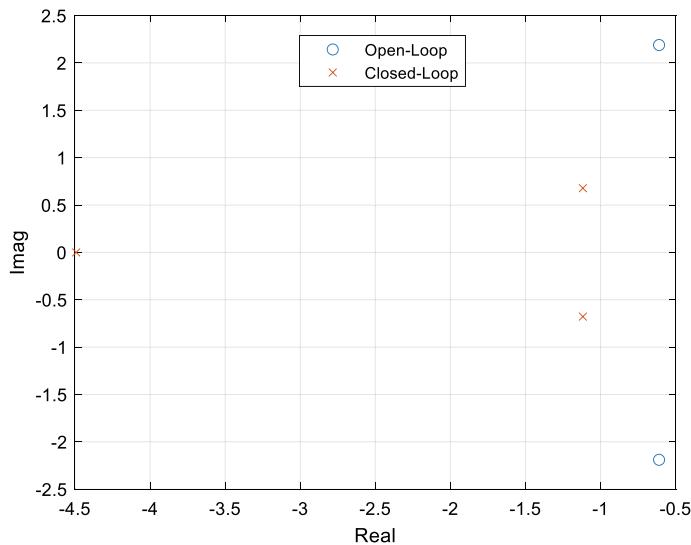


Fig. 13.5 Open-loop (o) and closed-loop (x) eigenvalues in Example 13.2

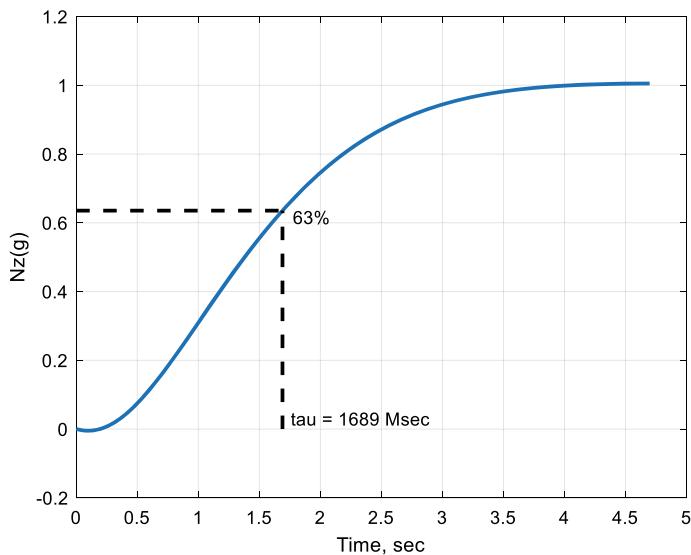


Fig. 13.6 Vertical acceleration step-input response in Example 13.2

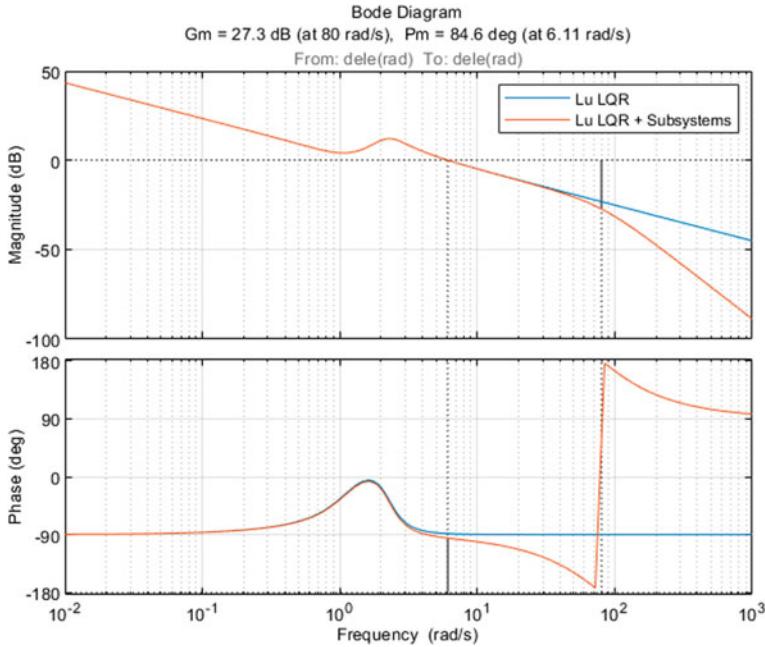


Fig. 13.7 Elevator loop gain and stability margins in Example 13.2

required values. Note that for a SISO system, the MIMO margins are equivalent to the system vector margins. Without the actuator dynamics, the vector gain and phase margins are as predicted by the theory. The MIMO gain margin is infinite, and the MIMO phase margin is above 60°. With the actuator dynamics included, the MIMO gain margin is about 20 dB, while the MIMO phase margin remains above 60°.

The presented time domain and frequency domain metrics indicate a satisfactory design for the baseline LQR PI servo-controller.

The second phase of the overall design process starts with the squaring-up of the system B matrix. We use the method from [8] to compute \bar{B} , as defined in Table 13.1.

$$\bar{B} = (B \ B_2) = \left(\underbrace{\begin{pmatrix} 1.0012 \\ -0.04284 \\ -7.7957 \end{pmatrix}}_B \underbrace{\begin{pmatrix} 0.045284 & 7.7957 \\ 7.8596 & -0.03984 \\ -0.03984 & 1.0014 \end{pmatrix}}_{B_2} \right)$$

Following the design steps from Table 13.1, we select a sufficiently small value for the tuning parameter v , define ARE weights (Q_v, R_v) (13.106), with (Q_0, R_0)

set to identity matrices, solve ARE (13.103) for P_v and compute the CRM feedback gain L_v (13.109).

$$\begin{aligned} v = 0.1 \Rightarrow & Q_v = \begin{pmatrix} 680.55 & 0 & 0 \\ 0 & 680.55 & 0 \\ 0 & 0 & 680.55 \end{pmatrix}, \quad R_v = \begin{pmatrix} 0.091 & 0 & 0 \\ 0 & 0.091 & 0 \\ 0 & 0 & 0.091 \end{pmatrix} \\ \Rightarrow & P_v = \begin{pmatrix} 7.9243 & 0.50869 & -0.024194 \\ 0.50869 & 7.7907 & -0.35307 \\ -0.024194 & -0.35307 & 7.3407 \end{pmatrix} \\ \Rightarrow & L_v = \begin{pmatrix} 87.168 & 5.5956 & -0.26614 \\ 5.5956 & 85.698 & -3.8837 \\ -0.26614 & -3.8837 & 80.747 \end{pmatrix} \end{aligned}$$

Figure 13.8 shows closed-loop simulation data with the actuator dynamics turned on and without uncertainties.

The system-regulated (actual) output is able to follow the output of the CRM driven by external step-input commands. Response differences between the ideal reference model and the CRM are negligible. The baseline control input required for command tracking is shown in Fig. 13.9.

The commanded elevator position and rate data are clearly within reasonable and achievable actuation bounds.

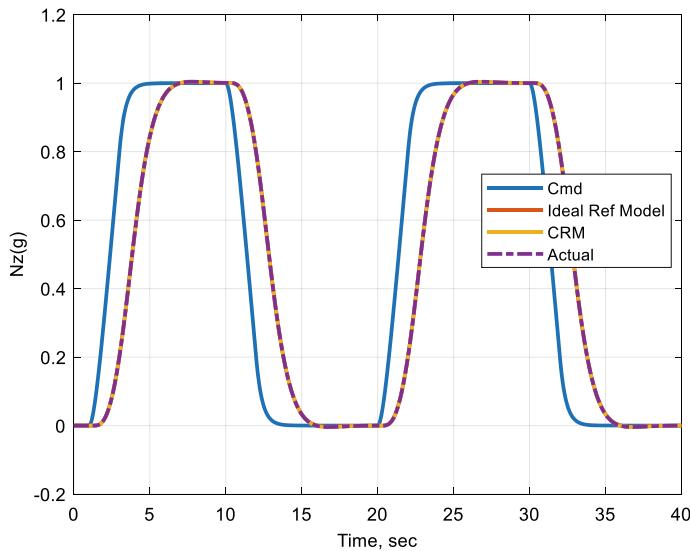


Fig. 13.8 Closed-loop simulation data without uncertainties in Example 13.2

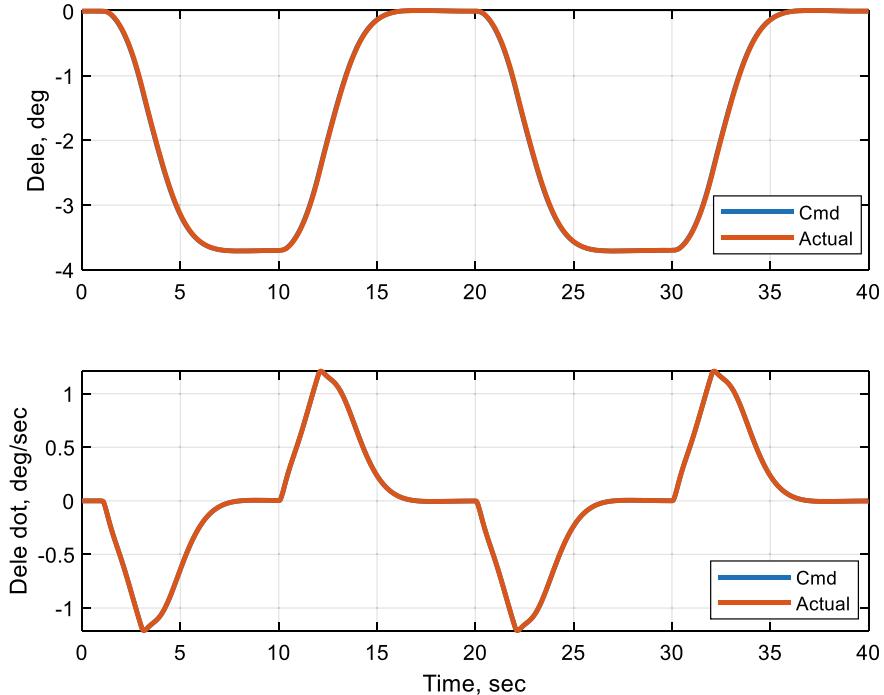


Fig. 13.9 Baseline elevator position (Delete) and elevator rate (Delete dot) data in Example 13.2

The third and final phase in the CRM-based MRAC design consists of selecting adaptation rates $\Gamma_\Theta = \Gamma_\Theta^T > 0$ and the regressor vector $\Phi(x)$ for adaptive laws (13.117).

$$\Gamma_\Theta = \underbrace{92.747}_{\|L_v\|} \begin{pmatrix} 0 & 0 \\ 0 & I_{11 \times 11} \end{pmatrix}, \quad \Phi(x) = \Phi \begin{pmatrix} \alpha, q_b \\ \underbrace{x_p} \end{pmatrix} = \begin{pmatrix} \alpha \\ q_b \\ \phi_1(\alpha) \\ \vdots \\ \phi_N(\alpha) \end{pmatrix}$$

The adaptation rates are set to be proportional to the norm of L_v [9]. This gives a smooth transient-free closed-loop command tracking performance. Adaptation gain on the integrated tracking error is zeroed out since matched uncertainties in the system dynamics (13.87) depend only on the states of the original system.

The regressor vector $\Phi(x)$ has eleven components. It contains two system states (α, q_b) and nine Radial Basis Functions (RBFs).

$$\phi_k(\alpha) = e^{-\frac{(\alpha-\alpha_k)^2}{2\sigma^2}}, \quad k = 1, \dots, (N = 9)$$

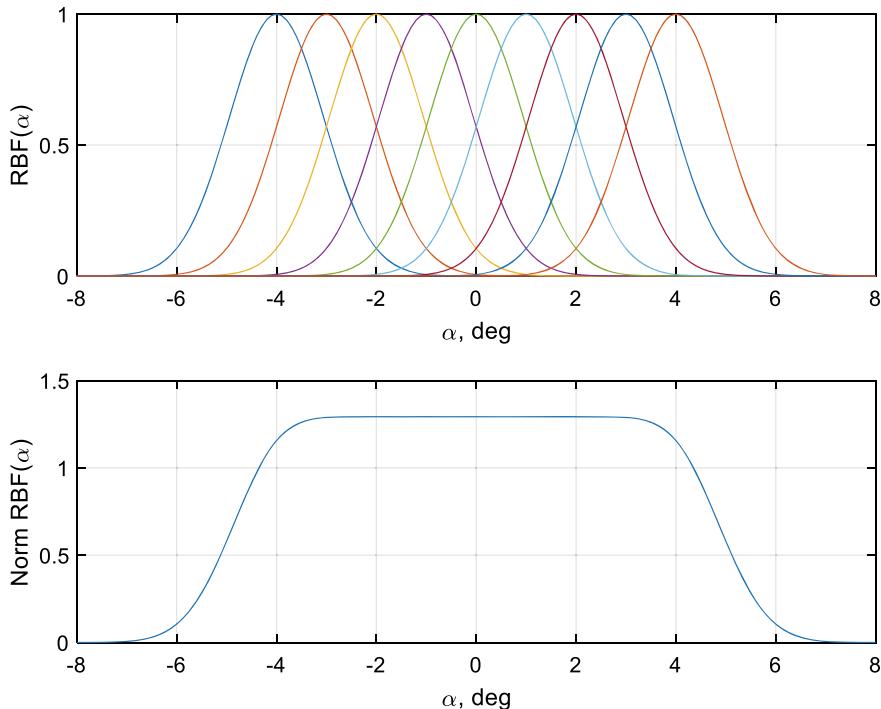


Fig. 13.10 RBF data in Example 13.2

The corresponding nine RBF centers α_k are selected to cover the interval $[-4, 4]$ degree, spaced evenly at 1° from each other. The RBF widths are set to be equal $(\frac{1}{2\sigma^2})$ and are defined such that the “total gain through” the network is constant within the activation area, while it gradually fades out outside of the interval. The RBF data are shown in Fig. 13.10.

For testing purposes, the system matched uncertainties Λ and $d(x_p)$ in (13.87) are selected as follows.

$$\Lambda = 0.5, \quad d\left(\underbrace{\alpha, q_b}_{x_p}\right) = -0.14701\alpha - 0.21475q_b + 0.2\Phi_d(\alpha)$$

This represents a fifty percent loss of control effectiveness, a thirty percent reduction in the LQR feedback gains on α and q_b , as well as inherently nonlinear effects due to α -dependent additive uncertainty $\Phi_d(\alpha)$, Fig. 13.11.

Figure 13.12 presents closed-loop system dynamics operating in the presence of the above defined uncertainties but without adaptation.

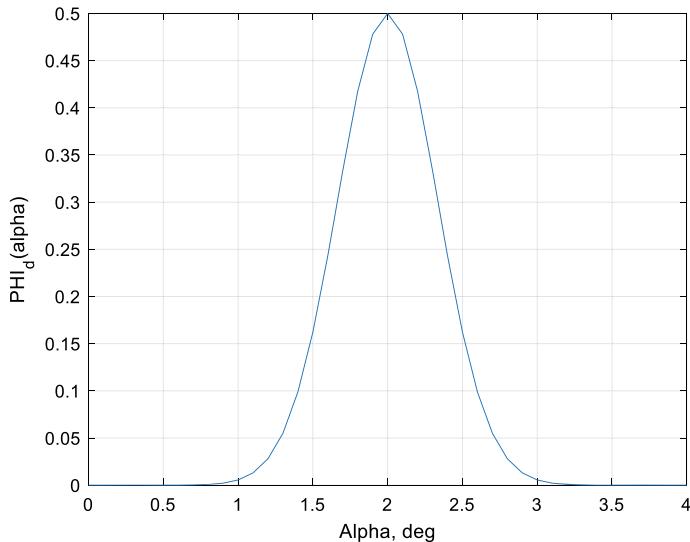


Fig. 13.11 Nonlinear α -dependent additive uncertainty in Example 13.2

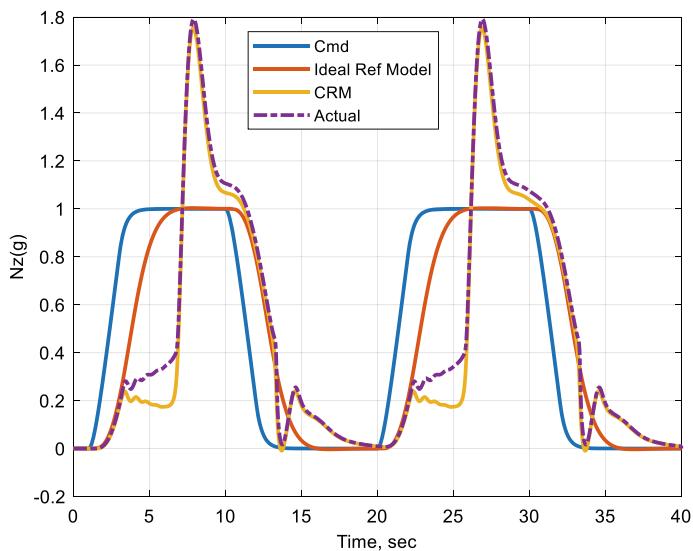


Fig. 13.12 Command tracking in the presence of uncertainties, without adaptive control in Example 13.2

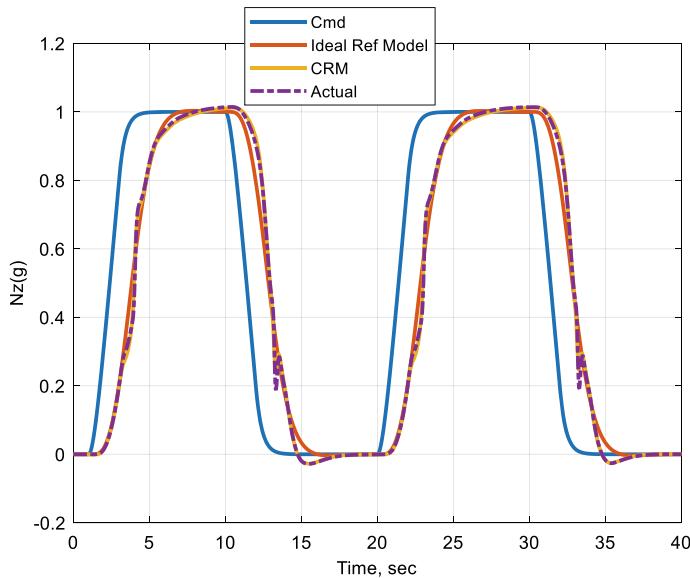


Fig. 13.13 Command tracking in the presence of uncertainties, with adaptive control in Example 13.2

The data show a significant degradation in the closed-loop tracking performance. Turning adaptation on restores the system baseline tracking performance (without uncertainties) sufficiently close, Fig. 13.13.

The total control activity has increased as compared to the baseline, Fig. 13.14. This is the expected behavior. Operating under uncertainties, the control system must find an online solution to stabilize the inner-loop dynamics and to restore baseline command tracking performance without undesirable transients. Achieving the latter is the main benefit of using the CRM modification in MRAC systems. ■

In conclusion, we would like to note that the CRM-based state feedback MRAC design developed in this chapter, with an observer-like reference model modification, can be extended to adaptive output feedback controllers [6]. This topic will be addressed in the next chapter.

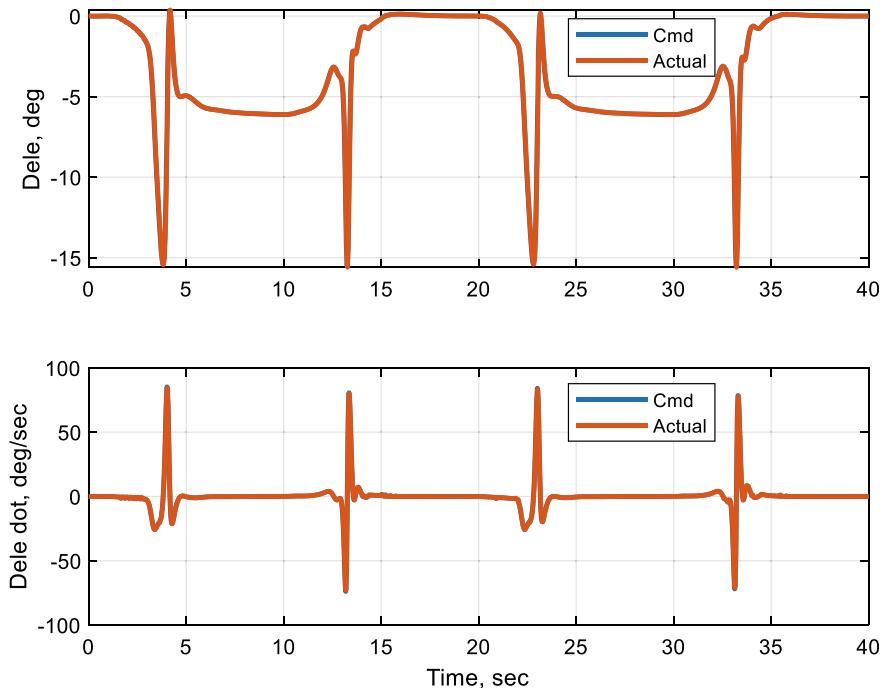


Fig. 13.14 Elevator position (Dele) and elevator rate (Dele dot) data with adaptive control in Example 13.2

13.7 Summary

This section is devoted to the development and analysis of an observer-like (aka CRM) modification to the reference model dynamics formulation, within the MRAC state feedback framework. We drew a parallel between the derived modification and the theory of Luenberger observers. The CRM modification allowed us to quantify and influence transient dynamics for a class of MIMO dynamical systems with adaptive controllers. Overall, the derived design represents a numerically efficient technique for reducing unwanted transient oscillations in state feedback MRAC systems.

13.8 Exercises

Exercise 13.1 Show that if the external command $r(t)$ in (13.1) is continuously differentiable, and its rate $\dot{r}(t)$ is uniformly bounded in time, then the signal $\varphi(t)$ in (13.8) asymptotically tends to zero, as $t \rightarrow \infty$. (*Hint:* Differentiate the error dynamics (13.5) and show that $\ddot{e}(t)$ is uniformly bounded. Then use Barbalat's lemma to establish asymptotic convergence of $\dot{e}(t)$ to zero.)

Exercise 13.2 Prove (13.36). (*Hint:* Show that if (13.42) holds true then $\|z(t)\|$ in (13.40) asymptotically in time tends to zero. Use [3, Lemma 9.6, p. 355] to aid in the proof.)

Exercise 13.3 Show that for the extended dynamics (13.87), driven by the MRAC controller (13.98), the smoothness requirement on the command y_{cmd} can be removed, yet the signal $\varphi(t)$ in (13.136) will tend to zero asymptotically in time. This formally proves validity of using $o(1)$ in the asymptotic relation (13.137).

Exercise 13.4 Consider the aircraft dynamics and the MRAC design from Example 12.2 (Chap. 12). For the same system, design and simulate an MRAC controller with an observer-like dynamics. Compare and discuss the two designs.

References

1. Luenberger DG (1964) Observing the state of a linear system. IEEE Trans Mil Electron MIL-8:74–80
2. Whitaker HP, Yamron J, Kezer A (1958) Design of model reference control systems for aircraft. Rep. R-164, Instrumentation Laboratory, MIT, Cambridge, MA
3. Khalil H (1996) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, p 07458
4. Kevorkian J, Cole JD (1996) Multiple scale and singular perturbation methods. Appl Math Sci 114
5. Murray JD (1984) Asymptotic analysis. Appl Math Sci 48
6. Lavretsky E (2012) Adaptive output feedback design using asymptotic properties of LQG/LTR controllers. IEEE Trans Autom Control 57(6):1587–1591
7. Kailath T (1980) Linear systems. Prentice-Hall Inc., Englewood Cliffs
8. Misra P (1998) Numerical algorithms for squaring-up non-square systems. Part II: General case. In: Proceedings of American control conference, San Francisco, CA
9. Gibson TE, Qu Z, Annaswamy AM, Lavretsky E (2015) Adaptive output feedback based on closed-loop reference models. IEEE Trans Autom Control 60(10):2728–2733



Output Feedback Servomechanism with Observer-Based Loop Transfer Recovery and Adaptive Augmentation

14

In this chapter, we introduce an observer-based adaptive output feedback augmentation design of a baseline robust servo-controller for multi-input–multi-output controllable and observable dynamical systems with matched uncertainties and bounded disturbances. This material is central to our book. It represents a culmination of all previously discussed control methods, with formal quantifiable guarantees of closed-loop stability, robustness, and tracking performance in the presence of system uncertainties. The emphasis is on adaptive controllers that operate based on available output feedback signals (measurements), as oppose to state feedback connections. We assume that the number of the system measured outputs (sensors) is no less than the number of the control inputs (actuators). We call these systems “tall”. If the number of inputs and outputs are the same (square systems), we would require that the system has relative degree one. Such an input–output property might be restrictive for a generic and broader class of tall systems. In what follows, we will be able to alleviate the relative-degree-one restriction by assuming that the system has more outputs than inputs, and that the corresponding output-to-input matrix has full rank. It turns out that in this case, the system can be “squared-up” (i.e., augmented) using pseudo-control signals to yield relative-degree-one minimum phase dynamics. Since the “squaring-up” problem is solvable for any controllable and observable triplet (A, B, C) [1], our proposed adaptive output feedback design is applicable to systems whose regulated output dynamics may be non-minimum phase or have a relative degree higher than one. In its core, our adaptive output feedback design is based on asymptotic properties of linear quadratic Gaussian regulators with loop transfer recovery [2]. In essence, this method combines robust and adaptive controllers in a unified output feedback framework. The design is formally justified, that is, we will be able to formulate sufficient conditions to guarantee closed-loop stability and uniform ultimate boundedness of the corresponding tracking error dynamics. At the end of chapter,

we will offer a flight control design case study to demonstrate key features and benefits of the method. We call this method “Observer Loop Transfer Recovery with Adaptive Augmentation”, or in short OBLTRA.

14.1 Introduction

Adaptive control was at the forefront of automatic control technologies when it was first incorporated in aerospace applications. Unfortunately, the flight mishap of the X-15 aircraft caused a 30-year hiatus for onboard adaptive flight systems [3]. For state accessible systems, adaptive control is now a mature discipline [4, 5]. However, in flight applications, only outputs are available and the adaptive control structure becomes much more complicated. Often in aerospace and other industrial applications, systems are multi-input–multi-output (MIMO) and their dynamics would be “tall” that is the system has more output measurements (sensors) than that of the control inputs (non-square dynamics). Also, in aerospace systems, flight sensors are installed to provide measurements of accelerations and angular rates. An acceleration signal of an aerial vehicle with a tail-mounted surface will have non-minimum phase dynamics from the corresponding control input to the vehicle acceleration, which in many cases will also represent the system controlled output. For tall systems with non-minimum phase dynamics, adaptive control design methods are scarce [4, 5]. For this class of systems, a recently developed adaptive output feedback control methodology is presented [2, 6–9]. These controllers solve the state accessible matching problem when only outputs are accessible and when the system is not necessarily square. The system single-input–single-output (SISO) dynamics from control to the regulated output or to an output measurement are allowed to be non-minimum phase. However, the tall MIMO system transmission zeros must be stable. One of the most important features of this adaptive solution is that it can augment an existing linear baseline controller, a necessary feature for any flight application.

The chapter is organized as follows. Section 14.2 introduces an optimal control problem that will later be used in its dual form to find an optimal observer gain. Section 14.3 discusses the asymptotic nature of the associated optimal control cost function. Section 14.4 analyzes the asymptotic structure of the *Algebraic Riccati Equation* (ARE) associated with the optimal control problem. Section 14.5 introduces and analyzes the stability properties of the proposed adaptive controller. Section 14.6 closes with a brief discussion.

Notation: Throughout the text, R^n represents the Euclidean n -dimensional space and $R^{n \times m}$ denotes the space of all n -by- m matrices, where n and m are integers. For any $x \in R^n$, $\|x\|$ is the Euclidean vector norm of x , and for any $A \in R^{n \times m}$, $\|A\|$ is the induced matrix norm. \mathbb{C}^- symbolizes the open left half of the complex plane. For a parameter-dependent matrix $A_v \in R^{n \times m}$ with $v > 0$, the asymptotics $A_v = A_0 + A_1 v + O(v^2)$ with $v \rightarrow 0$, imply

$\lim_{v \rightarrow 0} \|A_v - A_0 - A_1 v\| = \lim_{v \rightarrow 0} O(v^2) = 0$. It also means that the convergence rate $O(v^2)$ is no slower than v^2 , where the “Big O” symbol comes from the Bachmann–Landau asymptotic-order notation [10].

14.2 Optimal Control with “Cheap” Input and “Expensive” Output

Consider a linear time-invariant (LTI) controllable system in the classical form,

$$\dot{x} = Ax + Bu, \quad y = Cx \quad (14.1)$$

whereby $x \in R^n$ denotes the n -dimensional system state, $u \in R^m$ is the vector of controls, $y \in R^p$ is the regulated output, and the constant matrices (A, B, C, D) are of the corresponding dimensions.

Assumption 14.1 For the system in (14.1):

1. (A, B) is controllable, and (A, C) is observable.
2. The number of the regulated outputs equals the number of the system control inputs, $p = m$.
3. The system vector relative degree is unity: $\det(C^T B) \neq 0$.
4. The transfer function from u to y has all of the transmission zeros in the open left half of the complex plain.

Of interest is the following optimal linear quadratic (LQ) control problem formulation: Find the control policy to minimize the quadratic cost,

$$J_v = \min_u \int_0^\infty (x^T Q_v x + u^T R_v u) dt \quad (14.2)$$

along the trajectories of the system dynamics (14.1) and with the weight matrices,

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) C^T C, \quad R_v = \left(\frac{v}{v+1} \right) R_0 \quad (14.3)$$

defined based on a sufficiently small positive constant $0 < v \ll 1$. Introduce

$$\rho_v = \frac{v}{v+1}. \quad (14.4)$$

Then the v -dependent optimal cost function J_v in (14.2) can be written as

$$J_v = \min_u \int_0^\infty \left(x^T Q_0 x + \frac{1}{\rho_v} y^T y + \rho_v u^T R_0 u \right) dt \quad (14.5)$$

From the definitions (14.2), (14.3), (14.4), and (14.5) it is easy to see that small values of v allow for large (i.e., “cheap”) control efforts to be applied to the system dynamics (14.1) in order to regulate an “expensive” output, such as y in (14.1). In that sense, a fixed value of v defines an optimal balance between cheap control efforts and expensive values of the regulated output.

14.3 Optimal Cost Asymptotic Analysis

What happens to the cost J_v as v gets small? To answer that question, consider the modified cost which is only “cheap” in control,

$$\bar{J}_v = \rho_v J_v = \min_u \int_0^\infty (x^T (\rho_v Q_0 + C^T C)x + \rho_v^2 u^T R_0 u) dt \quad (14.6)$$

for $\rho_v \leq 1$. Clearly for any $0 < \rho_v < 1$, \bar{J}_v exists and is a non-increasing function of v , bounded from below by zero. Thus \bar{J}_v converges to a finite limit, as v tends to zero. To better understand the asymptotic nature of \bar{J}_v , consider the following optimal control cost problem:

$$\bar{J}_0 = \min_u \int_0^\infty (x^T C^T C x) dt \quad (14.7)$$

For the cost function in (14.7) satisfying Assumption (14.1) $\bar{J}_0 = 0$ [1], Fact (4). Furthermore, the limit of \bar{J}_v is equal to the cost in (14.7): $\lim_{v \rightarrow 0} \bar{J}_v = \bar{J}_0$ [1], Facts (1) and (2). In other words, the “maximal achievable accuracy” under the cheap LQ optimal control in (14.6) is zero, which in turn implies $\lim_{v \rightarrow 0} \rho_v x^T(0) P_v x(0) = 0$, for any initial condition $x(0)$ and using the corresponding solution P_v of an Algebraic Riccati Equation (ARE), to be formalized at a later time. Furthermore, it can be shown that (see [1], Eq. 40)

$$\rho_v B^T P_v = O(\rho_v) \quad (14.8)$$

as v tends to zero. This implies that $P_v = O(1)$ and thus

$$\lim_{v \rightarrow 0} P_v = P_0 \quad (14.9)$$

has a well-defined limit P_0 that has full rank. This in turn implies that the maximal achievable cost under the balanced control output LQ problem formulation (14.2), (14.3) is nonzero. The implications of that in the context of the corresponding ARE will be investigated in the next section.

14.4 ARE Asymptotic Analysis

The ARE associated with the LQ problem (14.1), (14.2), (14.3) is

$$P_v A + A^T P_v - \left(1 + \frac{1}{v}\right) P_v B R_0^{-1} B^T P_v + Q_0 + \left(1 + \frac{1}{v}\right) C^T C = 0 \quad (14.10)$$

or, equivalently

$$v(P_v A + A^T P_v + Q_0) + (v+1)\left(C^T C - P_v B R_0^{-1} B^T P_v\right) = 0 \quad (14.11)$$

For $v = 0$,

$$P_0 B R_0^{-1} B^T P_0 = C^T C \quad (14.12)$$

where P_0 is the limit defined in (14.9), and we wish to derive the next term in the asymptotic expansion of P_v .

$$P_v = P_0 + \boxed{\varphi_1(v)P_1} + o(\varphi_1(v)) \quad (14.13)$$

as $v \rightarrow 0$, where the scalar function $\varphi_1(v)$ and the matrix P_1 needs to be determined such that the remainder $(P_v - P_0 - \varphi_1(v)P_1)$ is of order $o(\varphi_1(v))$, that is,

$$\lim_{v \rightarrow 0} \frac{\|P_v - P_0 - \varphi_1(v)P_1\|}{\varphi_1(v)} = 0 \quad (14.14)$$

Using (14.13), consider the following quadratic term expansion:

$$\begin{aligned} P_v B R_0^{-1} B^T P_v &= (P_0 + \varphi_1(v)P_1 + o(\varphi_1(v)))B R_0^{-1} B^T (P_0 + \varphi_1(v)P_1 + o(\varphi_1(v))) \\ &= P_0 B R_0^{-1} B^T P_0 + \varphi_1(v)\left(P_1 B R_0^{-1} B^T P_0 + P_0 B R_0^{-1} B^T P_1\right) \\ &\quad + o(\varphi_1(v)) \end{aligned} \quad (14.15)$$

Substituting (14.13) and (14.15) into (14.11) gives

$$\begin{aligned} v\left((P_0 + \varphi_1(v)P_1 + o(\varphi_1(v)))A + A^T(P_0 + \varphi_1(v)P_1 + o(\varphi_1(v))) + Q_0\right) \\ + (v+1)\underbrace{\left(C^T C - P_0 B R_0^{-1} B^T P_0\right)}_0 - \varphi_1(v)\left(P_1 B R_0^{-1} B^T P_0 + P_0 B R_0^{-1} B^T P_1\right) \\ + o(\varphi_1(v)) \end{aligned} = 0 \quad (14.16)$$

which in turn simplifies to,

$$\begin{aligned} & \left(\frac{v}{v+1} \right) (P_0 A + A^T P_0 + Q_0) + \left(\frac{v}{v+1} \right) \varphi_1(v) (P_1 A + A^T P_1) \\ & - \varphi_1(v) (P_1 B R_0^{-1} B^T P_0 + P_0 B R_0^{-1} B^T P_1) = o(\varphi_1(v)) \end{aligned} \quad (14.17)$$

as $v \rightarrow 0$. Consider two choices for selecting $\varphi_1(v)$.

(1) $\varphi_1(v) = o\left(\frac{v}{v+1}\right)$. In this case as $v \rightarrow 0$,

$$\left(\frac{v}{v+1} \right) (P_0 A + A^T P_0 + Q_0) = o\left(\frac{v}{v+1}\right) \quad (14.18)$$

which is a contradiction, since $\frac{v}{v+1} \neq o\left(\frac{v}{v+1}\right)$.

(2) $\varphi_1(v) = O\left(\frac{v}{v+1}\right)$. Substituting this into (14.17) gives

$$\begin{aligned} & \left(\frac{v}{v+1} \right) (P_0 A + A^T P_0 + Q_0) = o(\varphi_1(v)) - O\left(\left(\frac{v}{v+1}\right)^2\right) (P_1 A + A^T P_1) \\ & - O\left(\frac{v}{v+1}\right) (P_1 B R_0^{-1} B^T P_0 + P_0 B R_0^{-1} B^T P_1) = O\left(\frac{v}{v+1}\right) \end{aligned} \quad (14.19)$$

which is an acceptable asymptotic relation for $0 < v \ll 1$. Although (14.19) does not define $\varphi_1(v)$ uniquely, it specifies the function order of magnitude.

Based on the above arguments, choose $\varphi_1(v) = \frac{v}{v+1}$ and rewrite (14.13).

$$P_v = P_0 + \frac{v}{v+1} P_1 + o\left(\frac{v}{v+1}\right) \quad (14.20)$$

From (14.17),

$$\begin{aligned} & \left(\frac{v}{v+1} \right) (P_0 A + A^T P_0 + Q_0 - P_1 B R_0^{-1} B^T P_0 - P_0 B R_0^{-1} B^T P_1) \\ & + \left(\frac{v}{v+1} \right)^2 (P_1 A + A^T P_1) + o\left(\frac{v}{v+1}\right) = 0 \end{aligned} \quad (14.21)$$

and so

$$P_0 A + A^T P_0 + Q_0 - P_1 B R_0^{-1} B^T P_0 - P_0 B R_0^{-1} B^T P_1 = 0 \quad (14.22)$$

This ARE appears in [6], Eq. (13.52). Here, P_0 is the unique positive-definite symmetric solution and proves the validity of the asymptotic expansion.

$$P_v = P_0 + \frac{v}{v+1} P_1 + o\left(\frac{v}{v+1}\right) \quad (14.23)$$

Note that since $\frac{v}{v+1} = v + O(v^2)$, $o\left(\frac{v}{v+1}\right) = o(v)$ then as $v \rightarrow 0$, (14.23) can be written as

$$P_v = P_0 + v P_1 + O(v^2) + o(v) = P_0 + O(v) \quad (14.24)$$

These formal arguments justify the existence (but not necessarily uniqueness) of an asymptotic expansion in the form of (14.24) for the ARE unique positive-definite symmetric parameter-dependent solution P_v .

14.5 Adaptive Output Feedback Design and Analysis

In this section, we are going to present the adaptive output feedback design methodology that was originally developed in [2] and further improved in [6], where global asymptotic stability of the corresponding tracking error was established. That was a significant improvement of the original design where only uniform ultimate boundedness (UUB) of the tracking error was claimed.

As it stands, the control design method is applicable to achieving bounded command tracking for tall MIMO systems with matched uncertainties, while using output measurements only. The dynamics of interest are,

$$\begin{aligned} \dot{x} &= A x + B \Lambda(u + \Theta^T \Phi(x)) + B_{\text{ref}} y_{\text{cmd}}, \\ y &= C x, \quad y_{\text{reg}} = C_{\text{reg}} x \end{aligned} \quad (14.25)$$

where $A \in R^{n \times n}$, $\Lambda = I_{m \times m}$, $C \in R^{p \times n}$, and $C_{\text{reg}} \in R^{m \times n}$ are known matrices. The system state is $x \in R^n$, and the control input is $u \in R^m$. The system measurements are grouped into $y \in R^p$, the regulated output is $y_{\text{reg}} \in R^m$, and $y_{\text{cmd}} \in R^m$ denotes an external bounded time-varying command for the regulated output y_{reg} to follow. The regulated output dynamics are allowed to be non-minimum phase and have a vector relative degree greater than unity. The system uncertainties are represented by a constant unknown non-singular positive diagonal matrix $\Lambda \in R^{m \times m}$, a constant unknown matrix $\Theta \in R^{N \times m}$, and a known regressor vector $\Phi(x) \in R^N$. It is assumed that the regressor is globally Lipschitz-continuous in x , that is, there exists a finite positive known constant $0 < L_\Phi < \infty$, such that

$$\|\Phi(x_1) - \Phi(x_2)\| \leq L_\Phi \|x_1 - x_2\| \quad (14.26)$$

for any $x_1, x_2 \in R^n$. Other suppositions enabling the design are given below.

Assumption 14.2

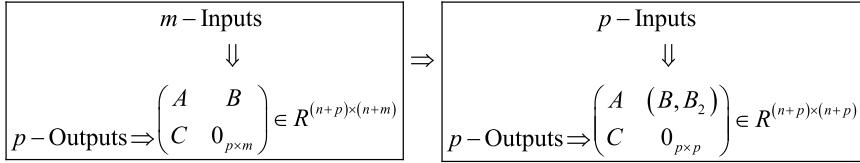
1. (A, B) is controllable and $\text{rank } B = m$ (B has full column rank).
2. (A, C) is observable and $\text{rank } C = p$ (C has full row rank).
3. The number of measured outputs is greater than the number of control inputs ($p > m$), and $\text{rank}(C B) = m$.
4. The MIMO dynamics (tall) from input to regulated and measured outputs are minimum phase.

The first two assumptions are standard in dynamics and control. The third assumption is very common for most practical systems in aerospace, automotive, and other industries, where the outputs (sensors) and the inputs (actuators) are defined by the vehicle designer. The sensor/actuator suites are placed at specific locations on the vehicle, in order to achieve desired input–output characteristics [11, 12]. The assumed full rank condition does not constitute a restriction on the system regulated output $y_{\text{reg}} = C_{\text{reg}} x$. These limitations are placed on the system measured output signals that are selected by the system architects to produce a controllable and observable vehicle configuration. On the other hand, the system regulated output is often not selectable, and its dynamics are allowed to be non-minimum phase and have a high relative degree.

The formulation (14.25) implies that the regulated output is also available as a measurement and that its tracking error dynamics have been embedded into the system. For example, adding an integrated regulation error would yield a system like (14.25). Other formulations leading to similar dynamics are possible. The regulated output can be excluded from the set of the system output measurements. Such an extension is very similar to the design of observer-based control of unknown outputs in observable linear systems, and as such, it will not be covered here.

Under the first and the third assumptions, the squaring-up method from [13] can be applied to find a constant matrix $B_2 \in R^{n \times (p-m)}$, such that $\det(C \bar{B}) \neq 0$ with $\bar{B} = (B \ B_2)$, and the transfer function $C(s I_{n \times n} - A)^{-1} \bar{B}$ becomes minimum phase (i.e., the system transmission zeros are located in \mathbb{C}^-). The squaring-up problem for a non-square linear MIMO system, with m inputs and ($m < p$) outputs, can be stated as follows [13]: “Given the state matrix $A \in R^{n \times n}$, the input matrix $B \in R^{n \times m}$, and the output matrices $C \in R^{p \times n}$, $D \in R^{p \times m}$, with ($n > m$, $p > m$), determine pseudo-input matrices $B_2 \in R^{n \times (p-m)}$ and $D_2 \in R^{p \times (p-m)}$, such that the resulting square system with p inputs and p outputs has its transmission zeros in the open left half complex plane, \mathbb{C}^- ”. In [13], several constructive algorithms for solving the squaring-up problem are given. For systems without feedforward connections, $D = 0_{p \times m}$, and in this case, the squaring-up problem is reduced to finding a pseudo-input matrix $B_2 \in R^{n \times (p-m)}$, such that the resulting square system has its transmission zeros in \mathbb{C}^- (Fig. 14.1).

The added pseudo-control columns B_2 are “fictitious”, in the sense that they do not represent physical inputs in the system. Squaring-up problems may have multiple solutions. Reference [13] gives two sufficient conditions for a solution to

**Fig. 14.1** Squaring-up problem: a conceptual diagram

exist. They are: (a) (A, B) to be controllable and (b) $\text{rank}(C B) = m$. Observe that when a squaring-up solution is found, the resulting system transfer function matrix becomes square, minimum phase, and has unity relative degree. However, the squaring-up paradigm does not restrict the system regulated output to be minimum phase, nor does it require unity relative degree of the regulated output dynamics.

The squaring-up procedure represents one of the essential components in the design process for adaptive output feedback controllers with observer-like reference models. The design proceeds as follows. Let $Q_0 \in R^{n \times n}$ and $R_0 \in R^{m \times m}$ be symmetric and positive definite. Choose $v > 0$, define two symmetric positive-definite parameter-dependent weight matrices,

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) \bar{B} \bar{B}^T, \quad R_v = \frac{v}{v+1} R_0 \quad (14.27)$$

and consider the parameter-dependent Algebraic Riccati Equation (ARE).

$$P_v A^T + A P_v - P_v C^T R_v^{-1} C P_v + Q_v = 0 \quad (14.28)$$

Under the stated assumptions, the ARE possesses the unique symmetric positive-definite solution P_v , for any positive v , [14]. The asymptotic behavior of P_v , as the positive constant parameter v tends to zero, is derived and justified in Sect. 14.4. In [6], Chap. 13, Corollary 13.1, using the same exact asymptotic expansion $P_v = P_0 + P_1 v + O(v^2)$, it is shown that the asymptotic relations

$$\begin{aligned} P_v &= P_0 + O(v), \quad P_v^{-1} = P_0^{-1} + O(v), \\ P_v^{-1} \bar{B} &= \underbrace{C^T R_0^{-\frac{1}{2}} W}_{P_0^{-1} \bar{B}} + O(v) \end{aligned} \quad (14.29)$$

take place as $v \rightarrow 0$, with a constant symmetric positive-definite matrix P_0 and $W = (U V)^T$, where the two unitary matrices, U and V , are defined by the singular value decomposition $\bar{B}^T C^T R_0^{-\frac{1}{2}} = U \Sigma V$, and Σ is the diagonal matrix of the corresponding singular values. The first and the second relations in (14.29)

guarantee strict positive definiteness of P_v and P_v^{-1} uniformly in v , while the third relation implies

$$P_v^{-1} B = C^T R_0^{-\frac{1}{2}} W S^T + O(v) \quad (14.30)$$

where $S = (I_{m \times m} \ 0_{(p-m) \times m})$. The total control input is

$$u = u_{bl} + u_{ad} \quad (14.31)$$

where u_{bl} is the baseline controller and u_{ad} is the adaptive incremental control input. With (14.31), the system dynamics (14.25) can be written as,

$$\dot{x} = A x + B u_{bl} + B \Lambda \left(u_{ad} + \underbrace{\Theta^T \Phi(x) + (I_{m \times m} - \Lambda^{-1}) u_{bl}}_{\bar{\Theta}^T \underbrace{\begin{pmatrix} \Phi(x) \\ u_{bl} \end{pmatrix}}_{\bar{\Phi}(x, u_{bl})}} \right) + B_{ref} y_{cmd} \quad (14.32)$$

or equivalently

$$\dot{x} = A x + B u_{bl} + B \Lambda \left(u_{ad} + \bar{\Theta}^T \bar{\Phi}(x, u_{bl}) \right) + B_{ref} y_{cmd} \quad (14.33)$$

Based on (14.33), the state observer dynamics are introduced,

$$\dot{\hat{x}} = A \hat{x} + B u_{bl} + B \hat{\Lambda} \left(u_{ad} + \hat{\Theta}^T \bar{\Phi}(\hat{x}, u_{bl}) \right) + L_v \left(y - \underbrace{C \hat{x}}_{\hat{y}} \right) + B_{ref} y_{cmd} \quad (14.34)$$

where “hats” signify estimated quantities and $\bar{\Theta}$ and $\bar{\Phi}(x, u_{bl})$ represent the extended unknown constant matrix of parameters and the known extended regressor, correspondingly. Also in (14.34), $\hat{x} \in R^n$, $\hat{y} \in R^p$, and $L_v \in R^{n \times p}$ are the observer state, the predicted measured output, and the observer error feedback gain, respectively. The observer gain L_v in (14.34) is chosen optimally,

$$L_v = P_v C^T R_v^{-1} \quad (14.35)$$

where $P_v = P_v^T > 0$ is the unique solution of the Algebraic Riccati Equation (ARE) (14.28). Furthermore because of (14.28), the observer matrix

$$A_v = A - L_v C = A - P_v C^T R_v^{-1} C \quad (14.36)$$

satisfies

$$P_v \underbrace{\left(A - \underbrace{P_v C^T R_v^{-1} C}_{L_v} \right)^T}_{A_v} + \underbrace{\left(A - \underbrace{P_v C^T R_v^{-1} C}_{L_v} \right)}_{A_v} P_v + P_v C^T R_v^{-1} C P_v + Q_v = 0 \quad (14.37)$$

or, equivalently

$$P_v A_v^T + A_v P_v = -P_v C^T R_v^{-1} C P_v - Q_v < 0 \quad (14.38)$$

Consequently, A_v is Hurwitz for any $(\eta, v) > 0$ and

$$A_v^T P_v^{-1} + P_v^{-1} A_v = -C^T R_v^{-1} C - P_v^{-1} Q_v P_v^{-1} < 0 \quad (14.39)$$

The observer dynamics (14.34) motivate the selection of the adaptive signal,

$$u_{ad} = -\hat{\Theta}^T \bar{\Phi}(\hat{x}, u_{bl}) \quad (14.40)$$

in which case these dynamics simplify and turn out to be linear.

$$\dot{\hat{x}} = A \hat{x} + B u_{bl} + L_v(y - \hat{y}) + B_{ref} y_{cmd} \quad (14.41)$$

At this point, the baseline control component is selected in the form of a linear observer-based feedback,

$$u_{bl} = -K_{bl}^T \hat{x} \quad (14.42)$$

with the constant gain $K_{bl} \in R^{n \times m}$ chosen such that

$$A_{ref} = A - B K_{bl}^T \quad (14.43)$$

becomes Hurwitz and has the desired modal characteristics. Using (14.42) in (14.41) gives the closed-loop observer dynamics,

$$\dot{\hat{x}} = A_{ref} \hat{x} + L_v(y - \hat{y}) + B_{ref} y_{cmd} \quad (14.44)$$

which naturally lead to the definition of an exponentially stable ideal (open-loop) reference model in the form

$$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} y_{\text{cmd}}, \quad y_{\text{ref}} = C_{\text{reg}} x_{\text{ref}} \quad (14.45)$$

whose output y_{ref} would be designed via a proper selection of K_{bl} to adequately track bounded commands y_{cmd} with sufficiently small errors. It is evident that the state observer (14.41) does not explicitly depend on $\hat{\Lambda}$. Hence the only parameter to be estimated is $\hat{\Theta}$, which appears in the adaptive control input formulation (14.40). Substituting the latter into the system (14.33) results in

$$\dot{x} = A x + B u_{\text{bl}} - B \Lambda \left(\hat{\Theta}^T \bar{\Phi}(\hat{x}, u_{\text{bl}}) - \bar{\Theta}^T \bar{\Phi}(x, u_{\text{bl}}) \right) + B_{\text{ref}} y_{\text{cmd}} \quad (14.46)$$

The two design parameters $(L_v, \hat{\Theta})$ are selected such that the state \hat{x} of the observer (14.41) tracks the state x of the system (14.46) with bounded errors. Also, it can be shown that \hat{x} tracks the state of the ideal reference model (14.45) x_{ref} with bounded errors, which in turn implies that x tracks x_{ref} , and y_{reg} tracks y_{cmd} , both with bounded errors. This is the overall design outline. In order to get into the specifics, let

$$e_x = \hat{x} - x \quad (14.47)$$

denote the state observation error, which is not available for control design. However, the output observation error

$$e_y = \hat{y} - y = C(\hat{x} - x) = C e_x \quad (14.48)$$

represents the known measured online quantity. Subtracting (14.46) from (14.41) gives the observer error dynamics,

$$\begin{aligned} \dot{e}_x &= \underbrace{(A - L_v C)}_{A_v} e_x + B \Lambda \left(\hat{\Theta}^T \bar{\Phi}(\hat{x}, u_{\text{bl}}) - \bar{\Theta}^T \bar{\Phi}(x, u_{\text{bl}}) \right) \\ &= A_v e_x + B \Lambda \left(\hat{\Theta}^T \bar{\Phi}(\hat{x}, u_{\text{bl}}) - \bar{\Theta}^T \bar{\Phi}(\hat{x}, u_{\text{bl}}) + \bar{\Theta}^T \bar{\Phi}(\hat{x}, u_{\text{bl}}) - \bar{\Theta}^T \bar{\Phi}(x, u_{\text{bl}}) \right) \\ &= A_v e_x + B \Lambda \left(\underbrace{(\hat{\Theta} - \bar{\Theta})^T}_{\Delta \bar{\Theta}} \bar{\Phi}(\hat{x}, u_{\text{bl}}) + \bar{\Theta}^T \underbrace{(\bar{\Phi}(\hat{x}, u_{\text{bl}}) - \bar{\Phi}(x, u_{\text{bl}}))}_{\Delta \bar{\Phi}} \right) \\ &= A_v e_x + B \Lambda \left(\Delta \bar{\Theta}^T \bar{\Phi}(\hat{x}, u_{\text{bl}}) + \bar{\Theta}^T \Delta \bar{\Phi} \right) \end{aligned} \quad (14.49)$$

with the observer gain L_v defined as in (14.35). Next, stable adaptive laws are derived via a quadratic Lyapunov function candidate in the form,

$$V(e_x, \Delta\bar{\Theta}) = e_x^T P_0^{-1} e_x + \text{trace}\left(\Lambda \Delta\bar{\Theta}^T \Gamma_\Theta^{-1} \Delta\bar{\Theta}\right) \quad (14.50)$$

where $\Gamma_\Theta = \Gamma_\Theta^T > 0$ is the adaptation rate matrix, P_0 is defined in (14.29), and

$$\Delta\bar{\Theta} = \hat{\bar{\Theta}} - \bar{\Theta} \quad (14.51)$$

denotes the matrix of parameter estimation errors. Such a Lyapunov function candidate is originally proposed in [15]. It allows to prove asymptotic stability of the error dynamics. Next, an analogous proof but with a slight modification is presented. Differentiating (14.50) along the trajectories of (14.49) gives

$$\begin{aligned} \dot{V}(e_x, \Delta\bar{\Theta}) &= e_x^T \left(A_v^T P_0^{-1} + P_0^{-1} A_v \right) e_x + 2 e_x^T P_0^{-1} B \Lambda \left(\Delta\bar{\Theta}^T \bar{\Phi} + \bar{\Theta}^T \Delta\bar{\Phi} \right) \\ &\quad + 2 \text{trace}\left(\Lambda \Delta\bar{\Theta}^T \Gamma_\Theta^{-1} \dot{\hat{\bar{\Theta}}}\right) \end{aligned} \quad (14.52)$$

Since

$$e_x^T P_0^{-1} B = \underbrace{e_x^T C^T R_0^{-\frac{1}{2}}}_{{e_y^T}} W S^T = e_y^T R_0^{-\frac{1}{2}} W S^T \quad (14.53)$$

then

$$\begin{aligned} \dot{V}(e_x, \Delta\bar{\Theta}) &= e_x^T \left(A_v^T P_0^{-1} + P_0^{-1} A_v \right) e_x + 2 e_y^T R_0^{-\frac{1}{2}} W S^T \Lambda \left(\Delta\bar{\Theta}^T \bar{\Phi} + \bar{\Theta}^T \Delta\bar{\Phi} \right) \\ &\quad + 2 \text{trace}\left(\Lambda \Delta\bar{\Theta}^T \Gamma_\Theta^{-1} \dot{\hat{\bar{\Theta}}}\right) \end{aligned} \quad (14.54)$$

Recalling the asymptotics from (14.29) $P_v^{-1} = P_0^{-1} + O(v)$, while adding and subtracting $e_x^T (A_v^T O(v) + O(v) A_v) e_x$, further gives

$$\begin{aligned} \dot{V}(e_x, \Delta\bar{\Theta}) &= e_x^T \left(A_v^T P_v^{-1} + P_v^{-1} A_v \right) e_x - 2 e_x^T O(v) A_v e_x \\ &\quad + 2 e_y^T R_0^{-\frac{1}{2}} W S^T \Lambda \left(\Delta\bar{\Theta}^T \bar{\Phi} + \bar{\Theta}^T \Delta\bar{\Phi} \right) + 2 \text{trace}\left(\Lambda \Delta\bar{\Theta}^T \Gamma_\Theta^{-1} \dot{\hat{\bar{\Theta}}}\right) \end{aligned} \quad (14.55)$$

The ARE (14.39) can be written as,

$$A_v^T P_v^{-1} + P_v^{-1} A_v = -C^T R_v^{-1} C - P_v^{-1} Q_v P_v^{-1} \quad (14.56)$$

Substituting (14.56) into (14.55) yields

$$\begin{aligned}\dot{V}(e_x, \Delta\bar{\Theta}) &= -e_y^T R_v^{-1} e_y - e_x^T P_v^{-1} Q_v P_v^{-1} e_x - 2 e_x^T O(v) A_v e_x \\ &\quad + 2 e_y^T R_0^{-\frac{1}{2}} W S^T \Lambda (\Delta\bar{\Theta}^T \bar{\Phi} + \bar{\Theta}^T \Delta\bar{\Phi}) + 2 \text{trace} \left(\Lambda \Delta\bar{\Theta}^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \right)\end{aligned}\quad (14.57)$$

Note that,

$$A_v e_x = (A - P_v C^T R_v^{-1} C) e_x = A e_x - \frac{v+1}{v} P_v C^T R_0^{-1} e_y \quad (14.58)$$

and consequently

$$\begin{aligned}\dot{V}(e_x, \Delta\bar{\Theta}) &= -e_y^T R_v^{-1} e_y - e_x^T P_v^{-1} Q_v P_v^{-1} e_x - 2 e_x^T O(v) A e_x \\ &\quad + 2 e_x^T O(1) P_v C^T R_0^{-1} e_y \\ &\quad + 2 e_y^T R_0^{-\frac{1}{2}} W S^T \Lambda (\Delta\bar{\Theta}^T \bar{\Phi} + \bar{\Theta}^T \Delta\bar{\Phi}) \\ &\quad + 2 \text{trace} \left(\Lambda \Delta\bar{\Theta}^T \Gamma_\Theta^{-1} \dot{\hat{\Theta}} \right)\end{aligned}\quad (14.59)$$

Since,

$$e_y^T R_0^{-\frac{1}{2}} W S^T \Lambda \Delta\bar{\Theta}^T \bar{\Phi}(\hat{x}, u_{bl}) = \text{trace} \left(\Lambda \Delta\bar{\Theta}^T \bar{\Phi}(\hat{x}, u_{bl}) e_y^T R_0^{-\frac{1}{2}} W S^T \right) \quad (14.60)$$

then using the projection-based adaptive laws [16],

$$\dot{\hat{\Theta}} = \text{Proj} \left(\hat{\Theta}, -\Gamma_{\bar{\Theta}} \bar{\Phi}(\hat{x}, u_{bl}) e_y^T R_0^{-\frac{1}{2}} W S^T \right) \quad (14.61)$$

results in

$$\begin{aligned}\dot{V}(e_x, \Delta\bar{\Theta}) &= -e_y^T R_v^{-1} e_y - e_x^T P_v^{-1} Q_v P_v^{-1} e_x - 2 e_x^T O(v) A e_x \\ &\quad + 2 e_x^T O(1) P_v C^T R_0^{-1} e_y + 2 e_y^T R_0^{-\frac{1}{2}} W S^T \Lambda \bar{\Theta}^T \Delta\bar{\Phi}\end{aligned}\quad (14.62)$$

and consequently

$$\begin{aligned}\dot{V}(e_x, \Delta\bar{\Theta}) &\leq -\frac{v+1}{v} \lambda_{\min}(R_0^{-1}) \|e_y\|^2 - \left(\lambda_{\min}(P_v^{-1} Q_v P_v^{-1}) + 2 O(v) \|A\| \right) \|e_x\|^2 \\ &\quad + 2 \underbrace{\left(O(1) \|R_0^{-1} C P_v\| + \|R_0^{-\frac{1}{2}} W S^T \Lambda \bar{\Theta}^T\|_{L_{\bar{\Phi}}} \right)}_{c(v)} \|e_x\| \|e_y\|\end{aligned}\quad (14.63)$$

where $c(v)$ is a v -dependent constant of order $O(v)$. Inequality (14.63) can also be written as

$$\dot{V}(e_x, \Delta\bar{\Theta}) \leq -z^T \underbrace{\begin{pmatrix} \frac{v+1}{v} \lambda_{\min}(R_0^{-1}) & -c(v) \\ -c(v) & (\lambda_{\min}(P_v^{-1} Q_0 P_v^{-1}) + 2 O(v) \|A\|) \end{pmatrix}}_{T_v} z \quad (14.64)$$

with $z = (\|e_y\| \|e_x\|)^T$. It is easy to see that for sufficiently small positive values of v , the symmetric matrix T_v becomes strictly positive definite. In fact, it is sufficient to select v such that

$$(v+1)\lambda_{\min}(R_0^{-1})(\lambda_{\min}(P_v^{-1} Q_0 P_v^{-1}) + 2 O(v) \|A\|) > v c^2(v) \quad (14.65)$$

Based on (14.65), a conservative but simplified condition can also be imposed.

$$\lambda_{\min}(R_0^{-1})(\lambda_{\min}(P_v^{-1} Q_0 P_v^{-1}) + 2 O(v) \|A\|) > v c^2(v) \quad (14.66)$$

Since (14.66) is true for $v = 0$ and both sides of the inequality are continuous in v , there must exist $0 < v_0 \ll 1$ such that for all $0 < v \leq v_0$, the inequality (14.66) takes place, and so the time derivative of the Lyapunov function (14.50), along the trajectories of the error dynamics (14.49), is globally non-positive.

$$\dot{V}(e_x, \Delta\bar{\Theta}) \leq -(\|e_y\|^2 + \|e_x\|^2)\lambda_{\min}(T_v) \leq 0, \quad \forall 0 < v \leq v_0 \ll 1$$

This argument proves Lyapunov stability of the coupled system (14.49), (14.61) equilibrium and UUB of all internal signals. Asymptotic stability of the estimation error follows from application of the Barbalat's lemma. Consider dynamics of the observer tracking error,

$$e = \hat{x} - x_{\text{ref}} \quad (14.67)$$

Subtracting (14.45) from (14.44) gives

$$\dot{e} = A_{\text{ref}} e - L_v C e_x \quad (14.68)$$

Since $e_x(t) \xrightarrow[t \rightarrow \infty]{} 0$ and A_{ref} is Hurwitz then the observer tracking error asymptotically tends to zero, $e(t) \xrightarrow[t \rightarrow \infty]{} 0$. Finally,

$$\|x(t) - x_{\text{ref}}(t)\| = \left\| \underbrace{(\hat{x}(t) - x(t))}_{e_x(t)} - \underbrace{(\hat{x}(t) - x_{\text{ref}}(t))}_{e(t)} \right\| \leq \|e_x(t)\| + \|e(t)\| \xrightarrow[t \rightarrow \infty]{} 0 \quad (14.69)$$

and consequently, the state $x(t)$ of the original system (14.33) tracks the state $x_{\text{ref}}(t)$ of the open-loop reference model (14.45), globally and asymptotically.

The design synopsis, encapsulating the system dynamics and the control equations, is given in Table 14.1.

This is a direct adaptive model reference output feedback control augmentation solution for a class of tall MIMO dynamics, whose transmission zeros are required to be stable but individual (SISO) transfer functions are allowed to be non-minimum phase. Next, we shall demonstrate applicability and practicality of this method via a case study.

Table 14.1 Robust and adaptive output feedback control design summary

Open-loop plant	$\dot{x} = Ax + B \Lambda(u + \Theta^T \Phi(x)) + B_{\text{ref}} y_{\text{cmd}}$
Measured outputs	$y = Cx$
Baseline control	$u_{\text{bl}} = -K_{\text{bl}}^T \hat{x}$
State observer	$\dot{\hat{x}} = A\hat{x} + Bu_{\text{bl}} + L_v(y - \hat{y}) + B_{\text{ref}} y_{\text{cmd}}, \quad \hat{y} = C\hat{x}$
Observer gain	$L_v = P_v C^T R_v^{-1}$
Squared-up B -matrix	$\bar{B} = \begin{pmatrix} B & B_2 \end{pmatrix} \Rightarrow \begin{cases} \det(C \bar{B}) \neq 0 \\ \text{zeros}[C(sI_{n \times n} - A)^{-1} \bar{B}] \in \mathbb{C}^- \end{cases}$
ARE weights	$Q_v = Q_0 + \left(\frac{v+1}{v}\right) \bar{B} \bar{B}^T, \quad R_v = \frac{v}{v+1} R_0$
Algebraic Riccati equation	$P_v A^T + A P_v - P_v C^T R_v^{-1} C P_v + Q_v = 0$
Output tracking error	$e_y = \hat{y} - y$
Extended regressor	$\bar{\Phi}(x, u_{\text{bl}}) = (\Phi^T(x) \ u_{\text{bl}}^T)^T$
Output selection matrix for adaptive laws	$S = \begin{pmatrix} I_{m \times m} & 0_{m \times (p-m)} \end{pmatrix}$
Singular value decomposition	$\bar{B}^T C^T R_0^{-\frac{1}{2}} = U \Sigma V$
Unitary matrix	$W = (U \ V)^T$
Projection-based MRAC laws	$\dot{\hat{\Theta}} = \text{Proj}\left(\hat{\Theta}, -\Gamma_{\bar{\Theta}} \bar{\Phi}(\hat{x}, u_{\text{bl}}) e_y^T R_0^{-\frac{1}{2}} W S^T\right)$
Adaptive increment	$u_{\text{ad}} = -\hat{\Theta}^T \bar{\Phi}(\hat{x}, u_{\text{bl}})$
Total control input	$u = u_{\text{bl}} + u_{\text{ad}}$

14.6 Adaptive Flight Control of a Flexible Transport Aircraft

In this case study, we shall design robust and adaptive output feedback controllers for longitudinal dynamics of a large transport aircraft with flexible structure. Specifically, we will design, analyze, and simulate an observer-based loop transfer recovery controller (OBLTR) with a direct adaptive model reference control augmentation. The OBLTR design details and mathematical foundations were presented in Chap. 6. We also gave a summary of the OBLTR design at the beginning of this chapter.

The aircraft data (wings-level cruise configuration) are taken from [17], where all linear displacements, velocities, and accelerations are given in meters (m), m/s, and m/s², while all angles and angular rates are in radians (rad) and rad/s, respectively.

The vehicle open-loop model (plant) includes a short-period mode and four structural bending modes. Each mode is described by a complex-conjugate pair of eigenvalues. The system state $x_p \in R^{10 \times 1}$ consist of the vehicle angle of attack α , body axis pitch rate q_b , four structural mode positions $(\xi_i)_{i=1,\dots,4}$, and their rates $(\eta_i)_{i=1,\dots,4}$.

$$x_p = (\alpha \ q_b \ \xi_1 \ \eta_1 \ \xi_2 \ \eta_2 \ \xi_3 \ \eta_3 \ \xi_4 \ \eta_4)^T$$

There are two horizontal control surfaces available: (1) the elevator (δ_e) (an aft-mounted tail surface) and (2) the canard (δ_c) (a forward-mounted surface). The aircraft dynamics also depend on the vertical gust velocity vector $w_g \in R^{3 \times 1}$. The gust enters the plant at three different locations, along the vehicle center line. So, the aircraft longitudinal dynamics

$$\dot{x}_p = A_p x_p + B_p \delta + B_g w_g$$

are driven by the two-dimensional control input $\delta = (\delta_e \ \delta_c)^T$ and by the three-dimensional gust input w_g .

We have modified the original data to make the open-loop system unstable in pitch. This necessitates a control action to restore and maintain basic stability of the vehicle. The open-loop plant matrices are

$$A_p = \begin{pmatrix} -1.60 & 1 & -1.1811 & -0.1181 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6.57 & -2.446 & -1.8130 & 1.1805 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -7.196 & -0.445 & -56.82 & -5.53 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1.349 & 0.2466 & 0 & 0 & -231.52 & -1.712 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -2.093 & 0.242 & 0 & 0 & 0 & 0 & -408.86 & -2.679 & -10.71 & -0.518 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0.3073 & 0.05588 & 0 & 0 & 0 & 0 & -1.24 & -0.176 & -390.1 & -0.474 \end{pmatrix}$$

$$B_p = \begin{pmatrix} -0.070 & 3.726 & 0 & 0.572 & 0 & -0.465 & 0 & -0.582 & 0 & -0.112 \\ -0.006 & -0.28 & 0 & 0.019 & 0 & -0.054 & 0 & -0.0532 & 0 & 0.035 \end{pmatrix}^T$$

$$B_g = \begin{pmatrix} 0.0042 & 0.06 & 0 & 0.0105 & 0 & 0.0065 & 0 & -0.0045 & 0 & -0.0021 \\ 0.0037 & -0.0417 & 0 & 0.0393 & 0 & 0.0039 & 0 & 0.0101 & 0 & -0.0009 \\ 0.0012 & -0.056 & 0 & -0.0086 & 0 & 0.0059 & 0 & 0.0064 & 0 & 0.0012 \end{pmatrix}^T$$

The system output measurements include the pitch rate q and vertical accelerations $(a_{z_i})_{i=1,2,3}$ from three distinct nodes on the vehicle centerline. The pitch rate is measured near the vehicle center of gravity (cg). The first vertical acceleration is taken near the tip of the aircraft nose, the second is near cg, and the third acceleration measurement comes from an aft cg location. These are the same three locations where the vertical gust w_g enters the system dynamics. So, the system measured output vector is,

$$y_p = C_p x_p + D_p u + D_{p,g} w_g$$

with

$$y_p = (q_b \ a_{z1} \ a_{z2} \ a_{z3})^T$$

and the output matrices shown below.

$$C_p = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -32.65 & -28.04 & 49.93 & 2.37 & -1405.41 & -10.39 & -1707.61 & -13.09 & -4472.18 & -7.53 \\ -11.69 & -35.68 & 65.02 & 6.27 & 117.33 & 0.87 & 895.25 & 5.94 & 191.7 & 1.34 \\ 3.68 & -38.04 & 153.63 & 17.68 & -739.42 & -5.47 & -821.74 & -5.66 & 672.66 & -1.83 \end{pmatrix}$$

$$D_p = \begin{pmatrix} 0 & -1.02 & 0 & -1.2 \\ 0 & -12.55 & -1.23 & 3.69 \end{pmatrix}^T, \quad D_{p,g} = \begin{pmatrix} 0 & 0.072 & -0.978 & -2.184 \\ 0 & 0.47948 & 1.2092 & 2.0474 \\ 0 & 0.3424 & 1.3224 & 2.448 \end{pmatrix}^T$$

The system regulated output is the second acceleration measurement near cg a_{z2} ,

$$\begin{aligned} y_{\text{reg}} = a_{z_2} &= \underbrace{C_p(2, :) x_p}_{C_p \text{ reg}} + \underbrace{D_p(2, :) u}_{D_p \text{ reg}} + \underbrace{D_{p,g}(2, :) w_g}_{D_{p,g} \text{ reg}} \\ &= C_p \text{ reg } x_p + D_p \text{ reg } u + D_{p,g} \text{ reg } w_g \end{aligned}$$

where $(2, :)$ refers to the second rows of the above matrices.

In order to regulate y_{reg} , we shall blend the two control surfaces, the elevator δ_e and the canard δ_c , and create a single longitudinal virtual control input.

$$u = 0.5 (\delta_e - \delta_c)$$

Such a control mixing is standard in aerospace systems. It is called “control allocation”. To incorporate u into the system dynamics, we introduce control allocation matrix $G = (0.5 \ -0.5)^T$ and arrive at the open-loop plant dynamics in the form,

$$\dot{x}_p = A_p x_p + (B_p G)u + B_g w_g$$

with the virtual control input u and with its corresponding B -matrix.

$$(B_p G) = (-0.032 \ 2.003 \ 0 \ 0.2765 \ 0 \ -0.2055 \ 0 \ -0.2644 \ 0 \ -0.0385)^T$$

Our first step is to create the desired reference model dynamics, and our tool of choice is the OBLTR method. In particular, we shall design a baseline output feedback controller $u (= u_{\text{bl}})$, such that $y_{\text{reg}} = a_{z_2}$ adequately tracks its commanded value $y_{\text{cmd}} = a_{z_2 \text{ cmd}}$, while operating only on the system output measurements. Toward that, we augment the linear plant dynamics with the integrated tracking error signal,

$$\dot{e}_I = y_{\text{reg}} - y_{\text{cmd}}$$

and arrive at the extended open-loop system,

$$\underbrace{\begin{pmatrix} \dot{e}_I \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0 & C_p \text{ reg} \\ 0 & A_p \end{pmatrix}}_A \underbrace{\begin{pmatrix} e_I \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} D_p G \\ B_p G \end{pmatrix}}_B u + \underbrace{\begin{pmatrix} -1 \\ 0 \end{pmatrix}}_{B_{\text{cmd}}} y_{\text{cmd}} + \underbrace{\begin{pmatrix} D_{p,g} \\ B_{p,g} \end{pmatrix}}_{B_g} w_g$$

with the output measurements,

$$\begin{aligned} y &= \begin{pmatrix} e_I \\ y_p \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0_{1 \times 10} \\ 0_{4 \times 1} & C_p \end{pmatrix}}_C \underbrace{\begin{pmatrix} e_I \\ x_p \end{pmatrix}}_x + \underbrace{\begin{pmatrix} 0 \\ D_p G \end{pmatrix}}_D u + \underbrace{\begin{pmatrix} 0 \\ D_p G \end{pmatrix}}_{D_g} w_g \\ &= C x + D u + D_g w_g \end{aligned}$$

that include the integrated vertical acceleration tracking error and the original system outputs. Adding the integrated output error will allow us to design a control input without command-feedforward connections. In the context of the extended system, the regulated output can be expressed as

$$y_{\text{reg}} = \underbrace{\begin{pmatrix} 0 & C_{p \text{ reg}} \end{pmatrix}}_{C_{\text{reg}}} x + \underbrace{D_{p \text{ reg}}}_D u + \underbrace{D_{p g \text{ reg}}}_D w_g = C_{\text{reg}} x + D_{\text{reg}} u + D_{g \text{ reg}} w_g$$

We now proceed to the design of an LQR (Proportional + Integral) (PI) state feedback controller. After several design iterations, we have selected the LQR weights as

$$Q_{\text{lqr}} = \text{diag}(0.005 \ 0 \ 0 \ 0 \ 0.0001 \ 0 \ 0.0001 \ 0 \ 0.0001 \ 0 \ 0.0001), \quad R_{\text{lqr}} = 1$$

Next, we solve the ARE,

$$P_{\text{lqr}} A + A^T P_{\text{lqr}} - P_{\text{lqr}} B R_{\text{lqr}}^{-1} B^T P_{\text{lqr}} + Q_{\text{lqr}} = 0$$

for P_{lqr} , compute the LQR state feedback gains,

$$K_{\text{lqr}} = R_{\text{lqr}}^{-1} B^T P_{\text{lqr}}$$

and examine closed-loop primary poles of the resulting closed-loop A -matrix,

$$A_{\text{ref}} = A - B K_{\text{lqr}}$$

to ensure that adequate natural frequencies (less than 1 Hz) and damping ratios (no less than 0.6) are achieved.

The next step is the design of a full-order state observer. Per OBLTR design, we exclude output measurements with nonzero D -matrices and select the integrated tracking error e_I and the pitch rate q_b as the two measurements for the observer design. Consequently, the extended open-loop plant has one input u and two outputs y . According to our design methodology, we need to square-up the system dynamics, that is to say: We need to add one pseudo-input by building a matrix $B_2 \in \mathbb{R}^{11 \times 2}$ to enforce the squaring-up conditions

$$\bar{B} = (B \ B_2) \Rightarrow \begin{cases} \det(C \bar{B}) \neq 0 \\ \text{zeros}[C(s I_{n \times n} - A)^{-1} \bar{B}] \in \mathbb{C}^- \end{cases}$$

Using the squaring-up method from Sect. 14.5 gives

$$\begin{aligned}\bar{B}^T &= \begin{pmatrix} B & B_2 \end{pmatrix}^T \\ &= \begin{pmatrix} -1.23 & -0.032 & 2.003 & 0 & 0.2765 & 0 & -0.2055 & 0 & -0.2644 & 0 & -0.0385 \\ -1.5631 & 0.076391 & -0.95987 & -0.04296 & 0.024962 & -0.04992 & 0.91006 & -0.172 & -0.11698 & -0.013159 & -1.212 \end{pmatrix}\end{aligned}$$

We can now verify that the squared-up system,

$$\begin{array}{c} \text{2-Inputs} \\ \Downarrow \\ \text{2-Outputs} \Rightarrow \begin{pmatrix} A & \bar{B} \\ C & 0_{2 \times 2} \end{pmatrix} \in R^{13 \times 13} \end{array}$$

with one fictitious pseudo-control column B_2 , satisfies the two squaring-up conditions: (1) The system is minimum phase; and (2) the system relative degree is unity.

With the selected matrix \bar{B} , we choose the observer weights similar to (14.27),

$$Q = Q_0 + \left(\frac{\nu + 1}{\nu} \right) \bar{B} \bar{B}^T, \quad R = \frac{\nu}{\nu + 1} R_0$$

with,

$$Q_0 = I_{11 \times 11}, \quad R_0 = \begin{pmatrix} 1000 & 0 \\ 0 & 1 \end{pmatrix}, \quad \nu = 0.04$$

solve the ARE

$$P_\nu A^T + A P_\nu - P_\nu C^T R_\nu^{-1} C P_\nu + Q = 0$$

for P_ν , compute the steady-state Kalman filter gain $L_\nu = R_\nu^{-1} C P_\nu$, and write the state observer dynamics,

$$\begin{aligned}\dot{\hat{x}} &= A \hat{x} + B u_{\text{bl}} + B_{\text{cmd}} y_{\text{cmd}} + L_\nu (y - \hat{y}) \\ \hat{y} &= C \hat{x} + D u_{\text{bl}}\end{aligned}$$

with the baseline control input,

$$u_{\text{bl}} = -K_{\text{lqr}}^T \hat{x}$$

while utilizing the LQR-optimal state feedback gains K_{lqr} applied to the observer states \hat{x} .

Let us comment on the choice of \bar{B} and on our selection of the tuning parameter ν . As we have previously discussed, \bar{B} turns the extended open-loop system with one input and two outputs into a (2×2) minimum phase system whose relative

degree is unity. This is the OBLTR squaring-up procedure, where we have added one fictitious input into the second column of \bar{B} . The squaring-up modification allows us to recover the LQR state feedback margins at the plant-input breakpoint and, at the same time, enforce the needed (for adaptive laws) asymptotics (14.29). The latter is achieved by setting the tuning knob v to be sufficiently small. However, if v becomes too small, then the observer gains may grow large, which is undesirable since the system noise sensitivity may increase. This is a tradeoff in our design: We need to find v small enough but not too small. Also note that our selection of (\bar{B}, v) is by no means unique, yet it presents a straightforward way to recover optimal LQR state feedback margins (at the system input), with the assigned crossover frequencies, obtain reasonably small observer gains, and enforce the asymptotic relation (14.29).

With the selected pair (\bar{B}, v) , we recover the gain and phase margins of the optimal LQR state feedback controller (Fig. 14.2).

For generality, we included second-order actuator dynamics as a subsystem into analysis but not in the design. The actuator natural frequency and damping ratio are: $\omega_{act} = 60$ rps and $\xi_{act} = 0.8$. The gain and phase margins at the virtual input (including the observer dynamics) are very close to those of the LQR state feedback controller. Also, values of the achieved margins are quite reasonable for the selected transport aircraft configuration.

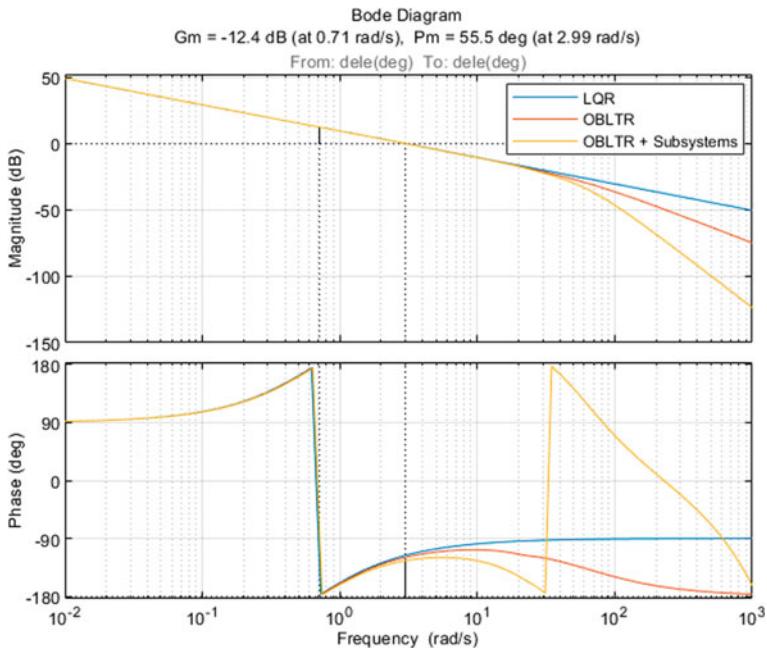


Fig. 14.2 LQR and OBLTR loop gain margins at virtual control input in Sect. 14.6

Note that the loop gain of the OBLTR controller rolls off faster than the loop gain of the LQR state feedback system. So, the dynamic OBLTR solution is less sensitive to modeling uncertainties, and it has better disturbance rejection properties than the LQR state feedback. Of course, the faster roll-off at high frequencies in the OBLTR design is attributed to adding the dynamics of the state observer.

With the baseline OBLTR controller in the loop, the closed-loop system becomes,

$$\begin{aligned} \text{Closed-Loop System : } & \dot{x} = A x - B K_{\text{lqr}}^T \hat{x} + B_{\text{cmd}} y_{\text{cmd}} + B_g w_g \\ \text{State Observer : } & \dot{\hat{x}} = (A - B K_{\text{lqr}}^T) \hat{x} + B_{\text{cmd}} y_{\text{cmd}} + L_v(y - \hat{y}) \\ \text{Measurements : } & y = (C - D K_{\text{lqr}}^T)x \\ \text{Estimated Measurements : } & \hat{y} = (C - D K_{\text{lqr}}^T)\hat{x} \\ \text{Regulated Output : } & y_{\text{reg}} = C_{\text{reg}} x - D_{\text{reg}} K_{\text{lqr}}^T \hat{x} + D_g w_g \end{aligned}$$

and the resulting closed-loop eigenvalues are placed well within practical bounds that would be representative of a large transport aircraft in a cruise configuration.

Without uncertainties, the closed-loop system tracking performance is satisfactory. Representative data are shown in Fig. 14.3.

The baseline OBLTR controller forces the regulated output (vertical acceleration near cg) $y_{\text{reg}} = a_{z_2}$ to track its commanded values, and the required control effort lies well within the bandwidth of a typical aircraft actuation system.

We have also tested the OBLTR baseline system closed-loop tracking using other command shapes. All the results have shown adequate performance. For all these reasons, the closed-loop baseline system becomes our reference model for adaptive control to achieve and maintain, if and when uncertainties are present in the system dynamics. Specifically, we define the reference model matrices,

$$\begin{aligned} A_{\text{ref}} &= A - B K_{\text{lqr}}^T, & B_{\text{ref}} &= B_{\text{cmd}} \\ C_{\text{ref}} &= C - D K_{\text{lqr}}^T, & D_{\text{ref}} &= 0 \end{aligned}$$

and write the reference model dynamics in the form of (14.45).

$$\dot{x}_{\text{ref}} = A_{\text{ref}} x_{\text{ref}} + B_{\text{ref}} z_{\text{cmd}}, \quad y_{\text{ref}} = C_{\text{reg}} x_{\text{ref}}$$

These are the desired dynamics. In other words, this is exactly how we want our system to respond to external commands. Note that the state observer remains the same, as in (14.41).

$$\dot{\hat{x}} = A_{\text{ref}} \hat{x} + L_v(y - \hat{y}) + B_{\text{ref}} y_{\text{cmd}}$$

Before proceeding further, a remark is in order. As derived, our adaptive MRAC design is applicable to systems whose measured and regulated output have no

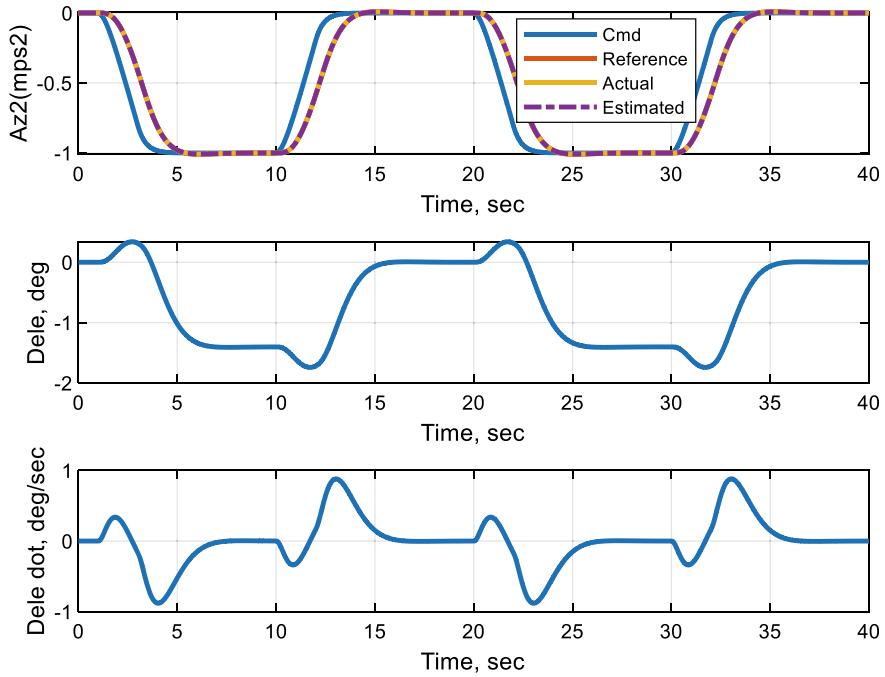


Fig. 14.3 Baseline OBLTR output tracking performance (without uncertainties) in Sect. 14.6

direct feedforward control connections, that is, our method is applicable to systems with the zero D -matrix in the measurements but not in the regulated outputs. It is possible to broaden our design methodology to cover systems with feedforward control connections in their outputs. Derivations of such a method are similar to what we have presented in this chapter, and because of their similarity, we choose to omit formal proofs.

Let us now test the baseline system performance in the presence of uncertainties. We reduce the baseline controller gains by 70%, set $\Lambda = 0.5$, and also introduce a matched nonparametric alpha-dependent uncertainty in the form of a Gaussian (an RBF), centered at $\alpha_c = 2^\circ$, with the RBF sigma-width set to 0.011636, and with the function peak magnitude of -0.25 .

With these uncertainties active and operating under the baseline controller only, the closed-loop system tracking performance degrades significantly (Fig. 14.4).

It is evident from the test data shown in Fig. 14.3 that the baseline OBLTR controller is unable to adequately track the reference acceleration command signal.

In order to recover the desired baseline closed-loop performance, we add an adaptive output feedback u_{ad} , as shown in Table 14.1. We select adaptation rate matrix $\Gamma_{\bar{\Theta}}$ to be diagonal. Then, the adaptive laws can be written as,

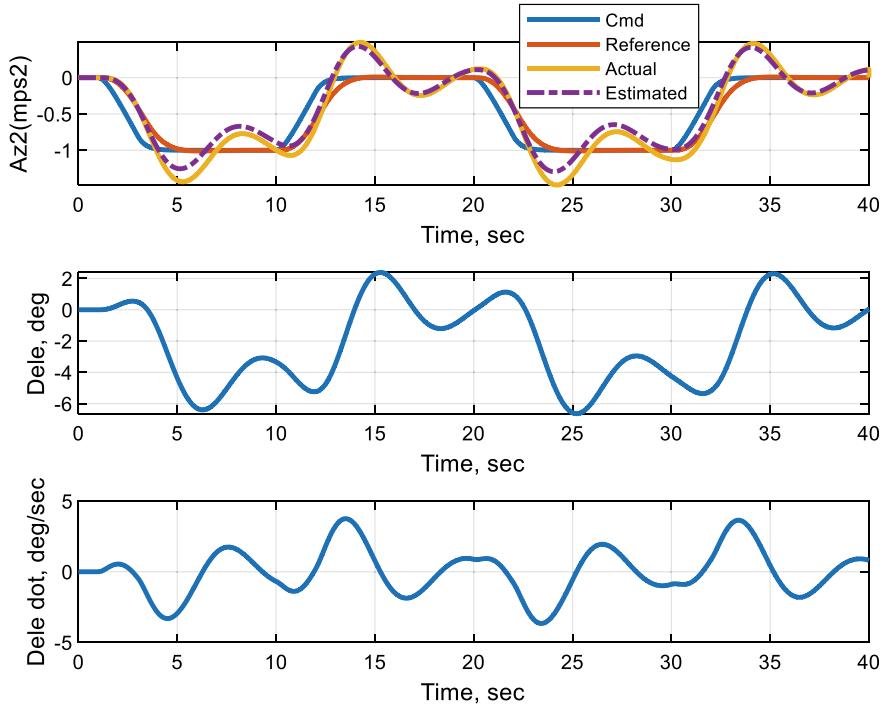


Fig. 14.4 Baseline OBLTR controller tracking with uncertainties in Sect. 14.6

$$\begin{aligned}\dot{\hat{\Theta}} &= \text{Proj} \left(\hat{\Theta}, -\Gamma_{\Theta} \Phi(\hat{x}) e_y^T R_0^{-\frac{1}{2}} W S^T \right) \\ \dot{\hat{K}}_u &= \text{Proj} \left(\hat{K}_u, -\Gamma_u u_{\text{bl}} e_y^T R_0^{-\frac{1}{2}} W S^T \right)\end{aligned}$$

where $(\Gamma_{\Theta}, \hat{K}_u)$ are the rates of adaptation for the original adaptive parameters $\hat{\Theta}$ and for the baseline control component u_{bl} , respectively. In this case, the adaptive control augmentation component is

$$u_{\text{ad}} = -\hat{\Theta}^T \Phi(\hat{x}) - \hat{K}_u \underbrace{(-K_{\text{lqr}} \hat{x})}_{u_{\text{bl}}}$$

It is interesting to note that according to (14.32), the adaptive gain \hat{K}_u serves as an estimate of the constant parametric uncertainty $(I_{m \times m} - \Lambda^{-1})$. Hence, an estimate $\hat{\Lambda}$ of the system control effectiveness Λ can be reconstructed.

$$\hat{\Lambda} = (I_{m \times m} - \hat{K}_u)^{-1}$$

For the design study, we define rates of adaptation to be diagonal and directly proportional to the matrix norm of the observer gain: $\|L_v\| = 67.812$. This enforces transient-free adaptation dynamics [18, 19].

We also define an α -dependent RBF regressor vector on the grid of breakpoints $[-10 \ 10] \frac{\pi}{180}$, in two degree increments from each other. The RBF sigma-widths are set to $(\frac{2}{3} \ \frac{2\pi}{180})$. This value allows to position individual RBFs such that any two consecutive functions overlap.

With the (Baseline OBLTR + Adaptive) controller operating in the presence of the reduced control effectiveness, the scaled-down baseline control gains, and the α -dependent uncertainty, the closed-loop system tracking performance is recovered rather well, using reasonable control deflections and rates (Fig. 14.5).

Achieved vertical load factors (negatives of vertical accelerations in g s) are shown in Fig. 14.6.

The data clearly confirm a graceful recovery of the baseline system performance. In addition, we look at the time domain evolution of adaptive gains: $\hat{A}(t) = (1 - \hat{K}_u(t))^{-1}$, $\|\hat{K}(t)\|$ and $\|\hat{\Theta}(t)\|$. The data are shown in Fig. 14.7.

The simulation indicates that system improvements mostly come through RBF-based $\hat{\Theta}$ adaptation, with small changes to other adaptive parameters.

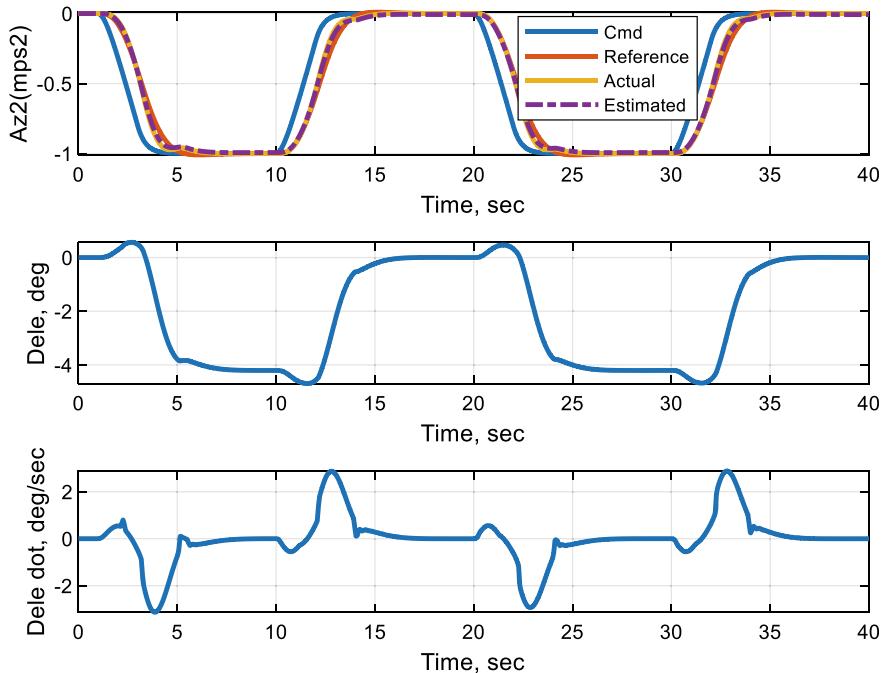


Fig. 14.5 (Baseline OBLTR + Adaptive) controller tracking with uncertainties in Sect. 14.6

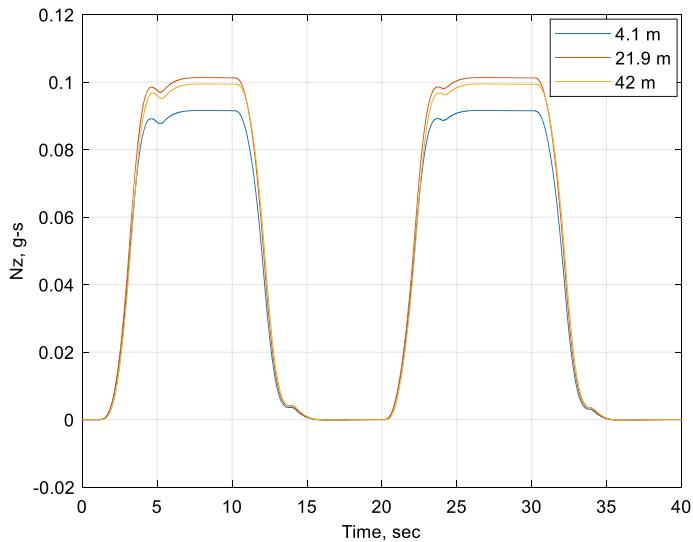


Fig. 14.6 Norms of adaptive gains in Sect. 14.6

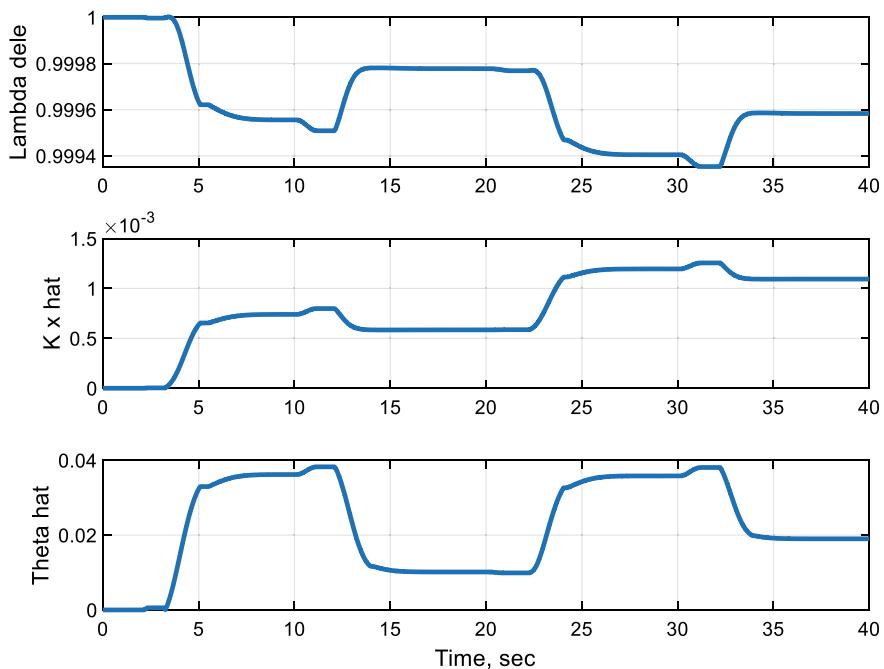


Fig. 14.7 Norms of adaptive parameters in Sect. 14.6

In order to add realism into simulation testing, we employ the gust model from [17]. The model is driven by a random noise, and it generates three separate gust profiles (w_{g_i}) _{$i=1,2,3$} , according to the block diagram shown in Fig. 14.8.

The intent of the model is to emulate time delays in gust propagation along the length of the aircraft.

For simulation testing, we select normally distributed zero-mean noise with standard deviation set to one and generate three light vertical gust profiles (Fig. 14.9).

Then, we evaluate closed-loop (Baseline OBLTR + Adaptive) system tracking and gust rejection performance, in the presence of uncertainties (Fig. 14.10).

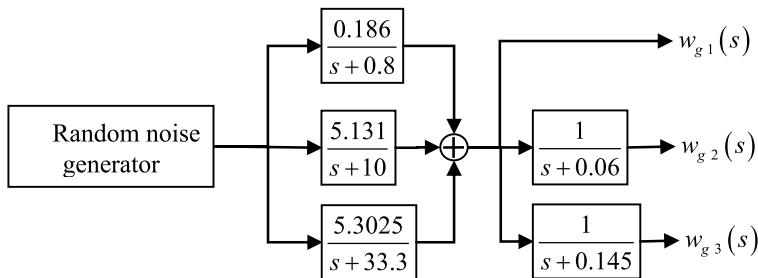


Fig. 14.8 Gust dynamic model in Sect. 14.6

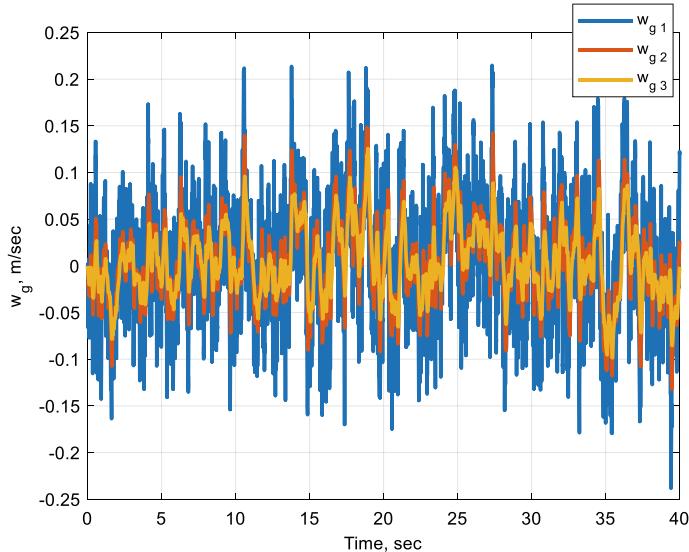


Fig. 14.9 Vertical gust velocities in Sect. 14.6

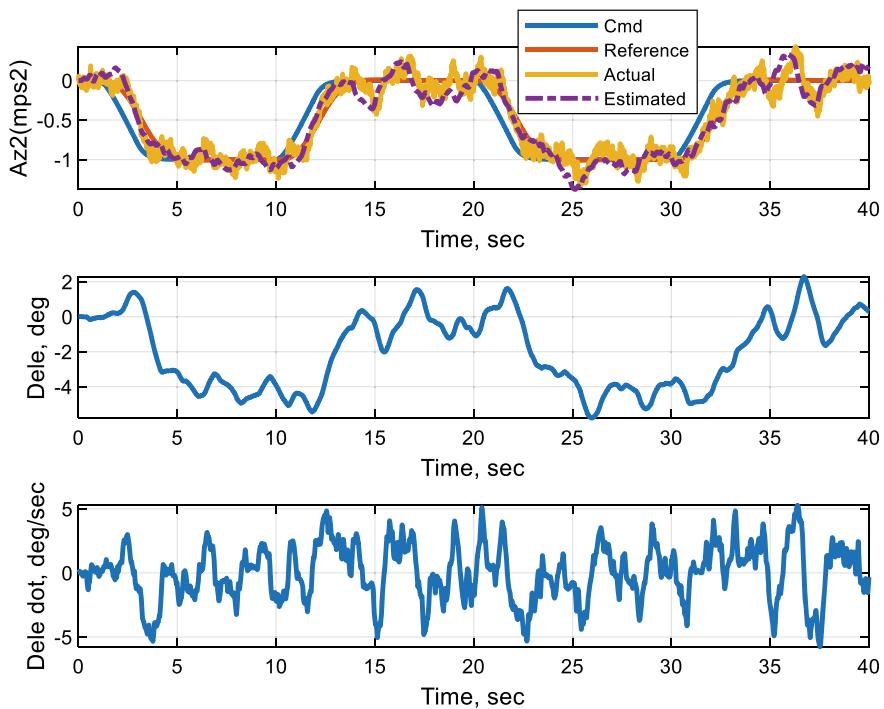


Fig. 14.10 Closed-loop tracking and gust rejection with uncertainties in Sect. 14.6

Clearly, the system is able to attenuate gust effects sufficiently enough in order to maintain closed-loop command tracking abilities in the presence of both gust and aerodynamic uncertainties.

We have also tested the exact same controller (without further retuning) in the presence of medium-to-high gust, nonlinear-in-parameter uncertainties, actuator dynamics, and with various pitch rate commands. All of the simulation trials have resulted in satisfactory and resilient to uncertainties closed-loop system tracking performance. It is interesting to note that collected simulation data show that the adaptive system requires about the same level of control effort as the baseline OBLTR controller, yet the adaptive controller is able to cope quickly and efficiently with a variety of “unknown unknowns” in the system dynamics.

14.7 Design Case Study: Control of the “Respect the Unstable” Dynamics

In this study, we design, analyze, and simulate an OBLTR baseline controller with output feedback adaptive augmentation using the problem motivation from the Inaugural IEEE Bode Lecture “Respect the Unstable”, presented by Dr. Gunter

Stein in 1989 and subsequently published in [20]. In order to carry out the case study, we will use the X-29A aircraft data from the NASA technical report [21].

The open-loop aircraft dynamics are unstable and non-minimum phase with respect to the selected regulated output. For such a system, control designers must learn to “respect the unstable” nature of the underlying control problem. It presents a formidable design challenge on the reachable levels of stability, robustness, and performance via feedback control. These limitations are due to the Bode’s integral theorem [22], also known in the control’s literature as the “waterbed effect”, which enacts hard limitations on the achievable disturbance attenuation for open-loop systems with unstable poles and right half plane zeros.

For this class of systems, the Bode’s integral formula can be proven,

$$\int_0^\infty \ln \left(\left| \det \underbrace{(I + L(i\omega))^{-1}}_{S(i\omega)} \right| \right) d\omega = \int_0^\infty \ln(|\det(S(i\omega))|) d\omega = \pi \sum_{k=1}^q \lambda_k^{\text{ol}} \quad (14.70)$$

where $(\lambda_k^{\text{ol}})_{k=1,\dots,q \leq n} \in \mathbb{C}^+$ are the open-loop system unstable eigenvalues, $S(i\omega) = (I + L(i\omega))^{-1}$ and $L(i\omega)$ are the sensitivity and the loop gain transfer function matrices, computed at the system input or at the output, and I is the identity matrix of the corresponding dimensions.

It is important to immediately note that the Bode’s formula (14.70) is valid if the loop gain $L(i\omega)$ decays to zero (norm-wise) faster than $\frac{1}{s}$, [22].

$$\lim_{s \rightarrow \infty} \|s L(i\omega)\| = 0 \quad (14.71)$$

When the open-loop system is stable, the Bode’s formula (14.70) simplifies.

$$\int_0^\infty \ln(\det|S(i\omega)|) d\omega = 0 \quad (14.72)$$

The Bode’s formulas (14.70) and (14.72) define the “waterbed effect”, also known as a “conservation law”. Making the system sensitivity small via feedback in a selected frequency band forces large sensitivity values outside of the band, since the total integral must remain constant. However, if the system loop gain does not satisfy (14.71), it is possible to reduce the system sensitivity in a selected low frequency region without making it increase elsewhere at high frequencies [22].

We begin with the X-29A open-loop aircraft longitudinal short-period dynamics

$$\boxed{\underbrace{\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix}}_{\dot{x}_p} = \underbrace{\begin{pmatrix} \frac{Z_\alpha}{V_0} & 1 + \frac{Z_q}{V_0} \\ M_\alpha & M_q \end{pmatrix}}_{A_p} \underbrace{\begin{pmatrix} \alpha \\ q \end{pmatrix}}_{x_p} + \underbrace{\begin{pmatrix} \frac{Z_{\delta_e}}{V_0} \\ M_{\delta_e} \end{pmatrix}}_{B_p} \underbrace{\delta_e}_{u}} \Rightarrow \boxed{\dot{x}_p = A_p x_p + B_p u} \quad (14.73)$$

where the system state vector x_p has two components, α and q . They represent the aircraft angle of attack (rad) and the Bode axis pitch rate (rad/s), correspondingly. The system control input is the virtual elevator δ_e (rad), which is composed as a linear combination of the available actual control surfaces. The linear model stability and control data are taken from [21], with respect to the selected equilibrium (1g wings-level trim) flight conditions.

$$V_0 = 977 \left(\frac{\text{ft}}{\text{s}} \right), \quad \text{Alt} = 8000(\text{ft}), \quad \alpha = 2.8(\text{degree})$$

$$A_p = \begin{pmatrix} -2.241 & 0.9897 \\ 44.74 & -0.9024 \end{pmatrix}, \quad B_p = \begin{pmatrix} -0.2331 \\ -45.93 \end{pmatrix} \quad (14.74)$$

Our selected regulated output is defined as a linear combination of the body axis vertical acceleration $A_z \left(\frac{\text{ft}}{\text{s}^2} \right)$ and the pitch rate.

$$y_{\text{reg}} = \underbrace{\left(-\frac{A_z}{g} \right)}_{N_z(g)} + K_q q \quad (14.75)$$

Normalizing A_z with respect to the gravity acceleration $g = 32.1741 \left(\frac{\text{ft}}{\text{s}^2} \right)$ defines the aircraft vertical load factor $N_z(g)$ measured in g s. The pitch rate coefficient, $K_q = 12.4$, is selected to prioritize A_z over q at and above high-speed conditions of interest. In aerospace applications, the regulated output (14.75) is called the “C-star”. It is a widely used signal in aircraft flight control applications.

The regulated output (14.75) can be written in terms of the system states.

$$y_{\text{reg}} = \underbrace{\left(66.5 \ 12.745 \right) x_p}_{C_{p \text{ reg}}} + \underbrace{8.586 u}_{D_{p \text{ reg}}} = C_{p \text{ reg}} x_p + D_{p \text{ reg}} u \quad (14.76)$$

It is straightforward to verify that the open-loop system (14.73), (14.74), (14.75), (14.76) is unstable and non-minimum phase (yellow highlighted values in Table 14.2) with respect to the selected regulated output y_{reg} .

Therefore, depending on our choice of the controller, the Bode’s formula (14.70) may apply and restrict achievable design targets.

We shall use the (OBLTR + Adaptive) control step-by-step design as defined in Table 14.1. Essentially, we will perform the following four steps:

Table 14.2 Open-loop short-period dynamics poles and zeros in X-29A design study

Poles	Zeros
-8.2595	74.412
5.1161	-7.5815

- (1) LQR state feedback baseline servo-controller for extended system.
- (2) Squaring-up extended system.
- (3) OBLTR design and analysis.
- (4) Adaptive augmentation design and simulation.

Since this process is similar to Sect. 14.6, we will shorten definitions and only discuss specific design data in each design step.

Step 1: LQR State Feedback Baseline Servo-controller for Extended System We add the integrated tracking error signal $e_i = \frac{y_{\text{reg}} - y_{\text{cmd}}}{s}$ to the open-loop system (14.73) to define the extended open-loop dynamics whose state is $x = (e_I \alpha q)^T$. After that, we select LQR weight matrices,

$$Q_{\text{lqr}} = \begin{pmatrix} 0.0001 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad R_{\text{lqr}} = 1$$

solve the Algebraic Riccati Equation, and compute the resulting baseline controller LQR gains (see Sect. 14.6 for definitions).

$$K_{\text{lqr}} = \begin{pmatrix} \underbrace{-0.01}_{K_{\text{lqr}I}} & \underbrace{-1.4472}_{K_{\text{lqr}\alpha}} & \underbrace{-0.24008}_{K_{\text{lqr}q}} \end{pmatrix}$$

This allows us to define the baseline (Proportional + Integral) state feedback-driven elevator servo-controller.

$$\delta_e = \boxed{u = -K_{\text{lqr}}x} = K_{\text{lqr}I} \left(\frac{y_{\text{cmd}} - y_{\text{reg}}}{s} \right) - K_{\text{lqr}\alpha} \alpha - K_{\text{lqr}q} q$$

Closed-loop system time domain performance is tuned by increasing/decreasing the (1, 1)-element of Q_{lqr} to achieve a reasonable rise time (~ 1 s) when tracking step-input commands (Fig. 14.11).

The loop gain frequency response shows more than adequate stability margins at the system input.

That is why we use LQR state feedback design with its guaranteed stability and robustness properties (Sect. 14.4). For robustness testing, we also added a second-order actuator dynamics with the natural frequency $\omega_{n,\text{act}} = 70(\frac{\text{rad}}{\text{s}})$ and the damping ratio $\xi_{\text{act}} = 0.7$. The red line in Fig. 14.12 shows extra attenuation at high frequencies due to the actuator dynamics. The loop gain-crossover frequency is around 10 rad/s which is expected in this case due to the large open-loop unstable pole. Note that without the actuator dynamics, the LQR loop gain would have had an infinite positive gain margin and at least a 60° of phase margin.

Closed-loop system sensitivity $S_{y_{\text{reg}}}$ and cosensitivity $T_{y_{\text{reg}}}$ transfer functions at the regulated output, without actuator dynamics in the loop, are shown in Fig. 14.13.

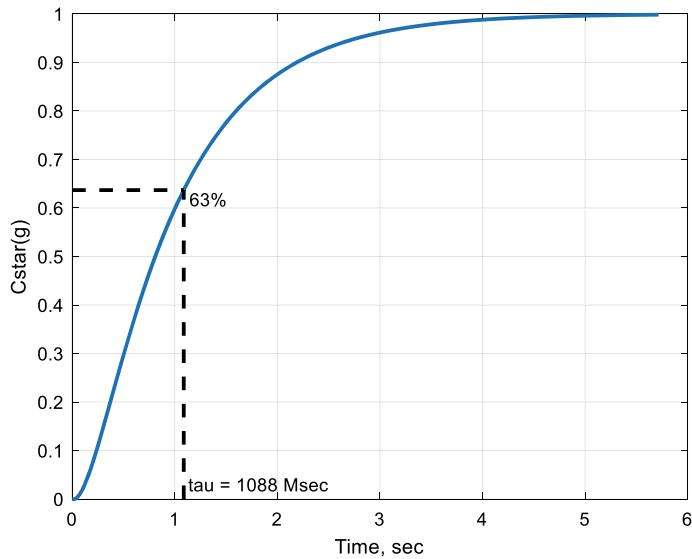


Fig. 14.11 Step-input response with an LQR state feedback controller in X-29A design study

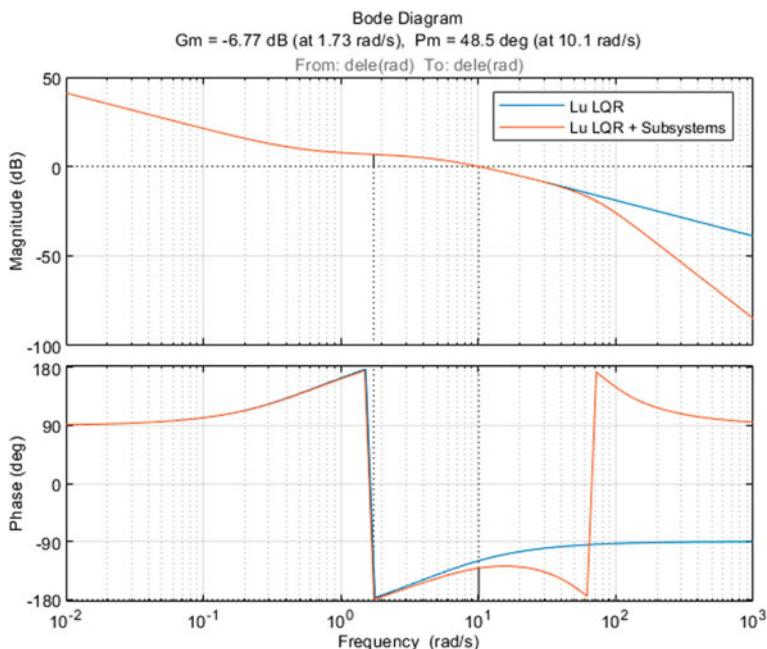


Fig. 14.12 Loop gain and stability margins with an LQR state feedback controller in X-29A design study

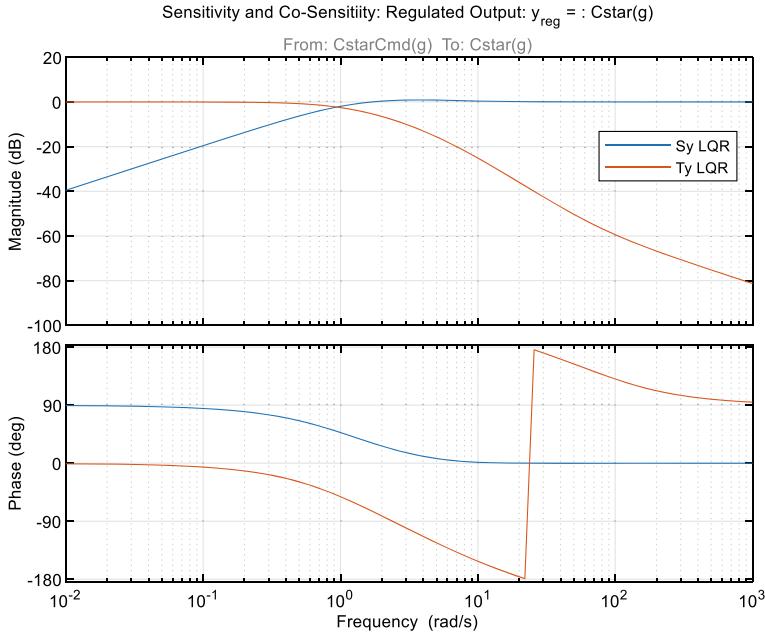


Fig. 14.13 Closed-loop sensitivity (blue) and cosensitivity (red) at regulated output in X-29A design study

These two transfer functions define the system closed-loop dynamics from external commands to the regulated output $T_{y_{reg}}$ and to the regulated output error $S_{y_{reg}}$, respectively. Bode data show the integral action embedded into the control system. Per design, the baseline LQR state feedback controller provides Type 1 tracking performance, with minimal peaking above the zero-dB line.

The important point to discuss here is the application of the Bode's formula (14.70) to the system output sensitivity transfer function matrix $S_y \in R^{3 \times 3}$, which in turn is computed based on the system loop gain transfer function matrix $L_y \in R^{3 \times 3}$, with both quantities defined at the system output breakpoint, $S_y = (I_3 + L_y)^{-1}$.

Without the actuator dynamics in the loop, the “waterbed effect” *does not* apply to the LQR state feedback controller. That is expected since the LQR loop gain roll-off is $O(\frac{1}{s})$. However, that is not a realistic assumption, since subsystems, such as control actuators and output measurement sensors, are always present in practice.

With the second-order actuator dynamics added for analysis (but not for design), the system loop gain roll-off is $O\left(\frac{1}{s^3}\right)$ and the Bode’s formula is in effect, since it applies to loop gains with relative degree two or greater. Indeed, we can verify that the predicted value,

$$\int_0^\infty \ln(|\det(S(i\omega))|) d\omega = \pi \sum_{k=1}^{q=1} \lambda_k^{\text{ol}} = \pi \underbrace{(5.12)}_{\text{Unstable Open-Loop Pole}} = 16.085$$

from the formula (14.70) matches very closely the numerically computed area of the output sensitivity determinant under the log. The same result can be produced if instead of using sensitivity at the output, the input sensitivity transfer function $S_u(s) = (I_m + L_u(s))^{-1}$ is used, where $m = 1$ and $L_u(s)$ is the system loop gain, computed at the input breakpoint Fig. 14.14.

The LQR input sensitivity without the actuator dynamics (blue curve) has no peaking. However, with the actuator dynamics in the loop, the peaking above the zero-dB line in the LQR input sensitivity transfer function (red curve) is evident. That is the “waterbed effect” as predicted and defined by the Bode’s formula (14.70).

To reiterate, the Bode’s formula value is driven by unstable open-loop poles, and in that sense, the Bode’s integral is independent of a control design selection. Because of that, the formula embeds quantifiable limitations on achievable control design goals, one of which is the shaping of input or output sensitivity peaks to

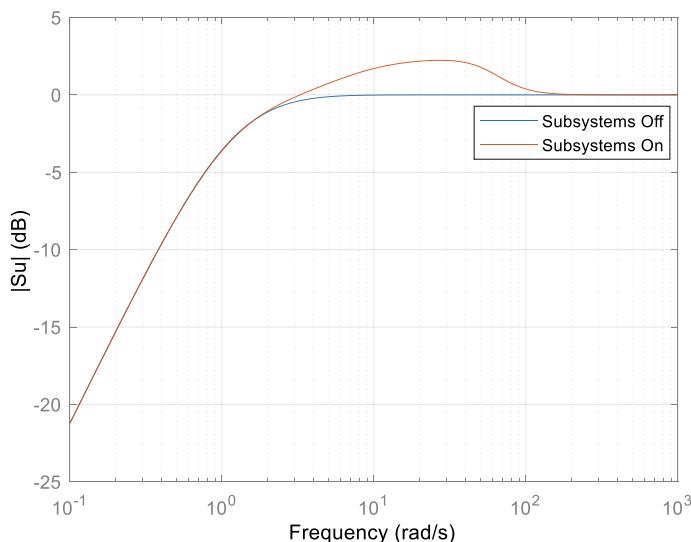


Fig. 14.14 LQR sensitivity with and without actuator dynamics at the system input breakpoint in X-29A design study

minimize the system response due to load disturbances, sensor noise, or any other imperfections that may exist at the control input or at the system measured output.

Step 2: Squaring-up Extended System We perform the squaring-up of the extended dynamics using the process as defined in Chap. 6 and summarized in Table 14.1. The resulting B -matrix data for the squared-up extended system are shown below.

$$\bar{B} = (B \ B_2) = \begin{pmatrix} 8.5863 & -33 \\ -0.23307 & -32.495 \\ -45.926 & -6.1607 \end{pmatrix}$$

The squaring-up method yields a stable transmission zero at (-73.6) .

Step 3: OBLTR Design and Analysis The OBLTR design tuning “knob” is set to unity, $v = 1$. The initial weights, Q_0 and R_0 , for the ARE (14.28) are set to identity matrices of the corresponding dimensions. The ARE unique symmetric positive-definite solution is

$$P_v = \begin{pmatrix} 56.867 & 30.75 & 10.227 \\ 30.75 & 22.249 & 8.3251 \\ 10.227 & 8.3251 & 48.7 \end{pmatrix}$$

Figure 14.15 shows the loop transfer recovery property of the OBLTR output feedback controller.

The recovered stability margins at the plant input are satisfactory. Loop gain data show the expected trends, whereby the OBLTR loop gain (red) adds an extra (-20 dB/decade) roll-off to the LQR loop gain (blue). Also, with subsystems (second-order actuator) accounted for in the analysis, the OBLTR loop gain (yellow) exhibits another (-40 dB/decade) roll-off. So, the optimal state feedback margins at the system input breakpoint are recovered. Overall, the OBLTR controller has a much better disturbance attenuation properties than that of the optimal LQR state feedback system.

The closed-loop system response data are shown in Fig. 14.16.

The data confirm smooth regulated output tracking performance with realistic control inputs. Note that since the open-loop system is unstable, the control surfaces move trailing edge down (positive) to yield upward motion of the aircraft. This is to be compared with an open-loop stable aircraft pitch dynamics, where a negative elevator deflection (trailing edge up) produces a positive vertical load factor, thus making the vehicle move upward. In short, “counterintuitive” elevator movements are clear indications of the open-loop system instabilities.

Step 4: Adaptive Augmentation Design and Simulation First, we test the baseline OBLTR controller performance in the presence of matched uncertainties, which are represented by 50% reduction in elevator control power, ($\Lambda = 0.5$), and a nonlinear in angle-of-attack control increment function $f(\alpha)$ (Fig. 14.17).

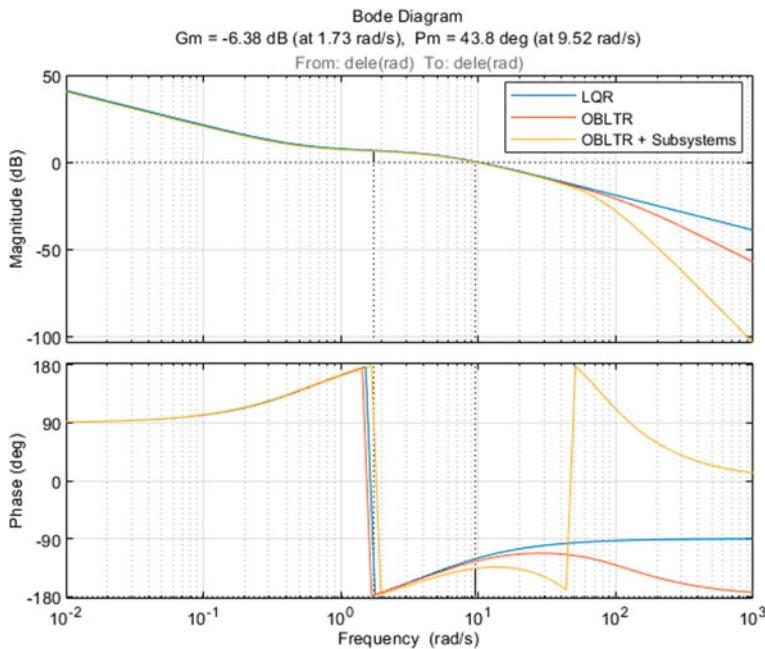


Fig. 14.15 Stability margins recovery with OBLTR controller in X-29A design study

With these uncertainties turned on, the baseline OBLTR controller yields a significant deterioration of the closed-loop system output tracking performance (Fig. 14.18).

Under the selected uncertainties, the OBLTR controller is able to maintain closed-loop system stability but its ability to track external commands degraded.

Next, we define rates of adaptation directly proportional to the norm of the observer gain ($0.05\|L_v\|$) and select the regressor vector $\Phi(\alpha)$, whose components are defined as radial basis functions with equally spaced centers in the α -interval (Fig. 14.19). Also, the overlap between the regressor components is chosen to enforce a constant gain through the regressor.

With the adaptive augmentation turn on yields a much-improved command tracking performance (Fig. 14.20).

Evolution of adaptive parameter norms is shown in Fig. 14.21.

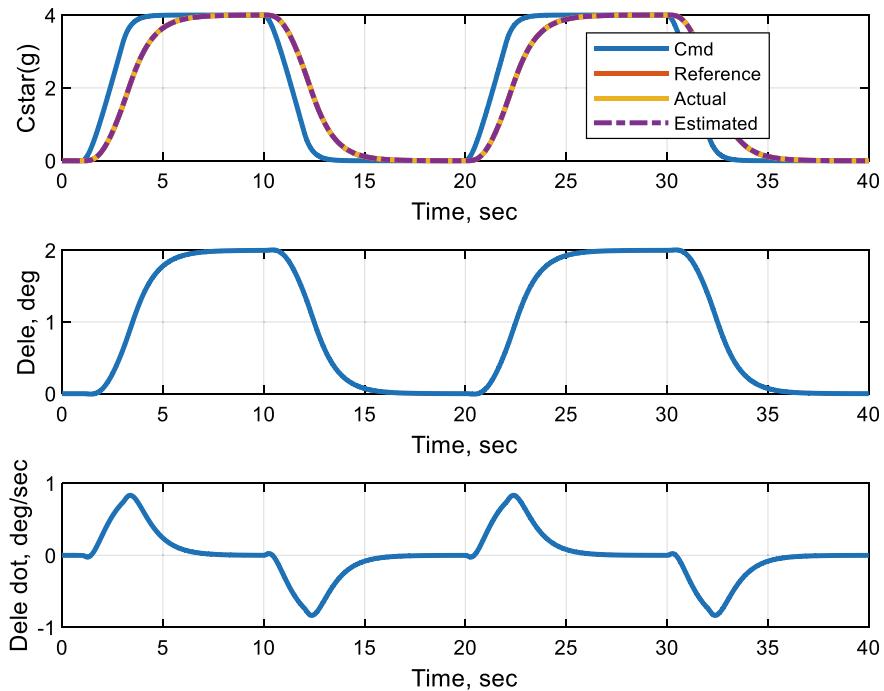


Fig. 14.16 Closed-loop simulation data with baseline OBLTR controller in X-29A design study

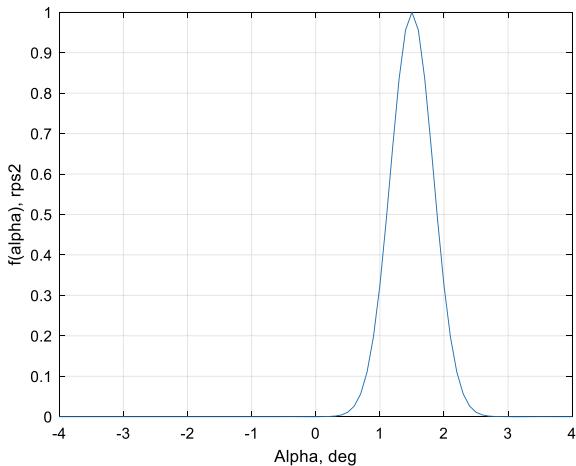


Fig. 14.17 Nonlinear matched uncertainty in X-29A design study

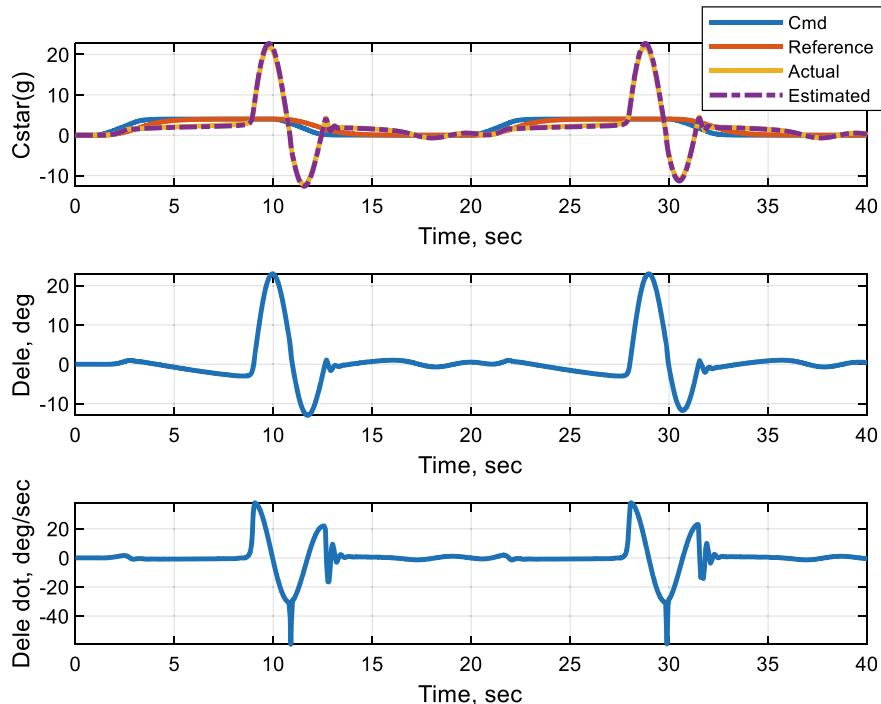


Fig. 14.18 Tracking performance deterioration with OBLTR and matched uncertainties in X-29A design study

As previously discussed, smoothness of adaptive parameters is predicated by the design and in particular, by the selection of the rates of adaptation, whose values are set directly proportional to the norm of the observer gain.

Finally, Fig. 14.22 shows the matched uncertainty estimation data.

The external command is selected to persistently excite the system dynamics, and in that case, the adaptive control augmentation can be decomposed into the corresponding uncertainty estimation terms. The actual (blue) and the estimated (red) data are shown versus time and as functions of α . Both plots confirm close approximation of the system uncertainties. We immediately note that the ability to identify “unknown unknowns” is *not* guaranteed by our design. What is warranted here is the recovery of the baseline system closed-loop stability and output tracking performance, all without relying on system identification. That is the essence of our method to design output feedback direct adaptive controllers with closed-loop reference models, which in turn represent the OBLTR state observer component of the baseline robust LQR PI output feedback controller. ■

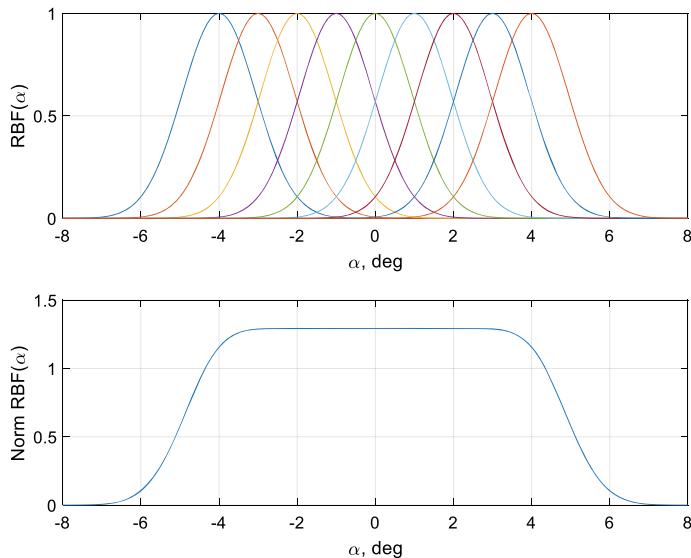


Fig. 14.19 Regressor with radial basis function components in X-29A study

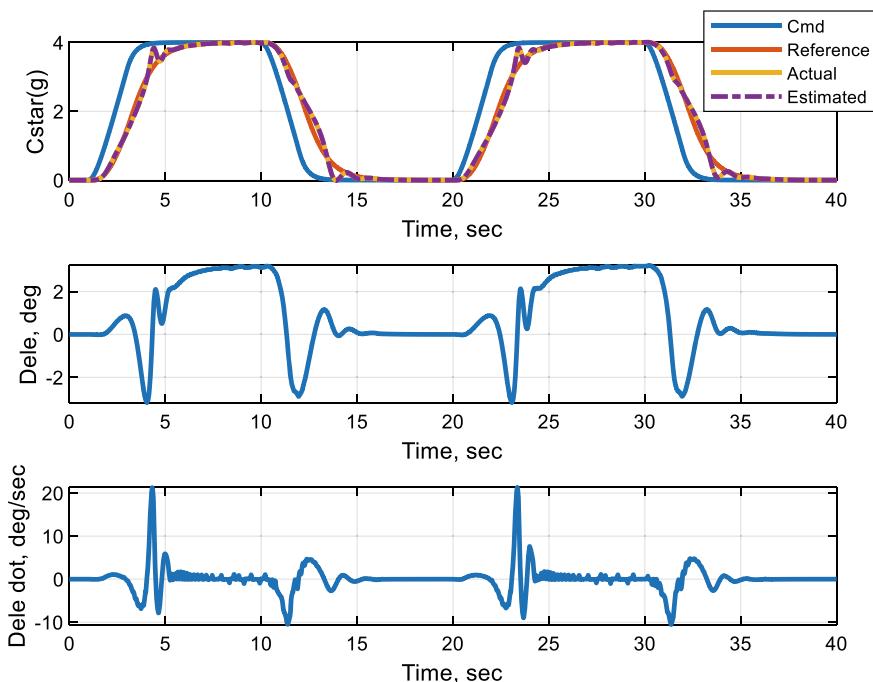
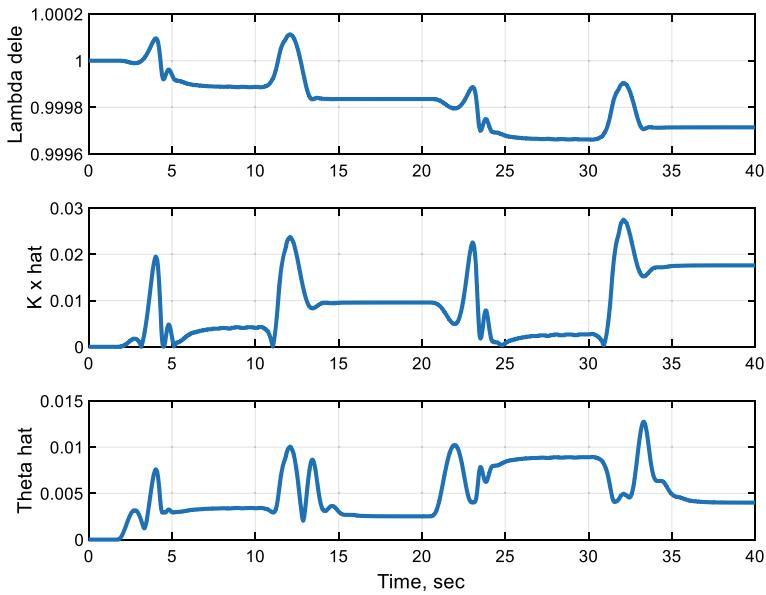
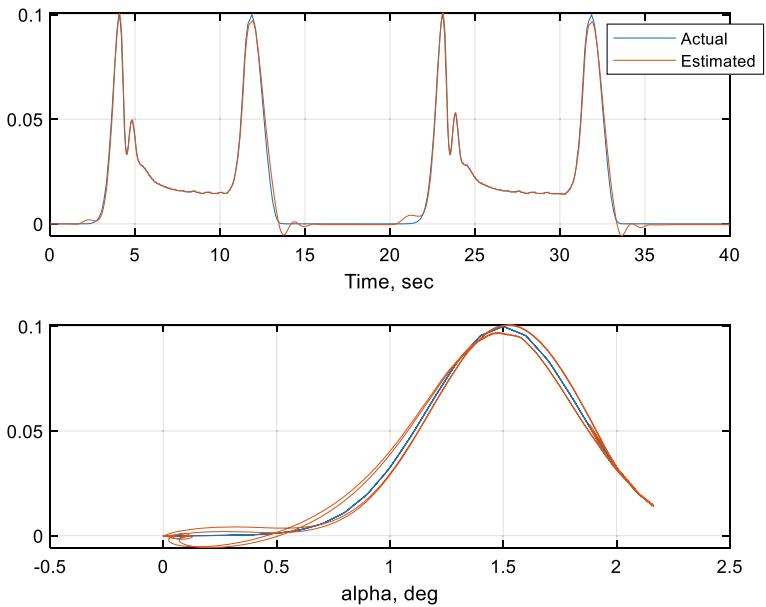


Fig. 14.20 Tracking performance recovery with (OBLTR + Adaptive) controller and matched uncertainties in X-29A design study

**Fig. 14.21** Norms of adaptive parameters in X-29A study**Fig. 14.22** Uncertainty estimation data in X-29A study

14.8 Design Case Study: (OBLTR + Adaptive) Flight Control of Aircraft MIMO Roll–Yaw Dynamics

The main purpose of this study is to demonstrate (OBLTR + Adaptive) control design and analysis steps for a MIMO system, such as the lateral-directional (a.k.a. “roll–yaw”) aircraft dynamics, as defined in Chap. 1.

$$\underbrace{\begin{pmatrix} \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \end{pmatrix}}_{\dot{x}_p} = \underbrace{\begin{pmatrix} \frac{Y_\beta}{V_0} & \frac{Y_{p_s}}{V_0} & \frac{Y_{r_s}}{V_0} - 1 \\ L_\beta & L_{p_s} & L_{r_s} \\ N_\beta & N_{p_s} & N_{r_s} \end{pmatrix}}_{A_p} \underbrace{\begin{pmatrix} \beta \\ p_s \\ r_s \end{pmatrix}}_{x_p} + \underbrace{\begin{pmatrix} \frac{Y_{\delta_{ail}}}{V_0} & \frac{Y_{\delta_{rud}}}{V_0} \\ L_{\delta_{ail}} & L_{\delta_{rud}} \\ N_{\delta_{ail}} & N_{\delta_{rud}} \end{pmatrix}}_{B_p} \underbrace{\begin{pmatrix} \delta_{ail} \\ \delta_{rud} \end{pmatrix}}_u$$

The systems state x_p includes the aircraft sideslip angle β (rad), as well as the vehicle stability axis roll and yaw rates (rad/s), p_s and r_s . The control input u is represented by the aileron and the rudder deflections (rad), δ_a and δ_r . The regulated output of interest consists of the aircraft roll rate p_s (rad/s) and the lateral load factor $N_y = -\frac{A_y}{g}$ (g s), where A_y is the vehicle lateral acceleration in body axis (ft/s²) and $g = 32.174$ is the gravitational acceleration (ft/s²).

$$y_{\text{reg}} = (p_s \ N_y)^T = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ \frac{Y_\beta}{g} & \frac{Y_{p_s}}{g} & \frac{Y_{r_s}}{g} \end{pmatrix}}_{C_{p \text{ reg}}} x_p + \underbrace{\begin{pmatrix} 0 & 0 \\ \frac{Y_{\delta_{ail}}}{g} & \frac{Y_{\delta_{rud}}}{g} \end{pmatrix}}_{D_{p \text{ reg}}} u = C_{p \text{ reg}} x_p + D_{p \text{ reg}} u$$

So, we have a (2×2) open-loop system, with two control inputs and two regulated outputs. Once aileron and rudder controls are designed, the aircraft will be able to perform coordinated rolls, while minimizing lateral acceleration and thus sideslip angle excursions. In addition, the vehicle would be able to fly steady-heading sideslip maneuvers with a commanded lateral acceleration and a near-zero roll rate. By design, such a control system will also allow to execute stable flight trajectories with a roll rate and a lateral acceleration commands issued simultaneously.

The roll–yaw model data are defined via stability and control derivatives that are computed numerically in a high-fidelity aircraft flight simulation environment aFltSim (see Chap. 1 and Appendix A). Specifically, the model parameters are defined using numerical linearization with respect to a 1g-level flight trim at the selected flight conditions.

$$V_0 = 717.17 \left(\frac{\text{ft}}{\text{s}} \right), \quad \text{Alt} = 25,000(\text{ft}), \quad \alpha = 4.5627(\text{degree})$$

$$A_p = \begin{pmatrix} -0.11794 & 0.00085 & -1.0001 \\ -7.0113 & -1.4492 & 0.22059 \\ 6.3035 & 0.06511 & -0.41172 \end{pmatrix}, \quad B_p = \begin{pmatrix} 0 & 0.015257 \\ -7.9662 & 2.6875 \\ 0.60926 & -2.3577 \end{pmatrix}$$

$$C_{p\text{ reg}} = \begin{pmatrix} 0 & 1 & 0 \\ -2.6049 & 0.018724 & 0.067695 \end{pmatrix}, \quad D_{p\text{ reg}} = \begin{pmatrix} 0 & 0 \\ 0 & 0.33698 \end{pmatrix}$$

Similar to previous design studies, we shall follow the same steps (Table 14.1) to design and analyze an (OBLTR + Adaptive) dynamic output feedback controller for the MIMO roll–yaw dynamics.

Step 1: LQR State Feedback Baseline Servo-controller for Extended System We define two integrated tracking errors,

$$e_i = \frac{y_{\text{reg}} - y_{\text{cmd}}}{s} = \begin{pmatrix} \frac{p_s - p_s \text{ cmd}}{s} \\ \frac{N_y - N_y \text{ cmd}}{s} \end{pmatrix}$$

and add them to the roll–yaw dynamics to form the extended open-loop dynamics with five dimensional state x , two control inputs u , two regulated outputs y_{reg} , and two external commands y_{cmd} for y_{reg} to follow.

$$\underbrace{\begin{pmatrix} \dot{e}_I p_s \\ \dot{e}_I N_y \\ \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} 0_{2 \times 2} & C_{p\text{ reg}} \\ 0_{2 \times 3} & A_p \end{pmatrix}}_A \underbrace{\begin{pmatrix} e_I p_s \\ e_I N_y \\ \beta \\ p_s \\ r_s \end{pmatrix}}_x + \underbrace{\begin{pmatrix} D_{p\text{ reg}} \\ B_p \end{pmatrix}}_B u + \underbrace{\begin{pmatrix} -I_2 \\ 0_{3 \times 2} \end{pmatrix}}_{B_{\text{cmd}}} \underbrace{\begin{pmatrix} p_s \text{ cmd} \\ N_y \text{ cmd} \end{pmatrix}}_{y_{\text{cmd}}}$$

We shall design an LQR PI state feedback controller to achieve a closed-loop with a sufficiently fast roll rate and a slower lateral acceleration. We would also like to minimize overshoot, undershoot, and achieve sufficiently large stability margins at the system input breakpoints, defined for SISO and MIMO loop gains. These are our control design requirements.

After several design iterations, we defined LQR diagonal weight matrices, $(Q_{\text{lqr}}, R_{\text{lqr}})$, and computed the corresponding optimal LQR PI state feedback gains K_{lqr} .

$$Q_{\text{lqr}} = \text{diag}(1.025 \ 1.0289 \ 0 \ 0 \ 1.6021), \quad R_{\text{lqr}} = \text{diag}(1 \ 0.49129)$$

$$K_{\text{lqr}} = \begin{pmatrix} -1.0107 & -0.059923 & 0.25106 & -0.36827 & -0.23965 \\ 0.085329 & -1.4446 & 1.5166 & -0.05602 & -2.206 \end{pmatrix}$$

Figure 14.23 shows the closed-loop system response to step-input commands.

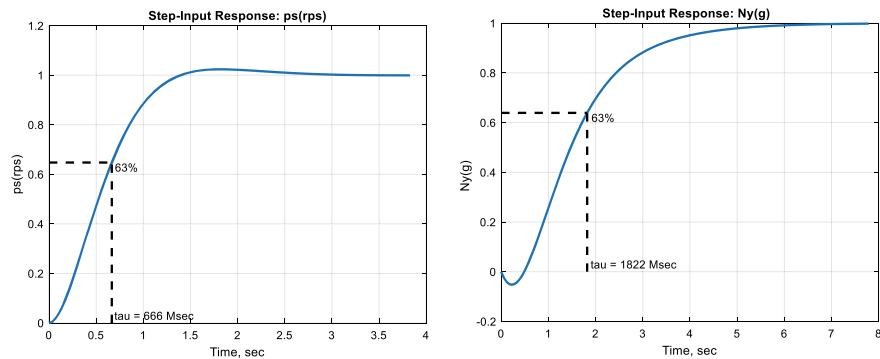


Fig. 14.23 Step-input response simulation data with LQR PI state feedback in roll–yaw control study

As seen from the simulation data, the closed-loop system exhibits a sufficiently small roll-rate rise time which is about three times faster than that for the lateral acceleration. This is acceptable. Also, the overshoot and undershoot characteristics in both channels are satisfactory.

Primary response and cross-output dependencies are shown in Fig. 14.24.

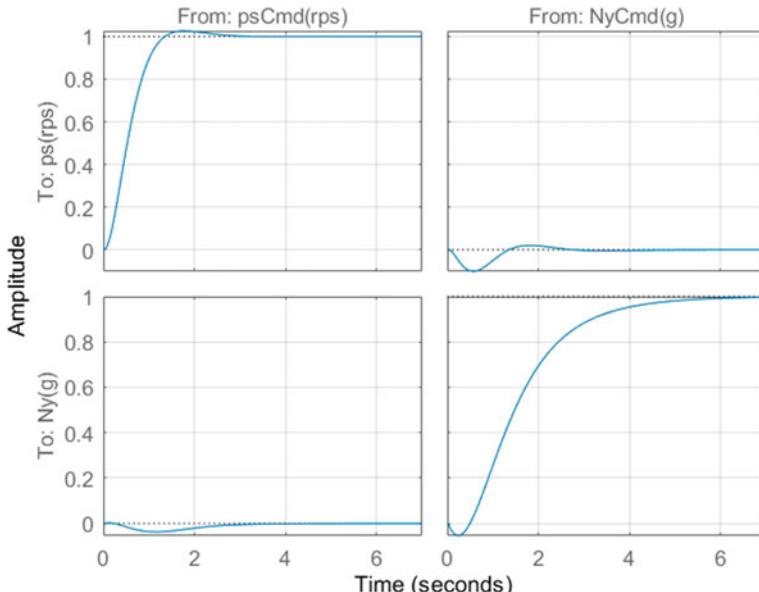


Fig. 14.24 Closed-loop system primary and cross-output simulation data in roll–yaw study

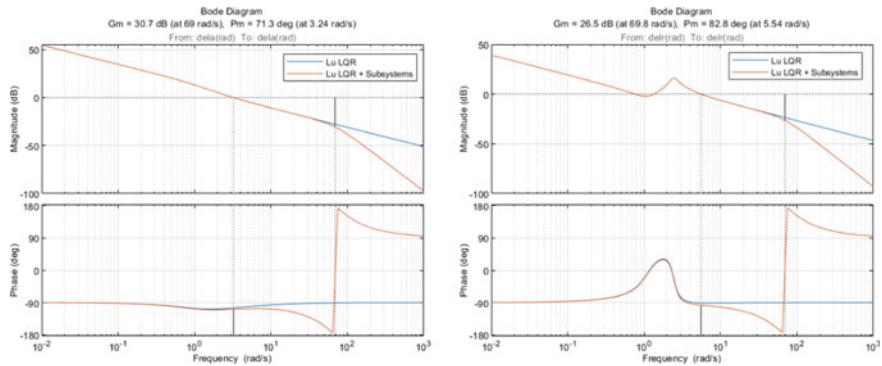


Fig. 14.25 SISO loop gains and stability margins in roll-yaw control study

The system tracking performance due to step-input commands is acceptable and near-decoupled, which is due to the use of integral feedback within the LQR PI-control solution.

Similar to the previous design studies, we added second-order actuators at the aileron and rudder inputs. This is done to add a realism into stability analysis. The actuators natural frequencies and the damping ratios were set exactly the same as in the previous section. With the aileron and rudder actuator dynamics accounted for, the system SISO loop gains and the corresponding stability margins at individual control input breakpoints are shown in Fig. 14.25.

The SISO gain and phase margins are well above the typically required values (6 dB and 45°). The aileron crossover frequency is typical of an aircraft of this type. The rudder crossover frequency is slightly higher than that of the aileron channel. It could be further reduced if needed. The actuator dynamics is visible in both plots, whereby the loop gains show an extra (pole 0 dB) roll-off above 70 rps, which is due to the selected actuators model order and their natural frequency values.

Figure 14.26 shows sensitivity and cosensitivity transfer functions computed at the regulated output.

The negligible peaking in the system frequency response data is indicative of well-damped closed-loop dynamics, with a minimal influence of external disturbances on the system behavior.

Figure 14.27 shows the closed-loop system return difference frequency response with and without actuator dynamics.

These data and further analysis reveal more than adequate SVD-based MIMO stability margins (see Chap. 6), which in turn verifies relative stability of the system in the MIMO sense. In other words, large MIMO margins always imply large SISO margins, and in addition, the MIMO data confirm that there is no undesirable cross-coupling between the aileron and the rudder control inputs.

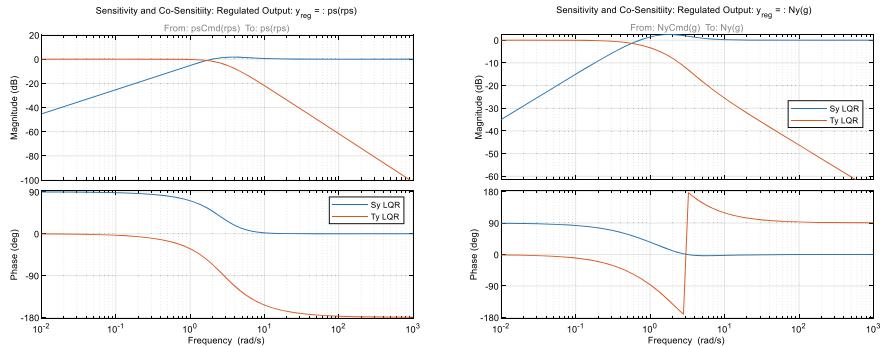


Fig. 14.26 Sensitivity and cosensitivity transfer functions at regulated output in roll–yaw control study

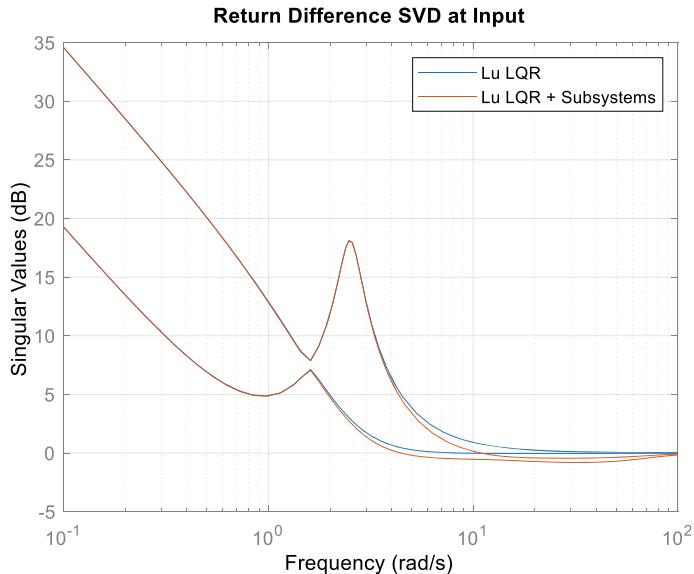


Fig. 14.27 Return difference frequency response data in roll–yaw control study

Step 2: Squaring-up Extended System The squaring-up design follows the steps as outlined in Table 14.1. In this case, we have five states n , four outputs p , and two control inputs m . Consequently, one transmission zero ($n - p$) can be assigned via an addition of two column vectors ($p - m$) to the extended open-loop system

B -matrix.

$$\bar{B} = (B \ B_2) = \begin{pmatrix} 0 & 0 & -2.036 & 0.1124 \\ 0 & 0.337 & -0.1531 & -1.452 \\ 0 & 0.015257 & 0.1482 & 1.406 \\ -7.9662 & 2.6875 & -0.001833 & -0.01739 \\ 0.60926 & -2.3577 & -0.02397 & -0.2274 \end{pmatrix}$$

$\underbrace{\hspace{1cm}}_B \quad \underbrace{\hspace{1cm}}_{B_2}$

Within the squaring-up method, we used the LQR design to place a single stable transmission zero at (-1.68) . Also, the columns of B_2 were rescaled to match the norm of B . This is a numerical conditioning procedure. It does not change the location of transmission zeros, nor it is required by the squaring-up method. Through many design trades, we have found it to be helpful in achieving loop transfer recovery without excessive gains in the corresponding state observer dynamics.

Step 3: OBLTR Design and Analysis The OBLTR tuning knob v is set to 0.1, while the rest of the required for design data are set to unity. This is a default setting. In most cases it yields close-to-final outcome. After several iterations, we scaled R_0 matrix by 1.4. The resulting filter ARE solution P_v and the Luenberger observer gain matrix L_v are shown below.

$$P_v = \begin{pmatrix} 2.4298 & 0.050272 & -0.04782 & 0.10275 & -0.003409 \\ 0.050272 & 2.0503 & -1.6859 & 0.19994 & -0.37132 \\ -0.04782 & -1.6859 & 2.0443 & -0.13025 & 0.0084635 \\ 0.10275 & 0.19994 & -0.13025 & 9.699 & -1.2443 \\ -0.003409 & -0.37132 & 0.0084635 & -1.2443 & 2.5576 \end{pmatrix}$$

$$L_v = \begin{pmatrix} 19.091 & 0.395 & 0.80731 & -0.0268 \\ 0.395 & 16.11 & 1.5709 & -2.9175 \\ -0.37572 & -13.246 & -1.0234 & 0.0665 \\ 0.80731 & 1.5709 & 76.206 & -9.7766 \\ -0.02678 & -2.9175 & -9.7766 & 20.095 \end{pmatrix}$$

For comparison purposes, SISO loop gains and stability margins for LQR state feedback and OBLTR output feedback controllers are shown in Fig. 14.28.

The loop transfer recovery of the OBLTR controller is clearly visible. Making the tuning knob v smaller will further recover LQR margins at frequencies higher than the loop gain crossover. We do not have to do that. The LQR margins are already recovered, and the OBLTR extra loop gain attenuation is beneficial to enforce rejection of high frequency disturbances and model inaccuracies that may exist in that region. In that sense, the OBLTR dynamic output feedback controller is more robust than the LQR state feedback solution.

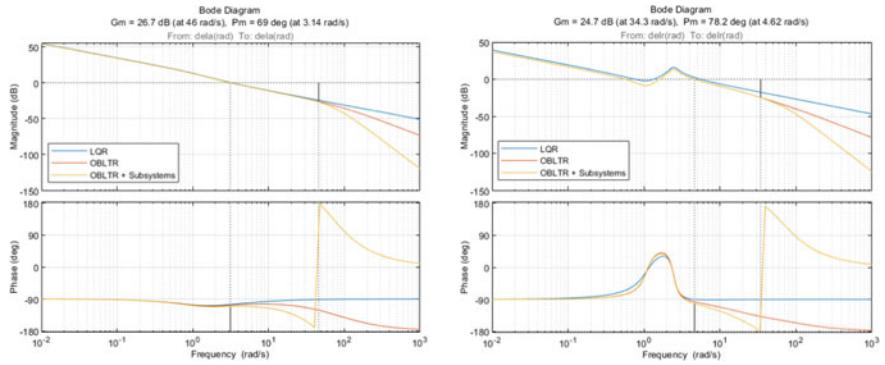


Fig. 14.28 Comparison of LQR and OBLTR stability margins in roll–yaw control study

Figure 14.29 shows MIMO SVD-based modal characteristics of the LQR and OBLTR return difference matrices.

Similar to Fig. 14.28, we observe the loop transfer recovery property of the OBLTR controller, as predicted by design. It can be verified that the corresponding MIMO margins are well above the required (6 dB and 45°) targets.

Closed-loop command tracking performance under the baseline OBLTR output feedback controller is shown in Fig. 14.30.

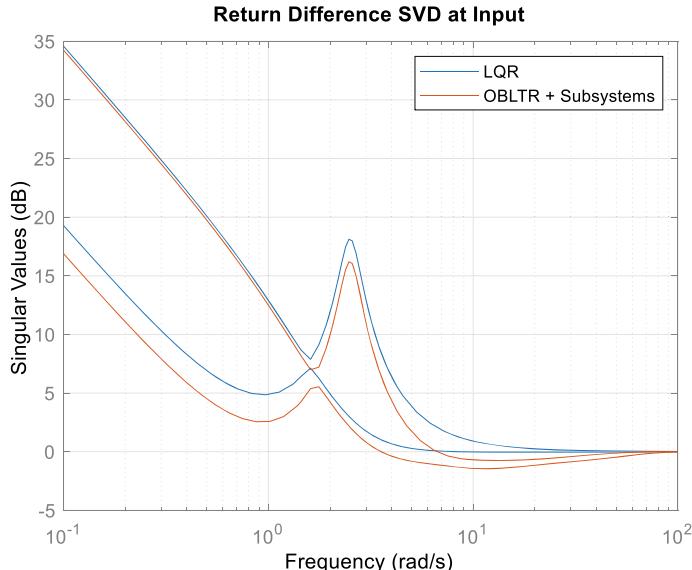


Fig. 14.29 MIMO return difference matrices comparison for LQR and OBLTR solutions in roll–yaw control study

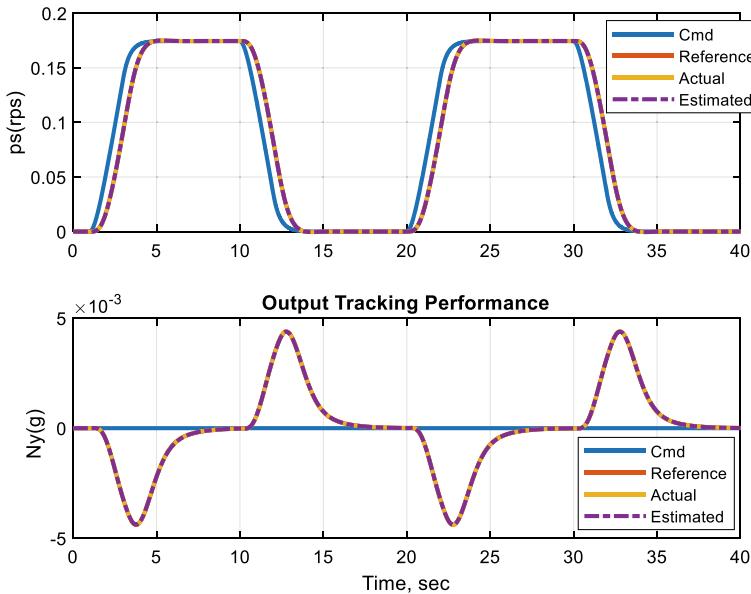


Fig. 14.30 Baseline OBLTR closed-loop tracking performance in roll-yaw control study

The data confirm that the system is able to perform coordinated roll maneuvers with near-zero lateral acceleration. This is a highly desirable property for any roll-rate tracking flight system, whether it is designed for a piloted or an autonomous vehicle. The OBLTR-driven control surface position and rates (Fig. 14.31) are smooth and within actuation limits.

Step 4: Adaptive Augmentation Design and Simulation For testing robustness of the OBLTR baseline controller tracking performance, the following uncertainties are inserted into the open-loop system dynamics:

- OBLTR proportional feedback gains are set to zero. This emulates matched uncertainties in sideslip angle β and stability axis angular rates ($p_s r_s$).
- Λ is set to 0.1, which equates to a 90% loss of control effectiveness in the aileron and the rudder channels.
- A nonlinear matched β -dependent uncertainty is added to the system control input, $f(\beta) = 1 - e^{-\left(\frac{\beta}{0.0233}\right)^2}$. This is our attempt to model state-dependent nonlinear uncertainties in the aircraft aerodynamic moments. The same function is added to both control components.

With these uncertainties turned on, closed-loop simulation data with the baseline controller in the loop are shown in Fig. 14.32.

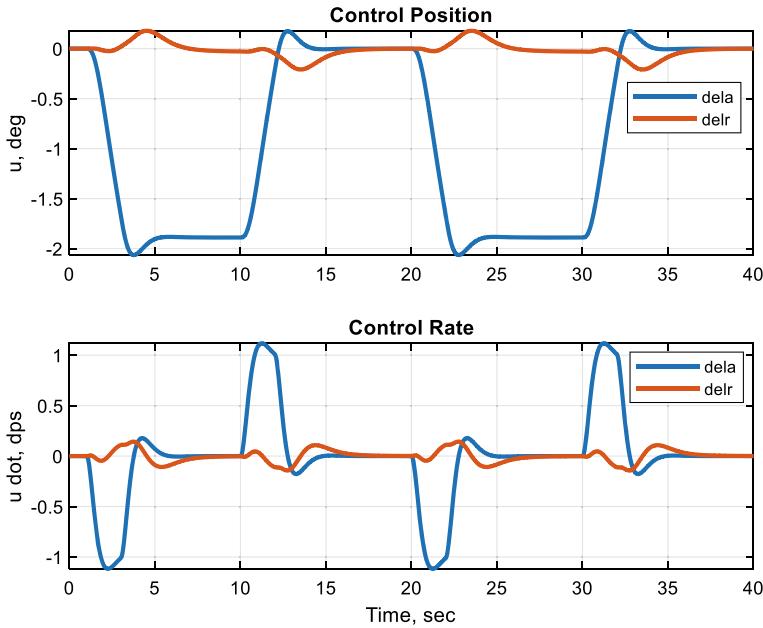


Fig. 14.31 Baseline OBLTR commanded aileron and rudder positions and rates in roll–yaw control study

The plant uncertainties are somewhat “severe” and perhaps not very realistic. Their values are selected to show a significant degradation of the system closed-loop tracking performance with OBLTR baseline controller in the loop. At the same time, these data also confirm robustness of the baseline system. Even without proportional feedback and with a 90% reduced integral feedback, the OBLTR controller is able to retain closed-loop system stability.

We select the rates of adaptation directly proportional to the norm of the state observer gain: $10\|L_v\|$. The adaptive regressor vector $\Phi(x_p)$ is selected to be the same as in the previous design study, except here it is defined to depend on the sideslip angle. This selection gives fast and smooth recovery of the baseline tracking performance (Fig. 14.33).

The simulated control input positions and rates are still within their actuation limits (Fig. 14.34).

Time evolution of the closed-loop data is shown in Fig. 14.35.

As predicted by the design, state components of the ideal reference (blue) model, the original system (red), and the state estimator (yellow) are very close to each other, indicating fast and smooth recovery of the baseline tracking performance. In addition, one can verify that the adaptive parameters are also smooth and well behaved.

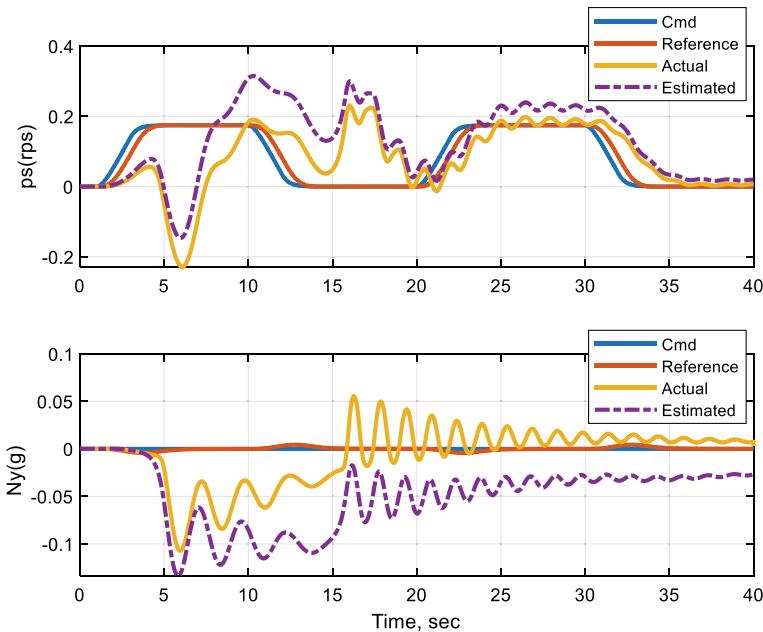


Fig. 14.32 Baseline OBLTR tracking performance degradation with plant uncertainties in roll-yaw control study

Figure 14.36 shows the system uncertainties and their estimated values, for both the aileron and the rudder control channels.

The data are plotted versus time and the sideslip angle. Similar to the previous design study, in this case the external commands are also persistently exciting the system dynamics, which in turn leads to a sufficiently close estimation of the unknown functions that are present in the aileron and the rudder control channels. As a reminder to the reader, we emphasize again that the (OBLTR + Adaptive) system estimation performance is not guaranteed, nor it is required to track external commands in the presence of unknown imperfections in the system dynamics.

The aim of this control design study was to demonstrate practicality and numerical efficiency of the (OBLTR + Adaptive) output feedback controller for a fully coupled MIMO system, represented by the roll–yaw dynamics of a fixed-wing aircraft. The main take-away point here is that there are no significant differences in applying our design to SISO or MIMO dynamics. We show that as long as the system remains controllable, it is possible to construct a dynamic output feedback solution, such as the (OBLTR + Adaptive) controller, to maintain closed-loop stability, robustness and, at the same time, recover the desired baseline target tracking performance, while operating under significant uncertainties, modeling errors, and control failures. ■

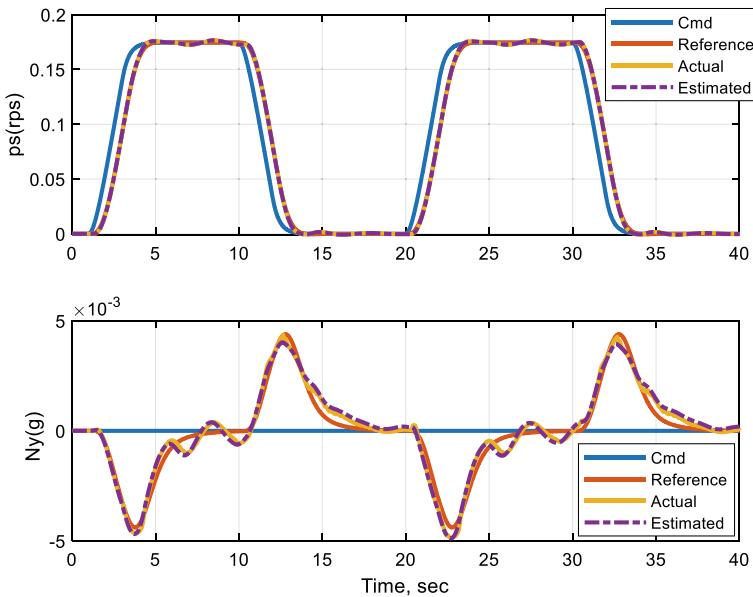


Fig. 14.33 Tracking performance recovery with adaptive augmentation turned on in roll–yaw control study

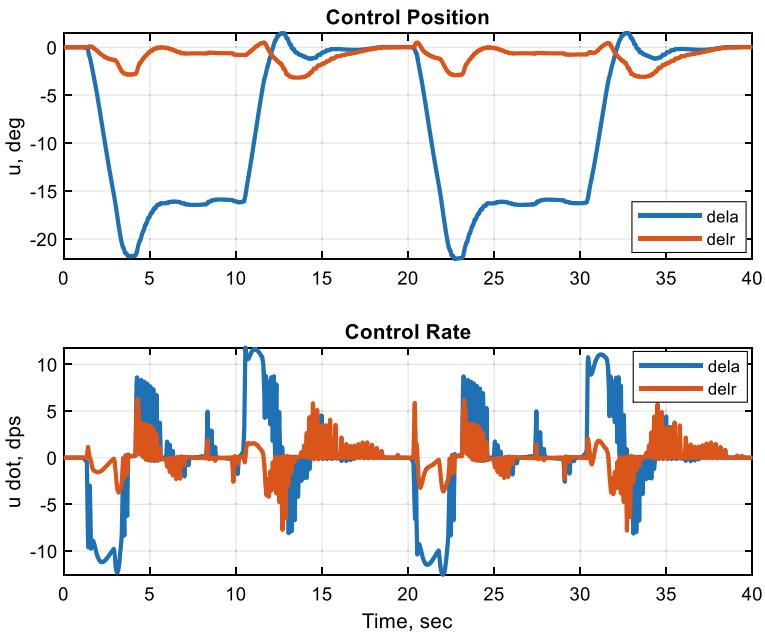


Fig. 14.34 Aileron and rudder data with adaptive augmentation turned on in roll–yaw control study

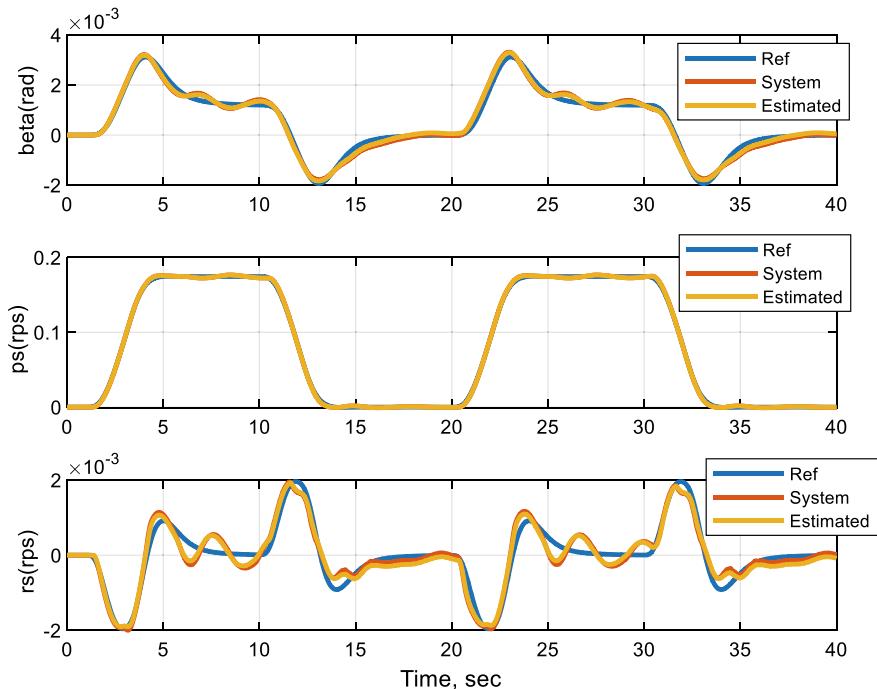


Fig. 14.35 Closed-loop system states in roll–yaw control study

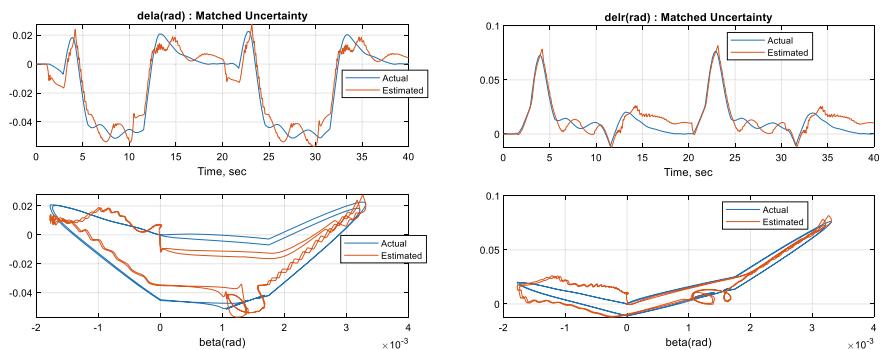


Fig. 14.36 System uncertainties and their estimated values in roll–yaw control study

14.9 Conclusions

In this chapter, we have presented an adaptive output feedback augmentation design for MIMO dynamical systems with matched uncertainties, and with the number of output measurements exceeding the number of control inputs. The system regulated output dynamics are allowed to be non-minimum phase and/or have a high relative degree. We have also developed a detailed design case study related to flight control of a large-size transport aircraft, with prominent structural dynamics, aerodynamic uncertainties, and environmental disturbances. In addition, we demonstrated the design capabilities using an aircraft pitch dynamics model representative of an unstable non-minimum phase vehicle, operating under parametric uncertainties and using partial output measurements available for control.

It is worthwhile to reflect back on the design procedure and summarize the quintessence of our method. After all proofs, derivations, and formal statements, it all comes down to the design of a baseline LQR/OBLTR dynamic output feedback controller, for the original system and without uncertainties. The corresponding linear system, with the baseline OBLTR controller in the loop, defines the desired reference model. The key step to the design of an adaptive augmentation is the introduction of a small positive constant (the so-called tuning knob) into the observer ARE, followed by exploitation of the ARE asymptotic properties with respect to that tuning knob. It turns out that for a sufficiently small value of the tuning knob, one can create a linear combination of the system outputs, use it in the adaptive law, and then augment the baseline OBLTR controller with an adaptive output feedback in such a way that all matched uncertainties in the system dynamics are mitigated. We have also noted that the derived design can be easily extended to a class of nonparametric and non-matched uncertainties, including bounded process noise.

In summary, the (Baseline OBLTR + Adaptive) output feedback design consists of the following four main steps:

1. Using the system without uncertainties, design and analyze an LQR PI state feedback baseline controller.
2. Square-up open-loop extended input–output dynamics.
3. Select a sufficiently small tuning knob, design an OBLTR dynamic output feedback controller, and construct a linear combination of the system outputs for adaptive laws.
4. Compute an adaptive output feedback augmentation and add it to the baseline OBLTR controller.

What is interesting here is the fact that the design of a robust OBLTR controller paves the way to the design of a robust adaptive output feedback. Together, the two controllers provide seamless mitigation of a large class of uncertainties in the system dynamics, while relying only on the system output measurements. So, in a way, we have built an “output feedback bridge” between robust and adaptive control methodologies.

14.10 Exercises

Exercise 14.1 Suppose that all states are accessible, that is, let $y = x$. Write down the adaptive output feedback laws and compare them to a direct state feedback MRAC. Comment on the similarity of the derived solution with the observer-like CRM design from Chap. 13.

Exercise 14.2 Derive an adaptive output feedback solution, similar to the one from Sect. 14.3, but without a baseline linear controller.

Exercise 14.3 Consider the system dynamics (14.25), with the linear regressor vector $\Phi(x) = K_\Phi^T x$ and with an unknown constant matrix $K_\Phi \in R^{n \times m}$. Suppose that $B_{\text{ref}} = 0_{n \times m}$. Using the techniques from Sect. 14.3 and relying only on the system output measurements y , derive an adaptive controller to force the system regulated output y_{reg} track bounded commands.

Exercise 14.4 With the aircraft data from Sect. 14.6, perform a trade study in selecting appropriate values for the tuning knob v . Show numerically that as v gets smaller, the asymptotic relation (14.29) takes place. Plot $\left\| P_v^{-1} \bar{B} - C^T R_0^{-\frac{1}{2}} W \right\|$ versus v . Also show that for small values of v , the observer gains get large, eventually leading to high gain effects. Explain this phenomenon. Compute and plot (vs. v) the associated crossover frequencies and MIMO gain/phase margins at the output breakpoint of the nominal system. Comment on your results.

Exercise 14.5 For the aircraft data from Sect. 14.6, use the squaring-up method to select your own (\bar{B}, v) , redesign the controller, rerun simulation tests, and compare your data with the original results. Test and comment on the system tracking and gust rejection performance. Select a matched nonparametric uncertainty and, without retuning the controller, rerun simulation tests with increasing gust magnitudes. Comment on your results.

Exercise 14.6 Using the aircraft data from Sect. 14.6, replace the system regulated output a_{z2} with q_b . Design a (Baseline OBLTR + Adaptive) output feedback controller to track bounded commands. Simulate the closed-loop system using various uncertainties (parametric and nonparametric), as well as the gust model from Sect. 14.6. Comment on your results.

Exercise 14.7 Use the model data from Sect. 14.8. Change the regulated output to $y_{\text{reg}} = (p_s \beta)^T$. Assume that all three states in the corresponding MIMO system are available for control design. Repeat the four steps from the study in Sect. 14.8 to design, analyze, and simulate (OBLTR + Adaptive) controller. Note: The squaring-up design must be performed without placing transmission zeros. Explain why it is

the case here. Compare your design data to that of the baseline LQR PI state feedback controller. Comment on your results.

References

1. Kwakernaak H, Sivan R (1972) The maximally achievable accuracy of linear optimal regulators and linear optimal filters. *IEEE Trans Autom Control* 17(1):79–86
2. Lavretsky E (2012) Adaptive output feedback design using asymptotic properties of LQG/LTR controllers. *IEEE Trans Autom Control* 57(6):1587–1591
3. Dydek ZT, Annaswamy AM, Lavretsky E (2010) Adaptive control and the NASA X-15 program: a concise history, lessons learned, and a provably correct design. *IEEE Control Syst Mag*
4. Narendra KS, Annaswamy AM (2005) Stable adaptive control. Dover, New York
5. Ioannou P, Fidan P (2006) Adaptive control tutorial. In: SIAM, Advances in design and control, SIAM, PA
6. Lavretsky E, Wise KA (2013) Robust and adaptive control with aerospace applications. In: Advanced textbooks in control and signal processing. Springer, London. ISBN: 978-1-4471-4395-6 (print), 978-1-4471-4396-3 (online)
7. Lavretsky E (2011) Reference dynamics modification in adaptive controllers for improved transient performance. In: Proceedings of AIAA guidance, navigation and control conference, Portland, OR
8. Gibson TE (2014) Closed-loop reference model adaptive control with application to very flexible aircraft. PhD thesis, Massachusetts Institute of Technology
9. Gibson TE, Qu Z, Annaswamy AM, Lavretsky E (2015) Adaptive output feedback based on separation principle and closed-loop reference models. *IEEE Trans Autom Control*
10. Kevorkian J, Cole JD (1996) Multiple scale and singular perturbation methods. *Appl Math Sci* 114
11. Balas G, Young P (2003) Sensor selection via closed-loop control objectives. *IEEE Trans Autom Control* 7(6):692–704
12. Wal M, Jager B (2001) A review of methods for input/output selection. *Automatica* 37:487–510
13. Misra P (1998) Numerical algorithms for squaring-up non-square systems. Part II: General case. In: Proceedings of American control conference, San Francisco, CA
14. Anderson BDO, Moore JB (1990) Optimal control, linear quadratic methods. Dover, Mineola
15. Gibson TE, Qu Z, Annaswamy AM, Lavretsky E (2015) Adaptive output feedback based on closed-loop reference models. *IEEE Trans Autom Control* 60(10):2728–2733
16. Lavretsky E, Gibson TE (2011) Projection operator in adaptive systems. Arxiv preprint, arXiv:1112.4232
17. McLean D (1978) Gust-alleviation control systems for aircraft. *Proc Inst Electr Eng Control Sci* 125(7):675–685
18. Gibson TE, Annaswamy AM, Lavretsky E (2013) Closed-loop reference model adaptive control. Part I: Transient performance. In: American control conference
19. Gibson TE, Annaswamy AM, Lavretsky E (2013) Closed-loop reference model adaptive control: composite control and observer feedback. In: 11th IFAC International workshop on adaptation and learning in control and signal processing
20. Stein G (2003) Respect the unstable. *Control Syst Mag* 23(4):12–25
21. Bosworth JT (1992) Linearized aerodynamic and control models of the X-29A airplane and comparison with flight data. NASA technical memo, 4356
22. Åström KJ, Murray RM (2008) Feedback systems: an introduction for scientists and engineers. Princeton University Press, Princeton, p 08540

23. Krstic M, Kokotovic PV (1993) Transient-performance improvement with a new class of adaptive controllers. *Syst Control Lett* 21:451–461
24. Datta A, Ioannou P (1994) Performance analysis and improvement in model reference adaptive control. *IEEE Trans Autom Control* 39(12)
25. Zang Z, Bitmead R (1994) Transient bounds for adaptive control systems. *IEEE Trans Autom Control* 39(1)
26. Luenberger DG (1964) Observing the state of a linear system. *IEEE Trans Mil Electron MIL-8*:74–80
27. Khalil HK (1996) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, p 07458
28. Gibson TE, Annaswamy AM, Lavretsky E (2012) Improved transient response in adaptive control using projection algorithms and closed loop reference models. In: AIAA guidance navigation control conference



Robust and Adaptive Output Feedback Control for Square Non-Minimum Phase Systems

15

We introduced the observer-based loop transfer recovery (OBLTR) methodology in Chap. 6 and discussed OBLTR controllers with adaptive augmentation in Chaps. 13 and 14. These methods are applicable to a wide class of linear time-invariant (LTI) multi-input–multi-output (MIMO) systems with matched uncertainties and with the number of output measurements exceeding that of the control inputs (tall dynamics). In this chapter, we will extend the OBLTR design to a class of square LTI MIMO systems, with matched uncertainties, whereby the nominal (no uncertainties) LTI MIMO system dynamics are square, with equal number of control inputs and output measurements that also represent the regulated outputs of interest. The nominal system is allowed to be non-minimum phase and possess arbitrary relative degree from the system control input to the regulated output. Nonlinear state-dependent uncertainties enter the system dynamics through the range of the nominal control effectiveness matrix. Their effects are matched by the presence of the system control input. The developed control methodology consists of the OBLTR design with a direct adaptive augmentation. The OBLTR portion of the design employs the squaring-up procedure from Chap. 6, which is reviewed here for completeness and then extended to encompass non-minimum phase square systems with arbitrary relative degree. For the squared-up nominal system dynamics, we design a robust baseline linear dynamic output feedback controller, whose gain and phase margins at the plant input recover the margins of a state feedback optimal linear quadratic regulator (LQR). In order to extend robustness of the baseline OBLTR controller to matched state-dependent nonlinear uncertainties, we add a direct model reference adaptive output feedback augmentation, followed by a concise step-by-step summary of the overall process to build a (Robust + Adaptive) dynamic output feedback servo-controller. As we already mentioned, this design is applicable to a large class of square non-minimum phase systems with arbitrary relative degree. Throughout the chapter, relative degree zero

aircraft longitudinal dynamics are utilized to elucidate the key design features, as they sequentially appear in the text. The chapter ends with a brief summary and a set of conclusions.

15.1 Introduction

In aerospace systems, typical measurements for flight control design include: air-speed, linear accelerations, angular rates, and aerodynamic angles, such as angle of attack and angle of sideslip. All these signals are measured by a suite of sensors mounted on an aircraft. They define the vehicle body-fixed degrees-of-freedom. In flight systems, the angular rate sensors are the most reliable. They are the “backbone” of almost all flight controllers that are in operation today. In addition, aerodynamic angles play an important role in the design of flight critical systems.

Toward that end, a challenging question arises: “Can an aerial vehicle be reliably controlled without knowing angular rates and aerodynamic angles?” The ideas put forward in this chapter suggest that the answer is a Yes. For example, it is shown that a vertical acceleration command-following flight control system can be designed for an aerial system based only on the vehicle vertical acceleration sensor. This is a very challenging control design task. The longitudinal dynamics of an aircraft are often non-minimum phase with a non-unit relative degree. For an aircraft, these two properties take place if an aft-mounted surface (such as an elevator) is the primary control input and the vertical acceleration is the regulated output. In that case, the system relative degree is zero. Moreover, modern agile aircraft are often designed to be open-loop unstable and have state-dependent aerodynamic uncertainties.

In general, output feedback design is a longstanding problem of interest in dynamics and control. A comprehensive survey on the subject can be found in [1]. The seminal concept of a state observer was first introduced in [2]. Textbooks such as [3–6] cover robust control theoretical background as well as many applications. It is fair to say that a vast majority of robust output feedback designs focus on achieving closed-loop stability and command tracking performance. Fewer designs exist that guarantee relative stability. Among those, the linear quadratic Gaussian synthesis with loop transfer recovery (LQG/LTR) [3, 7, 8] represents one of the most often utilized (in practice) methodologies for robust output feedback. In addition to closed-loop stability and tracking, this particular design guarantees relative stability with quantifiable margins. It is interesting to note that the origins of LQG/LTR can be traced back to the now-classical textbook by Kwakernaak and Sivan [9], where an observer-based control design to recover output margins was first proposed in Sect. 5.6. Perhaps due to the section asterisk marking (which indicates optional reading), the original concept of margins recovery with observer-based controllers went largely unnoticed until a decade later, when the LQG/LTR concept appeared in the literature.

Overall, what makes the LQG/LTR method appealing to industrial applications is its programmability. In other words, software tools can be written to automate the development of production-ready industrial controllers for systems that may have thousands of design points. In this case, using classical control design (or any other method that requires manual tuning) is time-cost consuming and thus prohibitive.

Original ideas and framework for the design of adaptive output feedback controllers can be found in [10, 11]. Many theoretical advancements were made since. Three notable contributions include: (a) adaptive backstepping [12]; (b) adaptive control with high gain observers [13, 14]; and (c) multiple model adaptive control [15]. Analysis of transients in adaptive systems is reported in [16]. The development of adaptive controllers with minimal state observers was first reported in [17, 18]. Recent extensions of that method were focused on asymptotic stability analysis and quantification of transients [19–23].

The ideas described in this chapter are formulated in generic control-theoretic terms for a class of MIMO dynamical systems whose baseline minimal linear dynamics are perturbed by a set of nonlinear matched static uncertainties. The main focus is to construct a command-following servo-control system using regulated output as the only measurement available for control synthesis. Specifically, robust and adaptive output feedback control design methods are presented for a class of MIMO dynamical systems with matched linear and state-dependent uncertainties. The developed control methodology consists of OBLTR baseline controller and a direct adaptive augmentation. The OBLTR portion of the design employs the squaring-up procedure [24] which is reviewed and then extended to non-minimum phase square systems with arbitrary relative degree. The fictitious inputs introduced via the squaring-up procedure are not used for control design. They define a Q-matrix for the observer design with the emphasis on recovering stability margins at the system input. In order to extend robustness of the baseline OBLTR controller to matched and possibly state-dependent nonlinear uncertainties, a direct model reference adaptive output feedback augmentation is added, followed by a concise step-by-step summary of the overall process to build a (Robust + Adaptive) dynamic output feedback servo-controller.

15.2 Problem Motivation

In industrial applications, a control design often begins with the definition of a finite-dimensional dynamical model that represents the process/system of interest, within a specified accuracy, and inside of a predetermined operational envelope. Alongside the model, control engineers would define the system control inputs, the regulated outputs, and a set of output measurements (the “sensors”) for implementing selected control policies. But first and foremost, a set of control requirements is stated to formalize the desired process specifications, achievable via admissible

control actions. Among common control requirements, there are three “must-haves” with precise quantifiable measures for any selected control policy to attain. They are:

- (1) Closed-loop stability.
- (2) Relative stability, such as gain and phase margins at the plant inputs and at control-critical sensors.
- (3) Command tracking performance, such as rise and settling time, percent undershoot and overshoot.

Whether a specific controller achieves these design targets would be verified via formal analysis and later validated through numerous tests in a high-fidelity simulation and a hardware-in-the-loop environments. These verification and validation steps are essential to reducing risks in fielding a control system.

Modeling for control design in a realistic engineering environment typically begins with the formulation of a finite set of linear time-invariant (LTI) open-loop plants in the classical state-space form,

$$\begin{aligned}\dot{x}_p &= A_p x_p + B_p u, \quad y_{\text{meas}} = C_p \text{meas} x_p + D_p \text{meas} u, \\ y_{\text{reg}} &= C_p \text{reg} x_p + D_p \text{reg} u,\end{aligned}\tag{15.1}$$

whereby each linear system is verified to be controllable and observable (that is minimal), while the corresponding dynamics approximate the evolution of the true process around its selected equilibrium. In (15.1), $x_p \in R^{n_p}$ denotes the n_p -dimensional state, $u \in R^m$ is the m -dimensional vector of controls, $y_{\text{meas}} \in R^{n_{y_{\text{meas}}}}$ combines the system output measurements, and $y_{\text{reg}} \in R^{n_{y_{\text{reg}}}}$ represents the regulated output. These output signals are recorded online (e.g., during the system operation) by a suit of physical devices, called the sensors.

In what follows, the regulated outputs are considered to be the only measurements available in the system, and the number of controls is assumed to be equal to the number of the measured outputs $n_{y_{\text{meas}}} = n_{y_{\text{reg}}} = m$, that is, the system is square, and

$$y_{\text{meas}} = y_{\text{reg}} = C_p \text{reg} x_p + D_p \text{reg} u.\tag{15.2}$$

This is an “extreme” and the most challenging case to deal with in controls engineering. Often, the process has more sensed outputs than controls, that is, the system is “tall”. Robust and adaptive control solutions for tall systems are presented in Chaps. 6, 13, and 14, where asymptotic arguments are extensively utilized [25] to analyze transient dynamics in the time domain and relative stability properties in the frequency domain.

Rationale for considering tall systems comes primarily from aerospace and other industrial applications where the tallness of the system dynamics can be achieved via sensor placement [26, 27].

In what follows, the system open-loop dynamics ($u = 0$) are allowed to be unstable, have arbitrary relative degree from the control input to the regulated output, and possess non-minimum phase (that is unstable) finite transmission zeros.

Definition 15.1 A complex number z_0 is the finite transmission zero of (15.1), (15.2) if the system Rosenbrock matrix $R(s) = \begin{pmatrix} A_p - s I_{n_p \times n_p} & B_p \\ C_{p \text{ reg}} & D_{p \text{ reg}} \end{pmatrix}$ drops rank at $s = z_0$ [1, 4]

$$\underbrace{\text{rank} \begin{pmatrix} A_p - z_0 I_{n_p \times n_p} & B_p \\ C_{p \text{ reg}} & D_{p \text{ reg}} \end{pmatrix}}_{R(z_0)} < \max_s \underbrace{\text{rank} \begin{pmatrix} A_p - s I_{n_p \times n_p} & B_p \\ C_{p \text{ reg}} & D_{p \text{ reg}} \end{pmatrix}}_{R(s)} \leq (n_p + m). \quad (15.3)$$

The system is called minimum phase if its finite transmission zeros are located in the open left half complex plane \mathbb{C}^- .

It is well known that unstable non-minimum phase systems with arbitrary relative degree are hard to control via output feedback [28], that is, the desired stability, robustness, and performance specifications may not always be achievable. To further motivate the upcoming discussion, consider a rather simple yet very realistic example from the aerospace industry, whereby an unstable non-minimum phase system with zero relative degree is of prime interest.

Example 15.1 Longitudinal Short-Period Dynamics of an Aircraft In Chap. 1, we show that the longitudinal short-period dynamics of an aircraft near the vehicle equilibrium can be approximately represented as a second-order minimal system in the LTI form [29].

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V_0} & 1 + \frac{Z_q}{V_0} \\ M_\alpha & M_q \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} \frac{Z_{\delta_e}}{V_0} \\ M_{\delta_e} \end{pmatrix} \delta_e. \quad (15.4)$$

Within the first few seconds, these dynamics fairly accurately describe the aircraft motion due to an aft-mounted movable horizontal surface, called the elevator, whose deflection is denoted by δ_e (rad). Moving the elevator trailing edge up (negative) or down (positive) will primarily induce changes in the aircraft aerodynamic pitching moment, and as a result, the vehicle would rotate and pitch its nose up or down, respectively.

As it is shown in Fig. 15.1, the angle between the aircraft nose and the horizon is the pitch angle θ (rad), $\dot{\theta} = q$ is the pitch rate (rad/sec), α is the angle of attack (rad), and V_0 denotes the trimmed airspeed in feet per second (feet/sec). Also, the constant quantities $(Z_\alpha, Z_q, Z_{\delta_e}, M_\alpha, M_q, M_{\delta_e})$ represent partial derivatives of the vehicle aerodynamic forces and moments due to their subscripted variables. The

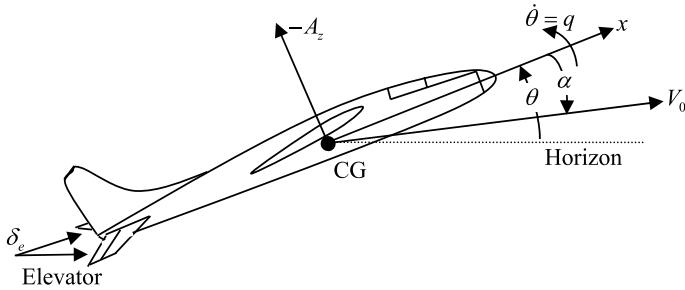


Fig. 15.1 Aircraft longitudinal dynamics nomenclature

regulated output of interest is the aircraft vertical acceleration A_z (ft/s², positive down), computed at the vehicle center of gravity (CG). Its direction is perpendicular to the longitudinal x -axis, such that negative values of A_z indicate an upward motion tendency of the aircraft. According to [29],

$$A_z = Z_\alpha \alpha + Z_{\delta_e} \delta_e, \quad (15.5)$$

where Z_α and Z_{δ_e} are partial derivatives of the aircraft vertical force due to changes in the vehicle angle of attack α and the elevator deflection δ_e , correspondingly. Nominal values for Z_{δ_e} are strictly negative. Because of that, when the elevator is deflected upward (negative) to rotate the aircraft and climb, the vehicle CG would initially lose altitude prior to executing its commanded climb maneuver. This phenomenon is common to all tail-driven mechanical devices, and it can be explained by the fact that the vehicle $\delta_e \rightarrow A_z$ dynamics are non-minimum phase, that is, the system open-loop transfer function has at least one unstable zero. For an open-loop unstable short-period aircraft dynamics, Fig. 15.2 shows a typical change in the vehicle vertical acceleration, when the elevator trailing edge moves up to execute a step-input command.

The model parameters used in this simulation are

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} -1.0527 & 1.0 \\ 0.2912 & -0.1292 \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} -0.0343 \\ -1.1684 \end{pmatrix} \delta_e, \\ A_z = -346.48 \alpha - 11.29 \delta_e. \quad (15.6)$$

Note that formally speaking the system relative degree is zero. A second-order transfer function in the form

$$\frac{\delta_e}{\delta_{cmd}} = \frac{\omega_{act}^2}{s^2 + 2\xi_{act}\omega_{act}s + \omega_{act}^2}, \quad (15.7)$$

can be added to emulate the elevator actuator dynamics from the commanded position δ_{cmd} to the actual/achieved deflection δ_e , with the actuator natural frequency $\omega_{act} =$

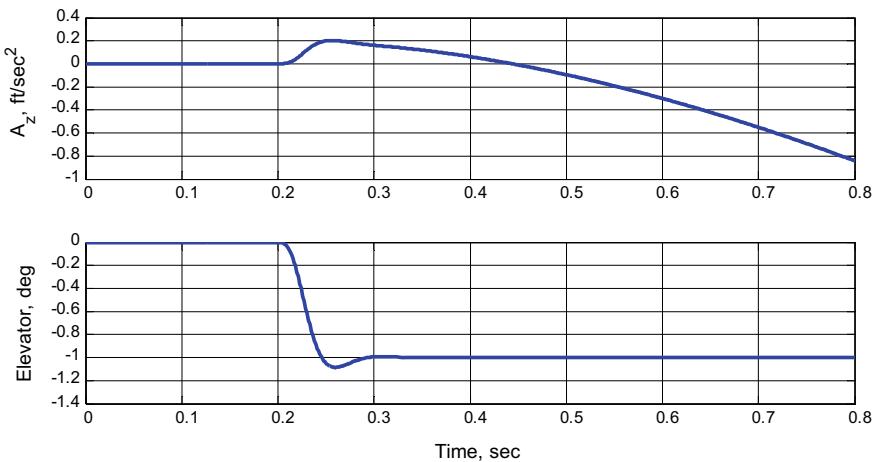


Fig. 15.2 Open-loop non-minimum phase vertical acceleration response due to elevator deflection

125.66 rad/s (20 Hz), and the actuator damping ratio $\xi_{\text{act}} = 0.7$. These are typical values, representative of an aircraft elevator surface actuation device. As seen from Fig. 15.2, when the elevator starts to move its trailing edge up (at 0.2 s), the aircraft vertical acceleration A_z momentarily increases in the “wrong” direction and then reverses its path to finally execute the commanded climb maneuver (a negative value). Such an undesired trend is typical of a system with non-minimum phase dynamics. In this particular case, the open-loop system is unstable and has two real zeros, one of which is positive. Table 15.1 shows the system unstable poles and zeros highlighted in bolditalic.

Clearly, the selected open-loop short-period aircraft dynamics are unstable and non-minimum phase. Moreover, even though the system relative degree is formally zero, it represents a “weak” relative degree condition. Those readers who are familiar with aerospace applications should immediately recognize that the vertical acceleration of an aircraft would be controlled via a change of the aircraft angle of attack α by moving elevator δ_e , changing the aircraft pitch rate q through the B matrix, and as result, forcing α increase or decrease. In other words, aircraft vertical acceleration would not be regulated directly through the D -matrix. What it means is that the “strong” relative degree of this system is two, not zero. That is the main reason why we selected (in this example) the aircraft vertical acceleration to be the regulated

Table 15.1 Open-loop short-period dynamics poles and zeros

Poles	Zeros
-1.3012	5.9484
0.11927	-6.0775

The unstable zero is faster than the unstable pole

output. It demonstrates one of the most challenging yet very practical applications in aerospace. ■

In practice, the system parameters are rarely known exactly. In other words, the dynamics (15.1) could represent the so-called nominal system, which in turn may substantially differ from the real process. In order to bring a realism into the model definition, it is plausible to introduce a set of structured uncertainties, while preserving the model controllability and observability properties. For clarity sake, only matched uncertainties [10, 11] will be considered. In this case, the perturbed model dynamics become

$$\begin{aligned}\dot{x}_p &= A_p x_p + B_p \Lambda \left(u + \Theta^T \Phi(x_p) \right), \\ y_{\text{reg}} &= C_p \text{reg} x_p + D_p \text{reg} \Lambda \left(u + \Theta^T \Phi(x_p) \right),\end{aligned}\quad (15.8)$$

where $\Lambda \in R^{m \times m}$ is a constant diagonal matrix with unknown strictly positive diagonal elements, $\Theta \in R^{N \times m}$ is an unknown constant matrix of parameters, and $\Phi(x_p) \in R^{N \times 1}$ is the known state-dependent regressor vector with Lipschitz-continuous components. Rationale for selecting these uncertainties can be found in early chapters. Suffices to say, these definitions stem from the observed in practice model inaccuracies for many aerial vehicles.

The design goal addressed in this chapter is that of finding a control policy u for the system regulated output y_{reg} to closely track bounded time-varying commands y_{cmd} , while minimizing transients and maximizing overall robustness of the selected control to model imperfections. In addition, the control design task must be accomplished based on the available system measurements only, which in this case implies using the output signal y_{reg} to synthesize the control input u . In essence, the task at hand is to design a servomechanism logic for a square MIMO system (15.8) with matched uncertainties, whose open-loop nominal linear dynamics (15.1) are allowed to be unstable, non-minimum phase, and have an arbitrary relative degree from the system control input to the selected regulated output.

The developed, in this chapter, control methods rely on the design of a baseline output feedback OBLTR controller augmented by a direct model reference adaptive control (MRAC), Chaps. 6, 13, 14.

Throughout the chapter, we will use the aircraft dynamics from Example 15.1 to demonstrate the design steps and its essential features, as they appear in the text.

15.3 The Squaring-Up Design for Non-Minimum Phase Systems with Arbitrary Relative Degree

As we have already discussed in Chap. 6, the squaring-up method developed in [24] applies to the conventional LTI dynamics.

$$\dot{x}_p = A_p x_p + B_p u, \quad y = C_p x_p + D_p u,$$

$$x_p \in R^{n_p}, \quad u \in R^m, \quad y_p \in R^p. \quad (15.9)$$

If the number of the system output measurements p is greater than the number of the control inputs m , the system is called “tall”. If on the other hand $p < m$, the system is “wide”. Finally, if $p = m$ then the system is “square”. Originally developed for a class of wide systems, the squaring-up method is easily restated for tall systems, via a duality argument and with the corresponding set of sufficient conditions.

1. (A_p, C_p) is observable, and C_p has full row rank p .
- 2.

$$\text{rank}(C_p B_p) = m < p. \quad (15.10)$$

3. Finite transmission zeros of (15.9) (if any) are stable.

By and large, squaring-up a tall system entails finding $(p - m)$ columns $B_{p2} \in R^{n_p \times (p-m)}$ such that all finite transmission zeros of the newly formed $(p \times p)$ square system, with the added $(p - m)$ “fictitious” inputs, are placed at the prescribed locations in \mathbb{C}^- . A total of $(n_p - p)$ transmission zeros can be assigned via the squaring-up method. Columns in B_{p2} can be thought of as extra input directions for process disturbances entering the system dynamics. We discussed that feature in Chap. 6. As a result of the squaring-up, the modified system with the newly added inputs becomes $(p \times p)$ square, minimum phase, and relative degree one, that is,

$$\begin{aligned} \text{rank} \left(\begin{array}{c|c} A_p - z_0 I_{n \times n} & \bar{B} \\ \hline C_p & 0_{p \times p} \end{array} \right) &< n + p \\ \Rightarrow z_0 \in \mathbb{C}^- \quad \wedge \quad \det(C_p \bar{B}) &\neq 0 \end{aligned} \quad (15.11)$$

Under the above-stated assumptions (15.10), the theoretical proof and the corresponding numerical algorithm from [24] provide an efficient solution to the squaring-up problem. We presented detailed analysis and control-oriented applications of the squaring-up method in Chap. 6, where the OBLTR concept was introduced.

However, this method cannot be directly applied to the already square non-minimum phase dynamics (15.1), (15.2) whose relative degree may not be unity. In this case, the second and the third sufficient conditions for squaring-up may no longer hold. In fact, often for a single-input-single-output (SISO) LTI non-minimum phase system, the regulated output can be defined such that its relative degree is zero or greater than unity. In the latter case $C_p B_p = 0$, and the original squaring-up method is bound to fail. For example, the double integrator $\ddot{x} = u$, with the position measurement $y_{\text{meas}} = x$, is not a tall system nor does it satisfy the squaring-up conditions (15.10).

In what follows, we will formulate an extension modification to the open-loop system dynamics to circumvent the second and the third assumptions in (15.10). The proposed change starts with the redefinition of the system output measurements. Toward that end, two new outputs are introduced. The first one represents the tracking error integrator,

$$\dot{e}_{yI} = y_{\text{reg}} - y_{\text{cmd}} \quad (15.12)$$

and the second signal defines a low pass filtered version of the system control input,

$$\dot{u}_f = \frac{(u - u_f)}{\tau_u} \quad (15.13)$$

with the filter time constant $\tau_u > 0$.

Before we proceed any further, several remarks are in order. It is customary in controls engineering to embed an error integrator such as (15.12) into the design, so that constant external commands can be tracked with zero errors, while constant disturbances are rejected with zero errors as well. Another common practice in controls is to add a filter such as (15.13) into the loop, so as to mitigate measurement noise, or to restrict the closed-loop system bandwidth and prevent excitation of unmodeled high-order dynamics, such as structural modes. As we show later, these techniques result in a redefined system output whose D -matrix is zero.

In general control applications, an open-loop dynamics modification such as (15.12)–(15.13) is also known as the “frequency shaping” design [4]. It is often employed to construct optimal LQR controllers via frequency-dependent cost weights.

Merging (15.12), (15.13) with the open-loop plant dynamics (15.8), but without the matched uncertainties ($\Lambda = I_{m \times m}$, $\Theta = 0_{N \times m}$), gives the nominal extended open-loop tall MIMO system,

$$\begin{aligned} \underbrace{\begin{pmatrix} \dot{e}_{yI} \\ \dot{u}_f \\ \dot{x}_p \end{pmatrix}}_{\dot{x}} &= \underbrace{\begin{pmatrix} 0_{m \times m} & 0_{m \times m} & C_p \text{ reg} \\ 0_{m \times m} & -\frac{1}{\tau_u} I_{m \times m} & 0_{m \times n_p} \\ 0_{n_p \times m} & 0_{n_p \times m} & A_p \end{pmatrix}}_A \underbrace{\begin{pmatrix} e_{yI} \\ u_f \\ x_p \end{pmatrix}}_x \\ &+ \underbrace{\begin{pmatrix} D_p \text{ reg} \\ \frac{1}{\tau_u} I_{m \times m} \\ B_p \end{pmatrix}}_B u + \underbrace{\begin{pmatrix} -I_{m \times m} \\ 0_{m \times m} \\ 0_{n_p \times m} \end{pmatrix}}_{B_{\text{cmd}}} y_{\text{cmd}} \\ y_{\text{reg}} &= \underbrace{\begin{pmatrix} 0_{m \times m} & 0_{m \times m} & C_p \text{ reg} \end{pmatrix}}_{C_{\text{reg}}} \underbrace{\begin{pmatrix} e_{yI} \\ u_f \\ x_p \end{pmatrix}}_x + \underbrace{D_p \text{ reg}}_{D_{\text{reg}}} u \end{aligned} \quad (15.14)$$

with m controls and $2m$ output measurements.

$$\bar{y}_{\text{meas}} = \underbrace{\begin{pmatrix} I_{m \times m} & 0_{m \times m} & 0_{m \times n_p} \\ 0_{m \times m} & I_{m \times m} & 0_{m \times n_p} \end{pmatrix}}_{\bar{C}} \underbrace{\begin{pmatrix} e_y I \\ u_f \\ x_p \end{pmatrix}}_x = \bar{C} x \in R^{2m \times 1}. \quad (15.15)$$

As seen from (15.15), the original measurement (that is the regulated output) $y_{\text{meas}} = y_{\text{reg}}$ is replaced by the integrated tracking error $e_y I$. Also, the filtered control signal u_f is added as yet another output measurement. This procedure gives $p = 2m$ output measurements in the extended system with m control inputs. Clearly, the newly formed extended system remains observable, and its first Markov parameter has full column rank.

$$\text{rank}(\bar{C} B) = \text{rank} \left(\begin{pmatrix} I_{m \times m} & 0_{m \times m} & 0_{m \times n_p} \\ 0_{m \times m} & I_{m \times m} & 0_{m \times n_p} \end{pmatrix} \begin{pmatrix} D_{p \text{ reg}} \\ \frac{1}{\tau_u} I_{m \times m} \\ B_p \end{pmatrix} \right) = \text{rank} \left(\begin{pmatrix} D_{p \text{ reg}} \\ \frac{1}{\tau_u} I_{m \times m} \\ B_p \end{pmatrix} \right) = m. \quad (15.16)$$

So, the first and the second assumptions in (15.10) are both satisfied.

We can also show that the third assumption in (15.10) is valid. In fact, we will prove that the tall extended system (15.14), (15.15) has no finite transmission zeros, that is, the system Rosenbrock matrix,

$$\begin{aligned} \text{rank}(R_0(s)) &= \text{rank} \left(\begin{array}{cc} A - s I_{n \times n} & B \\ \bar{C} & 0_{2m \times 2m} \end{array} \right) \\ &= \text{rank} \left(\begin{array}{ccc|c} -s I_{m \times m} & 0_{m \times m} & C_{p \text{ reg}} & D_{p \text{ reg}} \\ 0_{m \times m} & -\left(\frac{1}{\tau_u} + s\right) I_{m \times m} & 0_{m \times n_p} & \frac{1}{\tau_u} I_{m \times m} \\ 0_{n_p \times m} & 0_{n_p \times m} & A_p - s I_{n_p \times n_p} & B_p \\ \hline A - \lambda I_{(n_p+2m) \times (n_p+2m)} & & & B \\ \boxed{I_{m \times m} \ 0_{m \times m} \ 0_{m \times n_p}} & & & 0_{2m \times m} \\ \boxed{0_{m \times m} \ I_{m \times m} \ 0_{m \times n_p}} & & & \bar{C} \end{array} \right) \\ &= n_p + 3m \end{aligned} \quad (15.17)$$

has full column rank for all $s \in \mathbb{C}$.

Suppose (15.17) is not true. Then there must exist a nonzero $s \in \mathbb{C}$ and a nonzero vector,

$$\begin{aligned} z &= (z_1^T, z_2^T, z_3^T, z_4^T)^T \in \mathbb{C}^{(3m+n_p) \times 1}, z_1 \in \mathbb{C}^{m \times 1}, \\ z_2 &\in \mathbb{C}^{m \times 1}, z_3 \in \mathbb{C}^{n_p \times 1}, z_4 \in \mathbb{C}^{m \times 1} \end{aligned} \quad (15.18)$$

such that

$$R_0(s)z = 0 \quad (15.19)$$

However, from the last two m -dimensional rows in (15.17) it follows that $z_1 = z_2 = 0_{n_p \times 1}$. Then the second row in (15.17) requires $z_4 = 0_{m \times 1}$. That in turn implies,

$$\begin{pmatrix} C_p \text{reg} \\ A_p - s I_{n_p \times n_p} \end{pmatrix} z_3 = 0 \quad (15.20)$$

Since the pair (C_p, A_p) is assumed to be observable then $z_3 = 0_{n_p \times 1}$, which is a contradiction to the argument. Therefore, (15.17) holds true, and all of the conditions in (15.10) are satisfied.

We now turn our attention back to squaring-up the tall system (15.14), (15.15). Adding a set of extra m columns $B_2 \in R^{(n_p+2m) \times m}$ to the system input matrix $B \in R^{(n_p+2m) \times m}$ gives the Rosenbrock $(n_p + 4m) \times (n_p + 4m)$ square matrix,

$$\begin{aligned} R(s) &= \begin{pmatrix} A - s I_{n \times n} & (B \ B_2) \\ \bar{C} & 0_{2m \times 2m} \end{pmatrix} \\ &= \left(\begin{array}{cc|c|c} -s I_{m \times m} & 0_{m \times m} & C_p \text{reg} & D_p \text{reg} \\ 0_{m \times m} & -\left(\frac{1}{\tau_u} + s\right) I_{m \times m} & 0_{m \times n_p} & \frac{1}{\tau_u} I_{m \times m} \\ 0_{n_p \times m} & 0_{n_p \times m} & A_p - s I_{n_p \times n_p} & B_p \\ \hline A - \lambda I_{(n_p+2m) \times (n_p+2m)} & & & B \\ \boxed{\begin{array}{ccc} I_{m \times m} & 0_{m \times m} & 0_{m \times n_p} \\ 0_{m \times m} & I_{m \times m} & 0_{m \times n_p} \end{array}} & & 0_{2m \times m} & 0_{2m \times m} \\ \hline \bar{C} & & & B_2 \end{array} \right) \end{aligned} \quad (15.21)$$

where $B_2 = (* * *)^T \in R^{(n_p+2m) \times m}$ needs to be found such that $R(s)$ drops rank at the prescribed transmission zeros in \mathbb{C}^- . According to [1], we can assign $(n_p + 2m) - 2m = n_p$ finite transmission zeros in a $(2m \times 2m)$ square system with the zero D -matrix and a non-singular first Markov parameter (relative degree one dynamics).

Introduce matrix partitioning,

$$\begin{aligned}\bar{B}_1 &= \begin{pmatrix} D_p \text{reg} & * \\ \frac{1}{\tau_u} I_{m \times m} & * \end{pmatrix}_{2m \times 2m}, \\ \bar{B}_2 &= (B_p \ B_{p2})_{n_p \times 2m}, \\ A &= \begin{pmatrix} \boxed{0_{m \times m}} & \boxed{0_{m \times m}} & \boxed{C_p \text{reg}} \\ \boxed{0_{m \times m}} & \boxed{-\frac{1}{\tau_u} I_{m \times m}} & \boxed{0_{m \times n_p}} \\ \boxed{A_{11}} & & \boxed{A_{12}} \\ \boxed{0_{n_p \times m}} & \boxed{0_{n_p \times m}} & \boxed{A_p} \end{pmatrix} \end{aligned} \quad (15.22)$$

and rewrite (15.21) as

$$R(s) = \begin{pmatrix} \boxed{A - s I_{n \times n}}_{(n_p+2m) \times (n_p+2m)} & \boxed{\bar{B}_1}_{2m \times 2m} \\ I_{2m \times 2m} & 0_{2m \times n_p} \\ 0_{2m \times 2m} & \boxed{\bar{B}_2}_{n_p \times 2m} \end{pmatrix} \quad (15.23)$$

Define

$$\bar{B}_1 = \begin{pmatrix} D_p \text{reg} & I_{m \times m} \\ \frac{1}{\tau_u} I_{m \times m} & 0_{m \times m} \end{pmatrix}_{2m \times 2m} \quad (15.24)$$

and immediately note that \bar{B}_1 is non-singular, with its well-defined inverse.

$$\bar{B}_1^{-1} = \begin{pmatrix} 0_{m \times m} & \tau_u I_{m \times m} \\ I_{m \times m} & -\tau_u D_p \text{reg} \end{pmatrix} \quad (15.25)$$

Via (15.22) and (15.24), the rank of $R(s)$ can be computed explicitly.

$$\begin{aligned}\text{rank}(R(s)) &= \text{rank} \begin{pmatrix} \boxed{A - s I}_{(n_p+2m) \times (n_p+2m)} & \boxed{\bar{B}_1}_{2m \times 2m} \\ I_{2m \times 2m} & 0_{2m \times n_p} \\ 0_{2m \times 2m} & \boxed{\bar{B}_2}_{n_p \times 2m} \end{pmatrix} \\ &= \text{rank} \left\{ \begin{pmatrix} I_{2m \times 2m} & 0 & 0 \\ -\bar{B}_2 \bar{B}_1^{-1} I_{n_p \times n_p} & & 0 \\ 0 & I_{2m \times 2m} & \end{pmatrix} \right. \\ &\quad \left. \begin{pmatrix} A_{11} - s I_{2m \times 2m} & A_{12} & \boxed{\bar{B}_1}_{2m \times 2m} \\ 0_{n_p \times 2m} & A_p - s I_{n_p \times n_p} & \boxed{\bar{B}_2}_{n_p \times 2m} \\ \boxed{I_{2m \times 2m}} & 0_{2m \times n_p} & 0_{2m \times 2m} \end{pmatrix} \right\} \end{aligned}$$

$$= \text{rank} \left(\begin{array}{cc|c|c} A_{11} - s I_{2m \times 2m} & & A_{12} & \boxed{\bar{B}_1} \\ * & \boxed{A_p - \bar{B}_2 \bar{B}_1^{-1} A_{12} - s I_{n_p \times n_p}} & & 0_{n_p \times 2m} \\ \hline & \boxed{I_{2m \times 2m}} & 0_{2m \times n_p} & 0_{2m \times 2m} \end{array} \right) \quad (15.26)$$

Since \bar{B}_1 is non-singular and the last ($2m$) rows of the matrix in the right-hand side of (15.26) are linearly independent with respect to each other and relative to the first ($2m$) rows, the rank of $R(s)$ must be no less than ($4m$). At the same time,

$$\boxed{n_p + 4m} = \max_s(\text{rank}(R(s))) = 4m$$

$$+ \max_s \left(\text{rank} \left(\underbrace{\left(\underbrace{(A_p - \bar{B}_2 \bar{B}_1^{-1} A_{12})}_{A_{pcl}} - s I_{n_p \times n_p} \right)}_{\leq n_p} \right) \right) \quad (15.27)$$

Therefore, setting the prescribed transmission zeros in the extended system is equivalent to placing poles of

$$A_{pcl} = A_p - \bar{B}_2 \bar{B}_1^{-1} A_{12}, \quad (15.28)$$

at the desired transmission zero locations. Substituting (15.22) and (15.25) into (15.28) yields

$$\begin{aligned} A_{pcl} &= A_p - \bar{B}_2 \bar{B}_1^{-1} A_{12} = A_p - (B_p \ B_{p2}) \begin{pmatrix} 0_{m \times m} & \tau_u I_{m \times m} \\ I_{m \times m} & -\tau_u D_{p \text{reg}} \end{pmatrix} \begin{pmatrix} C_{p \text{reg}} \\ 0_{m \times n_p} \end{pmatrix}, \\ &= A_p - (B_p \ B_{p2}) \begin{pmatrix} 0_{m \times m} \\ C_{p \text{reg}} \end{pmatrix} = A_p - B_{p2} C_{p \text{reg}} \end{aligned} \quad (15.29)$$

Since $(A_p, C_{p \text{reg}})$ is observable then any desired set of n_p transmission zeros for A_{pcl} can be assigned through a proper definition of $B_{p2} \in R^{n_p \times m}$ in (15.29). The now proven result is summarized below.

Theorem 15.1 Any observable $(m \times m)$ LTI dynamics in the form

$$\begin{pmatrix} A_p & B_p \\ C_{p \text{reg}} & D_{p \text{reg}} \end{pmatrix} \sim \begin{cases} \dot{x}_p = A_p x_p + B_p u \\ y_{\text{reg}} = C_{p \text{reg}} x + D_{p \text{reg}} u, \end{cases} \quad (15.30)$$

with m controls u , n_p states x_p , and m regulated outputs y_{reg} can be squared-up such that the resulting extended $(2m \times 2m)$ system (15.14), written symbolically as,

$$\left(\begin{array}{c|cc} A & \bar{B} \\ \hline C & 0_{2m \times 2m} \end{array} \right) = \left(\begin{array}{ccc|cc|c} 0_{m \times m} & 0_{m \times m} & C_{p \text{ reg}} & D_{p \text{ reg}} & I_{m \times m} \\ 0_{m \times m} & -\frac{1}{\tau_u} I_{m \times m} & 0_{m \times n_p} & \frac{1}{\tau_u} I_{m \times m} & 0_{m \times m} \\ 0_{n_p \times m} & 0_{n_p \times m} & A_p & B_p & B_{p2} \\ \hline & A & & B & B_2 \\ \hline & I_{m \times m} & 0_{m \times m} & 0_{m \times n_p} & 0_{2m \times m} & 0_{2m \times m} \\ & 0_{m \times m} & I_{m \times m} & 0_{m \times n_p} & & \end{array} \right) \quad (15.31)$$

with any constant $\tau_u > 0$ and $\bar{B} = (B \ B_2)$, has a non-singular first Markov parameter, $\det(\bar{C} \bar{B}) \neq 0$, and all of its n_p finite transmission zeros are assigned as desired. The zero assignment can be performed by placing poles of $(A_p - B_{p2} C_{p \text{ reg}})$ to the desired transmission zero locations, via $B_{p2} \in R^{n_p \times m}$ from (15.31), applied to the observable pair $(C_{p \text{ reg}}, A_p)$.

Remark 15.1 The proof of the theorem is constructive. It is based on the original ideas from [24] and gives a numerical algorithm to calculate B_{p2} to then form $B_2 = (I_{m \times m} \ 0_{m \times m} \ B_{p2}^T)^T$ such that the Rosenbrock matrix $R(s)$ in (15.21) drops rank exactly at the set of the n_p desired transmission zeros. Calculation of the squaring-up solution B_{p2} can be accomplished via direct pole placement or using the linear quadratic regulator (LQR) design. The only requirement for the algorithm to work is the observability of the original system.

Corollary 15.1 For any non-singular matrix $M \in R^{2m \times 2m}$, transmission zeros of $\left(\begin{array}{c|cc} A & \bar{B} M \\ \hline C & 0_{2m \times 2m} \end{array} \right)$ are the same as those of the system (15.31).

Proof of Corollary 15.1

As seen from (15.28), the squaring-up solution matrix $B_2 \in R^{(n_p+2m) \times m}$ is not unique. In fact, right-multiplying $\bar{B} = (B \ B_2)$ by any non-singular matrix $M \in R^{2m \times 2m}$ does not change $A_{p \text{ cl}}$ and proves the claimed scaling invariance of the assigned transmission zeros.

Remark 15.2 The scaling invariance property of the squaring-up technique can be utilized to improve numerical features (such as conditioning) of the system.

Example 15.2 Squaring-up the Aircraft Dynamics (15.6) This system is SISO ($p = m = 1$), unstable, relative degree zero, and non-minimum phase. We select the control filter time constant $\tau_u = 0.1$ and use (15.14)–(15.15) to write down the

nominal extended system matrices.

$$A = \begin{pmatrix} 0 & 0 & -346.48 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & 0 & -1.0527 & 1.0 \\ 0 & 0 & 0.2912 & -0.1292 \end{pmatrix},$$

$$B = \begin{pmatrix} -11.29 \\ 10 \\ -0.0343 \\ -1.1684 \end{pmatrix}, \bar{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, D = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (15.32)$$

This extended system is tall and has no finite transmission zeros. Applying Theorem 15.1 results to

$$A_p = \begin{pmatrix} -1.0527 & 1.0 \\ 0.2912 & -0.1292 \end{pmatrix}, C_{p\text{ reg}} = (-346.48 \ 0), \quad (15.33)$$

we use the LQR design to find B_{p2} such that the closed-loop matrix $(A_p - B_{p2} C_{p\text{ reg}})$ becomes Hurwitz and arrives at the resulting squared-up (2×2) system.

$$\left(\begin{array}{l} A = \begin{pmatrix} 0 & 0 & -346.48 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & 0 & -1.0527 & 1.0 \\ 0 & 0 & 0.2912 & -0.1292 \end{pmatrix}, \bar{B} = \begin{pmatrix} -11.29 & 9.765 \\ 10 & 11.02 \\ -0.0343 & -1.572 \\ -1.1684 & -3.075 \end{pmatrix} \\ \bar{C}_{\text{meas}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \bar{D}_{\text{meas}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \end{array} \right) \quad (15.34)$$

We verify that the squared-up system has stable transmission zeros $(-23.8, -1.29)$ and that the system first Markov parameter is non-singular: $\det(C B) = -222.1$. Consequently, the system relative degree is one. In (15.34), the squaring-up solution vector \bar{B}_2 is rescaled without any effect on the achieved transmission zeros (Corollary 15.1). In this example, the second column of \bar{B} is normalized and then right-multiplied by $\|B\| = 15.126$, without any effect on the prescribed transmission zeros. Note however that the zeros of the original system scalar transfer function are invariant and have not changed. In other words, the zeros of

$$P_{\text{reg}}(s) = C_{p\text{ reg}} (s I_{n_p \times n_p} - A_p)^{-1} B_p + D_{p\text{ reg}}, \quad (15.35)$$

remain the same (one of which is unstable), as shown in Table 15.1. ■

15.4 Observer-Based Loop Transfer Recovery (OBLTR) Servo-Controller for Square Systems

Using the extension of the squaring-up method from the previous section, we can now follow the OBLTR design steps (Chap. 6) to construct a robust output feedback servomechanism controller such that the regulated output y_{reg} of the nominal extended dynamics (15.14) asymptotically follows external constant commands y_{cmd} . The output feedback controller (OBLTR) includes a state feedback linear quadratic regulator (LQR) coupled with Luenberger full-order state observer. We introduced and presented details of the OBLTR control design in Chap. 6. Related information can also be found in [17]. In addition to guaranteed tracking performance, the OBLTR technique has the ability to recover stability margins of the LQR state feedback controller at the system input. The extension of the squaring-up paradigm developed in the previous section is the enabler for achieving robustness for this class of systems, that is, the loop transfer recovery (LTR) portion of the observer-based design is possible due to the fact that the nominal extended system (15.14) is constructed to be minimum phase, with the relative degree of unity and the zero D -matrix.

Later in this chapter, we will also address and analyze robustness margins of the developed servo-control solution at the original system input, as well as their dependence on the system parameters.

Assumption 15.1 The original plant (A_p, B_p) is controllable, and the open-loop system transfer function $P_{\text{reg}}(s)$ in (15.35) has no finite transmission zeros at the origin.

The above assumption is needed to preserve controllability of the extended system (15.31). Clearly, any controller that stabilizes these dynamics also solves the piecewise constant command tracking problem for the original regulated output y_{reg} . The overarching challenge here is to find a robust output feedback servomechanism solution. To address this problem, a control servosolution is obtained through the OBLTR methodology from Chap. 6.

The Design of an LQR State Feedback Controller for Extended System

Similar to the classical LQG/LTR method [1, 3, 4, 6], the first step in the OBLTR design is to construct an LQR-based state feedback controller with the desired quantifiable measures of performance and robustness. Specifically, starting with the extended open-loop dynamics (15.14),

$$\dot{x} = A x + B u + B_{\text{ref}} y_{\text{cmd}} \quad (15.36)$$

we compute an $(m \times n)$ LQR state feedback gain matrix,

$$K_{\text{lqr}} = R_{\text{lqr}}^{-1} B^T P_{\text{lqr}} \quad (15.37)$$

using the unique symmetric positive-definite solution P_{lqr} of the Algebraic Riccati Equation (ARE),

$$P_{\text{lqr}} A + A^T P_{\text{lqr}} + Q_{\text{lqr}} - P_{\text{lqr}} B R_{\text{lqr}}^{-1} B^T P_{\text{lqr}} = 0 \quad (15.38)$$

with the appropriately selected positive-definite symmetric weights $Q_{\text{lqr}} \in R^{n \times n}$ and $R_{\text{lqr}} \in R^{m \times m}$. In this case, the LQR-based state feedback control is of the form,

$$\begin{aligned} u_{\text{lqr}} &= -K_{\text{lqr}} x = -\begin{pmatrix} K_{e_y I} & K_{u_f} & K_{x_p} \end{pmatrix} \begin{pmatrix} e_y I \\ u_f \\ x_p \end{pmatrix} \\ &= \frac{K_{e_y I}}{s} (y_{\text{cmd}} - y_{\text{reg}}) - \frac{K_{u_f}}{\tau_u s + 1} u_{\text{lqr}} - K_{x_p} x_p \end{aligned} \quad (15.39)$$

where s denotes the Laplace transform (LT) complex variable.

Remark 15.3 Explicitly solving for u_{lqr} , it is interesting to see that the LQR solution,

$$u_{\text{lqr}} = \left(I_{m \times m} + \frac{K_{u_f}}{\tau_u s + 1} \right)^{-1} \left(\frac{K_{e_y I}}{s} (y_{\text{cmd}} - y_{\text{reg}}) - K_{x_p} x_p \right) \quad (15.40)$$

represents a lead-lag or a lag-lead controller, coupled with a (Proportional + Integral) state feedback (Fig. 15.3).

Consider the nominal open-loop system and the regulated plant transfer functions.

$$\begin{aligned} \text{Open-Loop System: } P(s) &= (s I_{n_p \times n_p} - A_p)^{-1} B_p \\ \text{Regulated Plant: } P_{\text{reg}}(s) &= C_p \text{ reg} P(s) + D_p \text{ reg} \end{aligned} \quad (15.41)$$

According to Fig. 15.3 and using the LQR solution (15.40), the loop gain dynamics,

$$u_{\text{out}} = -\underbrace{\frac{L g_u(s)}{\text{Loop Gain}}}_{\text{Loop Gain}} u_{\text{in}} \quad (15.42)$$

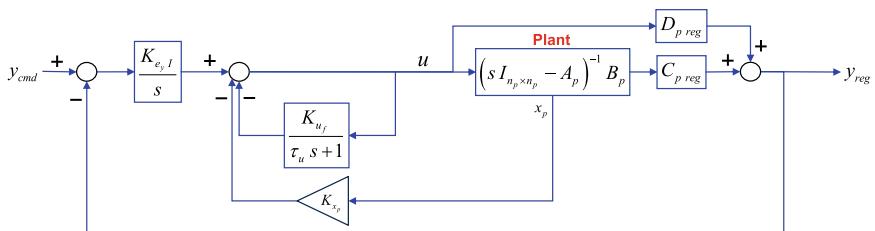


Fig. 15.3 LQR state feedback block diagram with the LQR state feedback controller (15.40)

can be derived at the plan input breakpoint. Setting $y_{\text{cmd}} = 0$ gives

$$\begin{aligned} u_{\text{out}} &= - \left(I_{m \times m} + \frac{K_{uf}}{\tau_u s + 1} \right)^{-1} \left(\frac{K_{e_y} I}{s} y_{\text{reg}} + K_{x_p} x_p \right) \\ &= - \underbrace{\left(I_{m \times m} + \frac{K_{uf}}{\tau_u s + 1} \right)^{-1} \left(\frac{K_{e_y} I}{s} P_{\text{reg}}(s) + K_{x_p} P(s) \right)}_{Lg_u(s)} u_{\text{in}} \end{aligned} \quad (15.43)$$

So, the LQR state feedback loop gain at the input breakpoint is

$$Lg_u(s) = \left(I_{m \times m} + \frac{K_{uf}}{\tau_u s + 1} \right)^{-1} \left(\frac{K_{e_y} I}{s} P_{\text{reg}}(s) + K_{x_p} P(s) \right) \quad (15.44)$$

This is the transfer function for margins analysis. When discussing such concepts as stability margins and loop transfer recovery for LTI dynamics, it is sufficient to restrict analysis and evaluation of the system transfer functions to the imaginary axis, that is, $s = j\omega$, with $\omega \in R$. The reason for that comes from Parseval's theorem and from the notion of the L_2 gain [13] (Theorem 5.4). Essentially, the L_2 gain of any MIMO LTI dynamics $y = G(s)u$ is the system H_∞ norm, which in turn equals to the largest singular value of the system transfer function, evaluated along the imaginary axis: $\sup_{\omega \in R} \sigma_{\max}(G(j\omega))$.

In Chap. 4, we showed that an LQR state feedback control solution such as (15.39), (15.40) guarantees optimal MIMO margins at the system input breakpoint.

Next, we will follow the OBLTR design guidelines from Chap. 6 to construct a full-order state observer and the corresponding OBLTR dynamic output feedback controller to recover optimal state feedback margins at the control input breakpoint for the original plant $P(s)$.

The Design of a Full State Luenberger Observer

The second step in the OBLTR design process is to convert the state feedback policy $u_{\text{lqr}} = -K_{\text{lqr}} x$ into an output feedback dynamic controller, while recovering the state feedback stability margins at the system input. The distinct feature of the OBLTR method, when compared to the classical LQG/LTR design [3, 6], is the selection of the v -parameterized weight matrices,

$$Q_v = Q_0 + \left(\frac{v+1}{v} \right) \bar{B} \bar{B}^T, \quad R_v = \frac{v}{v+1} R_0 \quad (15.45)$$

in the observer ARE,

$$P_v A^T + A P_v + Q_v - P_v \bar{C}^T R_v^{-1} \bar{C} P_v = 0 \quad (15.46)$$

where \bar{B} in (15.45) and onward is the solution of the squaring-up procedure (Theorem 15.1).

As seen from (15.45) and (15.46), the squaring-up method for transmission zero placement results in a special selection of Q_v to influence the ARE solution P_v . Specifically, choosing a sufficiently small $0 < v \ll 1$ leads to a recovery of the target LQR stability margins at the extended system input, with a non-singular symmetric positive-definite ARE solution,

$$P_v = P_0 + O(v) \quad (15.47)$$

where $P_0 = P_0^T > 0$ (Chap. 6). In contrast to the LQG/LTR design, the OBLTR-related ARE solution tends to a non-singular strictly positive-definite symmetric matrix, during the margin recovery process, while the following asymptotic relations take place as $v \rightarrow 0$:

$$P_v \bar{C}^T = \bar{B} W^T R_0^{\frac{1}{2}} + O(v), \quad P_v^{-1} \bar{B} = \bar{C}^T R_0^{-\frac{1}{2}} W + O(v) \quad (15.48)$$

where $W = (U V)^T$, the two unitary matrices U and V are defined by the singular value decomposition $\bar{B}^T \bar{C}^T R_0^{-\frac{1}{2}} = U \Sigma V$, and Σ is the diagonal matrix of the corresponding singular values. Equation (15.48) provides essential data for the OBLTR adaptive augmentation design that will be presented later.

Setting the observer gain to

$$L_v = P_v \bar{C}^T R_v^{-1} \quad (15.49)$$

the observer state dynamics are formulated,

$$\begin{aligned} \dot{\hat{x}} = & A \hat{x} + B u + B_{\text{cmd}} y_{\text{cmd}} \\ & + L_v (\bar{y}_{\text{meas}} - \hat{y}_{\text{meas}}), \quad \hat{y}_{\text{meas}} = \bar{C} \hat{x} \end{aligned} \quad (15.50)$$

with the baseline control signal,

$$u = -K_{\text{lqr}} \hat{x} \quad (15.51)$$

while using the LQR state feedback gains K_{lqr} from (15.37). Rewriting (15.50), (15.51) in the Laplace domain gives the dynamic output feedback OBLTR controller,

$$u = -K_{\text{lqr}} (s I_{n \times n} - A + B K_{\text{lqr}} + L_v \bar{C})^{-1} (B_{\text{cmd}} y_{\text{cmd}} + L_v \bar{y}_{\text{meas}}) \quad (15.52)$$

whose inputs are the commanded signal y_{cmd} and the extended system output measurements $\bar{y}_{\text{meas}} = (e_y^T I_u^T)^T$ from (15.15). Figure 15.4 shows the OBLTR block diagram, with the observer closed-loop matrix $A_{\text{obs}} = (A - B K_{\text{lqr}} - L_v \bar{C})$.

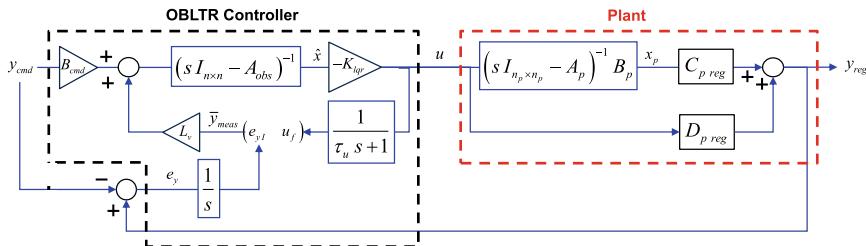


Fig. 15.4 OBLTR output feedback controller block diagram for square system

Remark 15.4 Although the two matrices $(A - B K_{lqr})$, $(A - L_v \bar{C})$ are Hurwitz by design, the observer closed-loop matrix $A_{obs} = (A - B K_{lqr} - L_v \bar{C})$ may have unstable eigenvalues [4]. This, perhaps an undesirable feature, is not due to the specifics of the OBLTR design. It rather persists in many observer-based controllers, such as LQG/LTR. Note however that the closed-loop extended system (15.36), with the OBLTR controller in the loop, is guaranteed to be stable. Nevertheless, for all practical engineering reasons, controller internal stability would have to be enforced, verified, and validated, in order to avoid utilization of unstable compensators in industrial systems.

In summary, the OBLTR design process consists of the three main steps:

- (1) Square-up the open-loop LTI dynamics and place transmission zeros at the target locations in \mathbb{C}^- .
- (2) Design a linear state feedback controller to satisfy the desired stability and robustness metrics.
- (3) Design a state observer to recover state feedback stability margins at the system input breakpoint.

With these design tuning “knobs” selected, it is possible to maintain the achieved state feedback command tracking performance and, at the same time, recover the LQR state feedback guaranteed stability margins at the system input. Note however that output margins are not guaranteed to be sufficient, and further tuning might be required to strike a reasonable balance between simultaneously enforcing adequate margins at the system input and at the selected control-critical sensors.

OBLTR Controller Interpretations

Using $\bar{y}_{\text{meas}} = \left(e_y^T I \ u_f^T \right)^T$, with its two components as defined in (15.12) and (15.13), the OBLTR control signal (15.52) can be written as,

$$u = -K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} \left(B_{\text{cmd}} y_{\text{cmd}} + \underbrace{\left(L_{v,y} \ L_{v,u} \right)}_{L_v} \left(\frac{\frac{y_{\text{reg}} - y_{\text{cmd}}}{s}}{\frac{u}{\tau_u s + 1}} \right) \right) \quad (15.53)$$

with the observer closed-loop matrix $A_{\text{obs}} = (A - B K_{\text{lqr}} - L_v \bar{C})$, and the observer gain $L_v = (L_{v,y} \ L_{v,u})$ partitioned into two matrices of the corresponding dimensions. Solving for u gives,

$$\begin{aligned} u &= - \left(I_{m \times m} + K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} \frac{L_{v,u}}{\tau_u s + 1} \right)^{-1} \\ &\quad K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} \left(B_{\text{cmd}} y_{\text{cmd}} + L_{v,y} \left(\frac{y_{\text{reg}} - y_{\text{cmd}}}{s} \right) \right) \end{aligned} \quad (15.54)$$

or equivalently,

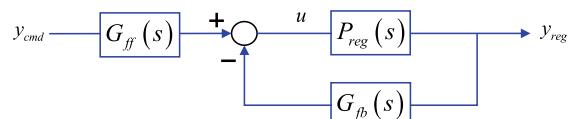
$$\begin{aligned} u &= \underbrace{\left[\left(I_{m \times m} + K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} \frac{L_{v,u}}{\tau_u s + 1} \right)^{-1} K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} \left(\frac{L_{v,y}}{s} - B_{\text{cmd}} \right) \right]}_{G_{ff}(s)} y_{\text{cmd}} \\ &\quad - \underbrace{\left[\left(I_{m \times m} + K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} \frac{L_{v,u}}{\tau_u s + 1} \right)^{-1} K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} \frac{L_{v,y}}{s} \right]}_{G_{fb}(s)} y_{\text{reg}} \\ &= \boxed{G_{ff}(s) y_{\text{cmd}} - G_{fb}(s) y_{\text{reg}}} \end{aligned} \quad (15.55)$$

where $G_{ff}(s)$ and $G_{fb}(s)$ are the command-feedforward and the regulated output feedback transfer function matrices, respectively. Then the OBLTR controller block diagram can be drawn as that of a classical closed-loop system (Fig. 15.5).

Based on (15.55) and (15.41), the system loop gains at the control input and at the regulated output are

$$Lg_u(s) = G_{fb}(s) P_{\text{reg}}(s), \quad Lg_{y_{\text{reg}}}(s) = P_{\text{reg}}(s) G_{fb}(s). \quad (15.56)$$

Fig. 15.5 OBLTR block diagram: a conventional control system with feedback and feedforward connections



Moreover, the closed-loop transfer function (the system cosensitivity) from the command to the regulated output, $y_{\text{reg}} = T_{y_{\text{cmd}} \rightarrow y_{\text{reg}}}(s) y_{\text{cmd}}$, has the form

$$T_{y_{\text{cmd}} \rightarrow y_{\text{reg}}}(s) = (I_{m \times m} + L g_{y_{\text{reg}}}(s))^{-1} P_{\text{reg}}(s) G_{\text{ff}}(s) \quad (15.57)$$

These transfer functions can be used for OBLTR robustness analysis. Next, we shall analyze asymptotic properties of the OBLTR loop gain.

15.5 Loop Transfer Recovery and OBLTR Stability Margins for Square Systems

For the extended open-loop model (15.36) operating under the OBLTR controller (15.53), the control block diagram is shown in Fig. 15.6.

The v -dependent OBLTR loop gain $L g_{u,v}(s)$ at the extended system input breakpoint can be written as

$$L g_{u,v}(s) = K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} L_v \bar{C} (s I_{n \times n} - A)^{-1} B \quad (15.58)$$

Let us define open-loop and closed-loop resolvent matrices for the extended system,

$$\Phi(s) = (s I_{n \times n} - A)^{-1}, \quad \Psi(s) = (s I_{n \times n} - A + B K_{\text{lqr}})^{-1} \quad (15.59)$$

and rewrite the LQR state feedback loop gain $L g_u(s)$ from (15.43).

$$L g_u(s) = K_{\text{lqr}}(s I_{n \times n} - A)^{-1} B = K_{\text{lqr}}\Phi(s)B \quad (15.60)$$

Next, we show that the OBLTR loop gain (15.58) asymptotically converges point-wise in s to the LQR loop gain (15.60), as $v \rightarrow 0$. Toward that end, a generic result on transfer function asymptotics is stated first.

Theorem 15.2 Consider an LTI dynamical system $(A, B, C, 0_{p \times m})$ with n states, m inputs, and $p \geq m$ outputs. Let $K \in R^{m \times n}$ be a constant matrix. Define,

$$\bar{B} = \begin{cases} B, & p = m \\ (B \ B_2), & p > m \end{cases} \quad (15.61)$$

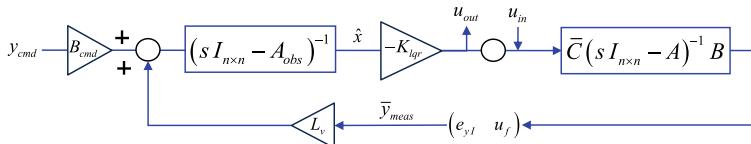


Fig. 15.6 OBLTR block diagram for square systems with a breakpoint at the extended system input

whereby for $p > m$, $B_2 \in R^{n \times (p-m)}$ denotes a constant matrix. Suppose that $S \subset \mathbb{C}$ is a set of complex numbers selected such that the eigenvalues of A , $(A - B K)$ and the finite transmission zeros of $C(s I_{n \times n} - A)^{-1} \bar{B}$ are not in S . Let $L_v \in R^{n \times p}$ be a parameter-dependent matrix and consider two transfer function matrices.

$$\begin{aligned} G_v(s) &= K(s I_{n \times n} - A + B K + L_v C)^{-1} L_v C(s I_{n \times n} - A)^{-1} B \\ G_0(s) &= K(s I_{n \times n} - A)^{-1} B \end{aligned} \quad (15.62)$$

Then the following two statements are true point-wise for any $s \in S$:

- (1) If for a constant non-singular matrix $Q \in R^{p \times p}$,

$$\lim_{v \rightarrow 0} (v L_v) = \bar{B} Q \quad (15.63)$$

then

$$\lim_{v \rightarrow 0} G_v(s) = G_0(s). \quad (15.64)$$

- (2) If for a constant non-singular matrix $Q \in R^{p \times p}$,

$$L_v = \frac{1}{v} \bar{B} Q + O(1), \quad v \rightarrow 0 \quad (15.65)$$

then

$$G_v(s) = G_0(s) + O(v), \quad v \rightarrow 0 \quad (15.66)$$

Remark 15.5 It is worth noting that neither controllability nor observability is required in stating Theorem 15.2. Also, the claimed recovery properties are not necessarily uniform in s . In fact, near the system open-loop and closed-loop poles, as well as around its transmission zeros, the recovery is definitely not uniform in the sense that $v = v(s)$. In other words, the required for recovery small values of v will certainly depend on a specific location of s in the complex plane \mathbb{C} .

Remark 15.6 The limiting expression (15.64) is less restrictive than the asymptotics (15.66). Clearly, the latter implies the former, point-wise in S . However, (15.66) quantifies an asymptotic point-wise in $s \in S$ recovery of the transfer function $G_0(s)$ in (15.66), with the recovery rate of $O(v)$, as $v \rightarrow 0$. This is a much stronger result than the limit in (15.64), which is to be expected since the asymptotic recovery predicates on a more restrictive sufficient condition, such as (15.65).

Proof of Theorem 15.2 The following statement is trivial: Given any two matrices X and Y of the corresponding dimensions, the two relations

$$(I + X Y) X = X (I + Y X), \quad X(I + Y X)^{-1} = (I + X Y)^{-1} X \quad (15.67)$$

hold, where I denotes the size-matching identity matrix.

Suppose that (15.63) holds. First, one needs to prove that all quantities in (15.62) are well defined and finite, which in turn implies that the matrix $A_v = s I_{n \times n} - A + B K + L_v C$ must be non-singular for any $s \in S$ and for all sufficiently small positive v

$$\forall s \in S \quad \exists v_0 > 0 \quad \forall 0 < v \leq v_0 : \quad \det(s I_{n \times n} - A + B K + L_v C) \neq 0 \quad (15.68)$$

Suppose the opposite, that is, for a given fixed $s \in S$, $\forall v > 0 \exists 0 < \mu(v) \leq v$ such that $A_{\mu(v)}$ is singular.

$$\begin{aligned} 0 &= \det(s I_{n \times n} - A + B K + L_{\mu(v)} C) \\ &= \det(I_{m \times m} + C(s I_{n \times n} - A + B K)^{-1} L_{\mu(v)}) \det(s I_{n \times n} - A + B K) \end{aligned} \quad (15.69)$$

Since $s \in S$ then $\det(s I_{n \times n} - A + B K) \neq 0$ and consequently

$$\det(\mu(v) I_{p \times p} + C(s I_{n \times n} - A + B K)^{-1} \mu(v) L_{\mu(v)}) = 0 \quad (15.70)$$

Taking the limit as $v \rightarrow 0$, while using (15.63) yields

$$\det(C(s I_{n \times n} - A + B K)^{-1} B) = 0 \quad (15.71)$$

and implies that s is the transmission zero for $(A - B K, B, C, 0_{m \times m})$. Since transmission zeros are invariant under state feedback, s is also the transmission zero for the original system $(A, B, C, 0_{m \times m})$, which gives a contradiction to the argument since $s \in S$ and proves (15.68).

Using the second equation in (15.67) and the notation from (15.59) with $K_{lqr} = K$ results in

$$\begin{aligned} G_v(s) &= K \left(\underbrace{(s I_{n \times n} - A + B K)}_{\Psi^{-1}(s)} + L_v C \right)^{-1} L_v C \underbrace{(s I_{n \times n} - A)^{-1}}_{\Phi(s)} \\ B &= K \left(\Psi^{-1}(s) + L_v C \right)^{-1} L_v C \Phi(s) B \\ &= K \left(I_{n \times n} + \underbrace{\Psi(s) L_v}_{X} \underbrace{C}_{Y} \right)^{-1} \underbrace{\Psi(s) L_v}_{X} C \Phi(s) \\ B &= K \underbrace{\Psi(s) L_v}_{X} \left(I_{m \times m} + \underbrace{C}_{Y} \underbrace{\Psi(s) L_v}_{X} \right)^{-1} C \Phi(s) B \end{aligned} \quad (15.72)$$

Therefore, for any $v > 0$

$$\begin{aligned} G_v(s) &= K \Psi(s) L_v (I_{m \times m} + C \Psi(s) L_v)^{-1} C \Phi(s) \\ B &= K \Psi(s) (v L_v) (v I_{m \times m} + C \Psi(s) (v L_v))^{-1} C \Phi(s) B \end{aligned} \quad (15.73)$$

and because of (15.63), the following limiting relation takes place:

$$\begin{aligned} \lim_{v \rightarrow 0} G_v(s) &= \lim_{v \rightarrow 0} (K \Psi(s) (v L_v) (v I_{m \times m} + C \Psi(s) (v L_v))^{-1} C \Phi(s) B) \\ &= K \Psi(s) \left[\lim_{v \rightarrow 0} (v L_v) \right] \left(C \Psi(s) \left[\lim_{v \rightarrow 0} (v L_v) \right] \right)^{-1} C \Phi(s) B \\ &= K \Psi(s) \bar{B} Q (C \Psi(s) \bar{B} Q)^{-1} C \Phi(s) B, \\ &= K \Psi(s) \bar{B} (C \Psi(s) \bar{B})^{-1} C \Phi(s) B \end{aligned} \quad (15.74)$$

Since $\Psi(s) = \Phi(s) (I_{n \times n} + B K \Phi(s))^{-1}$ then

$$\begin{aligned} \boxed{(K \Psi(s) \bar{B}) (\bar{C} \Psi(s) \bar{B})^{-1}} &= (I_{m \times m} + K \Phi(s) \bar{B})^{-1} (K \Phi(s) \bar{B}) \\ &\quad (\bar{C} \Phi(s) (I_{n \times n} + B K \Phi(s))^{-1} \bar{B})^{-1} \\ &= (I_{m \times m} + K \Phi(s) \bar{B})^{-1} (K \Phi(s) \bar{B}) \\ &\quad ((\bar{C} \Phi(s) \bar{B}) (I_{m \times m} + K \Phi(s) \bar{B})^{-1})^{-1} \\ &= (I_{m \times m} + K \Phi(s) \bar{B})^{-1} (K \Phi(s) \bar{B}) \\ &\quad (I_{m \times m} + K \Phi(s) \bar{B}) (\bar{C} \Phi(s) \bar{B})^{-1} \\ &= \boxed{(K \Phi(s) \bar{B}) (\bar{C} \Phi(s) \bar{B})^{-1}} \end{aligned} \quad (15.75)$$

Substituting (15.75) back into (15.74),

$$\begin{aligned} \lim_{v \rightarrow 0} G_v(s) &= (K \Phi(s) \bar{B}) (\bar{C} \Phi(s) \bar{B})^{-1} C \Phi(s) \\ \bar{B} \begin{pmatrix} I_{m \times m} \\ 0_{m \times m} \end{pmatrix} &= (K \Phi(s) \bar{B}) \begin{pmatrix} I_{m \times m} \\ 0_{m \times m} \end{pmatrix} = K \Phi(s) B = G_0(s) \end{aligned} \quad (15.76)$$

proves (15.64).

In order to prove the asymptotics (15.66), we rewrite (15.65) as

$$v L_v = \bar{B} Q + O(v), \quad v \rightarrow 0 \quad (15.77)$$

and substitute it into (15.73).

$$G_v(s) = K \Psi(s) (B Q + O(v)) (v I_{m \times m} + C \Psi(s) (B Q + O(v)))^{-1} C \Phi(s) B \quad (15.78)$$

Since,

$$v I_{m \times m} + C \Psi(s) (\bar{B} Q + O(v)) = C \Psi(s) \bar{B} Q + O(v) \quad (15.79)$$

then

$$(v I_{m \times m} + C \Psi(s) (\bar{B} Q + O(v)))^{-1} = (C \Psi(s) \bar{B} Q)^{-1} + O(v) \quad (15.80)$$

and consequently,

$$\begin{aligned} \boxed{G_v(s)} &= K \Psi(s) (\bar{B} Q + O(v)) \left((C \Psi(s) \bar{B} Q)^{-1} + O(v) \right) C \Phi(s) B \\ &= K \Psi(s) \bar{B} Q (C \Psi(s) \bar{B} Q)^{-1} C \Phi(s) B + O(v) \\ &= K \Psi(s) \bar{B} (C \Psi(s) \bar{B})^{-1} C \Phi(s) B + O(v) \\ &= (K \Phi(s) \bar{B}) (\bar{C} \Phi(s) \bar{B})^{-1} C \Phi(s) \bar{B} \begin{pmatrix} I_{m \times m} \\ 0_{m \times m} \end{pmatrix} + O(v) \\ &= K \Phi(s) B + O(v) = \boxed{G_0(s) + O(v)} \end{aligned} \quad (15.81)$$

which substantiates (15.66) and completes the proof of the theorem. ■

Theorem 15.3 Consider the extended system dynamics (15.36) with the OBLTR parameter-dependent controller (15.51)–(15.53). Let S denote any domain in \mathbb{C} defined such that the eigenvalues of A , $(A - B K_{lqr})$ and finite transmission zeros of $\bar{C}(s I_{n \times n} - A)^{-1} \bar{B}$ are not in S . Then as $v \rightarrow 0$, the v -parameter-dependent OBLTR loop gain $L_{g_{u,v}}(s)$ in (15.58) asymptotically recovers the LQR state feedback loop gain $L_{g_u}(s)$ in (15.43), and the corresponding limiting relation,

$$L_{g_u}(s) = \lim_{v \rightarrow 0} L_{g_{u,v}}(s) \quad (15.82)$$

is valid at the system control input breakpoint, point-wise for any fixed $s \in S$.

Proof of Theorem 15.3 Based on (15.48), asymptotics for the observer gain L_v in (15.49) can be derived

$$\begin{aligned} L_v &= P_v \bar{C}^T R_v^{-1} = \left(\frac{v+1}{v} \right) P_v \bar{C}^T R_0^{-1} = \left(\frac{v+1}{v} \right) \left(\bar{B} W^T R_0^{\frac{1}{2}} + O(v) \right) \\ R_0^{-1} &= \left(\frac{v+1}{v} \right) \bar{B} W^T R_0^{-\frac{1}{2}} + O(v) \end{aligned} \quad (15.83)$$

Consequently,

$$\lim_{v \rightarrow 0} (v L_v) = \bar{B} W^T R_0^{-\frac{1}{2}} \quad (15.84)$$

and the OBLTR loop gain (15.58) at the extended system input breakpoint (Fig. 15.6) is

$$\begin{aligned} Lg_{u,v}(s) &= K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} L_v \bar{C} (s I_{n \times n} - A)^{-1} \\ B &= K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} L_v \bar{C} \Phi(s) \bar{B} \begin{pmatrix} I_{m \times m} \\ 0_{m \times m} \end{pmatrix} \end{aligned} \quad (15.85)$$

where

$$A_{\text{obs}} = A - B K_{\text{lqr}} - L_v \bar{C} \quad (15.86)$$

Applying Theorem 4.1 with $K = K_{\text{lqr}}$, $C = \bar{C}$, $Q = W^T R_0^{-\frac{1}{2}}$ gives

$$\begin{aligned} \boxed{\lim_{v \rightarrow 0} Lg_{u,v}(s)} &= \lim_{v \rightarrow 0} (K_{\text{lqr}}(s I_{n \times n} - A_{\text{obs}})^{-1} L_v) \bar{C} \Phi(s) \bar{B} \begin{pmatrix} I_{m \times m} \\ 0_{m \times m} \end{pmatrix} \\ &= (K_{\text{lqr}} \Phi(s) \bar{B}) (\bar{C} \Phi(s) \bar{B})^{-1} \bar{C} \Phi(s) \bar{B} \begin{pmatrix} I_{m \times m} \\ 0_{m \times m} \end{pmatrix} \\ &= (K_{\text{lqr}} \Phi(s) \bar{B}) \begin{pmatrix} I_{m \times m} \\ 0_{m \times m} \end{pmatrix} = K_{\text{lqr}} \Phi(s) B = \boxed{Lg_u(s)} \end{aligned} \quad (15.87)$$

and proves (15.82). ■

Theorem 15.3 states sufficient conditions for LTR at the control input breakpoint, with the OBLTR controller in the loop. The sufficient conditions are enforced via the squaring-up method, as discussed in the Sect. 15.3. This is the robustness property of the baseline OBLTR controller we need in order to proceed to the design of an adaptive augmentation. But first, we will numerically demonstrate the vital steps and features of the derived nominal OBLTR design for square systems.

Example 15.3 LQR and OBLTR Design for the Square Non-Minimum Phase Aircraft Dynamics (15.6) For the extended open-loop system matrices (15.34), the LQR weights are selected as,

$$Q_{\text{lqr}} = \text{diag}(0.001 \ 0 \ 0 \ 1), \quad R_{\text{lqr}} = 1$$

with the three nonzero diagonal elements in Q_{lqr} . These are the cost weights assigned to the extended system state components (e_y, u_f, α, q). The unit value for the pitch rate weight is selected to enforce adequate pitch rate damping in the closed-loop design. These values can be optimized and further tuned in order to achieve the desired rise time (1–2 s) and a loop gain-crossover frequency ω_c (2–6 rad/s). The corresponding LQR state feedback solution,

$$K_{\text{lqr}} = (0.031623 \ 0 \ -6.6875 \ -3.5733) \quad (15.88)$$

has zero feedback gain on u_f . So, the filtered control component u_f will be used only as a measurement for the observer.

Also, in this example we add a control actuator with the second-order dynamics,

$$\frac{u}{u_{\text{cmd}}} = \frac{\omega_{\text{act}}^2}{s^2 + 2\xi_{\text{act}}\omega_{\text{act}}s + \omega_{\text{act}}^2}, \quad (15.89)$$

from the command u_{cmd} to the achieved value u , with the actuator natural frequency and damping ratio values set to $\omega_{\text{act}} = 20$ Hz and $\xi_{\text{act}} = 0.7$, correspondingly. The actuator dynamics are included into the analysis but not into the LQR design. In other words, the LQR gains are computed without explicitly accounting for the actuator dynamics, but the design stopping criteria (maintain adequate stability margins) is defined with the actuator dynamics accounted for.

Frequency characteristics of the resulting LQR state feedback loop gain at the original system input are shown in Fig. 15.7. The data exhibit adequate stability margins, even with the actuator second-order dynamics included into the analysis.

The corresponding closed-loop system step response (Fig. 15.8) shows command tracking performance of the LQR state feedback controller.

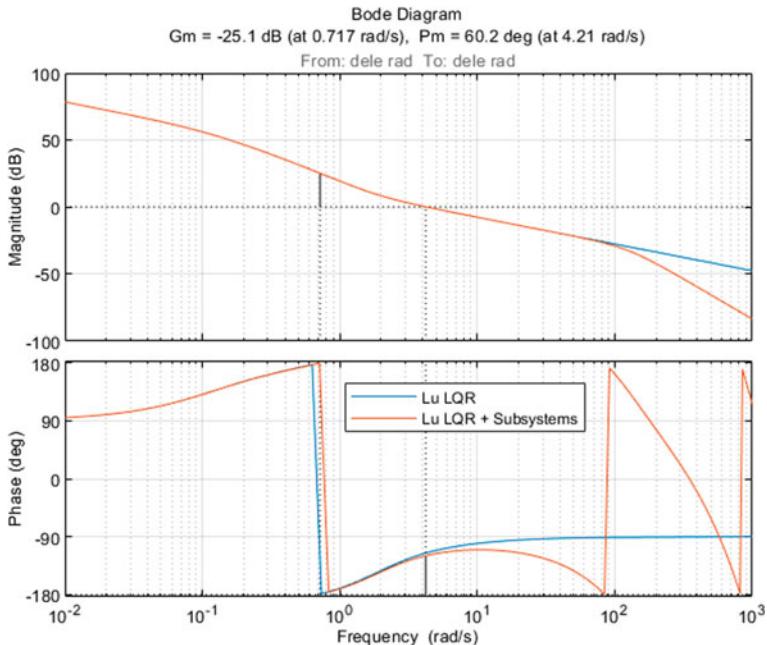


Fig. 15.7 LQR state feedback loop gain transfer function in Example 15.3 with actuator dynamics on and off

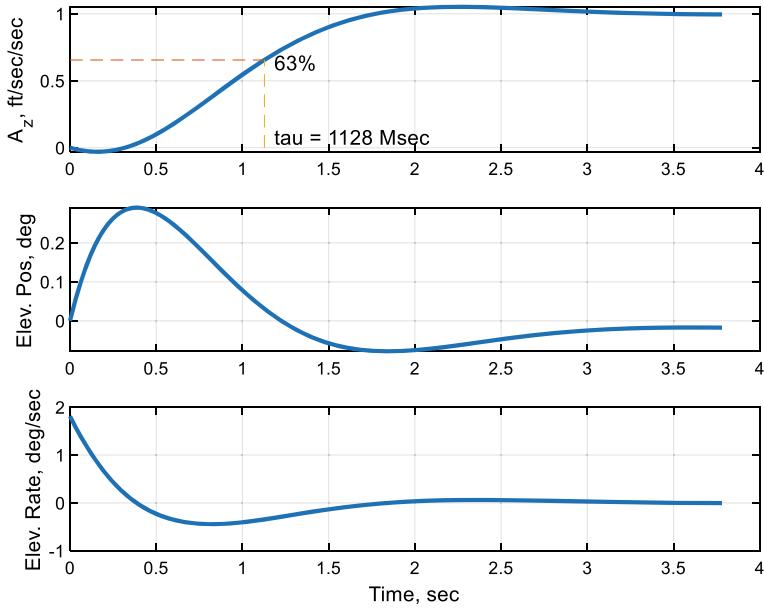


Fig. 15.8 Step-input response of the LQR state feedback controller in Example 15.3

The data are well behaved, as the system regulated output (the aircraft vertical acceleration A_z) asymptotically settles at the commanded 1 ft/s^2 step-input, with the rise time of 1.128 s.

The next step in the OBLTR design process is to construct an output feedback controller to: (a) recover the desired LQR state feedback stability margins at the system input and (b) maintain the achieved closed-loop system tracking performance.

We start with the squared-up extended open-loop dynamics (15.34) and define the observer weights according to (15.45) with

$$Q_0 = \text{diag}(1 \ 1 \ 40 \ 1), \quad R_0 = \text{diag}(1 \ 1),$$

and $v = 0.06$. Solving the ARE (15.46) gives the observer gain matrix (15.49).

$$L_v = \begin{pmatrix} 139.19 & 10.262 \\ 10.262 & 140.12 \\ -9.2 & -8.3916 \\ -11.815 & -22.413 \end{pmatrix}$$

OBLTR loop gain recovery at the system input breakpoint is shown in Fig. 15.9.

The LTR property at the system input is clearly visible. Per design, the OBLTR controller yields extra – 20 dB/decade roll-off when compared to the LQR state feedback loop gain.

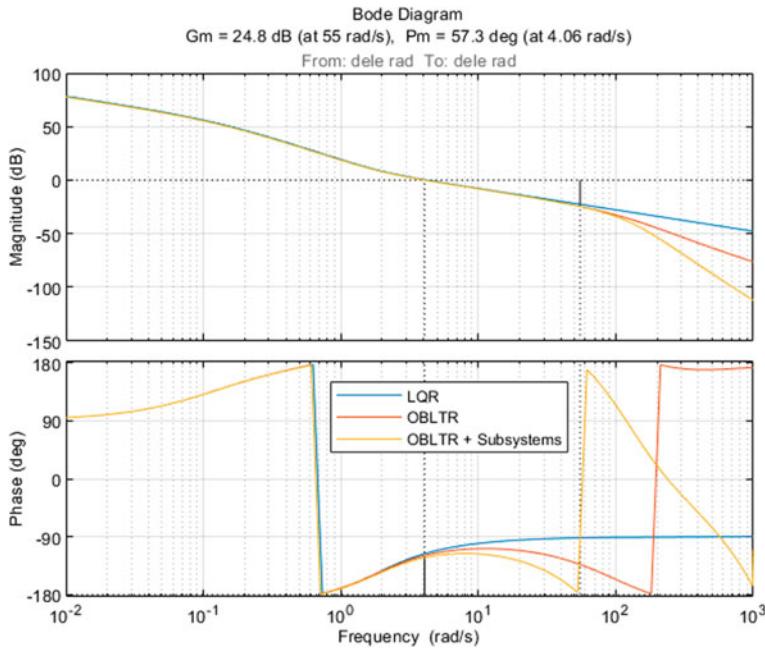


Fig. 15.9 OBLTR loop gain recovery at control input breakpoint in Example 15.3

It is also of interest to examine the asymptotic relation (15.48). The relevant matrices are shown below.

$$P_v^{-1} B = \begin{pmatrix} -0.61853 \\ 0.89458 \\ 0.90002 \\ 0.11495 \end{pmatrix}, \quad \bar{C}^T R_0^{-\frac{1}{2}} W = \begin{pmatrix} -0.74855 \\ 0.66308 \\ 0 \\ 0 \end{pmatrix}.$$

Choosing smaller v will force the difference between the two vectors go asymptotically to zero, whereas some of the corresponding observer eigenvalues would tend to infinity. This presents yet another tradeoff in the OBLTR design. Specifically, the loop gain recovery in Fig. 15.9 needs to be performed to exceed the LQR crossover frequency where the margins are defined, while keeping the observer eigenvalues within a reasonable range. In practice, preserving the inherent to the observer extra -20 dB/decade roll-off is beneficial, since the resulting controller would have better robustness to high frequency uncertainties in the plant, such as process disturbances and sensor noise.

In addition to the well-defined frequency domain characteristics, the OBLTR controller-driven closed-loop system tracking performance (0.5 g command) is shown in Fig. 15.10.

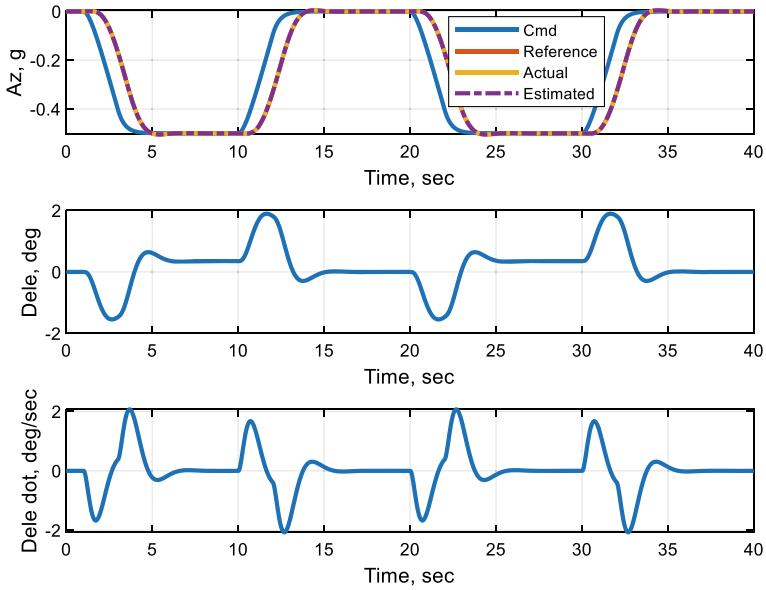


Fig. 15.10 Aircraft vertical acceleration tracking performance in Example 15.3

The system response is tested using a series of smoothed steps in the vertical acceleration command. As seen from the figure, the command tracking performance is well behaved, and it is achieved with the elevator control position and the rate that are within reasonable actuation limits.

15.6 OBLTR Adaptive Augmentation for Square Non-Minimum Phase Systems

When the integrated tracking error (15.12) and the filtered control input (15.13) are merged with the perturbed model dynamics (15.8), the resulting extended open-loop system

$$\dot{x} = A x + B \Lambda \left(u + \Theta^T \Phi(x) \right) + B_{\text{cmd}} y_{\text{cmd}} \quad (15.90)$$

can be viewed as the nominal extended open-loop dynamics (15.14) with matched uncertainties. The latter is comprised of the unknown constant diagonal control effectiveness matrix $\Lambda \in R^{m \times m}$, with strictly positive elements on the diagonal, and of the parametric state-dependent uncertainty $\Theta^T \Phi(x)$ acting directly in the control channel, with a constant unknown matrix of parameters $\Theta \in R^{N \times m}$ and a Lipschitz-continuous regressor vector $\Phi(x) \in R^{N \times m}$. The regressor is known. At

the same time, the system output measurements are

$$\bar{y}_{\text{meas}} = \begin{pmatrix} e_{yI} \\ u_f \end{pmatrix} = \begin{pmatrix} \frac{y_{\text{reg}} - y_{\text{cmd}}}{s} \\ \frac{s}{\tau_u s + 1} \end{pmatrix} = \bar{C} x \in R^{2m \times 1}, \quad (15.91)$$

as specified in (15.12), (15.13), and (15.15). As in the case of the baseline OBLTR controller, the outputs (15.91) are the only signals available for control synthesis, while the control task itself remains the same: track bounded commands y_{cmd} with bounded errors.

The challenge in this problem stems from the presence of the system matched uncertainties. This is mitigated and solved by an adaptive augmentation of the baseline OBLTR controller. The augmentation design follows the observer-based output feedback model reference adaptive control method from Chap. 14, where further details can be found, such as proofs of stability, analysis of transients, closed-loop tracking performance quantification, and utilization of adaptation logic with projection [30] to facilitate system robustness due to non-matched uncertainties. The applicability of this particular design approach is predicated on the squaring-up modification, Theorem 15.1. Specifically, it allows to convert the original square dynamics into a minimum phase square system of a larger dimension, with relative degree one. For completeness, the overall (OBLTR + Adaptive) control synthesis method for square controllable observable systems is summarized in Table 15.2.

The main feature of the (OBLTR + Adaptive) control design is the ability to seamlessly add an adaptive output feedback component to the robust baseline dynamic controller through the design of a Luenberger state observer. Essentially, tuning the state observer for loop transfer recovery of the baseline (LQR) state feedback margins, entails asymptotics (15.48) which in terms of adaptive control, allows construction of a stable adaptive law based on the available and estimated output signals only.

In this design, the state observer plays a dual role:

- (a) The observer generates state estimates for the baseline control feedback.
- (b) The observer acts as a closed-loop reference model for the adaptive controller [19–21], providing reference state and output estimates for the system to follow.

When the baseline robust OBLTR design is complete, an output feedback adaptive controller becomes also available as an optional element in the overall control block diagram shown in Fig. 15.11.

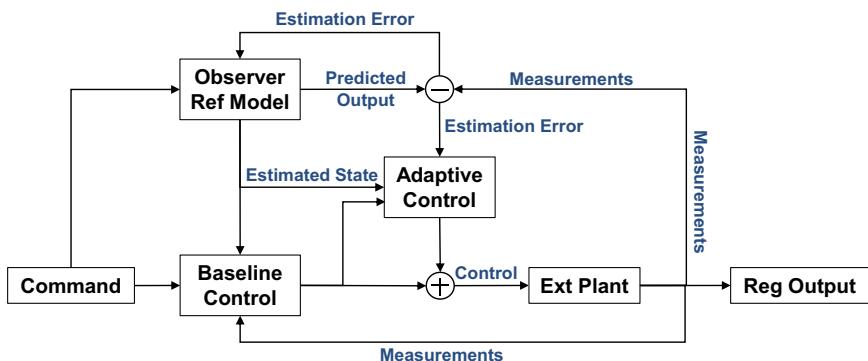
Table 15.2 OBLTR and adaptive output feedback control design summary from Chap. 14

MIMO square controllable observable plant	$\dot{x}_p = A_p x_p + B_p u, \quad y_{\text{meas}} = y_{\text{reg}} = C_{p \text{ reg}} x_p + D_{p \text{ reg}} u$
Integrated tracking error and filtered control	$\dot{e}_{yI} = y_{\text{reg}} - y_{\text{cmd}}, \quad \dot{u}_f = \frac{1}{\tau_u}(u - u_f)$
Extended tall system state and output measurements	$x = (e_{yI}^T, u_f^T, x_p^T)^T, \quad \bar{y}_{\text{meas}} = (e_{yI}^T, u_f^T)^T$
Extended tall system nominal dynamics data	$A = \begin{pmatrix} 0_{m \times m} & 0_{m \times m} & C_{p \text{ reg}} \\ 0_{m \times m} & -\frac{1}{\tau_u} I_{m \times m} & 0_{m \times n_p} \\ 0_{n_p \times m} & 0_{n_p \times m} & A_p \end{pmatrix}, \quad B = \begin{pmatrix} D_{p \text{ reg}} \\ \frac{1}{\tau_u} I_{m \times m} \\ B_p \end{pmatrix}, \quad B_{\text{cmd}} = \begin{pmatrix} -I_{m \times m} \\ 0_{m \times m} \\ 0_{n_p \times m} \end{pmatrix}$ $\bar{C}_{\text{meas}} = \begin{pmatrix} I_{m \times m} & 0_{m \times m} & 0_{m \times n_p} \\ 0_{m \times m} & I_{m \times m} & 0_{m \times n_p} \end{pmatrix}, \quad \bar{D}_{\text{meas}} = 0_{2m \times m}$
Open-loop extended plant with matched uncertainties	$\dot{x} = A x + B \Lambda(u + \Theta^T \Phi(x)) + B_{\text{cmd}} y_{\text{cmd}}$
Measured and regulated outputs	$\bar{y}_{\text{meas}} = \bar{C} x, \quad y_{\text{reg}} = C_{p \text{ reg}} x_p + D_{p \text{ reg}} u$
State observer	$\dot{\hat{x}} = A \hat{x} + B u_{\text{bl}} + B_{\text{cmd}} y_{\text{cmd}} + L_v (\bar{y}_{\text{meas}} - \hat{y}_{\text{meas}}), \quad \hat{y}_{\text{meas}} = \bar{C} \hat{x}$
Observer gain	$L_v = P_v \bar{C}^T R_v^{-1}$
B-matrix component selection for squaring-up and transmission zeros placement (Theorem 15.1)	$[B_{p2}: \lambda(A_p - B_{p2} C_{p \text{ reg}}) \in \mathbb{C}^-] \Rightarrow [B_2 = (I_{m \times m}, 0_{m \times m}, B_{p2}^T)^T]$
Squared-up B-matrix	$\bar{B} = (B \ B_2) \Rightarrow \begin{cases} \det(\bar{C} \bar{B}) \neq 0 \\ \text{zeros}[\bar{C} (s I_{n \times n} - A)^{-1} \bar{B}] \in \mathbb{C}^- \end{cases}$
ARE weights	$Q_v = Q_0 + \left(\frac{v+1}{v}\right) \bar{B} \bar{B}^T, \quad R_v = \frac{v}{v+1} R_0$
Algebraic Riccati Equation	$P_v A^T + A P_v - P_v \bar{C}^T R_v^{-1} \bar{C} P_v + Q_v = 0$
Output estimation error	$e_y = \hat{y}_{\text{meas}} - \bar{y}_{\text{meas}}$
Baseline control	$u_{\text{bl}} = -K_{\text{lqr}}^T \hat{x}$

(continued)

Table 15.2 (continued)

Extended regressor	$\bar{\Phi}(\hat{x}, u_{\text{bl}}) = \left(\Phi^T(\hat{x}) \ u_{\text{bl}}^T \right)^T$
Output selection matrix for adaptive laws	$S = \left(I_{m \times m} \ 0_{m \times m} \right)$
Singular value decomposition	$\bar{B}^T \bar{C}^T R_0^{-\frac{1}{2}} = U \Sigma V$
Unitary matrix	$W = (U \ V)^T$
Projection-based MRAC laws	$\dot{\hat{\Theta}} = \text{Proj}\left(\hat{\Theta}, -\Gamma_{\bar{\Theta}} \bar{\Phi}(\hat{x}, u_{\text{bl}}) e_y^T R_0^{-\frac{1}{2}} W S^T \right)$
Adaptive increment	$u_{\text{ad}} = -\hat{\Theta}^T \bar{\Phi}(\hat{x}, u_{\text{bl}})$
Total control input	$u = u_{\text{bl}} + u_{\text{ad}}$

**Fig. 15.11** Closed-loop block diagram of the OBLTR robust baseline controller with adaptive augmentation

Example 15.4 Adaptive Augmentation for the OBLTR Controller from Example 15.3 Continuing with the OBLTR baseline longitudinal flight controller in Example 15.3, a direct adaptive output feedback augmentation is added according to the equations in Table 15.2. Selected matched uncertainties include: (a) 50% reduction in control effectiveness ($\Lambda = 0.5$); (b) 70% reduction in the baseline feedback gains on α and q ; and (c) the nonlinear α -dependent Gaussian function,

$$f(\alpha) = \Theta^T \Phi(\alpha) = -0.04e^{-\frac{(\alpha-\alpha_c)^2}{2\sigma^2}}$$

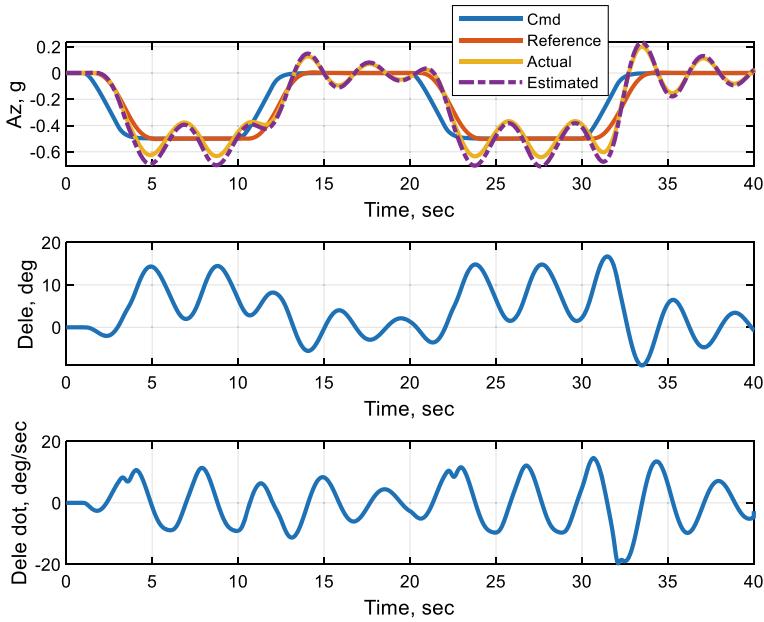


Fig. 15.12 Closed-loop system tracking performance in Example 15.4 with the baseline OBLTR controller turned on, adaptive augmentation off, actuators on, and with matched uncertainties

acting directly in the control channel, centered at $\alpha_c = 4 \frac{\pi}{180}$, with the width $\sigma = \frac{2}{3} \frac{\pi}{180}$. This uncertainty is introduced to emulate changes in the vehicle static stability, by way of modifying the slope of the aircraft aerodynamic pitching moment.

With these uncertainties turned on, the aircraft command tracking response is shown in Fig. 15.12.

The adaptive augmentation is turned off so that the closed-loop response can be compared to that of the system with the OBLTR baseline controller operating on the nominal linear dynamics. Tracking performance degradation due to the selected set of matched uncertainties is clearly visible. It is interesting to note that the state observer estimated vertical acceleration closely follows the vehicle actual response, yet both trajectories significantly differ from the commanded acceleration signal. Even though the tracking performance is not adequate, these data demonstrate robustness characteristics of the baseline OBLTR and its ability to maintain closed-loop stability in the presence of substantial changes in the system dynamics.

Figure 15.13 shows the radial basis functions that are selected for the OBLTR adaptive augmentation.

Identical separation distance and the overlap between the selected RBF centers yield uniform RBF total gain across the alpha interval of interest (the second subplot in Fig. 15.13). The RBF total gain at α is defined as $\sqrt{\sum_{i=1}^N \varphi_i^2(\alpha)}$, where $\varphi_i(\alpha)$ is the i th RBF. It turns out that the gain uniformity can be achieved by setting the RBF widths to be equal to the $4/3$ of the separation interval between the RBF centers.

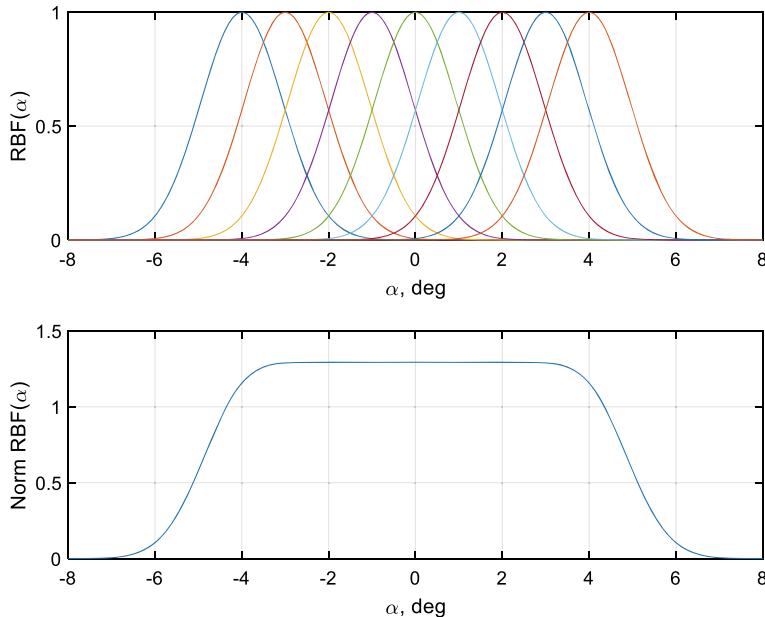


Fig. 15.13 Radial basis functions (RBFs) selected for the adaptive controller in Example 15.4

Simulation studies indicate that uniformity of the RBF gain significantly improves numerical features, such as transient smoothness in adaptive parameters in the overall design.

Rates of adaptation are shown in Table 15.3, next to their corresponding regressor components.

These values are selected to be proportional to the norm of the observer gain [19, 22] but no larger than the inverse of the simulation time step (0.01 s). Such a selection gives sufficiently fast adaptation dynamics with smooth transients, while avoiding instabilities due to numerical integration. Figure 15.14 shows the closed-loop simulation results with the adaptive OBLTR augmentation turned on.

We can see from the data that the nominal closed-loop dynamics are recovered without undesirable transients. This is a formally provable benefit of the OBLTR adaptive control design in Table 15.2. Transient analysis details can be found in [20, 21]. Elevator position and rate are well within the actuator capabilities. Baseline performance recovery of the original system state components is shown in Fig. 15.15.

Table 15.3 Rates of adaptation in Example 15.4

u_{bl}	$\hat{e}_y I$	\hat{u}_f	$\hat{\alpha}$	\hat{q}	$\varphi_i(\hat{\alpha})$
100	0	0	100	100	100

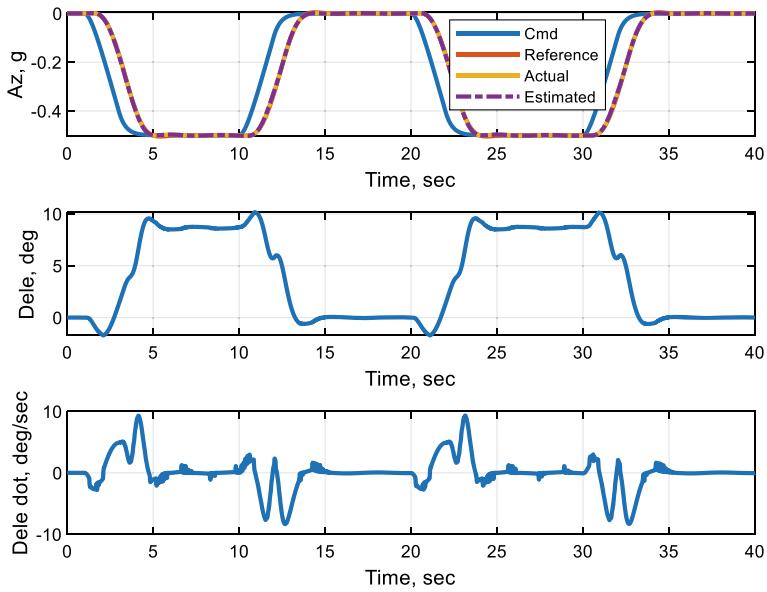


Fig. 15.14 Closed-loop system tracking performance in Example 15.4 with the baseline OBLTR controller turned on, adaptive augmentation on, with actuators and matched uncertainties

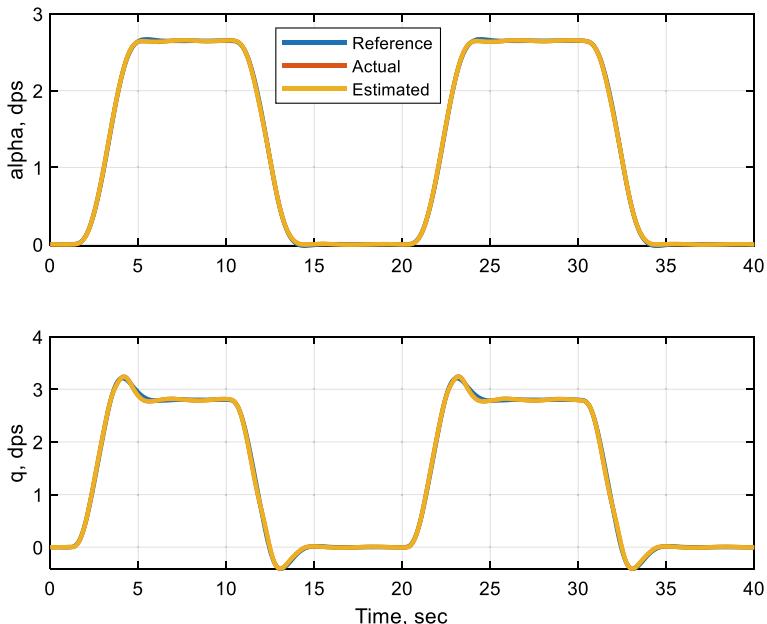


Fig. 15.15 System state recovery in Example 15.4 with the baseline OBLTR controller and adaptive augmentation turned on, in the presence of actuator dynamics and matched uncertainties

With the adaptive augmentation turned on, the state components in the closed-loop system are closely restored.

Norms of adaptive parameters are shown in Fig. 15.16.

The main point here to emphasize is the relative smoothness of the adaptive parameters, that is, the absence of high frequency transient oscillations during time periods of adaptation.

Figure 15.17 shows uncertainty estimation capabilities of the (OBLTR + Adaptive) controller

In this case and due to persistency of excitation (PE) induced by the external command into the system dynamics, the total matched uncertainty is estimated rather well. The uncertainty values are plotted versus time and versus the vehicle angle of attack, which is the dominating signal in this example. It is the well-known fact in adaptive control that parameter estimation requires PE, which is not necessarily guaranteed to take place. Nevertheless, provable stable closed-loop tracking performance is guaranteed, which is the main problem solved by the (OBLTR + Adaptive) dynamic output feedback controller. ■

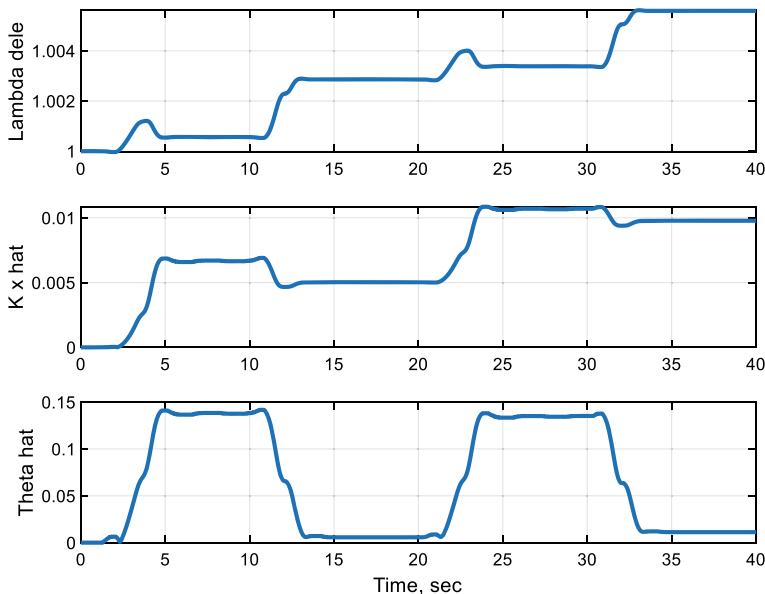


Fig. 15.16 Adaptive parameter norms in Example 15.4

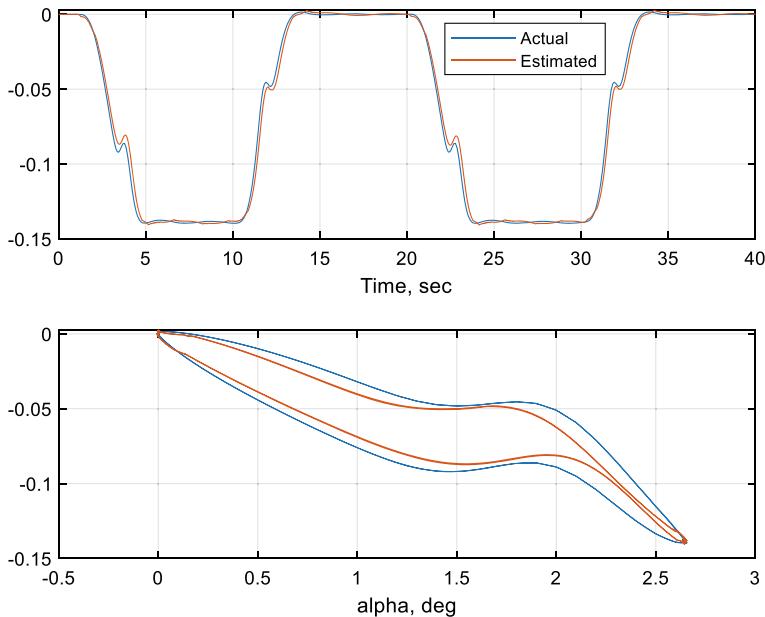


Fig. 15.17 Total matched uncertainty estimation in Example 15.4

15.7 Summary

The main question we addressed in this chapter is related to achieving closed-loop stability and tracking performance with adequate stability robustness margins via an output feedback observer-based control for linear time-invariant multi-input–multi-output systems whose open-loop dynamics are allowed to be unstable, non-minimum phase, and have arbitrary relative degree. We considered square systems with equal number of controls and sensed outputs. This is an extreme case since for tall dynamics (more output sensors than control inputs), the OBLTR method can be applied (Chap. 6) to design an output feedback robust servo-controller with stability margins approaching those of an optimal linear quadratic regulator in state feedback form. In this chapter, we extended the squaring-up and OBLTR methods to handle a large class of square non-minimum phase systems with arbitrary relative degree.

Using the above methods, we then constructed an adaptive output feedback augmentation of the baseline OBLTR controller. It was shown to recover the nominal closed-loop system performance in the presence of a wide class of matched state-dependent uncertainties. In that sense, the (OBLTR + Adaptive) control methodology presented in this chapter extends the state of the art in utilizing adaptive output feedback controllers for unstable non-minimum phase systems with arbitrary relative degree.

Throughout the chapter, we demonstrated main features and properties of the developed control design in a series of examples, all using the same aircraft longitudinal unstable open-loop dynamics with zero relative degree, with a single control input (elevator) and a single sensed regulated output (vertical acceleration). Controlling such a system using regulated output as the only sensed measurement available for control presents a formidable challenge in flight control applications. Even though the aircraft dynamics have zero relative degree, the vertical motion of the vehicle strongly depends on the vehicle angle of attack controlled through elevators, which in itself implies relative degree two, with respect to the system regulated output, such as vertical acceleration.

In general, output feedback control of unstable non-minimum phase systems with relative degree zero or greater than one is a challenging nontrivial task, and that is why we selected specific numerical examples in this chapter.

By now the reader should be able to recognize implications and practicality of the OBLTR control paradigm, with or without an adaptive augmentation. Throughout the textbook, we have shown that these output feedback control architectures are directly applicable to a large class of LTI MIMO controllable observable systems. The OBLTR design provides a baseline robust controller that allows to enforce design requirements within the model reference framework. The OBLTR controller contains a full-order state observer, which in turn plays a role of the closed-loop reference model for constructing a direct adaptive model reference augmentation. The LTR property of the OBLTR controller is the connection between robust and adaptive control designs, in the sense that the former enables the latter. And so, the control engineer now has the option of turning adaptive augmentation on and off, when needed and without a redesign of the baseline robust output feedback OBLTR controller.

15.8 Exercises

Exercise 15.1 Consider the aircraft short-period dynamics (15.6). Assume that the output measurements consist of the angle of attack, pitch rate, and vertical acceleration. That is, assume that states are available as output measurements. Use the (OBLTR + Adaptive) control method from Chap. 14 to design, analyze, and simulate the resulting controller. For simulation testing use the same inputs as in this chapter.

Exercise 15.2 Consider the aircraft short-period dynamics (15.6). Assume that the output measurements consist of the pitch rate and the vertical acceleration. Use the (OBLTR + Adaptive) control method from Chap. 14 to design, analyze, and simulate the resulting controller. For simulation testing use the same inputs as in this chapter.

Exercise 15.3 Consider the aircraft short-period dynamics (15.6) with the vertical acceleration as the only measurement output. Repeat all of the design and analysis steps from this chapter. Compare results with the data from Exercises 15.1 and 15.2.

References

1. Syrmos VL, Abdalla CT, Doratos JP, Grigoriadis K (1997) Static output feedback—a survey. *Automatica* 33(2):125–137. [https://doi.org/10.1016/S0005-1098\(96\)00141-0](https://doi.org/10.1016/S0005-1098(96)00141-0)
2. Luenberger DG (1964) Observing the state of a linear system. *IEEE Trans Mil Electron MIL-8*: 74–80. <https://doi.org/10.1109/TME.1964.4323124>
3. Maciejowski JM (1989) Multivariable feedback design. Addison-Wesley Publishers Ltd.<https://doi.org/10.1017/S0263574700007761>
4. Doyle JC, Francis BA, Tannenbaum AR (1992) Feedback control theory. Macmillan Publishing Company, New York, 10022
5. Anderson BDO, Moore JB (1980) Optimal control: linear quadratic methods. Dover, Mineola
6. Åström KJ, Murray RM (2008) Feedback control systems: an introduction for scientists and engineers. Princeton University Press, New Jersey, 08540
7. Doyle JC, Stein G (1981) Multivariable feedback design: concepts for a classical/modern synthesis. *IEEE Trans Automat Contr* 26(1):4–16. <https://doi.org/10.1109/TAC.1981.1102555>
8. Stein G, Athans M (1987) The LQG/LTR procedure for multivariable feedback control design. *IEEE Trans Automat Contr* 32(2):105–114. <https://doi.org/10.1109/TAC.1987.1104550>
9. Kwakernaak H, Sivan R (1972) Linear optimal control systems. New York: Wiley-Interscience
10. Narendra KS, Annaswamy AM (2005) Stable adaptive control. Dover, New York
11. Ioannou P, Fidan P (2006) Adaptive control tutorial, SIAM, advances in design and control. SIAM, PA. <https://doi.org/10.1137/1.9780898718652>
12. Krstic M, Kanellakopoulos I, Kokotovic P (1995) Nonlinear and adaptive control design. Wiley, New York
13. Khalil HK (1996) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, NJ 07458
14. Seshagiri S, Khalil HK (2000) Output feedback control of nonlinear systems using RBF neural networks. *IEEE Trans Neural Netw* 11(1):69–79. <https://doi.org/10.1109/72.822511>
15. Kuipers M, Ioannou P (2010) Multiple model adaptive control with mixing. *IEEE Trans Automat Contr* 55(8):1822–1836. <https://doi.org/10.1109/TAC.2010.2042345>
16. Zang Z, Bitmead R (1994) Transient bounds for adaptive control systems. *IEEE Trans Automat Contr* 39(1):171–178. <https://doi.org/10.1109/CDC.1990.203273>
17. Lavretsky E. Reference dynamics modification in adaptive controllers for improved transient performance. AIAA Paper 2011-6200. <https://doi.org/10.2514/6.2011-6200>
18. Lavretsky E (2012) Adaptive output feedback design using asymptotic properties of LQG/LTR controllers. *IEEE Trans Automat Contr* 57(6):1587–1591. <https://doi.org/10.1109/TAC.2011.2174692>
19. Gibson TE, Annaswamy AM, Lavretsky E. Improved transient response in adaptive control using projection algorithms and closed loop reference models. AIAA Paper 2012-4775. <https://doi.org/10.2514/6.2012-4775>
20. Gibson TE, Annaswamy AM, Lavretsky E (2013) Closed-loop reference model adaptive control, part I: transient performance. In: The proceedings of the American control conference. <https://doi.org/10.1109/ACC.2013.6580353>
21. Gibson TE, Annaswamy AM, Lavretsky E (2013) Closed-loop reference model adaptive control: composite control and observer feedback. In: The proceedings of the 11th IFAC international workshop on adaptation and learning in control and signal processing
22. Lavretsky E (2015) Transients in output feedback adaptive systems with observer-like reference models. *Int J Adapt Control Signal Process* 29:1515–1125. <https://doi.org/10.1002/acs.2557>
23. Gibson TE, Qu Z, Annaswamy AM, Lavretsky E (2015) Adaptive output feedback based on closed-loop reference models. *IEEE Trans Automat Contr* 60(10):2728–2733. <https://doi.org/10.1109/TAC.2015.2405295>
24. Misra P (1988) Numerical algorithms for squaring-up non-square systems, part II: general case. In: The proceedings of the American control conference, San Francisco, CA

25. Kevorkian J, Cole JD (1996) Multiple scale and singular perturbation methods. *Appl Math Sci* 114. <https://doi.org/10.1007/978-1-4612-3968-0>
26. Balas G, Young P (2003) Sensor selection via closed-loop control objectives. *IEEE Trans Automat Contr* 7(6):692–704. <https://doi.org/10.1109/TAC.2003.877996>
27. Wal M, Jager B (2001) A review of methods for input/output selection. *Automatica* 37:487–510
28. Stein G (2003) Respect the unstable. *IEEE Control Syst Mag* 4:12–25. <https://doi.org/10.1109/MCS.2003.1213600>
29. Etkin B (1982) Dynamics of flight. Stability and control, 2nd edn. Wiley. <https://doi.org/10.1063/1.3060977>
30. Lavretsky E, Gibson TE (2011) Projection operator in adaptive systems. [arXiv:1112.4232](https://arxiv.org/abs/1112.4232)

Appendix A: Aircraft Flight Simulation (aFltSim) Software

In this appendix, we describe the MATLAB®/Simulink® [1] software environment for high-fidelity flight simulation of a fixed-wing aircraft dynamics and control. The aircraft model is generic and representative of a medium-size turbo-jet aerial vehicle. The aircraft data originate from [2]. Aircraft dynamics equations are summarized here, and their software implementations are discussed. This material is supplemental to Chap. 1 and is used through the book to generate and simulate models in selected numerical examples. In order to make the appendix material self-contained, aircraft specific definitions and concepts from Chap. 1 are reused and summarized.

A.1 Aircraft Flight Dynamics Equations of Motion

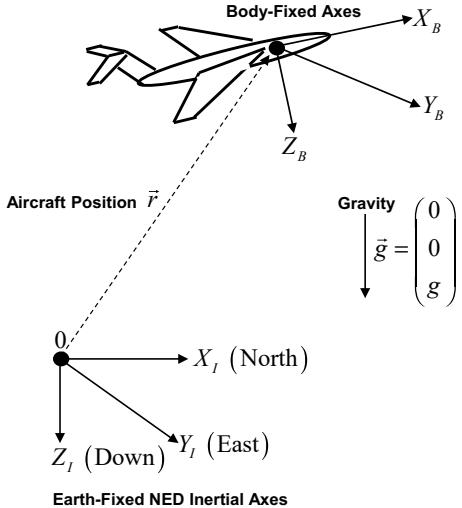
The aircraft dynamics are defined with respect to the vehicle body-fixed coordinate frame of reference (the body axes) which in turn undergoes translational and rotational displacements relative to the Earth-fixed inertial reference frame (the Earth axes) or equivalently, the North–East–Down (NED) frame of reference.

The $(X_I - Y_I)$ plane is parallel to the Earth's surface, while Z_I points down toward the Earth's center and parallel to the gravity vector \vec{g} . The NED origin 0 is fixed at an arbitrary location on the Earth's surface.

The origin of the body axis system is fixed at the aircraft center of gravity (CG), X_B points forward along the vehicle fuselage line, Y_B is positive out the right wing, and Z_B direction is positive downward out the bottom of the vehicle.

The gravity is assumed to be uniformly constant, which in turn implies the Flat-Earth assumption for simulation purposes and for modeling atmospheric flight dynamics of an aircraft on a short-to-medium time interval.

Fig. A.1 Earth-fixed and body-fixed reference frames



Rigid Aircraft Equations of Motion

Aircraft flight dynamics are derived based on the classical Newton–Euler formulations for the rigid body equations of motion, written with respect to the Earth-fixed inertial coordinate frame of reference (Fig. A.1) and expressed in the vehicle body-fixed coordinate system. In this context, the aircraft is viewed as a single rigid body moving through the three-dimensional Cartesian aerospace. The Newton–Euler equations describe the vehicle translational and rotational components of the overall motion in the Earth-fixed inertial coordinates.

$$\frac{d}{dt} \left(m \vec{V} \right)_I = \left(\vec{F} + m \vec{g} \right)_I, \quad \frac{d}{dt} (J \vec{\omega})_I = \left(\vec{M} \right)_I \quad (\text{A.1})$$

These relations define time-dependent changes in the aircraft momentum $\left(m \vec{V} \right)_I$ and its moment of momentum $(J \vec{\omega})_I$, subject to the total external force $\left(\vec{F} + m \vec{g} \right)_I$ and the total moment $\left(\vec{M} \right)_I$ acting on the vehicle. The index “I” implies that the time derivative operator $\frac{d}{dt}$ is applied to the total aircraft velocity vector \vec{V} and to the vehicle angular velocity vector $\vec{\omega}$ in the inertial frame of reference. The gravity vector is \vec{g} , the aircraft mass is m , and the inertia tensor (3×3) matrix is J .

We assume that \vec{g} , m and J are constant. Rewriting (A.1) in the aircraft body-fixed coordinates with the frame origin attached at the vehicle CG results in the

vehicle six-degrees-of-freedom (6-DoF) system of ordinary differential equations.

$$\begin{aligned}
 \text{Translational DOF: } m \underbrace{\begin{pmatrix} \dot{u}_b \\ \dot{v}_b \\ \dot{w}_b \end{pmatrix}}_{\dot{\vec{V}}} &= -m \left[\underbrace{\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix}}_{\vec{\omega}} \times \underbrace{\begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix}}_{\vec{V}} \right] \\
 &\quad + \underbrace{\begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix}}_{\vec{F} = \vec{F}_a + \vec{F}_T} + m \underbrace{\|\vec{g}\|}_{\vec{g}} \underbrace{\begin{pmatrix} -\sin \theta \\ \cos \theta \sin \varphi \\ \cos \theta \cos \varphi \end{pmatrix}}_{\vec{g}} \\
 \text{Rotational DOF: } J \underbrace{\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix}}_{\dot{\vec{\omega}}} &= - \left[\underbrace{\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix}}_{\vec{\omega}} \times J \underbrace{\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix}}_{\vec{\omega}} \right] + \underbrace{\begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix}}_{\vec{M}}
 \end{aligned} \tag{A.2}$$

The aircraft translational dynamics prescribe time-dependent changes of the velocity vector $\vec{V} = (u_b \ v_b \ w_b)^T$ with three body velocity components that represent the vehicle forward, lateral, and vertical speeds along the body-fixed axes, respectively. According to Newton's second law of motion, the vehicle translational dynamics are driven by the sum of all forces that are acting on the body externally. Those include the sum of aerodynamic and propulsive forces $\vec{F} = \vec{F}_a + \vec{F}_T$, as well as the gravitational force $m \vec{g}$, whereby the aircraft mass m and the gravity vector \vec{g} are assumed to be constant.

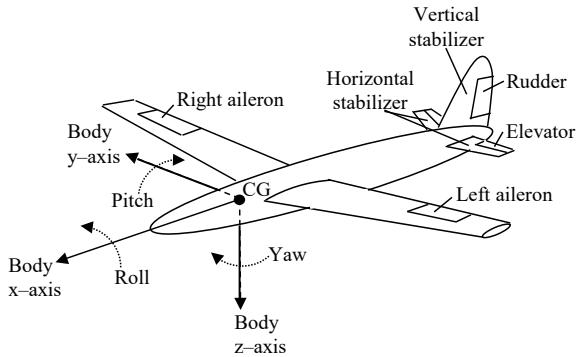
The aircraft rotational dynamics define evolution of the vehicle angular velocity vector $\vec{\omega} = (p_b \ q_b \ r_b)^T$ whose components are comprised of the body axis roll, pitch, and yaw angular rates, in that order. The system is driven by the total moment vector \vec{M} . It is also assumed that the aircraft inertia tensor matrix $J \in R^{3 \times 3}$ is constant.

Figure A.2 shows a sketch of a generic aircraft with its body-fixed coordinate frame of reference and conventional control surfaces such as ailerons, elevators, and rudder to control roll, pitch, and yaw angular displacements.

The aircraft 6-DoF equations of motion (A.2) define the vehicle dynamics with respect to the body-fixed reference frame as shown in Fig. A.2. This system depends on the gravity vector.

$$\vec{g} = g \begin{pmatrix} -\sin \theta \\ \cos \theta \sin \varphi \\ \cos \theta \cos \varphi \end{pmatrix} \tag{A.3}$$

Fig. A.2 Aircraft body-fixed coordinate frame



The three components of \vec{g} are written using the Euler bank, pitch, and yaw angles φ , θ , ψ . The triplet defines angular orientation of the aircraft body-fixed frame relative to the Earth-fixed inertial NED coordinate system. Euler angles transform NED frame into the body axis system via the three consecutive rotations about the Z_I , Y_I , and X_I axes, in that order. The rotation sequence is yaw–pitch–roll. It defines the three Euler angles ($\psi - \theta - \varphi$) and must be maintained for consistency of the definition (Fig. A.3).

Euler angles can be used to transform the aircraft velocity vector between the body-fixed axes and the NED inertial system. Based on the definition of the Euler rotations (Fig. A.3), velocity transformation from NED to the vehicle body axes is written first and according to the predetermined rotation sequence: $\psi \Rightarrow \theta \Rightarrow \varphi$.

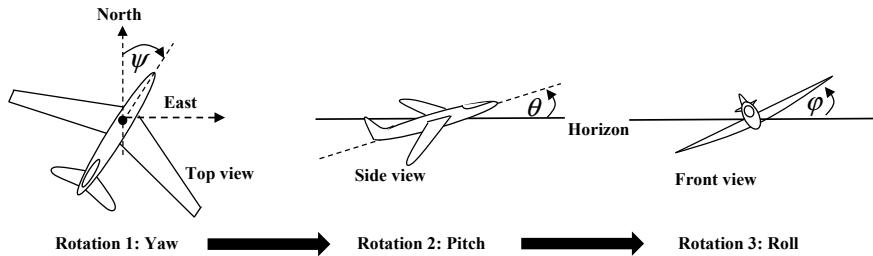


Fig. A.3 Rotation sequence of Euler angles: yaw → pitch → roll

$$\begin{aligned}
\begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \\
&= \underbrace{\begin{pmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi & \sin \varphi \cos \theta \\ \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi & \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi & \cos \varphi \cos \theta \end{pmatrix}}_{T_{I \Rightarrow B}} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \\
&= T_{I \Rightarrow B} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \tag{A.4}
\end{aligned}$$

In (A.4), $(\dot{x}, \dot{y}, \dot{z})$ are the north, east, and down velocity components, respectively, and $T_{I \Rightarrow B}$ is the total transformation matrix from body-fixed to inertial axes.

Inverting (A.4) yields transformation of the aircraft velocity vector from the aircraft body-fixed frame into the inertial NED system.

$$\begin{aligned}
\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} &= \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} \\
&= \underbrace{\begin{pmatrix} \cos \theta \cos \psi & \sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi \\ \cos \theta \sin \psi & \sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi & \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi \\ -\sin \theta & \sin \varphi \cos \theta & \cos \varphi \cos \theta \end{pmatrix}}_{T_{B \Rightarrow I}} \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} \\
&= T_{B \Rightarrow I} \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} \tag{A.5}
\end{aligned}$$

These are the vehicle navigation equations, and $T_{B \Rightarrow I} = T_{I \Rightarrow B}^{-1}$ is the transformation matrix from NED reference frame to the aircraft body axis system. Aircraft height above the ground (geometric altitude) is

$$h = -z \tag{A.6}$$

With (A.6), the navigation equations (A.5) can be rewritten.

$$\begin{aligned}
\dot{x} &= u_b \cos \theta \cos \psi + v_b (\sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi) \\
&\quad + w_b (\cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi) \\
\dot{y} &= u_b \cos \theta \sin \psi + v_b (\sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi) \\
&\quad + w_b (\cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi) \\
\dot{h} &= u_b \sin \theta - v_b \sin \varphi \cos \theta - w_b \cos \varphi \cos \theta \tag{A.7}
\end{aligned}$$

Rotational kinematics of Euler angles are defined below.

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \quad (\text{A.8})$$

Euler rotational kinematics equations (A.8) can be written explicitly.

$$\begin{aligned} \dot{\varphi} &= p_b + \tan \theta (q_b \sin \varphi + r_b \cos \varphi) \\ \dot{\theta} &= q_b \cos \varphi - r_b \sin \varphi \\ \dot{\psi} &= \frac{1}{\cos \theta} (q_b \sin \varphi + r_b \cos \varphi) \end{aligned} \quad (\text{A.9})$$

We note that the system (A.9) is not defined at $\theta = \pm 90^\circ$. This puts a restriction on modeling aircraft pitch angle dynamics to within 90° bounds. In that respect, the three-variable Euler orientation expression (A.9) is not directly applicable for simulating long haul flights around the Earth. Also, Euler rotational kinematics do not support simulations of high agility aircraft aerobatic maneuvers. Extensions are possible via quaternions [3] but that is not consequential within the book contents.

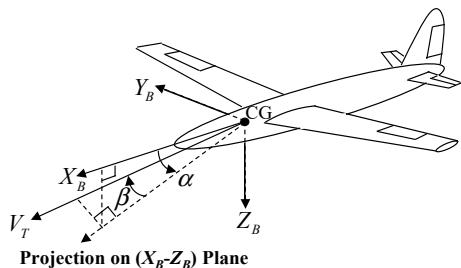
Aerodynamic Forces and Moments

The aircraft 6-DoF equations of motion (A.2) are driven by external forces $\vec{F} \in R^3$ and moments $\vec{M} \in R^3$. These quantities define the vehicle motion dependence on aerodynamics, propulsion, as well as on the gravity force $m \vec{g} \in R^3$ whose magnitude defines the aircraft total gross weight $W = \|m \vec{g}\| = m g$. Other models and subsystems can be added as needed to better represent the vehicle dynamics.

There are three core aerodynamic parameters that define aircraft motion in the body-fixed axes with respect to atmosphere. These airmass-related parameters are: true airspeed V_T , angle of attack (AOA) α , and angle of sideslip (AOS) β (Fig. A.4).

$$V_T = \sqrt{u_b^2 + v_b^2 + w_b^2}, \quad \alpha = \arctan\left(\frac{w_b}{u_b}\right), \quad \beta = \arcsin\left(\frac{v_b}{V_T}\right) \quad (\text{A.10})$$

Fig. A.4 True airspeed, angle of attack, and angle of sideslip



Inverting (A.10) gives the aircraft body axis velocity components in terms of the airmass parameters.

$$u_b = V_T \cos \alpha \cos \beta, \quad v_b = V_T \sin \beta, \quad w_b = V_T \sin \alpha \cos \beta \quad (\text{A.11})$$

Equations (A.11) are often used to compute the aircraft velocity vector based on the three airmass signals that in turn can be measured and/or estimated during flight.

Another two very important quantities that govern the aircraft motion are the vehicle dynamic pressure \bar{Q} and Mach number M .

$$\bar{Q} = \frac{\rho(h)V_T^2}{2}, \quad M = \frac{a(h)}{V_T} \quad (\text{A.12})$$

Their calculations are carried out based on the standard day air density $\rho(h)$ and the speed of sound $a(h)$. These two functions depend on the vehicle altitude h . They represent variations in the static atmospheric model over a range of geopotential heights referenced to Earth's mean sea level. Under the assumed Flat-Earth conditions, values for geopotential and geometric altitude h are equal to each other.

Excluding gravitational force ($m \vec{g}$), the external forces \vec{F} , and moments \vec{M} in the 6-DoF model (A.2) represent the sum of the aircraft aerodynamic and propulsive effects. As such, they depend on the airmass parameters (A.10), dynamic pressure and Mach number (A.12), altitude h , angular rates (p_b, q_b, r_b), and on the vehicle control input vector $\vec{\delta}$ often composed of ailerons, elevators, rudders, and thrust.

$$\vec{F}, \vec{M} = \vec{F}, \vec{M}(h, M, \bar{Q}, V_T, \alpha, \beta, p_b, q_b, r_b, \vec{\delta}) \quad (\text{A.13})$$

Dependencies in (A.13) can be modified to account for unsteady aerodynamic effects due to translational and rotational accelerations.

The aerodynamic forces and moments can be written in terms of non-dimensional coefficients that are defined in the vehicle body axes.

$$\vec{F}_a = \bar{Q} S_w \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix}, \quad \vec{M}_a = \bar{Q} S_w \begin{pmatrix} \bar{b} & 0 & 0 \\ 0 & \bar{c} & 0 \\ 0 & 0 & \bar{b} \end{pmatrix} \begin{pmatrix} C_l \\ C_m \\ C_n \end{pmatrix} \quad (\text{A.14})$$

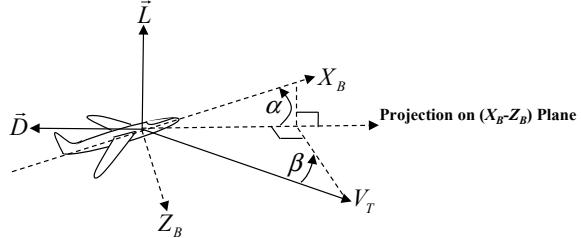
The factorization (A.14) depends on the wing reference data represented by wing area S_w , wing chord \bar{c} , and wing span \bar{b} . Definitions of the six non-dimensional aerodynamic coefficients are listed in Table A.1.

Coefficient primary dependencies are well-known from practice [1].

$$C_i = C_i(\alpha, \beta, M, \vec{\delta}), \quad i = x, y, z, l, m, n \quad (\text{A.15})$$

Table A.1 Six aerodynamic coefficients

C_x	Axial force coefficient
C_y	Lateral force coefficient
C_z	Vertical force coefficient
C_l	Rolling moment coefficient
C_m	Pitching moment coefficient

Fig. A.5 Aerodynamic lift and drag forces

The aerodynamic force \vec{F}_a is often decomposed into the aircraft lift \vec{L} and drag \vec{D} forces (Fig. A.5).

The lift and drag force decomposition is defined in the body axis $X_B - Z_B$. In other words, \vec{F}_a is projected onto $X_B - Z_B$ vertical plane and then decomposed. By definition, the lift force \vec{L} is perpendicular to true airspeed V_T and resides in the $X_B - Z_B$ vertical plane. The drag force component \vec{D} is in the same plane but parallel and opposite to the projection of the airspeed vector onto the vertical plane $X_B - Z_B$. Similar to (A.14), magnitudes for the aerodynamic lift and drag forces are written in terms of non-dimensional coefficients.

$$L = \overline{Q} S_w C_L, \quad D = \overline{Q} S_w C_D \quad (\text{A.16})$$

Based on their definition, we can calculate lift and drag coefficients from axial and vertical terms rotated with α .

$$\begin{aligned} C_L &= -C_Z \cos \alpha + C_x \sin \alpha \\ C_D &= -C_z \sin \alpha - C_x \cos \alpha \end{aligned} \quad (\text{A.17})$$

Non-dimensional aerodynamic coefficient data (C_L , C_D , C_M , C_l , C_m , C_n) are vehicle-specific although generic trends in their components can often be observed. The resulting (aerodynamic + propulsion) database provides foundations for aircraft high-fidelity flight simulation development.

A.2 High-Fidelity Flight Simulation Environment

Setting up a sufficiently precise flight simulation software architecture is one of the key steps in a model-based aircraft development framework. Modeling system dynamics with a set of ordinary differential equations (ODEs) starts with

the definition of the model states (degrees-of-freedom), external inputs (controls, disturbances), and outputs (measurements and other calculated online signals).

The system dynamics and output are

$$\dot{X} = F(X, Y, U), \quad Y = H(X, U) \quad (\text{A.18})$$

External inputs U may include actual controls that can be manipulated as well as disturbances that may be known or unknown. States X are internal signals. They define degrees-of-freedom for the system dynamics. The number of states is the same as the number of ODEs selected to model the system evolution in time. The outputs Y may include states and other signals that are computable from states and inputs. Some of the output components are fed back into the system while others are saved for further analysis. The triplet (X, Y, U) contains all signals that are required to compute and propagate the system state dynamics one step ahead in time. The functions $F(X, Y, U)$ and $H(X, U)$ can be defined analytically or they might be represented by interpolation of pre-existing database via lookup tables. Their definitions must ensure existence and uniqueness of the system trajectory solutions that arise from the set of ODEs selected to model the process dynamics. In other words, starting from an initial condition X_0 , the system (A.18) must have the unique solution forward in time.

An algorithm to compute the system time-dependent evolution is implemented numerically, and therefore the analytical integrator $\frac{1}{s}$ must be replaced by its discrete equivalent. For example, the forward Euler integration method can be used to implement iterations for the simulation process.

$$\begin{aligned} & \text{One-Step-Ahead Propagation: } X(k+1) = X(k) + \Delta t F(X(k), Y(k), U(k)), \\ & k = 0, 1, \dots, N \\ & \text{System Output: } Y(k) = H(X(k), U(k)) \\ & \text{State Initialization: } X(0) = X_0 \end{aligned} \quad (\text{A.19})$$

As defined in (A.19), the system state $X(k)$ at time t_k is propagated one step ahead to compute the next state value $X(k+1)$ at time $t_{k+1} = t_k + \Delta t$, with all calculations carried out based on the current state, output, and input values $(X(k), Y(k), U(k))$. Starting from an initial state X_0 , iterations (A.19) continue over a predetermined finite time interval $[t_0, t_N]$.

Based on (A.18), we combine the aircraft 6-DoF dynamics (A.2), the Euler kinematics (A.8), and the navigation equations (A.5) into a single twelve-degrees-of-freedom dynamical system.

$$\begin{aligned} \begin{pmatrix} \dot{u}_b \\ \dot{v}_b \\ \dot{w}_b \end{pmatrix} &= - \left[\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \times \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} \right] + \frac{1}{m} \left(\bar{Q} S_w \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \right) + g \begin{pmatrix} -\sin \theta \\ \cos \theta \sin \varphi \\ \cos \theta \cos \varphi \end{pmatrix} \\ \begin{pmatrix} \dot{p}_b \\ \dot{q}_b \\ \dot{r}_b \end{pmatrix} &= -J^{-1} \left[\begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \times J \begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \right] + J^{-1} \bar{Q} S_w \begin{pmatrix} \bar{b} C_l \\ \bar{c} C_m \\ \bar{b} C_n \end{pmatrix} + J^{-1} \begin{pmatrix} T_l \\ T_m \\ T_n \end{pmatrix} \end{aligned}$$

$$\begin{aligned} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} &= \begin{pmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \begin{pmatrix} p_b \\ q_b \\ r_b \end{pmatrix} \\ \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} &= \begin{pmatrix} \cos \theta \cos \psi & \sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi \\ \cos \theta \sin \psi & \sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi & \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi \\ -\sin \theta & \sin \varphi \cos \theta & \cos \varphi \cos \theta \end{pmatrix} \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} \end{aligned} \quad (\text{A.20})$$

The system state has twelve components.

$$X = (u_b \ v_b \ w_b \ p_b \ q_b \ r_b \ \varphi \ \theta \ \psi \ x \ y \ z)^T \quad (\text{A.21})$$

The aircraft control inputs are denoted by $\vec{\delta}$. For a conventionally configured aircraft the vehicle control inputs are aileron δ_a , elevator δ_e , rudder δ_r , and thrust δ_T .

$$\vec{\delta} = (\delta_a \ \delta_e \ \delta_r \ \delta_T)^T \quad (\text{A.22})$$

The thrust input δ_T may represent a position or an angle of the vehicle thrust (power) lever mechanical device. It controls the thrust output of the aircraft engines. For pure simulation purposes δ_T can also be defined as a percent of the thrust commanded.

Since aircraft forces and moments depend on altitude, Mach number, dynamic pressure, and three aero parameters (A.13), we need to add all of those as the model outputs. Calculations are conducted based on previously derived relations (A.10), (A.12) that are summarized below.

$$\begin{aligned} V_T &= \sqrt{u_b^2 + v_b^2 + w_b^2}, \quad \alpha = \arctan\left(\frac{w_b}{u_b}\right), \quad \beta = \arcsin\left(\frac{v_b}{V_T}\right) \\ h &= -z_I, \quad \overline{Q} = \frac{\rho(h)V_T^2}{2}, \quad M = \frac{a(h)}{V_T} \end{aligned} \quad (\text{A.23})$$

In (A.23), aircraft geometric altitude h is defined as the opposite to the system vertical displacement z_I in the NED reference frame. Then altitude-dependent values for the air density $\rho(h)$ and the speed of sound $a(h)$ can be computed online using standard atmospheric tables.

In addition to (A.23), we add three translational acceleration signals (A_x , A_y , A_z) to the simulation output vector.

$$A_x = \frac{F_x}{m g}, \quad A_y = \frac{F_y}{m g}, \quad A_z = \frac{F_z}{m g} \quad (\text{A.24})$$

We assume that the three accelerometers are located at the vehicle CG. The simulation output vector is defined in (A.25).

$$Y = (X^T \ h \ M \ \overline{Q} \ V_T \ \alpha \ \beta \ A_x \ A_y \ A_z)^T \quad (\text{A.25})$$

Other output signals can be added to (A.25) to support simulation, as well as control design, analysis, and performance evaluations. For example, time derivatives of the aero parameters $(\dot{V}_T, \dot{\alpha}, \dot{\beta})$ from (A.10) are often saved as outputs.

In flight, an aircraft will frequently encounter environmental disturbances such as slowly varying winds and rapidly changing gusts. Both translational \vec{v}_{wg} and rotational $\vec{\omega}_{wg}$ wind-gust velocities are important to consider.

$$\vec{v}_{wg} = (u_{wg} \ v_{wg} \ w_{wg})^T, \quad \vec{\omega}_{wg} = (p_{wg} \ q_{wg} \ r_{wg})^T \quad (\text{A.26})$$

In (A.26) wind-gust components are expressed in the aircraft body axis system. They directly affect the aerodynamic forces and moments acting on the vehicle since those are created by the aircraft motion with respect to the incoming airmass. In order to incorporate wind-gust effects into a flight simulation, we need to define the aircraft translational and rotational velocity vectors relative to the surrounding air.

$$\vec{v}_a = (u_b - u_{wg} \ v_b - v_{wg} \ w_b - w_{wg})^T, \quad \vec{\omega}_a = (p_b - p_{wg} \ q_b - q_{wg} \ r_b - r_{wg})^T \quad (\text{A.27})$$

In this case, using (A.27) the aerodynamic parameters (A.10) must be redefined.

$$V_T = \sqrt{u_a^2 + v_a^2 + w_a^2}, \quad \alpha = \arctan\left(\frac{w_a}{u_a}\right), \quad \beta = \arcsin\left(\frac{v_a}{V_T}\right) \quad (\text{A.28})$$

As a result, the aircraft forces and moments (A.13), the six aerodynamic coefficients (A.15) as well as the vehicle accelerations (A.24) all become dependent on the airmass-related velocities.

Adding wind-gust velocities (A.26) to the aircraft control inputs $\vec{\delta}$ gives the total vector of external inputs for simulation.

$$U = (\vec{\delta}^T \ \vec{v}_{wg}^T \ \vec{\omega}_{wg}^T)^T \quad (\text{A.29})$$

Internal states X (A.21), system outputs Y (A.25), and external inputs U (A.29) are now well-defined and ready to support the simulation block diagram shown in Fig. A.7.

The diagram flow reflects that of the generic simulation in Fig. A.6 but it makes the latter become aircraft specific. In addition to previously discussed signals and models, the atmosphere model is shown to depend on a temperature deviation ΔT from standard day to account for non-standard atmospheric changes in the model outputs. Other models can be added to simulate subsystems such as control actuators and sensors. Formally speaking, every model of a subsystem within the simulation framework has its own set of internal states, and those signals need to be appropriately initialized and managed.

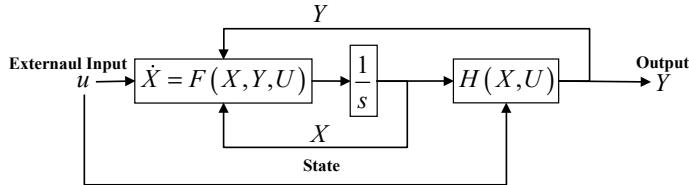


Fig.A.6 Generic simulation block diagram

Aircraft Trim Conditions

Aircraft flight simulation process begins with initialization of the system internal states and its external inputs. For an aircraft that means finding states and inputs such that the system trajectory starts at an equilibrium which is also called “trim”. In terms of a generic system (A.18), an equilibrium is a set of all state-control pairs $(X_{\text{eq}}, U_{\text{eq}})$ that satisfy the algebraic equations.

$$\begin{aligned}\dot{X}_{\text{eq}} &= F(X_{\text{eq}}, Y_{\text{eq}}, U_{\text{eq}}), \quad \dot{X}_{\text{eq}}, X_{\text{eq}} \in R^n, \quad U_{\text{eq}} \in R^m \\ Y_{\text{eq}} &= H(X_{\text{eq}}, U_{\text{eq}}) \in R^p\end{aligned}\quad (\text{A.30})$$

These are not ODEs. The equilibrium time derivative \dot{X}_{eq} and the output Y_{eq} are given. They represent the desired “targets” to achieve. There are $(n + p)$ nonlinear algebraic equations in (A.30), with $(n + m)$ unknowns $(X_{\text{eq}}, U_{\text{eq}})$. Equilibrium calculations are numerical in nature and are performed using an optimization method such as gradient descent with constraints. We use MATLAB®/Simulink® function *fmincon* to trim the aircraft model at any specified feasible flight conditions. The optimization finds the unknown equilibrium pairs $(X_{\text{eq}}, U_{\text{eq}})$ such that the resulting left-hand side values of (A.30) approach the desired targets $(\dot{X}_{\text{eq}}, Y_{\text{eq}})$, as close as possible and within optimization tolerances.

In aircraft dynamics, the most common equilibrium condition is the so-called Wings-Level 1g (WL1g) trim. It defines initial flight conditions for the vehicle to maintain a wings-level flight trajectory on a straight line with a constant airspeed and at a given fixed altitude. In order to do that, the aircraft must be able to generate a vertical force F which is equal and opposite to the gravitational force whose magnitude equals to the vehicle gross weight mg . Scaling the required vertical force by the aircraft mass implies that in order to maintain WL1g trim the aircraft must generate 1g acceleration in the vertical direction.

The WL1g flight trajectory requires finding “trim-with” set of states and controls such that six accelerations, vertical speed, angle of sideslip, and bank angle are equal to zero, while airspeed and altitude are given and held constant. These

are the trim targets, or equivalently the “trim-to” outputs.

$$\begin{aligned} \text{Trim-With: } & (u_b \ v_b \ w_b \ p_b \ q_b \ r_b \ \theta \ \delta_a \ \delta_e \ \delta_r \ \delta_T) \\ \Downarrow \\ \text{Trim - To: } & (\dot{u}_b = 0 \ \dot{v}_b = 0 \ \dot{w}_b = 0 \ \dot{p}_b = 0 \ \dot{q}_b = 0 \ \dot{r}_b = 0 \ \dot{\theta} = 0 \ M_0 \ \beta_0 \ \varphi_0 \ h_0) \end{aligned} \quad (\text{A.31})$$

In (A.31), four of the trim-to components, Mach number M_0 , angle of sideslip β_0 , bank angle φ_0 , and altitude h_0 , are given and fixed. They are denoted with the 0-subscript. The other components represent the 6-DoF trim-to acceleration target values. For WL1g trim, $\beta_0 = \varphi_0 = 0$. The heading angle ψ_0 does not influence aircraft 6-DoF equations, and thus it can be set to any constant value.

For any fixed heading angle ψ_0 , the WL1g 11-dimensional trim-to set and the 11-dimensional trim-with set uniquely define the 16-dimensional equilibrium pair $(X_{\text{eq}}, U_{\text{eq}})$ (A.21), (A.22) and the corresponding 12-dimensional state derivative vector in (A.30). Therefore, the WL1g (11×11) optimization problem is well-posed. The unique WL1g trim solution is found numerically via an iterative optimization method such as MATLAB®/Simulink® *fmicon*. The goal of the optimizer is to iteratively find a set of the selected trim-with inputs such that the target trim-to outputs are achieved within the user-specified numerical tolerances. An optimization block diagram can be drawn starting with the aircraft 6-DoF flight simulation illustration (Fig. A.7) and replacing the simulation integrator function with an optimizer logic.

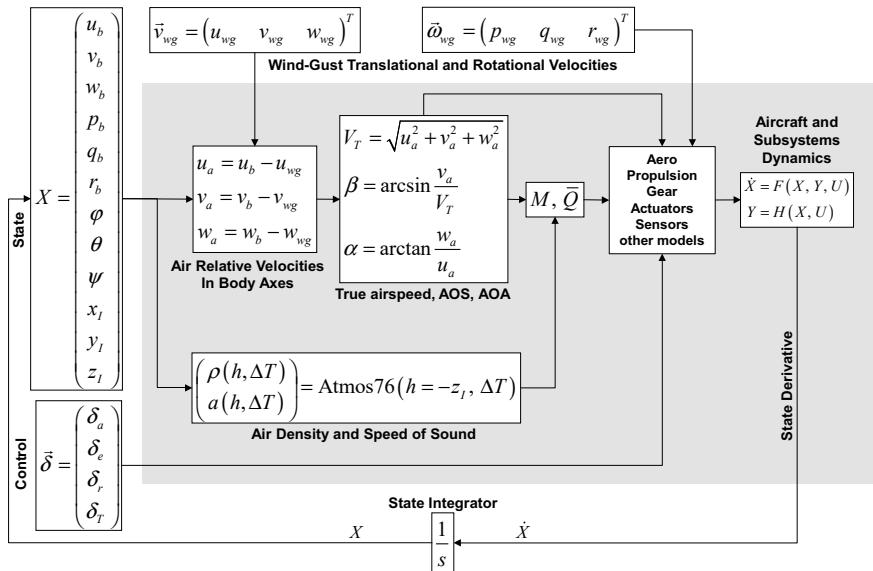


Fig. A.7 Aircraft flight simulation block diagram

Another commonly used aircraft trim configuration is called the steady-heading sideslip (SHSS) flight. This is a trimmed flight trajectory similar to WL1g except the vehicle flies at a constant nonzero sideslip angle $\beta_0 \neq 0$. The latter is defined in the trim-to set (A.31).

We often consider an aircraft flying in a steady turn while maintaining the desired altitude and constant airspeed. In flight dynamics this trim maneuver is referred to as the Wind-Up Turn (WUT). It is performed by finding trim-to inputs such that trim-with outputs are achieved with zero sideslip angle $\beta_0 = 0$ and a desired nonzero bank angle $\varphi_0 \neq 0$.

Linearization and Primary Flight Modes

Linearization of the nonlinear 6-DoF flight dynamics (A.20) is performed numerically using MATLAB®/Simulink® function *linmod*. The process of linearization is based on Taylor series expansions and numerical differentiation of the system right-hand side. Our preferred numerical differentiation method is the central difference.

System: $\dot{x} = f(x, u)$, $x \in R^n$, $u \in R^m$

Trim: $\dot{x}_{\text{eq}} = f(x_{\text{eq}}, u_{\text{eq}})$

$$\text{Linearization: } \dot{x} = f(x_{\text{eq}} + \Delta x, u_{\text{eq}} + \Delta u) \approx \underbrace{f(x_{\text{eq}}, u_{\text{eq}})}_{\dot{x}_{\text{eq}}} + A\Delta x + B\Delta u$$

$$\text{Jacobian w.r.t } x: A = \frac{\partial f(x_{\text{eq}}, u_{\text{eq}})}{\partial x} = \left[\frac{\partial f_i(x_{\text{eq}}, u_{\text{eq}})}{\partial x_j} \right]_{\substack{i=1:n \\ j=1:n}} \approx \left[\frac{f_i(x_{\text{eq}1}, \dots, (x_{\text{eq}j} + \Delta x_j), \dots, x_{\text{eq}n}, u_{\text{eq}}) - f_i(x_{\text{eq}1}, \dots, (x_{\text{eq}j} - \Delta x_j), \dots, x_{\text{eq}n}, u_{\text{eq}})}}{2\Delta x_j} \right]$$

$$\text{Jacobian w.r.t } u: B = \frac{\partial f(x_{\text{eq}}, u_{\text{eq}})}{\partial u} = \left[\frac{\partial f_i(x_{\text{eq}}, u_{\text{eq}})}{\partial u_j} \right]_{\substack{i=1:n \\ j=1:m}} \approx \left[\frac{f_i(x_{\text{eq}}, u_{\text{eq}1}, \dots, (u_{\text{eq}j} + \Delta u_j), \dots, u_{\text{eq}m}) - f_i(x_{\text{eq}}, u_{\text{eq}1}, \dots, (u_{\text{eq}j} - \Delta u_j), \dots, u_{\text{eq}m})}}{2\Delta u_j} \right] \quad (\text{A.32})$$

In (A.32), a nonlinear system $\dot{x} = f(x, u)$ is considered, and its trajectory deviations $\Delta \dot{x} = \dot{x} - \dot{x}_{\text{eq}}$ from equilibrium are approximated by the LTI dynamics via Taylor series expansions to the first order.

$$\Delta \dot{x} = A \Delta x + B \Delta u \quad (\text{A.33})$$

The resulting Jacobian matrices,

$$A = \frac{\partial f(x_{\text{eq}}, u_{\text{eq}})}{\partial x} \in R^{n \times n}, B = \frac{\partial f(x_{\text{eq}}, u_{\text{eq}})}{\partial u} \in R^{n \times m} \quad (\text{A.34})$$

are computed numerically by the central difference method as indicated in (A.32). Derivative approximations are performed iteratively, one state component at a time, while holding the others at their corresponding equilibrium values.

A.3 Simplified Flight Dynamics for Control Design

In order to achieve WL1g trim conditions for (A.20), we need to find a set of equilibrium states and controls such that the three translational and three angular accelerations are zeroed out,

$$\begin{aligned} \text{Translational DoF: } 0 &= -\left[\begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} u \\ v \\ w \end{pmatrix}\right] + \frac{1}{m} \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} + \vec{g} \\ \text{Rotational DoF: } 0 &= -\left[\begin{pmatrix} p \\ q \\ r \end{pmatrix} \times J \begin{pmatrix} p \\ q \\ r \end{pmatrix}\right] + \begin{pmatrix} \bar{L} \\ M \\ N \end{pmatrix} \end{aligned} \quad (\text{A.35})$$

An aircraft would have many distinct equilibriums throughout the vehicle flight operational envelope (Fig. A.8).

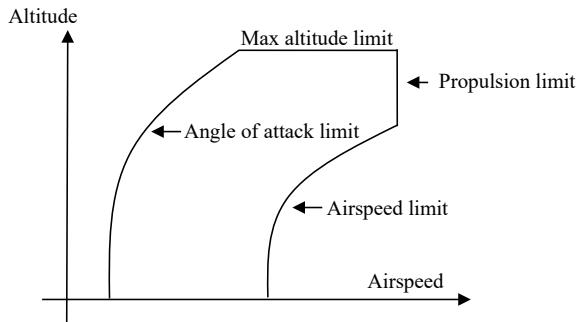
These trim points depend first hand on altitude and airspeed. Based on available trim flight conditions, the main idea behind constructing control-oriented models and then performing flight control design consists of several distinct steps. They are:

1. Cover the flight envelope with a dense set of trim points.
2. Derive simplified linear models around each of the trim point.
3. Use these dynamics to design fixed-point flight controllers per point.
4. Interpolate (i.e., gain schedule based on flight conditions) to combine linear controllers.

The result is a gain-scheduled flight control system that would be valid for the entire flight operational envelope. In what follows, we will concentrate on Step 2 and derive open-loop linear models (deviation dynamics from equilibrium) for a selected trim point.

When a conventional fixed-wing aircraft is trimmed wings-level at selected flight conditions, the vehicle dynamics naturally decouples into longitudinal and lateral-directional modes.

Fig. A.8 Aircraft operational flight envelope, as a function of altitude and airspeed



Longitudinal Dynamics

The aircraft longitudinal dynamics describe changes in forward, vertical, and pitching motion of the vehicle. These dynamics can be further decomposed into fast and slow components, or modes. The former is called the short-period, and the latter is the phugoid. Typically, there would be a timescale separation between the two modes. The short-period describes fast coupling between the aircraft angle of attack and the pitch rate. On the other hand, the phugoid represents a much slower (when compared to the short-period) dynamic interchange between the vehicle altitude and the airspeed, or equivalently, between the aircraft potential and kinetic energy levels.

The short-period and the phugoid modes can be revealed after the aircraft model is linearized around a trim point (an equilibrium). The aircraft linearized longitudinal equations of motion are

$$\begin{pmatrix} \dot{v}_T \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} X_V & X_\alpha & 0 & -g \cos \gamma_0 \\ \frac{Z_V}{V_0} & \frac{Z_\alpha}{V_0} & 1 + \frac{Z_q}{V_0} & -\frac{g \sin \gamma_0}{V_0} \\ M_V & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} v_T \\ \alpha \\ q \\ \theta \end{pmatrix} + \begin{pmatrix} X_{\delta_{th}} \cos \alpha_0 & X_{\delta_e} \\ -X_{\delta_{th}} \sin \alpha_0 & \frac{Z_{\delta_e}}{V_0} \\ M_{\delta_{th}} & M_{\delta_e} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_{th} \\ \delta_e \end{pmatrix} \quad (\text{A.36})$$

where V_0 is the trimmed airspeed, α_0 is trimmed angle of attack, $\gamma_0 = \theta_0 - \alpha_0$ is the trimmed flight path angle (), and θ_0 is the trimmed pitch angle. The model states (v_T, α, q, θ) and the control inputs (δ_{th}, δ_e) are incremental due to their trimmed values.

To shorten notation, we define $p = p_b$, $q = q_b$, $r = r_b$. The short-period mode of an aircraft is defined by angle of attack α and pitch rate q . Extracting those from the model (A.36) yields

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V_0} & 1 + \frac{Z_q}{V_0} \\ M_\alpha & M_q \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} \frac{Z_{\delta_e}}{V_0} \\ M_{\delta_e} \end{pmatrix} \delta_e \quad (\text{A.37})$$

These dynamics describe aircraft motion on a short interval of time, due to elevator input.

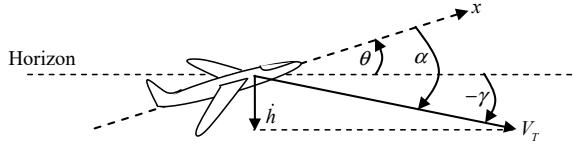
The aircraft phugoid motion can be derived by setting $\dot{\alpha} = \dot{q} = 0$ in (A.36), solving for the corresponding “fast steady-state” values (α, q) and substituting them into the remaining dynamic equations for v_T and θ .

The flight path angle γ is the angle between the aircraft airspeed vector and the horizon. For small angles, the following relationship links angle of attack α , pitch angle θ , and flight path angle γ .

$$\alpha = \theta - \gamma \quad (\text{A.38})$$

It is depicted in Fig. A.9.

Fig. A.9 Aircraft longitudinal motion and related angles



Multiplying both sides of (A.38) by true airspeed V_T gives the vertical speed in inertial space.

$$\dot{h} = V_T \gamma = V_T (\theta - \alpha) \quad (\text{A.39})$$

For small angles and assuming that the true airspeed is constant, we can differentiate (A.39) with respect to time, use α -dynamics from (A.37), and finally compute (approximately) the vertical acceleration a_z in body axes.

$$A_z \approx -\ddot{h} = -V (\dot{\theta} - \dot{\alpha}) = V (\dot{\alpha} - q) = Z_\alpha \alpha + Z_\delta \delta_e \quad (\text{A.40})$$

Throughout the book, this signal is often used as the regulated output for the aircraft longitudinal dynamics (A.37).

$$A_z = (Z_\alpha \ 0) \begin{pmatrix} \alpha \\ q \end{pmatrix} + Z_\delta \delta_e \quad (\text{A.41})$$

To add realism into simulation and control analysis (but not the design), we will use actuator models for control inputs, such as elevator. The actuator dynamics can be modeled by a second-order ordinary differential equation, with a specified natural frequency ω and a damping ratio ξ . The actuator model is driven by the elevator command δ_e^{cmd} , and the model dynamics are of second order.

$$\ddot{\delta}_e = -2\xi\omega\dot{\delta}_e + \omega^2(\delta_e^{\text{cmd}} - \delta_e) \quad (\text{A.42})$$

Lateral-Directional Dynamics

Consider the kinematics of the Euler roll equation from (A.9).

$$\dot{\phi} = p + \tan \theta (q \sin \varphi + r \cos \varphi) \quad (\text{A.43})$$

Let θ_0 denote the trimmed pitch angle. Then a linear approximation of (A.43) around $\varphi_0 = p_0 = q_0 = r_0 = 0$ can be written as

$$\dot{\phi} = p + r \tan \theta_0 \quad (\text{A.44})$$

Define stability axis roll and yaw rates (p_s, r_s). They are related to the body axis roll and yaw rates (p, r) in the following way:

$$p_s = p \cos \alpha + r \sin \alpha$$

$$r_s = r \cos \alpha - p \sin \alpha. \quad (\text{A.45})$$

Let α_0 denote the trimmed angle of attack (AOA). Then a linear approximation of (A.45) is of the form

$$\begin{aligned} p_s &= p \cos \alpha_0 + r \sin \alpha_0 \\ r_s &= r \cos \alpha_0 - p \sin \alpha_0 \end{aligned} \quad (\text{A.46})$$

Solving (A.46) for (p, r) yields

$$\begin{aligned} p &= p_s \cos \alpha_0 - r_s \sin \alpha_0 \\ r &= r_s \cos \alpha_0 + p_s \sin \alpha_0 \end{aligned} \quad (\text{A.47})$$

Substituting (A.47) into (A.44) results in

$$\begin{aligned} \dot{\phi} &= p_s \cos \alpha_0 - r_s \sin \alpha_0 + (r_s \cos \alpha_0 + p_s \sin \alpha_0) \tan \theta_0 \\ &= (\cos \alpha_0 + \sin \alpha_0 \tan \theta_0) p_s + (\cos \alpha_0 \tan \theta_0 - \sin \alpha_0) r_s \end{aligned} \quad (\text{A.48})$$

The following relation exists between the flight path angle, the pitch angle, and the angle of attack (at zero bank and sideslip angles):

$$\alpha_0 = \theta_0 - \gamma_0 \quad (\text{A.49})$$

Substituting (A.49) into (A.48) gives

$$\begin{aligned} \dot{\phi} &= \underbrace{(\cos \alpha_0 + \sin \alpha_0 \tan \theta_0)}_{\frac{\cos \gamma_0}{\cos \theta_0}} p_s + \underbrace{(\cos \alpha_0 \tan \theta_0 - \sin \alpha_0)}_{\frac{\sin \gamma_0}{\cos \theta_0}} r_s \\ &= \frac{\cos \gamma_0}{\cos \theta_0} p_s + \frac{\sin \gamma_0}{\cos \theta_0} r_s \end{aligned} \quad (\text{A.50})$$

Assuming small angles, the angle of sideslip dynamics can be written as,

$$\dot{\beta} = \frac{1}{V_0} (Y_\beta \beta + Y_p p_s + Y_r r_s + Y_{\delta_{ail}} \delta_{ail} + Y_{\delta_{rud}} \delta_{rud}) + \left(\frac{g \cos \theta_0}{V_0} \right) \varphi - r_s \quad (\text{A.51})$$

where the right-hand side of the equation depends on the derivatives of the side force Y , computed with respect to the lateral-directional states $(\beta, p_s, r_s, \varphi)$ and the control inputs $(\delta_{ail}, \delta_{rud})$. Using (A.50) and (A.51), the aircraft lateral-directional linearized dynamics are

$$\dot{\phi} = \frac{\cos \gamma_0}{\cos \theta_0} p_s + \frac{\sin \gamma_0}{\cos \theta_0} r_s$$

$$\begin{aligned}\dot{\beta} &= \frac{g \cos \theta_0}{V} \varphi + \frac{Y_\beta}{V} \beta + \frac{Y_p}{V} p_s + \left(\frac{Y_r}{V} - 1 \right) r_s + \frac{Y_{\delta_{ail}}}{V} \delta_{ail} + \frac{Y_{\delta_{rud}}}{V} \delta_{rud} \\ \dot{p}_s &= L_\beta \beta + L_p p_s + L_r r_s + L_{\delta_{ail}} \delta_{ail} + L_{\delta_{rud}} \delta_{rud} \\ \dot{r}_s &= N_\beta \beta + N_p p_s + N_r r_s + N_{\delta_{ail}} \delta_{ail} + N_{\delta_{rud}} \delta_{rud}\end{aligned}\quad (\text{A.52})$$

We can easily rewrite (A.52) in matrix form.

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{\cos \gamma_0}{\cos \theta_0} & \frac{\sin \gamma_0}{\cos \theta_0} \\ \frac{g \cos \theta_0}{V_0} & \frac{Y_\beta}{V_0} & \frac{Y_p}{V_0} & \frac{Y_r}{V_0} - 1 \\ 0 & L_\beta & L_p & L_r \\ 0 & N_\beta & N_p & N_r \end{pmatrix} \begin{pmatrix} \varphi \\ \beta \\ p_s \\ r_s \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \frac{Y_{\delta_{ail}}}{V_0} & \frac{Y_{\delta_{rud}}}{V_0} \\ L_{\delta_{ail}} & L_{\delta_{rud}} \\ N_{\delta_{ail}} & N_{\delta_{rud}} \end{pmatrix} \begin{pmatrix} \delta_{ail} \\ \delta_{rud} \end{pmatrix}\quad (\text{A.53})$$

When the airspeed is sufficiently high, the gravity term in (A.53) becomes negligible: $\frac{g \cos \theta_0}{V_0} \approx 0$. In this case, the bank dynamics can be eliminated.

$$\begin{pmatrix} \dot{\beta} \\ \dot{p}_s \\ \dot{r}_s \end{pmatrix} = \begin{pmatrix} \frac{Y_\beta}{V_0} & \frac{Y_p}{V_0} & \frac{Y_r}{V_0} - 1 \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{pmatrix} \begin{pmatrix} \beta \\ p_s \\ r_s \end{pmatrix} + \begin{pmatrix} \frac{Y_{\delta_{ail}}}{V_0} & \frac{Y_{\delta_{rud}}}{V_0} \\ L_{\delta_{ail}} & L_{\delta_{rud}} \\ N_{\delta_{ail}} & N_{\delta_{rud}} \end{pmatrix} \begin{pmatrix} \delta_{ail} \\ \delta_{rud} \end{pmatrix}\quad (\text{A.54})$$

The resulting fourth- and third-order lateral-directional linear models are suitable for control design and analysis.

A.4 aFltSim Block Diagram Architecture and Calling Sequence

The aFltSim software environment is implemented using MATLAB®/Simulink® block diagrams [1]. The underlying database for aerodynamics and engine models comes from [2]. The data are representative of a medium-size turbo-jet aircraft. MATLAB® scripts are set up to trim, linearize, initialize, and simulate the aircraft dynamics in the time domain. Figure A.10 shows the top-level aFltSim block diagram.

The green ovals are the Simulink® “imports”. They represent the input signal injection locations within the diagram. The blue ovals are the “outports”. They are the output collection locations. Simulink® uses these definitions to trim and linearize aFltSim. The rest of the input and output signals follow the naming convention as defined above and within this chapter.

Figure A.11 shows details of the “AircraftOpenLoopSimulation” simulation block. This is the core software within the aFltSim environment. Aircraft models, equations of motion, and output signals are encoded here.

The main run script to execute aFltSim is called “run_aFltSim_OpenLoop.m”. This script is menu-driven and gives the user options for selecting initial flight conditions, trimming, linearizing, and simulating the open-loop aircraft dynamics due to various inputs. Verifications of open-loop linear model responses versus

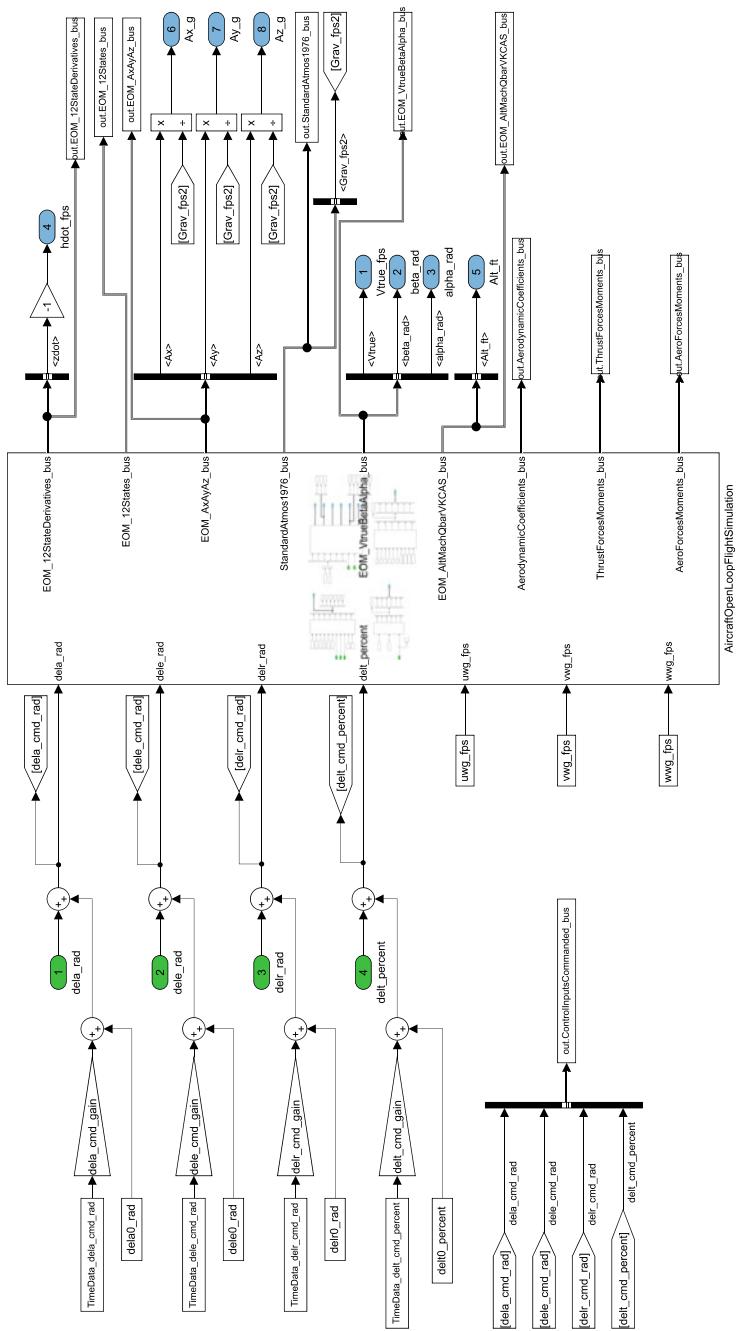


Fig. A.10 aFltSim top-level Simulink® block diagram

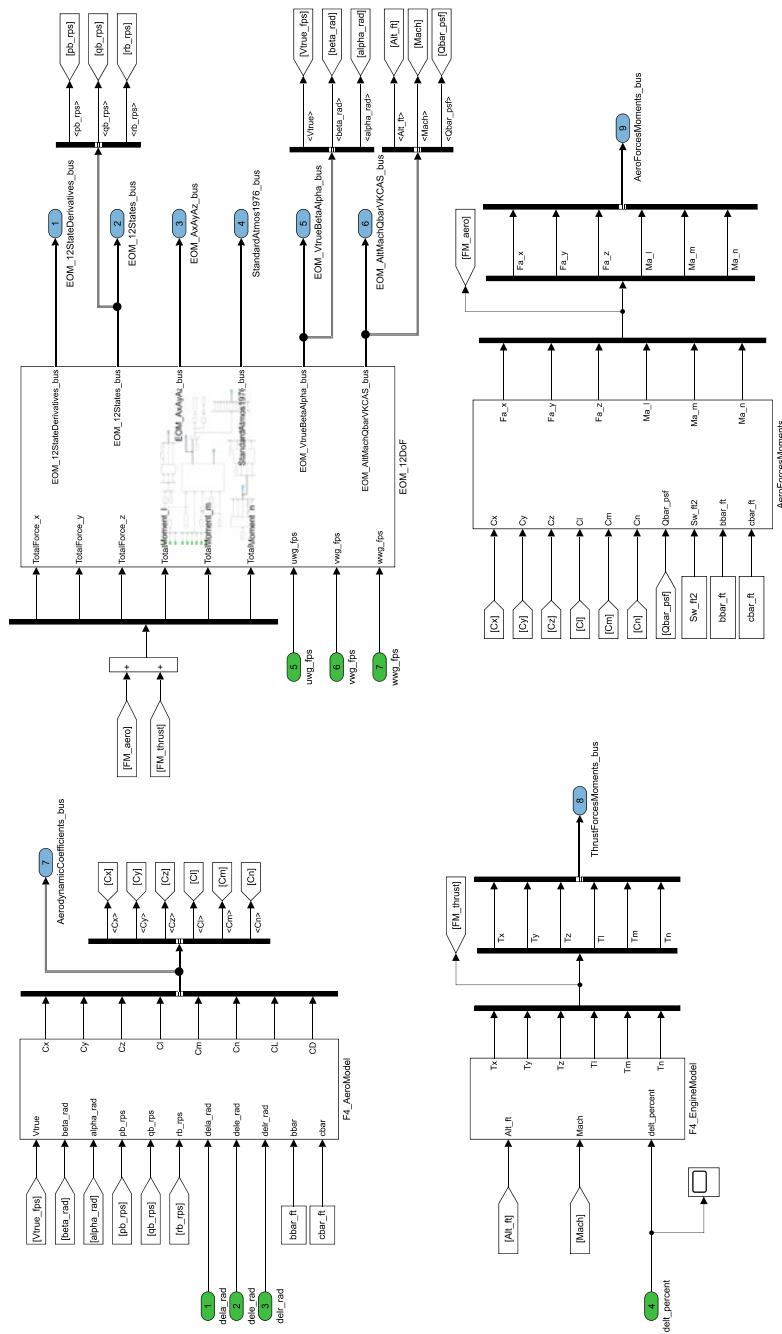


Fig. A.11 aFltSim second-level Simulink® block diagram

the 6-DoF output are performed within the main script by calling the respective subscripts. Each subscript within the main script is written to perform a distinct function. Detailed comments throughout aFltSim software should help the user navigate and better understand the sequence of calls from the main run script to other subscripts.

Adding control systems to aFltSim is seamless. Once the reader becomes familiar with the open-loop aFltSim architecture and studies the book material, a task such as adding a control system is assigned as a homework exercise.

References

1. MATLAB®/Simulink®, The MathWorks, Inc., Natick, Massachusetts, United States
2. Grauer JA, Morelli EA (2015) Generic global aerodynamic model for aircraft. *J Aircraft* 52(1)
3. Stevens BL, Lewis FL (1992) Aircraft control and simulation. Wiley, New York

Index

A

A+B Argument, 126

Actuator model

first-order, 488

natural frequency and damping of, 36, 452, 602

second-order, 34, 242, 489, 505, 699

uncertainty model for, 142

Adaptive control, 3, 5, 6, 32, 40, 41, 46, 97, 172, 173, 307, 332, 347, 368–370, 405, 408, 409, 413, 414, 442, 455, 463, 498, 507, 539, 547, 576–578, 582, 592, 603, 605, 619, 634, 641, 642, 671, 675, 677, 679

Adaptive control models

matched uncertainties, 40, 360, 616, 620

Aircraft control

ailerons, 8, 13, 16, 21, 152, 438, 498

elevator, 8, 13, 16, 21, 30, 41, 48, 131, 451, 531, 597

inner loop, 42, 51, 52, 224

outer loop, 51, 52

rudder, 8, 13, 16, 21, 42, 152, 361, 498

Aircraft dynamics

altitude, 10, 22, 24, 31, 51, 370, 528, 692, 698

angle of attack (AOA), 12, 24, 30–32, 41, 233, 270, 437, 451, 525, 527, 569, 611, 688, 698

angle of sideslip (AOS), 12, 688, 695

body axis rates, 6, 8, 32

body axis velocities, 13, 689

drag, 14, 690

equations of motion, 6, 32, 684, 701

Euler angles, 9, 686

lateral acceleration, 622

lateral-directional, 19, 26, 35, 494, 506, 622, 697, 700

lateral-directional dynamics, 19, 26, 34, 35, 37, 39, 42

lift, 14, 360, 527, 690

linearization, 25, 43, 622

longitudinal, 19, 26, 31–33, 41, 569, 597, 640, 643, 644, 698, 699

longitudinal dynamics, 31–33, 241, 353, 270, 597, 640, 698, 699

output measurements, 582, 634

short-period, 31, 32, 34, 39, 41, 101, 451, 467, 569, 570, 610, 643–645, 679

short period and phugoid modes, 31, 41, 698

sideslip, 494, 498, 622

simulation of, 20, 683

six-degrees-of-freedom (6-DoF), 7, 8, 11, 12, 16, 20, 23, 25, 28, 29, 685, 688, 689, 691, 695

stability axis rates, 270, 622

state vector, 26, 611

trim, 22, 24, 29–31, 41, 569, 694–696, 697, 701

vertical acceleration, 42, 51, 354, 569, 598

Aircraft flight simulation

linearization of, 25, 43, 622

noise and environmental disturbances, 40

trim-to components, 23, 695

wind-up-turn (WUT), 24, 696

wings-level 1g (WL1g), 22, 31, 43, 569, 697

Aircraft FLight Simulation (aFltSim) software environment, 20, 622, 683

Aircraft motion

air density, 13

angle-of-attack, 24, 31, 451, 452, 525, 527, 569, 611, 698

angle-of-sideslip, 12, 688

dynamic pressure, 13, 689

Mach number, 13, 689

speed of sound, 13, 689

- true air speed, 528
- Algebraic Riccati Equation (ARE), 152, 198–201, 204, 205, 207, 211, 234, 241, 244, 247, 248, 259, 261–264, 268, 295, 297, 351, 354, 459, 552–554, 556–558, 560, 562, 563, 568–570, 572, 573, 582, 584, 585, 587, 589, 591, 593, 596, 600, 601, 612, 616, 627, 634, 656, 672
- asymptotic properties of, 297, 303, 304, 552, 553, 634
- A-matrix
- eigenvalues of, 83, 89, 110, 114
 - eigenvectors of, 83, 85
 - modal form, 83
- Approximation-based adaptive control, 507
- ARE, *see* Algebraic Riccati Equation
- Artificial neural network
- approximation properties of, 515, 536
 - definition, 515
- Asymptotic expansions, 301, 302, 304, 306–308, 310, 311, 327, 354, 547–549, 553–555, 557, 558, 568, 585, 587, 589
- Asymptotic orders, 298, 299, 301, 306, 547, 548, 550
- B**
- Barbalat's lemma, 366, 404–407, 409, 419, 429, 436, 449, 462, 541, 543, 565, 578, 595
- Baseline control, 46, 172, 331, 452, 464, 528, 573, 591, 596, 601, 605, 606, 658, 671, 672
- Bode diagram, 103, 110
- Bode's integral theorem
- “waterbed” effect, 610, 614, 615
- C**
- Characteristic equation, 82–84, 90, 91, 202
- Closed-loop
- characteristic polynomial, 63, 115, 119, 124
- Closed-Loop Reference Model (CRM)
- adaptive output feedback, 577
 - observer-like control, 561, 568, 569, 577, 578, 635
- Command tracking, 3, 5, 48–51, 66, 67, 109, 172, 248, 254–256, 258, 267, 271, 350, 352, 359, 361, 362, 368, 371, 415, 416, 419, 421–424, 430, 431, 436, 445–447, 450, 454, 463, 479, 481, 493, 497, 499, 514–516, 521, 536, 570, 573, 574, 576, 577, 587, 609, 617, 628, 640, 642, 655, 659, 667, 670, 674
- Complementary sensitivity, 61, 63, 105, 106, 108, 109, 127–130, 136, 247, 255–259, 262, 263, 265–267, 346
- Controllability of LTI system
- Controllability matrix W_c , 90, 91
 - PBH eigenvector test, 90, 91
 - PBH rank test, 91
- Controller transfer functions
- butterworth filter, 72, 73
 - lag filter, 68, 69, 249, 257, 258
 - lead filter, 68, 69
 - leaky integrator, 68
 - notch filter, 71, 72, 249, 560
 - proportional-integral-derivative (PID), 65, 66
- D**
- ΔM analysis model, 116, 143, 146, 149, 153, 159, 167, 168
- Dyadic expansion, 113
- Dynamic inversion, 425, 426, 428
- Dynamic inversion MRAC for scalar dynamics, 425, 429–431, 443
- Dynamic output feedback control, 272, 285, 329, 623, 627, 634
- E**
- Equation of motion
- forces and moments for, 12, 688
 - gravity vector, 6, 7
 - non-dimensional coefficients, 13, 690
 - rotational, 395
 - translational, 6, 683, 689, 693
 - wind models, 12
- F**
- Feedback control
- gain design and analysis process, 46, 230, 232
- Feedforward control, 336
- Forward Euler integration, 15, 57, 97, 691
- Frequency domain analysis, 46, 47, 77, 79, 80, 103, 104, 106, 249, 288, 333–335, 342, 345
- Frobenius norm, 96, 113, 477
- Function
- continuous, 381, 402, 513, 514
 - convex, 483, 484, 488, 489, 492, 494, 521

- Lipschitz-continuous, 381, 419, 425, 432, 446, 449, 550
 negative-definite, 390
 positive-definite, 188, 190, 197, 390–392, 400, 401
 radial basis function (RBF), 510, 512, 536, 674, 675
 sigmoid, 510, 536
 uniformly-continuous, 366, 405–407, 410, 419, 449, 462, 565
- G**
 Gain-crossover frequency, 230, 237, 238, 241, 282, 352, 353, 666
 Gain-crossover point, 121
 Gain margin, 121–123, 129, 130, 139, 159, 162, 171, 203, 207, 210, 239, 282, 286, 334, 359, 572, 602, 612
- H**
 Hamiltonian matrix, 211, 212, 243, 261, 338
 Hamilton-Jacobi-Bellman partial differential equation, 177, 179, 183, 249
 H-inf optimal control
 algebraic riccati equation (ARE), 247, 248, 259, 261, 263, 268
 design model for, 257, 263, 266
 Hamiltonian for, 260
 Hamiltonian matrix for, 260
 performance index for, 247, 254–256, 359, 361
 regulated variables in, 248, 257, 262
 Hypercube, 161–167, 169, 505
- I**
 Input loop gain linear model, 329
 Integral control, 68, 106, 107, 130, 255, 276, 333, 445
 Internal model principle, 214
- L**
 Laplace Transform (LT)
 differentiation, 59
 exponential, 59, 86
 integration, 59, 219
 Step function, 59
 LaSalle–Yoshizawa theorem, 407, 409
 Legendre-Clebsch condition, 183
 Linear controller model, 3, 31, 40, 45, 445, 456, 458
 Linear plant model, 333, 342, 599
 Linear Quadratic Gaussian (LQG), 353, 581, 640, 641, 655, 657–659
 Linear Quadratic Regulator (LQR)
 design charts for, 240
 gain and phase margins of, 171, 203, 207, 238, 602
 return difference inequality for, 206, 207, 628
 Linear-Time-Invariant (LTI), 25, 26, 45, 54, 56–58, 60, 77, 81, 82, 84, 85, 87–93, 97, 103, 106, 108, 120, 121, 125, 126, 197, 200, 203, 211, 215, 217, 375, 396, 397, 410, 583, 696
 Lipschitz condition, 185, 378, 379, 381
 Loop gain transfer function, 76, 104, 105, 114, 203, 292, 326, 610, 614, 667
 Loop shaping
 block diagram of, 257
 weighting filters for, 257, 258
 Loop Transfer Recovery (LTR), 272, 288, 297, 324–326, 329–331, 337–339, 345–347, 349, 581, 582, 616, 627, 628, 640, 641, 655, 657–659, 661, 666, 668, 671, 679
 Low pass filter
 corner frequency of, 67, 68, 71
 time constant of, 66
 LQG, *see* Linear Quadratic Gaussian
 LQR, *see* Linear Quadratic Regulator
 LTR, *see* Loop Transfer Recovery
 Lyapunov algebraic equation, 199, 396, 397, 411, 500
 Lyapunov function
 candidate, 365, 392–396, 398, 404–406, 408, 418, 426, 434, 448, 461, 470, 477, 481, 519, 541, 543, 564, 593
 definition, 187
 geometric interpretation, 392
 radially unbounded, 190, 394, 395, 405, 408, 419, 434, 448, 449, 461, 462, 470, 519, 541
 Lyapunov's direct method, 188, 390–393, 396–398, 404, 408
 Lyapunov stability
 asymptotic, 385, 392, 394–397, 409, 418, 419, 472, 541, 543, 546
 definitions, 384, 387
 geometric interpretation, 386, 387
 global, 387
 historical roots of, 370
 local, 387
 of motion, 373
 theorems, 389

- uniform, 387, 392
- M**
- Matched uncertainties, 40, 360, 432, 442, 445, 451, 457, 460, 463, 466, 469, 528, 536, 568, 574, 575, 581, 587, 616, 619, 620, 629, 634, 639, 646, 648, 670–674, 676–678
- Matching conditions, 40, 360, 417, 433, 434, 437, 438, 450, 452, 454, 456, 517, 518, 528, 541, 559, 560
- Matrix norm, 96, 120, 298, 301, 347, 547, 548, 582, 606
- Matrix square root, 201, 306, 553
- Micchelli's theorem, 512
- Model Reference Adaptive Control (MRAC)
- augmentation of an optimal baseline controller, 456
 - direct, 359, 363, 368, 413
 - historical roots of, 370
 - indirect, 415
 - observer-like, 561
 - output feedback, 596
 - parameter convergence in, 369, 420, 440
 - scalar linear systems, 368
 - state-feedback, 413, 470, 473, 515, 568, 577, 578, 635
 - with closed-loop reference model (CRM), 619
 - with improved transient dynamics, 539
 - with integral feedback, 445, 451
 - with state-limiting constraints, 523
- Model reference control, 359–361, 597
- MRAC, *see* model reference adaptive control
- MRAC design examples
- aircraft lateral-directional dynamics, 494
 - aircraft short-period dynamics, 39
 - automatic landing system for transport aircraft, 524
 - control of a flexible transport aircraft, 597
 - DC-8 short-period dynamics, 463, 464
 - dead-zone mod for aircraft roll dynamics, 473
 - delta wing dynamics at high angle of attack, 436
 - e-mod mod for aircraft roll dynamics, 472
 - helicopter pitch dynamics, 430
 - sigma mod for aircraft roll dynamics, 479, 481
- N
- Norms, 95, 96, 136, 150, 247–252, 263, 306, 377, 471, 472, 474, 485, 510, 511, 513, 515, 549, 551–553, 574, 582, 607, 610, 617, 619, 621, 627, 630, 657, 675, 677
- matrix norms, 96, 120, 298, 301, 347, 547, 548, 582, 606
- vector norms, 96, 298, 415, 547
- Nyquist
- D-contour, 114
 - diagram, 103, 110, 116, 208
 - encirclements, 114–118, 124, 174
 - multivariable criterion, 114–118
- O**
- OBLTR, *see* Observer-Based loop Transfer Recovery
- Observability of LTI system
- observability matrix W_0 , 93, 94
 - PBH Eigenvector Test, 93, 94
 - PBH Rank Test, 94
- Observer-Based Loop Transfer Recovery (OBLTR), 272, 297, 325, 597, 639, 641, 646, 647, 655, 657–661, 665, 666, 668–679
- Open-loop characteristic polynomial, 63, 115, 119, 124, 213
- Optimal control
- algebraic Riccati equation (ARE), 199, 211, 241, 582
 - boundary conditions for, 183, 185, 196, 241
 - conditions on plant and performance index to solve, 201
 - eigenstructure of closed-loop, 211
 - guaranteed margins for, 203
 - Hamiltonian, 183, 184, 211, 212, 243, 260
 - Hamiltonian matrix, 211, 212, 243
 - Hamilton-Jacobi-Bellman (HJB) equation, 177, 179, 183
 - infinite-time LQR problem, 197
 - infinite-time performance index, 177
 - linear quadratic regulator, 177, 178, 197, 240, 243, 359, 552
 - loop transfer function, 203
 - LQR Hamiltonian, 195
 - optimal control policy, 180, 183, 192, 196, 241, 298
 - performance index, 177–180, 183, 187, 242
 - principle of optimality, 183, 186
 - quadratic performance index, 177, 178, 195, 197
 - return difference matrix, 239

-
- root locus, 203
state feedback control law, 259
- Ordinary differential equations
equilibrium of, 199
existence and uniqueness of solutions, 691
initial value problem, 196
- Orlicz's theorem, 381
- P**
- Peano's theorem, 377
- PE conditions, *see* persistency of excitation
- Persistency of Excitation (PE), 369, 420, 455, 474, 481, 677
- Phase crossover frequency, 121
- Phase crossover point, 123
- Phase margin, 49, 52, 69, 71, 76, 104, 110, 116, 120–123, 125, 129, 130, 133, 138, 139, 171, 174, 178, 203, 207, 210, 238, 258, 282, 286, 296, 325, 330, 331, 334, 359, 572, 602, 612, 625, 635, 639, 642
- Positive real
transfer function, 213
- Power signals, 250
- Principle of optimality, 181, 183, 186
- Projective control
dynamic output feedback, 271, 272, 285, 287
eigenstructure of, 271, 272, 285
residual dynamics in, 274, 275
static output feedback, 272, 276, 281
- Proportional plus integral control, 107, 225, 232
- Q**
- Quadratic cost function, 189, 197, 583
- R**
- Real parameter uncertainties
uncertainty models of, 145, 169
- Real stability margin, 103, 140, 159, 165
- Reference coordinate frames
earth-fixed inertial reference frame, 6, 683
North-East-Down (NED) reference frame, 6, 683
- Return difference dynamics, 105, 150, 239
- Ridge function. *See* sigmoid
- Robust MRAC design
dead-zone mod, 469, 472
e-mod, 472, 480
projection operator, 469, 482, 483, 486
- sigma-mod, 472
- Root locus, 61, 63, 64, 98, 103, 159, 171–173, 202, 211, 234, 235, 237, 271, 344
- S**
- Sensitivity, 28, 61, 62, 105, 106, 108, 109, 127–130, 136, 138, 139, 157, 159, 172, 176, 178, 183, 203, 204, 207, 247, 255–258, 262, 263, 265–267, 290, 346, 364, 388, 480, 602, 610, 612, 614, 615, 625, 626
- Separation principle, 272, 291, 292, 296, 325, 334, 337
- Servomechanism design model
controllability of, 220, 222
control law for, 146, 175, 219, 222, 458
loop gain crossover frequency, 569, 570
- LQR, *see* Linear Quadratic Regulator
- LQR performance index, 147, 152, 458, 459
- overshoot, 268, 569
- rise time (63%), 76, 569, 612
- settling time (95%), 49
- singular stability margins, 3, 129
- tracking constant commands, 3, 50, 215, 219, 370, 414
- undershoot, 49, 268, 539, 569
- Similarity transformation, 291, 316
- Singular
matrices, 108, 111, 204, 208, 234
perturbations, 205, 208, 546, 547, 549–551, 566, 567
- Singular value
decomposition, 111–113, 145, 148, 307, 311, 316, 327, 554, 557, 589, 596, 625, 628, 658, 673
maximum (2-norm), 96, 111, 113, 353
minimum, 111, 203, 235, 239, 326
robustness tests, 149
stability margins, 119, 124, 131, 136, 140, 153, 175, 203, 345, 348, 352, 353
- Small gain theorem, 140, 149, 151, 156–158, 169, 174, 175, 202
- Solution of state-space equations
forced response, 55
initial condition response, 55, 188
- Squaring-up
for tall systems, 315, 345, 581, 582, 587, 647, 650
- method, 563, 588, 601, 616, 627, 635, 646, 647, 655, 658, 666
- plant, 600, 602
- transmission zeros in, 325, 563, 582, 588, 658, 672

- Stability analysis model, 116, 140, 233, 235, 238
 Stability margins
 gain margin, 122, 159, 203, 210, 238, 334, 612
 phase margin, 178, 203, 207, 210, 239, 282, 331, 334, 612, 625, 642
 singular value-based margins, 120, 238
 time-delay margin, 140, 570
 vector margin, 120, 123, 171, 572
 State feedback control law, 175, 223, 247, 259, 261, 273, 293, 341
 State observers
 eigenvalues of, 272, 282, 288
 full-order, 288–290, 292, 294, 296, 312, 329
 global exponentially stability of, 290
 necessary and sufficient conditions for, 289
 State-space model
 controller model, 77, 132, 234
 loop gain at plant-input, 132, 235
 loop gain at plant-output, 234
 plant model, 78, 132, 333, 334
 State transition matrix
 infinite series expansion of, 55
 matrix exponential, 55
 properties of, 85
 Strictly positive real, 197, 204, 210, 308, 555
 Structured singular value, 103, 138, 140, 149–151, 155, 157–159, 176
 System type, 214
- T**
- Terminal manifold, 180
 Time delay
 approximation of, 75, 76
 Pade' approximation of, 75, 76
 series approximation of, 75, 76
 Transfer function
- U**
- Uncertainty
 additive, 141, 142, 575, 576
 dynamic models of, 142, 151, 159, 177, 192
 multiplicative, 125, 208, 210
 real parameter, 139, 140, 143, 145, 147, 159, 160, 169, 171, 176, 208
 Uniform ultimate boundedness
 comparison with Lyapunov stability, 399
 concept, 397–399
 definition, 398, 399
 Lyapunov-based analysis of, 400
- V**
- Vector norms
 1-norm, 96, 249
 2-norm, 96, 249
 inf-norm, 96, 249, 251, 252, 257
 p-norm, 96