

DISSERTATION

Efficient robust control design and optimization methods for flight control

ausgeführt zur Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften
unter der Leitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Martin Kozek

am Institut für Mechanik und Mechatronik
Arbeitsgruppe für Regelungstechnik und Prozessautomatisierung

eingereicht an der Technischen Universität Wien
Fakultät für Maschinenbau

von

Dipl.-Ing. Alexander Schirrer
Lerchenfelder Str. 13/6/61
1070 Wien
Matr.-Nr. 0125874

Wien, 2011

For my wife Olesea, my daughter Anna-Sofia,
my parents and my family,
with all of my love.

Kurzfassung

In dieser Dissertation werden einige wesentliche Aufgaben im Reglerentwurfsprozess für integrierte Flugregelungen von großen, flexiblen Flugzeugen beleuchtet und in den folgenden vier Kernbereichen erweitert: Zuerst werden mehrere Beiträge zur Methodik des konvexen Reglerentwurfs erbracht, darunter die Entwicklung eines leistungsfähigen Optimierungswerkzeuges und Lösungsansätze zur Synthese stabiler und stabilisierender Regler. Danach wird ein allgemeiner Ansatz zur Optimierung von Entwurfsparametern in komplexen Reglerentwurfsabläufen mithilfe genetischer Algorithmen formuliert, implementiert und getestet. Schließlich wird das Problem der Ordnungsreduktion von Reglern behandelt und zwei Formulierungen im Rahmen der frequenzgewichteten balancierten Reduktion vorgestellt, die die Beibehaltung der Regelleistung ermöglichen. Letztendlich wird die Parametrierung von Reglern, genauer die darin auftretende Interpolation lineardynamischer Systeme auf Stabilität untersucht. Ein neuer Ansatz zur modalen Interpolation mit Methoden der Geometrischen Algebra wird vorgestellt, der vielversprechende erste Resultate liefert.

Mehrere komplexe Fallstudien stellen die Leistung der entwickelten Methoden dar. Dazu zählen vor allem die erfolgreichen, komplexen Reglerentwürfe für die Lateralregelung eines großen, flexiblen Nurflügel-Flugzeugs mit hohen Anforderungen an die Regelleistung und Robustheit.

Die Arbeit beginnt mit einer umfassenden Einführung in die nötigen Grundlagen und den Stand der Technik bei modernen optimalen und robusten Entwurfsmethoden für Regler und dynamische Vorsteuerungen, gefolgt von einer kurzen Darstellung der Flugzeugmodelle, die als Basis für die späteren Fallstudien dienen. Die genannten neuen Beiträge samt der Entwurfs-Fallstudien bilden den Kern der Arbeit, dem eine Zusammenfassung und Ausblick folgen. Die Arbeit endet mit einem ausführlichen Anhang, der zusätzliches Material als direkte Referenz bereitstellt.

Die hier entwickelten Methoden zum Entwurf, zur Reduktion und zur Interpolation von Reglern ermöglichen die Beherrschung hochkomplexer Entwurfsaufgaben in der Flugregelung großer, flexibler Flugzeuge. Eindrucksvolle Ergebnisse konnten dabei in den vorgestellten Entwürfen der Lateralregelung für ein derartiges Nurflügel-Flugzeug erzielt werden.

Abstract

In this thesis, several essential tasks in the control design process for integrated flight control of large, flexible aircraft are investigated and extended in the following four core areas: First, several extensions to the convex control design methodology are proposed, including the development of an efficient numeric optimization tool and solution onsets to generate stable stabilizing controllers. Second, a general onset for design parameter optimization in complex control design flows by genetic algorithms is formulated, implemented, and tested. Third, the controller order reduction problem is considered and two frequency-weighted balanced reduction onsets to preserve control performance are proposed. Finally, the scheduling of control laws, specifically the task of interpolating linear system dynamics, is investigated for stability. A novel modal interpolation onset utilizing Geometric Algebra concepts is proposed which yields promising first results.

Numerous complex case studies demonstrate the performance of the developed methods. Specifically, the complex control designs for the lateral control of a large flexible blended wing body aircraft with high demands on control performance and robustness can be carried out successfully.

The elaboration starts with a comprehensive introduction into the fundamental concepts and into the state of the art of modern optimal and robust control design methods for feedback and feed-forward control. The aircraft models used in the subsequent case studies are introduced, followed by the core chapters which present the novel contributions as well as the associated case studies and examples. A summary and outlook on future research as well as an extensive appendix with additional material for quick reference conclude the work.

The methods and tools developed herein for control design, order reduction, and controller interpolation enable the engineer to master highly complex design tasks in the flight control design of large flexible aircraft. Impressive results could be obtained in the presented case studies on lateral flight control design for a flexible blended wing body aircraft.

Preface

This work represents my doctoral thesis and is the result of my research work in the years 2007 to 2011 at the Vienna University of Technology, Institute of Mechanics and Mechatronics, Workgroup of Control and Process Automation. In these years I enjoyed the inspiring and warm atmosphere in the work with my colleagues, but even more the strong support of my loving family. I sincerely express my gratitude for the support I received from my family, from my mentor and supervisor Prof. Martin Kozek as well as from Dr. Christian Benatzky who introduced me to the exciting field of robust flexible structure control. Besides the technical aspects, we have also always understood well on an informal and personal level. In the course of the EU-FP7 research project ACFA 2020 [1], which also builds the basis for many results in this thesis, I became acquainted to many of my current colleagues. I gratefully acknowledge the brilliant collaboration I had with my teammates Christian Westermayer and Mark Hemedi as well as with Prof. Martin Hromcik and Tomas Hanis from CTU Prague during these years. Having made many findings and experiences on the way, I am happy to look back, but also to look forward to future challenges.

The helpful collaboration also shaped the style of this thesis: I aimed for a well-understandable and in some sense sufficiently complete treatment of fundamentals in modern optimal and robust control methods so that it may be of helpful reference for future PhD students. Many detailed comments from and discussions with my colleagues during the thesis' preparation improved its methodological and structural quality. It is my hope that the contributions herein inspire, trigger, or support research efforts in the area. With best regards to the reader,

Alexander Schirrer
the author

Contents

Glossary	xi
I Introduction & State of the Art	1
1 Introduction	2
1.1 Overview and Motivation	2
1.2 Literature Review, State of the Art	3
1.2.1 Linear Algebra	3
1.2.2 Robust and Optimal Control	4
1.2.3 Flexible Structure Control	5
1.2.4 Flight Control Design	5
1.2.5 Further Research of the Author & Related Work	5
1.3 Contributions of this Work	6
1.4 Structure of this Work	7
2 Fundamentals	9
2.1 State Space Representation	9
2.1.1 General State Space Systems & Linearizations	9
2.1.2 Transfer Function Representation, Proper Systems	10
2.1.3 Gramians	12
2.2 Balanced Reduction	12
2.2.1 Frequency-Weighted Balanced Reduction (FWBR)	14
2.3 Robust Control Fundamentals	14
2.3.1 Robustness Analysis	15
2.3.2 Uncertainty Representation and Modeling	20
2.3.3 Robust Control Design Problem Statement	24
2.4 Optimization Fundamentals	24
2.4.1 Generic Optimization Problem	24
2.4.2 Convexity	25
2.4.3 Linear Programming	25
2.4.4 Quadratic Programming	26
2.4.5 Linear Matrix Inequalities (LMIs)	26
2.4.6 Genetic Algorithms	28
2.4.7 Evaluation Methodology “Goal Attainment”	29
2.5 Flexible-Structure Modeling	30
2.5.1 Nodal Models	30
2.5.2 Modal Form	31
2.5.3 Acceleration Sensors	32
2.6 Input / Output Selection	32

2.6.1	Information-based Criteria	34
2.6.2	Energy Criteria	34
2.6.3	Incorporation of Design Constraints by System Weighting	36
3	Optimal & Robust Control Design Methods	37
3.1	Standard problem formulation for \mathcal{H}_2 and \mathcal{H}_∞ feedback control design	37
3.1.1	\mathcal{H}_∞ weighting, stacking, and \mathcal{H}_∞ problem classes in standard form	37
3.1.2	Prerequisites for standard \mathcal{H}_2 and \mathcal{H}_∞ design	42
3.2	\mathcal{H}_∞ suboptimal controller design	43
3.3	\mathcal{H}_2 optimal and suboptimal controller design	44
3.4	LQ-based Optimal Control	46
3.4.1	Linear Quadratic Gaussian (LQG) Control	46
3.4.2	Output-weighted LQG Control Design	48
3.4.3	LQI Tracking Control Design	48
3.5	μ -Synthesis: DK-Iteration and DGK-Iteration Design	49
3.5.1	Complex perturbations & DK-Iteration	49
3.5.2	Real and complex perturbations & DGK-Iteration	50
3.6	Convex Controller Synthesis	51
3.6.1	Motivation & Overview	51
3.6.2	Initial controller design	53
3.6.3	Youla parametrization	54
3.6.4	Convex formulation of control design constraints and objectives	57
3.6.5	Selection of the filter basis	59
3.6.6	$Q - \mu$ -Synthesis for Robust Control Design	59
4	Flexible Aircraft Models	61
4.1	Introduction: Flight Dynamics and Control Design	61
4.2	Lateral Blended Wing Body (BWB) Aircraft Model	62
4.2.1	Open-Loop Analysis	63
4.2.2	Control Goals	65
4.3	Longitudinal Flexible Conventional Aircraft Model	65
4.3.1	Definition of Control Goals and Architecture	65
4.3.2	Scaling and Balanced Reduction	66
II	Novel Contributions	68
5	Extensions to Convex Control Design	69
5.1	Convex control design framework methodology	69
5.1.1	Adaptive constraint refinement	69
5.1.2	Object-Oriented Implementation of Framework	70
5.2	Strong Stabilization	72
5.2.1	Motivation & Introduction	73
5.2.2	Illustrating Examples	73
5.2.3	Strong Stabilization Constraint Formulation	79
5.2.4	Lyapunov Strong Stability Formulation	79
5.2.5	Heuristic Embedded Convex Region Constraint	80
5.2.6	Test results using a high-order model	84
5.3	Case Study 1: Strongly Stabilizing Lateral Feedback Controller for Flexible BWB Aircraft	86
5.3.1	Initial Feedback Control Design	86

5.3.2	Convex Feedback Controller Optimization	88
5.3.3	Results, Robustness & Discussion	88
5.3.4	Conclusions	89
5.4	Convex design of a multi-model scheduled feed-forward control law	89
5.4.1	Problem statement	90
5.4.2	Robustifying design algorithm	91
5.5	Case Study 2: Robust Scheduled Lateral Feed-Forward Controller for Flexible BWB Aircraft	92
5.5.1	Feed-Forward Control Goals and Control Architecture	92
5.5.2	Scheduling Concepts: Feed-Forward Multi-Model Control Design	92
5.5.3	Scheduling Results, Robustness & Discussion	93
6	Design Parameter Optimization	97
6.1	Introduction	97
6.1.1	Motivation for an Integrated Methodology	98
6.2	Methodology	98
6.2.1	Basic Idea	98
6.2.2	Generating the Design Plant	98
6.2.3	Generalized Design Plant Interconnection	99
6.2.4	Controller Synthesis	100
6.2.5	Controller Validation and Performance Evaluation	100
6.3	Formulating and Solving the Optimization Problem	101
6.3.1	Using Genetic Algorithms for Design Parameter Optimization	101
6.3.2	Implementation	102
6.4	Case Study 3: DPO of Lateral Control of Flexible BWB Aircraft	102
6.4.1	Lateral Feedback Control Architecture and Initial Design	103
6.4.2	DPO Formulation	104
6.4.3	DPO Results	105
6.5	Case Study 4: DPO of Longitudinal Control of Flexible Conventional Aircraft . .	108
6.6	Discussion & Conclusions	110
7	Extensions to Controller Order Reduction	112
7.1	Introduction	112
7.1.1	Literature Review of Controller Order Reduction Methods	112
7.2	Nominal Performance Frequency Weighting Derivation in $\mathbf{P} - \mathbf{K}$ - Form	113
7.3	Case Study: NP-Preserving Controller Order Reduction	116
7.4	Robust Performance Frequency Weighting Derivation: An Extension	118
7.4.1	Previous Results	118
7.4.2	Extension into an RP-preserving FWBR method	118
7.5	Case Study: RP-Preserving Controller Order Reduction	119
8	Gain Scheduling & Interpolation Studies	121
8.1	Overview	121
8.1.1	Gain Scheduling	121
8.1.2	Linear System Interpolation	122
8.2	LFR-based Scheduling of Compensators in Observer-based Representation	123
8.2.1	Interpolation of Full- and Augmented-Order Compensators	123
8.3	Stability Properties of Classic System Interpolation Methods	124
8.3.1	Problem statement	124
8.3.2	Companion form interpolation	125

8.3.3	Modal form interpolation	127
8.4	Geometric Algebra Interpolation	128
8.4.1	Basic Idea	128
8.4.2	Interpolation of the System Matrix	128
8.4.3	An onset for the interpolation of the B and C matrices	129
8.5	Example	130
8.6	Discussion	133
9	Summary and Outlook	135
9.1	Summary of the Results and Contributions	135
9.1.1	Extensions to Convex Control Design	135
9.1.2	Design Parameter Optimization	135
9.1.3	Controller Order Reduction	136
9.1.4	A Novel System Interpolation Onset	136
9.1.5	Results: Lateral Control for Large Flexible BWB Aircraft	136
9.2	An Outlook on Potential Future Research	137
III	Appendices	139
A	Linear Algebra Tools	140
A.1	Vector and Function Spaces	140
A.2	Some Matrix Properties	142
A.3	Singular Value Decomposition (SVD)	143
A.4	Norms	144
A.4.1	Vector- and Matrix Norms	144
A.4.2	Signal norms	146
A.4.3	System norms	146
A.5	Eigenvalues and Eigenvectors	149
A.5.1	Characteristic Polynomial	150
A.5.2	Theorem of Cayley-Hamilton	150
A.5.3	Multiple Eigenvalues/-vectors and their Computation	150
B	Observers & Observer-based Structures	151
B.1	Luenberger observer	151
B.2	State vector feedback control using a state observer	153
B.2.1	Separation principle	153
B.3	Observer-based Representation (OBR) of a feedback compensator	154
B.3.1	Transformation of Arbitrary Compensators to an OBR	155
C	LQ-based Lateral Control Designs for BWB Aircraft	158
C.1	Lateral LQG Control Design	158
C.1.1	LQG Architecture and Design Parameters	158
C.1.2	LQG Validation Results	159
C.2	Lateral LQI Control Design	160
C.2.1	Motivation	160
C.2.2	Design Parameters	160
C.2.3	LQI Results: Performance	161
C.2.4	LQI Results: Validation at various Mass Cases	161
C.3	Discussion	161
C.4	Conclusions	165

D Convex Control Design Framework Classes	166
D.1 Classes in the ConvexOptimizer Hierarchy	166
D.2 Classes in the Constraint/Objective Hierarchy	167
D.3 Auxiliary Classes	171
List of Figures	174
List of Tables	175

Glossary

Symbols

Notation	Description	Unit
$\mathbf{0}$	zero matrix	—
$\text{adj}(\cdot)$	adjoint of the matrix argument	—
\mathbb{C}	field of complex numbers	—
$\text{Conv}(\cdot)$	convex hull of the points given in the argument	—
$(\cdot)^\dagger$	Moore-Penrose pseudo inverse of the matrix	—
$\det(\cdot)$	determinant of the matrix argument	—
$\text{diag}(\cdot)$	diagonal matrix with the diagonal entries given by the argument	—
$\text{dom}(\cdot)$	domain of a function or space	—
$e, \exp(\cdot)$	Euler's number, exponential function	—
$E(\cdot)$	expected value operator	—
$\mathcal{F}_l(\cdot, \cdot)$	lower linear fractional transformation (LFT)	—
$\mathcal{F}_u(\cdot, \cdot)$	upper linear fractional transformation (LFT)	—
γ	bound on the \mathcal{H}_∞ norm of transfer functions	—
g_i	i th goal attainment ratio	—
$\text{grad}(\cdot)$	gradient operator	—
$(\cdot)^H$	complex-conjugate transpose	—
\mathbf{I}	identity matrix	—
$\Im(\cdot)$	imaginary part of the argument	—
j	imaginary unit	—
$\mathbf{\Lambda}$	eigenvalue matrix (diagonal or real-blockdiagonal)	—
$\lambda_i(\cdot)$	i th eigenvalue of the argument	—
$\mathcal{L}(\cdot)$	Laplace transform	—
$(\cdot)_m$	subscript for quantities in nodal coordinates	—
μ	structured singular value	—

Notation	Description	Unit
n_K	controller order	—
NL	subscript referring to quantities of the nonlinear system	—
$(\cdot)_n$	subscript for quantities in modal coordinates	—
n_ω	number of frequency points in a frequency gridding	—
n_Q	number of basis functions of the Youla parameter Q	—
n_{red}	reduced controller order	—
n_t	number of time points in a time-domain gridding	—
$\text{null}(\cdot)$	null space of the argument vectors	—
ω	angular frequency	rad/s
$\varphi_j(\cdot)$	j th interpolation weighting functions, activating functions	—
Q	state weighting matrix for LQ design	—
$\text{rank}(\cdot)$	matrix rank of the argument	—
$\Re(\cdot)$	real part of the argument	—
$(\cdot)_{\text{ref}}$	subscript for quantities related to reference signals	—
$\rho(\cdot)$	spectral radius of the argument	—
R	input weighting matrix for LQ design	—
\mathbb{R}	field of real numbers	—
s	Laplace variable	1/s
$\bar{\sigma}(\cdot)$	maximum singular value of the matrix argument	—
$\sigma_i(\cdot)$	i th singular value of the matrix argument	—
$\text{span}(\cdot)$	span of the argument vectors	—
T	non-singular state transformation matrix	—
t	time variable	s
θ	vector of weightings in the summation of the Youla parameter	—
Θ_{sstab}	set of parameters θ yielding a stable controller	—
$\text{trace}(\cdot)$	trace of the argument	—
$(\cdot)^T$	transpose	—
U	matrix of right eigenvectors	—
u	eigenvector	—
U	matrix of left singular vectors	—
v	eigenvector	—
V	process noise covariance matrix	—
V	matrix of desired input directions in eigenstructure assignment	—
V^H	matrix of right singular vectors	—
W_c	controllability Gramian	—
W	measurement noise covariance matrix	—
W	matrix of desired output directions in eigenstructure assignment	—
W_o	observability Gramian	—
x	vector of decision variables in an optimization problem	—

Notation	Description	Unit
Ξ	box-constrained (hypercube) set in the θ parameter space	–
X	solution of the controller algebraic Riccati equation	–
Y	solution of the filter algebraic Riccati equation	–

Signals & Systems

Notation	Description	Unit
\mathbf{A}	state-space system matrix	–
A_{IL}	aileron control surface	–
\mathbf{B}	input matrix	–
β	side slip angle	rad
\mathbf{C}	output matrix	–
$\hat{\mathbf{D}}$	uncertainty scaling matrix at a given frequency	–
\mathbf{D}	dynamic uncertainty scalings	–
\mathbf{D}	feed-through matrix	–
Δ	norm-bounded uncertainty matrix with or without structure	–
e	control error signal vector	–
\mathbf{G}	general or plant transfer function	–
\mathbf{K}	controller transfer function	–
\mathbf{K}_c	state vector feedback gain matrix	–
\mathbf{K}_f	observer gain matrix	–
\mathbf{L}	Kalman filter gain matrix	–
\mathbf{M}	nominal closed-loop transfer function	–
\mathbf{N}	perturbed closed-loop transfer function	–
$N_{y_{CG}}$	lateral acceleration at CG position	m/s ²
$N_{z_{CG}}$	vertical acceleration at CG position	m/s ²
$N_{z_{lat,law}}$	antisymmetric modal wing bending sensor signal	m/s ²
$N_{z_{law}}$	symmetric modal wing bending sensor signal	m/s ²
\mathbf{P}	plant transfer function	–
p	roll rate	rad/s
ϕ	roll angle	rad
\mathbf{Q}	Youla parameter transfer function	–
r	yaw rate	rad/s
ρ	scalar system parameter	–
$\boldsymbol{\rho}$	system parameter vector	–
RU	rudder control surface	–
\mathbf{S}	closed-loop sensitivity transfer function	–
\mathbf{S}_i	i th validation system	–
\mathbf{S}_{in}	input scaling matrix	–
\mathbf{S}_{out}	output scaling matrix	–

Notation	Description	Unit
T	closed-loop transfer function	–
u	control input signal vector	–
W	frequency-weighting transfer function	–
w	exogenous input signal vector	–
x	state vector	–
y	measurement output signal vector	–
y	exogenous performance output signal vector	–

Acronyms

Notation	Description
ARE	algebraic Riccati equation
BMI	Bi-Linear Matrix Inequality
BR	balanced reduction
BWB	blended wing body
CG	center of gravity
DC gain	static gain
DR mode	Dutch Roll mode
EFI	effective independence
FWBR	frequency-weighted balanced reduction
GA	Geometric Algebra
HSV	Hankel singular value
I/O	input/output
LFR	Linear Fractional Representation
LFT	Linear Fractional Transformation
LHP	left half plane
LMI	Linear Matrix Inequality
LP	linear program, linear programming
LPV	linear parameter-varying
LQ	Linear-Quadratic
LQG	Linear-Quadratic Gaussian
LQI	Linear-Quadratic Integral
LTI	linear time-invariant
MIMO	multi-input multi-output
NP	Nominal Performance
NS	Nominal Stability, Internal Stability
OBR	observer-based representation
PMI	Polynomial Matrix Inequality
QP	quadratic program, quadratic programming
RHP	right half plane
RP	Robust Performance
RS	Robust Stability

Notation	Description
SDP	semidefinite program/ programming
SISO	single-input single-output
SVD	Singular Value Decomposition

Part I

Introduction & State of the Art

Chapter 1

Introduction

1.1 Overview and Motivation

Robust and optimal control are widely applied disciplines with strong results and powerful numeric tools to meet the demands of challenging applications. This work is oriented particularly towards modern flight control designs for flexible aircraft. This field of applications requires a long, carefully built toolchain to support all involved tasks. These main tasks are dynamic system modeling, model manipulation and transformation, pre-processing for control design and the control design itself, followed by post-processing tasks such as validation, order-reduction and controller interpolation.

The demands for modern complex flight control design include not only the shaping of rigid-body flight dynamics, but also require the consideration of low-damped flexible vibration modes of the elastic lightweight aircraft structure. The control system should provide specified handling qualities for the pilot, but also reduce vibrations, maneuver-induced structural loads as well as those loads caused by wind gusts or turbulence. The control systems need to operate safely and effectively over a large range of the system's physical parameters, such as airspeed, altitude, or the aircraft's fuel or payload masses, which significantly influence its dynamic behavior. Finally, the physical modeling of aircraft dynamics is highly complex and differences between the design models and validation models or actual aircraft dynamics always exist, which results in challenging robustness requirements for the control laws.

Today, numerous reliable reference solutions and specific numeric tools exist to address most of these tasks. In view of their broad spectrum, their combined simultaneous treatment, however, unveils several vital gaps in the toolchain which need to be closed to fulfill the control system demands. In this work, some of these principal missing tools are therefore developed:

- For the convex control design (convex synthesis) methodology, an efficient and flexible, high-level optimization framework is developed and implemented. It serves as highly automated tool for the control engineer to formulate and solve the related optimization problems including their validation. The framework is extended to provide solutions to the challenging strong stabilization problem in feedback control design as well as to obtain robust scheduled feed-forward controllers via a multi-model design.
- A design parameter optimization framework is developed which enables efficient parameter search for complex control design problems. This way, the advantages of well-known design methods can be exploited and additional functionality such as validation results from nonlinear time-domain simulation can be directly integrated in the optimization process.
- Highly effective order reduction procedures for controllers are proposed which preserve

closed-loop nominal or robust performance. These are based on frequency-weighted balanced state reduction.

- System interpolation methods, relevant for modeling or controller scheduling, based on various canonical state-space representations are analyzed in terms of interpolation stability. Moreover, a novel state-space interpolation technique is sketched which exploits Geometric Algebra concepts and provides certain stability guarantees.

The effectiveness of these contributions is illustrated at numerous examples which range from specific academic problem settings to complex, large-scale flight control design problems. In particular, control designs for a large, flexible blended wing body (BWB) aircraft originating from the EU FP7 research project “ACFA 2020” (“Active Control for a Flexible 2020 Aircraft”) [1] are considered. They introduce additional challenges to conventional flight control design, for example due to open-loop instability and stringent performance requirements.

The main motivation for the underlying research is rooted both in theory and application: from a theoretical perspective, it is of interest to extend the scope of existing state-of-the-art methods and to address known problems in a novel, beneficial way. From an application point of view, it is advantageous to extend the applicability of existing methods and to remove or relax some of their limitations which otherwise may inhibit their use in a given relevant problem setting. Moreover, reliable algorithms, both conceptually and numerically, are needed to successfully and efficiently address challenging engineering problems.

This thesis provides some corresponding solutions which effectively support and integrate key tasks in the control design process for large, complex flexible structure systems, particularly for large flexible aircraft.

An overview on the most relevant literature related to the thesis’ topics is given in the following. Subsequently, the core contributions and the structure of the thesis are outlined in Sec. 1.3.

1.2 Literature Review, State of the Art

The control of multivariate systems is an extensive field of research. References to some important works are collected in the following to give a coarse overview on the topics treated in this thesis, but naturally this account cannot be complete. It should, however, point out the main sources that enable one to treat and develop the concepts that are elaborated herein. More detailed literature references are introduced throughout the text in each chapter or where appropriate to extend this picture.

1.2.1 Linear Algebra

Linear algebra forms the mathematical foundation of linear system and control theory. Of the many textbooks in the field, consider the following few references as a starting point: For a well-readable undergraduate text with a novel perspective, covering both standard linear algebra and geometric algebra fundamentals in a unified approach, refer to [92]. It concentrates on the conceptual understanding but, in turn, does not address numeric algorithms. On the contrary, the book [44] focuses entirely on the numeric aspects and algorithms arising in numeric linear algebra and is a well-known reference work in this regard. On the graduate level, the textbook [120] includes a thorough treatment of various linear algebra topics but requires a solid background in basic linear algebra.

1.2.2 Robust and Optimal Control

Well-known reference textbooks treating robust and optimal control topics have been written by Zhou et al. [174], Skogestad and Postlethwaite [143], and Glad and Ljung [41].

The excellent books [41] and [143] offer a well-readable treatment of multivariate linear (and some nonlinear) control theory as well as their practical application. Fundamental limitations in control are discussed extensively, modern optimal and robust control design methods are given ([143] now includes Linear Matrix Inequality (LMI) formulations of selected control problems) and illustrated via many academic and industrial examples. The target audience are graduate students, but also ambitious undergraduate students and control engineers and practitioners. In [174], special focus is laid on optimal and robust control methods in a more rigorous mathematical setting. It includes detailed material on control theory in high depth and is targeted at the graduate and post-graduate level from control engineering and/or applied mathematics background.

The Linear-Quadratic Gaussian (LQG) controller, a fairly old, central result in optimal control theory, is composed of a Kalman state estimator [45] and an optimal state vector feedback gain. While optimal in the nominal case, robustness could not directly be tuned or guaranteed. This paved the ground for \mathcal{H}_∞ -optimal control which provides suitable guarantees. Reference solutions to the \mathcal{H}_2 and \mathcal{H}_∞ control problems are given in [42] and [24].

The notion of robustness against a bounded and structured, but otherwise unknown set of perturbed plants is connected to \mathcal{H}_∞ control via the small gain theorem [21] and the quantity known as structured singular value μ [107]. By now, many tools to obtain numeric bounds on μ have been developed, but a direct and tractable computation of μ itself does not exist in the general cases. Hence the problem of finding a robust controller which obtains optimality in terms of a minimal μ , the so-called μ synthesis, is not solved. However, well-known approximations such as the DK-/DGK-iteration [6], [172] or the $Q - \mu$ -synthesis [18] are available and can lead to good results in practice. A textbook treating robust control design and analysis from an aerospace perspective is [7].

Optimization in Control

Some important problems in control or system analysis are unfortunately non-convex and sometimes non-smooth. Metaheuristic methods such as genetic algorithms are frequently employed in these cases (see [35]). However, also specific gradient-based search methods are often utilized such as in the \mathcal{H}_∞ fixed-order design toolbox HIFOO [50].

Luckily, many important control problems are in fact equivalent to convex optimization problems and thus can typically be solved efficiently. In the last decades, the convex LMI optimization problems rose to major importance because of their flexibility and because efficient solvers became available. The textbook [11] is directed towards control engineers and collects many LMI formulations of control design and analysis problems. A reference textbook on convex optimization with a more general perspective is [12] which gives a detailed account on properties and solution methods of convex optimization problems. One example of the versatile application areas of LMIs is the approximation of non-convex problems: a recent survey on LMI approximations to solve polynomial control problems is given in [16].

Convex control design (convex synthesis) onsets are shown in [11] and extended in [18] and [118] to obtain flight controllers for flexible aircraft. Thereby, frequency-domain (\mathcal{H}_∞ , \mathcal{H}_2) and time-domain constraint and objectives can be formulated and optimized for. The related feasibility problem, given a set of defined constraints, is discussed in [31]. In convex feedback control design, the closed loop is represented in Youla-parametrized form [171] which renders closed-loop transfer paths accessible to a convex problem formulation.

Scheduling of Control Laws

If the plant's dynamics is strongly parameter-dependent, which is the case in aircraft control, it is beneficial and sometimes inevitable to also parametrize the controller using online knowledge of these parameters, so that stability and performance demands can be met. Various forms of this "gain-scheduling" exist and can be categorized into linearization-based and quasi-LPV onsets as seen in the survey paper [122]. Linear Fractional Transformations (LFTs) can be utilized to express rational or polynomial parameter dependency and are also widely utilized in robust control design. One scheduling onset which utilizes a Linear Fractional Representation (LFR) of the plant model is found when representing the controller in an observer-based realization [3]. In [151], such scheduling is demonstrated for a flight controller designed for a flexible aircraft.

1.2.3 Flexible Structure Control

For flexible structure systems, active control concepts play an increasingly important role. With the rising demand for lightweight components and assemblies in many industrial areas active control concepts can help to meet the newly encountered challenges. These structures exhibit specific properties such as low-damped oscillatory modes and large compliance to loads which can be troublesome in typical operation cases, for example in terms of fatigue loads, large deformations, or comfort deterioration. The textbooks of Gawronski [39] and Preumont [117] treat control design specifically from the perspective of lightweight elastic structures and develop efficient methods that enable the treatment of large-scale problems.

At the beginning of the control design process for flexible structure systems, it is often possible to position or select the actuators or sensors that will be utilized for control. This choice significantly affects achievable control performance: a bad choice may impose severe fundamental limitations for the designed controller, whereas a good choice could circumvent these and thus enable significantly improved performance. A survey on the optimization in the input/output selection process is given in [152].

1.2.4 Flight Control Design

Fundamentals on classic control design for aircraft are given in the textbooks [13] (in German) and [145]. Recently, besides mastering the flight dynamics of rigid-body aircraft, a strong interest in the control of flexible aircraft is seen. The related, new challenges typically require modern control design methods, such as robust and/or scheduled multivariable designs. Control designs addressing both flight dynamics and aeroelasticity for flexible conventional aircraft have been studied in [79], [80], [140], [53], and [78]. More recently, an adaptive feed-forward controller for gust loads alleviation has been developed in [164] and combined with feedback approaches in [165].

In the strive for higher fuel efficiency and reduced emissions, novel concepts, particularly the transition to BWB aircraft configurations bear further potential [96]. However, these large light-weight flexible structures exhibit low-frequency elastic vibration modes, and coupling of those with the flight mechanic modes may occur. Moreover, the aircraft dynamics is significantly dependent on the flight parameters. Hence the task of developing robust and well-performing flight control laws faces significant challenges.

1.2.5 Further Research of the Author & Related Work

The author started working on robust control topics for flexible structure systems in 2007. An active vibration control concept for light-weight metro railcar bodies has been investigated and analyzed thoroughly [75], [125], [8], [132]. The work thereby focused on co-simulation studies for nonlinear control validation [126], [128], [127] which also included actuator positioning

optimization (input selection), as well as on control issues regarding the nonlinearities of the utilized piezo stack actuators [130] and their supporting structures [131]. Experimental validation of vibration control concepts at a scaled laboratory model of a metro railcar body has been performed in [114], [115].

The work group's research activity in the field of flight control for flexible aircraft started out with control designs for conventional flexible aircraft: The team designed a controller via DK-iteration for the hybrid control system presented in [165]; a 2DOF controller was designed by DK-iteration in [162] and optimized via design parameter optimization in [133] (see Sec. 6.5). Subsequently, BWB aircraft models became available within the ACFA2020 research project and methods for actuator and sensor positioning (often referred to as input / output selection) have been studied intensively, see [59]. An LQG-based closed-loop selection criterion is proposed in [60], energy-based selection criteria are proposed, applied, and compared to standard methods in [136], [158], and compared to information-based approaches in [161]. Recently, an efficient frequency-domain criterion has been formulated and successfully applied to BWB aircraft models in [61] and [62].

Initial LQ-based control designs for the flexible BWB aircraft model have been carried out in [134], see Chap. C. In [135], a DK-iteration controller for the same aircraft is designed and optimized via design parameter optimization, see Sec. 6.4. Convex lateral feedback and scheduled feedforward designs have been carried out in [138] (see Sec. 5.3) and [137] (see Sec. 5.5), respectively.

After a resizing of the BWB aircraft configuration, an integrated LPV control design [159] and a full-information \mathcal{H}_∞ scheduled feed-forward design [160] have been carried out on the new aircraft model, see the related doctorate thesis [157] which treats these control designs in great detail. A further study utilizing these models in a parametrized LFR form is given in [139] which reports on the encountered difficulties and found solutions with respect to DK-/DGK-iteration control design.

Finally, and most recently, the interpolation of systems are being studied and novel interpolation onsets are being investigated [129], see Chap. 8.

1.3 Contributions of this Work

This work documents the author's research results focused on developing, assessing, and extending various state-of-the-art and new methodologies for feedback and feed-forward control designs related to flight control. The core contributions close some important gaps in the complex toolchain of modeling, control design, and validation. An overview of these contributions, together with the relevant publications by the author is given in the following:

Extensions to Convex Control Design (Chap. 5): A novel, high-level optimization framework to formulate and carry out convex control design is developed and implemented in MATLAB[®] by the author. Time- and frequency-domain objectives and constraints as well as tools for both feedback and feed-forward control designs are implemented. Effective algorithms for an efficient, adaptive formulation of large-scale problems are developed. Utilizing this framework, several onsets are developed to obtain a strongly stabilizing (that is, stable and stabilizing) controller in the context of convex optimization. The method is successfully tested and applied to a feedback control design case study in which a lateral control design for a large flexible blended wing body aircraft is performed. These results are published in [138]. Moreover, an onset to perform robust scheduled feed-forward control design with multiple models is investigated and successfully applied in a subsequent case study to the same BWB aircraft model [137]. This application particularly benefits from the efficient problem formulation which retains the size of the actually solved

optimization problems sufficiently small for current solvers.

Design Parameter Optimization (Chap. 6): A design parameter optimization framework is developed, again in MATLAB[®]. Its goal is to freely combine well-defined, parametrized control design tasks and their validation with a suitable formulation of an optimization problem over these parameters. Two case studies demonstrate the effectiveness of this approach: the optimization of a longitudinal 2DOF control design for a flexible conventional aircraft (taken from [162]) is performed in [133], and the optimization of a lateral design for a flexible BWB aircraft model is shown in [135].

Extensions to Controller Order Reduction (Chap. 7): Frequency-weighted balanced state reduction forms the basis for two novel methods for controller order reduction. Firstly, a formulation is found which preserves nominal performance in the standard robust control feedback interconnection architecture. Secondly, a recent result for robust controller reduction is extended to the frequency-weighted case. Both novel methods show significantly improved reduction performance over the unweighted case at a flight control case study and at an example taken from literature.

Gain Scheduling & System Interpolation: A Novel Onset (Chap. 8): Several methods for the interpolation of linear dynamic systems, which are relevant for modeling as well as gain-scheduling, are reviewed with respect to their system stability properties. These significantly differ for the considered canonical system representations. Moreover, a novel state-space interpolation method is sketched which exploits Geometric Algebra concepts to consistently treat geometric relations in the interpolation onset. Moreover, it provides important local stability guarantees. This part is published by the author in [129].

This work should, besides transporting the main contributions, also serve as a reasonably self-contained guide through the control design process for the interested reader. Its limited length does not allow for a complete textbook-style treatment, but this thesis should represent a helpful starting point and succeed in explaining the main lines of research in robust and optimal control (LPV design methods are not treated, however).

1.4 Structure of this Work

The thesis is organized as follows: an extensive chapter on Fundamentals is given (Chap. 2), including basics on system formulation and processing (linearization, state-space definitions, balanced reduction), robust control fundamentals (robust stability and performance, μ calculus, uncertainty modeling) and an overview on some optimization problems and related tools. The key aspects of the modeling of flexible structures for control, as well as the related task of input (actuator) / output (sensor) selection is provided. The author's research focus is tied to this type of control problems as can be seen by the chosen case studies and examples. Moreover, state-of-the-art optimal and robust control design methods are summarized in Chap. 3, including \mathcal{H}_∞ -, \mathcal{H}_2 - and LQG-control formulations and their relationships, and recent convex control design methods and their formulation are provided.

The flexible aircraft models utilized in most case studies — a flexible conventional transport aircraft as well as a flexible BWB aircraft model — are introduced with a short characterization of their main properties in Chap. 4.

The main contributions are given in Chapters 5 (extensions to convex control design), 6 (design parameter optimization), 7 (controller order reduction), and 8 (interpolation studies).

A discussion, conclusions, and an outlook on further studies is given in Chap. 9. Finally, additional supplementary material is included in the appendices: Fundamentals of linear algebra

and linear system theory are summarized in a compact form as a quick reference in Chap. A. Moreover, a summary of observers and the observer-based realization of compensators is given from according literature in Chap. B. LQ-based control designs for the introduced BWB aircraft model are given in Chap. C. A listing of all classes of the developed convex control design framework implementation with their properties and methods is provided in Chap. D.

Chapter 2

Fundamentals

This chapter gives an overview on major, fundamental concepts in linear control design: the state space representation, the robust control design problem and its basic terminology, as well as uncertainty modeling and basics in optimization. A quick reference of relevant topics from linear algebra are collected in the Appendix of this work.

The exposition is oriented at main results from standard textbooks in linear control design (such as [143] and [174]) and optimization in control, see [12]. Specific references are given in the text.

2.1 State Space Representation

2.1.1 General State Space Systems & Linearizations

The dynamics of many general nonlinear dynamic systems can be written as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}(t)) & \mathbf{x}(t) \dots (n \times 1) \text{ state vector} \\ \mathbf{y}(t) &= \mathbf{G}(\mathbf{x}(t), \mathbf{u}(t)) & \mathbf{u}(t) \dots (n_u \times 1) \text{ input vector} \\ & & \mathbf{y}(t) \dots (n_y \times 1) \text{ output vector} \\ & & \boldsymbol{\theta}(t) \dots (n_\rho \times 1) \text{ parameter vector.} \end{aligned} \quad (2.1)$$

The functions $\mathbf{F} : \mathbb{R}^{n \times n_u \times n_\rho} \mapsto \mathbb{R}^n$ and $\mathbf{G} : \mathbb{R}^{(n \times n_u)} \mapsto \mathbb{R}^{n_y}$ are generally non-linear and in this work assumed to be continuously differentiable. This enables one to apply Taylor / Jacobian linearization at any fixed (stationary) equilibrium point $(\mathbf{x}_0, \mathbf{u}_0, \boldsymbol{\theta}_0)$ where

$$\dot{\mathbf{x}}|_0 = \mathbf{F}(\mathbf{x}_0, \mathbf{u}_0, \boldsymbol{\theta}_0) = \mathbf{0}. \quad (2.2)$$

The chosen tuple of system state \mathbf{x}_0 , input \mathbf{u}_0 , and parameter values $\boldsymbol{\theta}_0$ fulfilling (2.2) is frequently called operating point (used in the following), linearization point, stationary point, or equilibrium point.

Remarks:

- It is also possible to perform off-equilibrium linearizations about a trajectory $(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t), \tilde{\boldsymbol{\theta}}(t))$ in which $\dot{\tilde{\mathbf{x}}} = \mathbf{F}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t), \tilde{\boldsymbol{\theta}}(t)) \neq \mathbf{0}$. Then, the linearized dynamics can be interpreted as error dynamics for small perturbations of the system off this trajectory. Time-invariant and time-dependent linearizations along system trajectories are investigated in [14]. However, only linearizations at stationary operating points will be considered in this work.
- A formal condition based on Lie algebra for the existence of a locally valid linearization of non-linear systems is given in [77].

The linearized state-space system can be formulated by expanding (2.1) into its first-order Taylor approximation about the operating point:

$$\dot{\mathbf{x}}(t) \doteq \underbrace{\mathbf{F}(\mathbf{x}_0, \mathbf{u}_0, \boldsymbol{\theta}_0)}_{\dot{\mathbf{x}}_0=0} + \left. \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right|_0 (\mathbf{x}(t) - \mathbf{x}_0) + \left. \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \right|_0 (\mathbf{u}(t) - \mathbf{u}_0) + \left. \frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}} \right|_0 (\boldsymbol{\theta}(t) - \boldsymbol{\theta}_0) \quad (2.3)$$

$$\mathbf{y}(t) \doteq \underbrace{\mathbf{G}(\mathbf{x}_0, \mathbf{u}_0)}_{\mathbf{y}_0} + \left. \frac{\partial \mathbf{G}}{\partial \mathbf{x}} \right|_0 (\mathbf{x}(t) - \mathbf{x}_0) + \left. \frac{\partial \mathbf{G}}{\partial \mathbf{u}} \right|_0 (\mathbf{u}(t) - \mathbf{u}_0) \quad (2.4)$$

Note that the partial derivative values are constant for a fixed operating point, so they can be represented as constant matrices:

$$\mathbf{A} = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right|_0 = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{array} \right]_0 \quad \mathbf{B} = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_{n_u}} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_{n_u}} \end{array} \right]_0 \quad (2.5)$$

$$\mathbf{C} = \left[\begin{array}{ccc} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_{n_y}}{\partial x_1} & \cdots & \frac{\partial g_{n_y}}{\partial x_n} \end{array} \right]_0 \quad \mathbf{D} = \left[\begin{array}{ccc} \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_{n_u}} \\ \vdots & & \vdots \\ \frac{\partial g_{n_y}}{\partial u_1} & \cdots & \frac{\partial g_{n_y}}{\partial u_{n_u}} \end{array} \right]_0, \quad (2.6)$$

where f_i , g_i are the i th scalar functions in \mathbf{F} , \mathbf{G} , respectively, and \mathbf{A} is the Jacobian of the system. Its eigenvalues $\boldsymbol{\lambda}$ and eigenvectors $\boldsymbol{\Phi}$ describe the homogenous eigendynamics of the system close to the operating point $(\mathbf{x}_0, \mathbf{u}_0, \boldsymbol{\theta}_0)$. This local dynamics can be stable, corresponding to a stable operating point of the non-linear system, or it can be unstable, which implies that the operating point is locally unstable. Especially in the latter case one has to bear in mind that the linearization is generally only valid in a small neighborhood of the operating point.

The partial derivative with respect to the parameter vector $\frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}}$ is usually either omitted (by assuming operation at the parameter values used for linearization, i.e. $\boldsymbol{\theta} = \boldsymbol{\theta}_0$) or seen as additional system states or inputs and therefore merged into \mathbf{A} or \mathbf{B} , respectively [168].

The linearized system and output equations can be stated concisely as

$$\begin{aligned} \Delta \dot{\mathbf{x}}(t) &= \mathbf{A} \Delta \mathbf{x}(t) + \mathbf{B} \Delta \mathbf{u}(t) \\ \Delta \mathbf{y}(t) &= \mathbf{C} \Delta \mathbf{x}(t) + \mathbf{D} \Delta \mathbf{u}(t), \end{aligned} \quad (2.7)$$

where $\Delta \mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_0$, $\Delta \mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}_0$, and $\Delta \dot{\mathbf{x}}(t) = \dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_0$ are the *deviation variables*.

2.1.2 Transfer Function Representation, Proper Systems

When applying the Laplace transform to (2.7) one obtains the input/output-equivalent multi-input multi-output (MIMO) transfer function matrix

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{D}, \quad \Delta \mathbf{y}(s) = \mathbf{G}(s) \Delta \mathbf{u}(s). \quad (2.8)$$

For each pair of input j and output k , a single-input single-output (SISO) transfer function is defined as

$$G_{kj} = [c_{k1}, \dots, c_{kn}] (s\mathbf{I} - \mathbf{A})^{-1} \begin{bmatrix} b_{1j} \\ \vdots \\ b_{nj} \end{bmatrix} + d_{kj} = \frac{B(s)}{A(s)}, \quad (2.9)$$

where $B(s)$ and $A(s)$ are the numerator and the denominator polynomials of order m and n , respectively. While transfer functions can also be defined for $m > n$, causal systems and in particular state-space systems always yield $m \leq n$.

Definition 2.1.1: Proper and bi-proper systems, SISO case

- A system $G(s)$ is *strictly proper* if $G(j\omega) \rightarrow 0$ as $\omega \rightarrow \infty$. It can be represented as state-space system and has no feedthrough term.
- A system $G(s)$ is *bi-proper* or *semi-proper* if $G(j\omega) \rightarrow d, d \in \mathbb{R} \setminus \{-\infty, 0, \infty\}$ as $\omega \rightarrow \infty$. It can be represented as state-space system with finite, nonzero feedthrough term d .
- A system $G(s)$ which is strictly proper or bi-proper is *proper*.
- A system $G(s)$ is *improper* if $G(j\omega) \rightarrow \infty$ as $\omega \rightarrow \infty$. Improper systems cannot be represented as state-space systems.

Def. 2.1.1 directly extends to MIMO systems. State space systems are always proper. If $\mathbf{D} \neq \mathbf{0}$, they are bi-proper, otherwise they are strictly proper.

Example 2.1.1: Interpretation of non-zero feedthrough matrix \mathbf{D}

Given a flexible mechanical structure, its dynamics from force excitation to displacement or velocities of structure points can be modeled [39]. If the outputs are linear combinations of the states only (i.e. displacements or velocities), and all degrees of freedom of the system are modeled (or unmodeled dynamics is truncated), the resulting state-space system has no feed-through term $\mathbf{D} = \mathbf{0}$. However, if accelerations are formulated as output signals, feedthrough terms appear that essentially capture the instantaneous collocation feedthrough of force inputs to the collocated acceleration measurements. Also, non-zero \mathbf{D} -entries arise when the system dynamics is reduced and the DC reduction errors are compensated (residualization, see Sec. 2.2 and [143]).

2.1.3 Gramians

Definition 2.1.2: Gramians of state-space systems [97]

Consider a set of continuous, real-valued functions

$$\{f_i, i = 1, \dots, n\} \quad (2.10)$$

which are defined on an interval $[t_0, t_f]$. With the standard inner product of functions, the Gramian $\mathbf{W} = [W_{ij}]$ of (2.10) is defined as

$$W_{ij} = \int_{t_0}^{t_f} f_i(\tau) f_j(\tau) d\tau, \quad (2.11)$$

which is a symmetric, positive-semidefinite matrix.

For a stable state-space system with matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$, the Controllability Gramian \mathbf{W}_c is defined as

$$\mathbf{W}_c = \int_0^{\infty} e^{\mathbf{A}t} \mathbf{B} \mathbf{B}^T e^{\mathbf{A}^T t} dt \quad (2.12)$$

and is simultaneously the unique symmetric positive-semidefinite solution to the Lyapunov equation

$$\mathbf{A} \mathbf{W}_c + \mathbf{W}_c \mathbf{A}^T + \mathbf{B} \mathbf{B}^T = \mathbf{0}. \quad (2.13)$$

Analogously, the Observability Gramian \mathbf{W}_o is defined as

$$\mathbf{W}_o = \int_0^{\infty} e^{\mathbf{A}^T t} \mathbf{C}^T \mathbf{C} e^{\mathbf{A}t} dt \quad (2.14)$$

and is the unique symmetric positive-semidefinite solution to the Lyapunov equation

$$\mathbf{A}^T \mathbf{W}_o + \mathbf{W}_o \mathbf{A} + \mathbf{C}^T \mathbf{C} = \mathbf{0}. \quad (2.15)$$

The eigenstructure of these Gramians quantify state controllability and observability, respectively, as illustrated in [143].

2.2 Balanced Reduction

One widely applied standard method for system order reduction of linear time-invariant (LTI) state-space systems is the balanced state reduction (truncation or residualization, see [143], [174]) which is closely connected to principal component analysis [97]. An error bound for the unweighted case and an extension by frequency-domain weightings has been developed in [26]. Later, a error bounds for the one- and two-sided weighted cases have been found [173],[73], [144], and an extension ensuring stability in the case of two-sided frequency weightings is proposed in [155].

The basic onset for balanced reduction is as follows: The system is first transformed (by a state transformation) into the so-called balanced realization in which the Controllability and the Observability Gramians \mathbf{W}_c and \mathbf{W}_o (see Def. 2.1.2) are equal and diagonal. Their diagonal

entries are the so-called Hankel singular values (HSVs) σ_i ($i = 1, \dots, n$) which directly reflect the importance of the corresponding new state variables in the system input-output behavior:

$$\mathbf{W}_c = \mathbf{W}_o = \text{diag}(\sigma_i), \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0. \quad (2.16)$$

Then, a state partitioning of these balanced states,

$$\mathbf{x}_{b1} = \begin{bmatrix} x_{b,1} \\ \vdots \\ x_{b,k} \end{bmatrix}, \quad \mathbf{x}_{b2} = \begin{bmatrix} x_{b,k+1} \\ \vdots \\ x_{b,n} \end{bmatrix} \quad (2.17)$$

is chosen and all state-space matrices of the balanced realization $\mathbf{G}_{\text{bal}} = \left[\begin{array}{c|c} \mathbf{A}_b & \mathbf{B}_b \\ \hline \mathbf{C}_b & \mathbf{D}_b \end{array} \right]$ with $\mathbf{D} = \mathbf{D}_b$) are partitioned accordingly:

$$\begin{bmatrix} \dot{\mathbf{x}}_{b1} \\ \dot{\mathbf{x}}_{b2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{b11} & \mathbf{A}_{b12} \\ \mathbf{A}_{b21} & \mathbf{A}_{b22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{b1} \\ \mathbf{x}_{b2} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{b1} \\ \mathbf{B}_{b2} \end{bmatrix} \mathbf{u} \quad (2.18)$$

$$\mathbf{y} = \left[\mathbf{C}_{b1} \mid \mathbf{C}_{b2} \right] \begin{bmatrix} \mathbf{x}_{b1} \\ \mathbf{x}_{b2} \end{bmatrix} + \mathbf{D}_b \mathbf{u} \quad (2.19)$$

Those partitions corresponding to \mathbf{x}_{b1} are preserved while those corresponding to \mathbf{x}_{b2} are removed. Depending on how this removal is performed, two main variants are distinguished:

Balanced Truncation: Merely disregarding the blocks associated to \mathbf{x}_{b2} is called balanced truncation and yields the reduced system

$$\mathbf{G}_{\text{trunc}} = \left[\begin{array}{c|c} \mathbf{A}_{b11} & \mathbf{B}_{b1} \\ \hline \mathbf{C}_{b1} & \mathbf{D}_b \end{array} \right] \quad (2.20)$$

with the following properties:

- The DC gains of \mathbf{G} and $\mathbf{G}_{\text{trunc}}$ differ in general.
- The high-frequency behavior in terms of the limit for infinite frequency is the same:

$$\lim_{\omega \rightarrow \infty} \mathbf{G}(j\omega) = \lim_{\omega \rightarrow \infty} \mathbf{G}_{\text{trunc}}(j\omega) \quad (2.21)$$

Balanced Residualization: Another reduction variant is called balanced residualization in which the static solution for \mathbf{x}_{b2} is assumed to hold. By setting $\dot{\mathbf{x}}_{b2} = \mathbf{0}$, these solutions can be obtained algebraically if \mathbf{A}_{b22} is nonsingular as $\mathbf{x}_{b2,\text{static}} = -\mathbf{A}_{b22}^{-1} \mathbf{A}_{b21} \mathbf{x}_{b1} - \mathbf{A}_{b22}^{-1} \mathbf{B}_{b2} \mathbf{u}$ and inserted into the system equation (2.18) for $\dot{\mathbf{x}}_{b1}$ and into the output equation (2.19). The residualized reduced system matrices are thus

$$\mathbf{G}_{\text{resid}} = \left[\begin{array}{c|c} \mathbf{A}_{b11} - \mathbf{A}_{b12} \mathbf{A}_{b22}^{-1} \mathbf{A}_{b21} & \mathbf{B}_{b1} - \mathbf{A}_{b12} \mathbf{A}_{b22}^{-1} \mathbf{B}_{b2} \\ \hline \mathbf{C}_{b1} - \mathbf{C}_{b2} \mathbf{A}_{b22}^{-1} \mathbf{A}_{b21} & \mathbf{D}_b - \mathbf{C}_{b2} \mathbf{A}_{b22}^{-1} \mathbf{B}_{b2} \end{array} \right], \quad (2.22)$$

and the system shows these properties:

- The DC gains of \mathbf{G} and $\mathbf{G}_{\text{resid}}$ are equal.
- The high-frequency behavior of \mathbf{G} and $\mathbf{G}_{\text{resid}}$ are different, particularly, $\mathbf{G}_{\text{resid}}$ is generally bi-proper.

This procedure is motivated by the fact that the \mathcal{H}_∞ error between the original system $\mathbf{G} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array} \right]$ and the reduced system \mathbf{G}_{red} (for either variant, $\mathbf{G}_{\text{trunc}}$ or $\mathbf{G}_{\text{resid}}$),

$$\|\mathbf{G} - \mathbf{G}_{\text{red}}\|_\infty \quad (2.23)$$

should be kept small.

In fact, in this unweighted case, an analytic reduction error bound for both, truncation and residualization, is known [26], [143]:

$$\|\mathbf{G} - \mathbf{G}_{\text{red}}\|_\infty \leq 2 \sum_{j=k+1}^n \sigma_j \quad (2.24)$$

where \mathbf{G} is the original system, \mathbf{G}_{red} is the reduced system, and σ_j is the j th HSV of \mathbf{G} .

Remark: To preserve the global system dynamics characteristics, oscillatory pole pairs must not be split by the reduction partitioning. This is eased by using well-scaled systems, because then the two balanced states that correspond to a low-damped pole pair also typically have closely lying HSVs (compare the discussion on the almost-balanced form in [39]).

2.2.1 Frequency-Weighted Balanced Reduction (FWBR)

The (unweighted) balanced reduction is suitable to keep the related unweighted \mathcal{H}_∞ reduction error (2.24) small. A generalization is provided in [26] by the so-called frequency-weighted balanced reduction (FWBR) based on a frequency-weighted error measure

$$\varepsilon := \|\mathbf{W}_{\text{out}}(\mathbf{G} - \mathbf{G}_{\text{red}})\mathbf{W}_{\text{in}}\|_\infty. \quad (2.25)$$

Thereby, \mathbf{W}_{out} and \mathbf{W}_{in} are frequency-weighting functions which have to be

- stable and
- stably invertible (which implies that they have to be square, minimum-phase, and bi-proper with full-rank feedthrough matrix).

Although these restrictions do limit the choice of weightings, the FWBR approach is a powerful tool in practice if the error measure of interest requires shaping in the frequency domain. Novel derivations of FWBR weightings preserve various closed-loop performance measures when performing order reduction of a feedback controller are derived and demonstrated in Chap. 7.

The obtained solution is not necessarily optimal in the sense of a minimal weighted error ε , and unfortunately no analytic error bounds as in the unweighted case are known. However, improved reduction performance (in terms of small weighted actual reduction error) is observed in many well-conditioned problems.

The FWBR algorithm utilizes frequency-weighted Gramians and has been described in [26] and later improved in accuracy by the authors of [153]. Today, the MATLAB[®] command `balancmr` (or, alternatively, `reduce`) provides FWBR functionality via the `Weights` parameter.

2.3 Robust Control Fundamentals

Robust control can conceptually be split into robustness analysis and robust controller synthesis. However, the synthesis procedures utilize robustness analysis tools and properties throughout.

Robustness analysis studies the consequences of system perturbations on the dynamic characteristics (stability and performance measures). Robust controller synthesis procedures aim to generate control laws that achieve favorable system properties (again, in terms of stability and performance) despite the presence of bounded system perturbations. Exploiting structural information on these perturbations is the key to designing well-performing robust controllers. The relevant robust control terminology is summarized in the following and covers the basic uncertain system representation and robustness quantities. For a detailed treatment, refer to [143] or [174].

2.3.1 Robustness Analysis

This section collects the main results on robustness analysis, that is, the analysis of systems that are subject to bounded and specifically structured perturbations. The reader should pay special attention to the quantities defined in the following. Corresponding comments point at important details, for example illustrating the relation of system quantities and quantities related to (complex) matrices.

Small Gain Theorem

The small gain theorem in the following two formulations (based on [143]) is a very general stability condition for feedback loops of linear systems. It is a fundamental building block in robust control theory and forms the basis for the definitions of robust stability, robust performance, and the structured singular value μ (see the following sections). A more general version of the small gain theorem is given in [21] which is applicable to the interconnection of nonlinear and/or distributed systems.

Definition 2.3.1: Spectral radius stability condition

Given a stable open-loop transfer function $\mathbf{L}(s)$, the closed-loop system is stable if

$$\rho(\mathbf{L}(j\omega)) < 1 \quad \forall \omega, \quad (2.26)$$

where ρ is the spectral radius as defined in Def. A.4.5.

The following formulation utilizes the fact that the spectral radius is a lower bound for all matrix norms (see (A.14)).

Definition 2.3.2: Small gain theorem

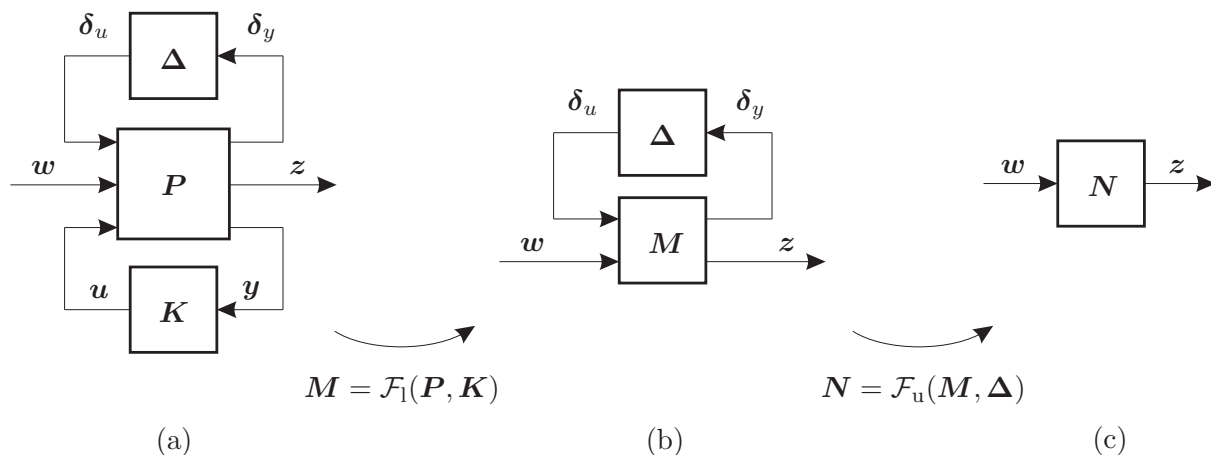
Given a stable open-loop transfer function $\mathbf{L}(s)$, the closed-loop system is stable if

$$\|\mathbf{L}(j\omega)\| < 1 \quad \forall \omega, \quad (2.27)$$

where $\|\cdot\|$ is any matrix norm (see Def. A.4.3).

Remarks:

- Note that the small gain theorem (2.27) is generally more conservative than the spectral radius condition (2.26). Both do not consider phase information, thus they are independent on the sign of $\mathbf{L}(s)$.

Figure 2.1: The $P - K - \Delta$ - form and LFTs

- Due to the validity of the multiplicative property (A.10), the small gain theorem can also be extended to loop transfer functions composed of several blocks: If $L = L_1 L_2$ is stable, then the closed-loop is stable if $\|L_1\| \|L_2\| < 1 \quad \forall \omega$. This important property is utilized to derive the robust stability condition in the following.

LFTs and the $P - K - \Delta$ - Form

A standard formulation of a system interconnection structure in robust control theory is the normalized $P - K - \Delta$ -form as in Fig. 2.1(a). A dynamic MIMO system P takes the input vector signals δ_u , w (exogenous or disturbance inputs), and u (control inputs) and generates the output vector signals δ_y , z (exogenous or performance outputs), and y (measured outputs). Closing the lower feedback loop with the controller K yields the nominal closed loop M . The upper feedback loop models system perturbation: The artificial outputs δ_y and inputs δ_u connect the uncertainty matrix Δ with the plant, whereas Δ is only known in terms of its qualitative structure (block-diagonal, complex- or real-valued LTI dynamics) and its norm bound $\|\Delta\|_\infty \leq 1$.

$$\Delta \in \Delta_B \Leftrightarrow \|\Delta\|_\infty \leq 1, \Delta \text{ structured.} \quad (2.28)$$

The feedback interconnections are formalized via lower and upper Linear Fractional Transformations (LFTs) \mathcal{F}_l and \mathcal{F}_u , respectively [143]:

$$M = \mathcal{F}_l(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}, \quad (2.29)$$

$$N = \mathcal{F}_u(M, \Delta) = M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12}. \quad (2.30)$$

The term performance path is frequently used for the transfer function from w to z .

The Structured Singular Value μ

The structured singular value μ is a generalization of the maximum singular value $\bar{\sigma}$, see [107].

Definition 2.3.3: Structured Singular Value μ

Consider the system interconnection in Fig. 2.1 where system M_{11} is the partitioning of M with inputs δ_u and outputs δ_y . Evaluating M_{11} at $s = j\omega$, the structured singular value μ of the complex-valued matrix $M_{11}(j\omega)$ is the nonnegative real scalar

$$\mu_{\Delta}(M_{11}(j\omega)) = \frac{1}{\min_{\Delta \in \Delta_B} \{k_m : \det(\mathbf{I} - k_m M_{11}(j\omega)\Delta) = 0\}}, \quad (2.31)$$

where $\Delta \in \Delta_B$ is structured in a known block-diagonal real or complex form and satisfies the norm bound $\bar{\sigma}(\Delta) \leq 1$. [143, 174].

Note that the structured singular value μ is a quantity formulated upon the complex transfer matrix of the system at a particular complex frequency. Sometimes it may be of interest to evaluate μ at other complex frequencies, e.g. at the unit circle (discrete-time case), or at more restrictive stability regions (relative or absolute stability).

Nominal Stability (NS) & Nominal Performance (NP)

The internal stability of the nominal closed-loop system M in Fig. 2.1 is a central requirement. In the context of robust control this property is called Nominal Stability (NS):

Definition 2.3.4: Nominal Stability (NS)

Consider a system M as in Fig. 2.1. It is nominally stable (NS) if it is internally stable, that is, if none of its components contain unstable hidden poles and if the injection of bounded input signals at any place in the system result in bounded output signals measured anywhere in the system. [143]

Utilizing the standardized $P-K-\Delta$ -form, a canonical way to formulate the desired system response in terms of the (closed-loop) transfer function from exogenous inputs (disturbances) to exogenous outputs is as follows¹:

Definition 2.3.5: Nominal performance (NP)

Consider an LTI system as in Fig. 2.1 and the case $\Delta = \mathbf{0}$. The system is said to achieve nominal performance if it is nominally stable (NS) and if

$$\bar{\sigma}(M_{22}(j\omega)) < 1 \quad \forall \omega \quad (2.32)$$

holds. Equivalently, the system achieves Nominal Performance (NP) if it is Nominal Stability, Internal Stability (NS) and

$$\|M_{22}\|_{\infty} < 1 \quad (2.33)$$

holds.

¹Other formulations exist in robust control design which define performance upon specific transfer functions, for example weighted sensitivity or complementary sensitivity functions. The NP formulation presented here is hinged on the performance path of the system and thus can depict a wide range of closed-loop transfer functions by appropriate structure of the augmented design plant P .

Note that performance formulations as in Def. 2.3.5 can be obtained for any frequency-domain specification of the form

$$\bar{\sigma}(\mathbf{G}(j\omega)) < \bar{\sigma}(\mathbf{W}(j\omega)) \quad \forall \omega \quad (2.34)$$

for stable transfer functions $\mathbf{G}(s)$ and $\mathbf{W}(s)$ where $\mathbf{W}(s)$ is invertible and the product $\mathbf{G}\mathbf{W}^{-1}$ exists. At each fixed ω , (2.34) is an inequality of two spectral matrix norms:

$$\|\mathbf{G}(j\omega)\|_{i_2} < \|\mathbf{W}(j\omega)\|_{i_2} \quad (2.35)$$

$$\Leftrightarrow \|\mathbf{G}(j\omega)\|_{i_2} \frac{1}{\|\mathbf{W}(j\omega)\|_{i_2}} < 1 \quad (2.36)$$

$$\Leftrightarrow \|\mathbf{G}(j\omega)\|_{i_2} \|\mathbf{W}^{-1}(j\omega)\|_{i_2} < 1. \quad (2.37)$$

Using the multiplicative property a statement can be made on the norm of the composed system (with some conservativeness):

$$\|\mathbf{G}(j\omega)\mathbf{W}^{-1}(j\omega)\|_{i_2} \leq \|\mathbf{G}(j\omega)\|_{i_2} \|\mathbf{W}^{-1}(j\omega)\|_{i_2} < 1. \quad (2.38)$$

Because (2.38) is valid for all values of ω , the equivalent system norm expression can be given the \mathcal{H}_∞ interpretation

$$\|\mathbf{G}\mathbf{W}^{-1}\|_\infty < 1. \quad (2.39)$$

Summing up, the frequency-domain requirement (2.34), which means that the singular value magnitude of \mathbf{G} is bounded above by the singular values magnitude of \mathbf{W} , is fulfilled if (2.39), the corresponding nominal performance formulation, is fulfilled.

Robust Stability (RS) and Robust Performance (RP)

In the presence of plant uncertainty $\Delta \neq \mathbf{0}$, the stability and performance specifications need to be fulfilled for all possible perturbations. These properties are called Robust Stability (RS) and Robust Performance (RP) of the uncertain system, respectively.

Definition 2.3.6: Robust Stability (RS)

Consider the uncertain closed-loop system $\mathbf{N} = \mathcal{F}_u(\mathbf{M}, \Delta)$ as in Fig. 2.1 where \mathbf{M} is the nominal closed loop and $\Delta \in \Delta_B$ is a stable, structured, uncertain matrix with unit norm bound $\|\Delta\|_\infty \leq 1$.

Then, Robust Stability (RS) is obtained if

$$\mathbf{M} \text{ is NS and } \mathbf{N} \text{ is stable } \forall \Delta \in \Delta_B \quad (2.40)$$

$$\Leftrightarrow \mathbf{M} \text{ is NS and } \mu_\Delta(\mathbf{M}_{11}(j\omega)) < 1 \quad \forall \omega. \quad (2.41)$$

Note that NS is a prerequisite and must be verified separately because the frequency-wise evaluation of μ does not contain this information.

Definition 2.3.7: Robust Performance (RP)

Given an uncertain closed-loop system \mathbf{N} as above, Robust Performance (RP) is obtained if

$$\mathbf{M} \text{ is NS and } \|\mathbf{N}\|_\infty < 1 \quad \forall \mathbf{\Delta} \in \mathbf{\Delta}_B \quad (2.42)$$

$$\Leftrightarrow \mathbf{M} \text{ is NS and } \mu_{\hat{\Delta}}(\mathbf{M}(j\omega)) < 1 \quad \forall \omega, \quad (2.43)$$

where $\hat{\Delta} = \text{diag}(\mathbf{\Delta}, \mathbf{\Delta}_{\text{perf}})$, and $\mathbf{\Delta}_{\text{perf}}$ is a full complex block compatible with the dimensions of \mathbf{w} and \mathbf{z} fulfilling $\|\mathbf{\Delta}_{\text{perf}}\|_\infty < 1$. Note that the block-diagonal extension of the uncertainty block by a full-complex “performance block” renders the problem of evaluating RP structurally equivalent to that of evaluating RS, thus the same computational tools can be used. Still, NS has to be verified separately.

These conditions, and with that the fulfilment of RS or RP, are proven or negated via a μ -analysis (and the verification of NS). Note that the RP condition effectively formulates whether the closed-loop transfer function from \mathbf{w} to \mathbf{z} is less than 1 in magnitude for all frequencies and $\mathbf{\Delta}$. Thus it reflects the robust fulfilment of appropriately scaled control performance objectives. Careful formulation of these objectives is a crucial part of the design.

Worst-Case Gain & Skew μ Value

One should be careful in the interpretation of the RP μ value. It does not, as one might expect, state the worst-case gain of the uncertain closed-loop system \mathbf{N} in Fig. 2.1 in the presence of the uncertainty $\mathbf{\Delta}$. Instead, reconsidering the definition of μ it follows that the RP μ value considers that **both** uncertainty blocks, $\mathbf{\Delta}$ and $\mathbf{\Delta}_{\text{perf}}$, are scaled by the common factor $k_m = \frac{1}{\mu_{\hat{\Delta}}(\mathbf{M}(j\omega))}$.

First, consider the case $\mu_{\text{RP}}(\mathbf{N}) := \sup_\omega \mu_{\hat{\Delta}}(\mathbf{M}(j\omega)) < 1$ and NS, so that RP is fulfilled. Then, $k_m > 1$ and a critical frequency ω_{crit} and a critical perturbation $k_m \hat{\Delta}_{\text{crit}}$, $\|\hat{\Delta}_{\text{crit}}\|_\infty$ exists at which the system is at the stability boundary ($\det(\mathbf{I} - k_m \mathbf{M}_{11}(j\omega_{\text{crit}}) \hat{\Delta}_{\text{crit}}) = 0$). If the uncertainty $\mathbf{\Delta}$, however, is left unscaled ($\|\mathbf{\Delta}\|_\infty \leq 1$), the worst-case gain can be less or equal to μ_{RP} .

Inversely, if NS is fulfilled, but RP is not, i.e., if $\mu_{\text{RP}}(\mathbf{N}) := \sup_\omega \mu_{\hat{\Delta}}(\mathbf{M}(j\omega)) > 1$, then the situation is reversed: μ is computed assuming that both $\mathbf{\Delta}$, $\mathbf{\Delta}_{\text{perf}}$ are scaled down by a common factor $k_m < 1$. If, however, $\mathbf{\Delta}$ remained unscaled at full norm, the worst case gain can be greater or equal to μ_{RP} .

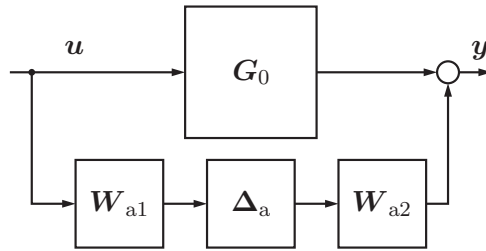
As a solution, the skew- μ value has been introduced [143]:

Definition 2.3.8: Skew- μ value

The skew- μ value of a complex transfer matrix $\mathbf{M}(j\omega)$ is defined as

$$\mu_{\hat{\Delta}}^s(\mathbf{M}(j\omega)) = \frac{1}{\min_{\hat{\Delta} \in \hat{\Delta}_B} \{k_m : \det(\mathbf{I} - \mathbf{K}_m \mathbf{M}(j\omega) \hat{\Delta}) = 0\}}, \quad (2.44)$$

where $\hat{\Delta} = \begin{bmatrix} \mathbf{\Delta} & \\ & \mathbf{\Delta}_{\text{perf}} \end{bmatrix}$, $\mathbf{K}_m = \begin{bmatrix} \mathbf{I} & \\ & k_m \mathbf{I} \end{bmatrix}$ is partitioned compatibly to $\hat{\Delta}$, so only $\mathbf{\Delta}_{\text{perf}}$ is scaled. It turns out that μ^s is always further from 1 than μ and can be interpreted as the worst-case gain. In MATLAB[®], this quantity is one of the outputs computed by the command `wcgain`, but also a skew- μ toolbox exists for more general robustness analysis [30].

Figure 2.2: Additive uncertainty model (uncertainty $\mathbf{W}_{a2}\Delta_a\mathbf{W}_{a1}$)

2.3.2 Uncertainty Representation and Modeling

Suitable uncertainty modeling is imperative to achieving robust stability and performance. Typical uncertainty models are additive uncertainty (commonly used to model neglected dynamics), multiplicative input or output uncertainty (modeling actuator or sensor magnitude or phase uncertainty), coprime-factor uncertainty (used to directly address pole uncertainty, also across the stability bound), and parametric uncertainties [174].

Unstructured Additive Uncertainty

Fig. 2.2 shows the standard model of an additive uncertainty. The nominal ($s \times r$) plant \mathbf{G}_0 has an ($r \times 1$) input vector \mathbf{u} and an ($s \times 1$) output vector \mathbf{y} . The parallel uncertain transfer path $\mathbf{W}_{a2}\Delta_a\mathbf{W}_{a1}$ must be of size ($s \times r$) to be compatible to \mathbf{G}_0 . The shown interconnection realizes an overall uncertain transfer $\mathbf{y} = (\mathbf{G}_0 + \mathbf{W}_{a2}\Delta_a\mathbf{W}_{a1})\mathbf{u}$.

The uncertain block Δ_a is considered as an LTI system with the following properties:

- it is stable,
- linear time-invariant,
- norm-bounded by $\|\Delta_a\|_\infty \leq 1$, and
- otherwise unknown, i.e. any coupling from its inputs to its outputs are possible, and phase relations are unknown.

Due to the last property, this uncertainty block is often called an unstructured complex uncertainty or a full complex uncertainty block, because any concrete realization of Δ_a yields a full complex-valued transfer matrix for any fixed frequency ω .

The weighting functions \mathbf{W}_{a1} and \mathbf{W}_{a2} , however, are assumed known and characterize the maximum possible magnitude of the uncertain paths over frequency. Due to the lack of structure in Δ_a it is typically sufficient to choose \mathbf{W}_{a1} and \mathbf{W}_{a2} square and diagonal, populated with simple transfer functions. Often, only one weighting is utilized. A sensible choice of these functions is not straightforward [174]. However, simple quantitative bounds on expected magnitudes of deviations over frequency, modeled by low-complexity weighting functions, typically suffice to obtain good analysis and design results.

The additive uncertainty model is typically utilized to depict neglected or unknown system dynamics in a simplified manner. This is relevant for example when the plant model utilized for control design is trusted only within a specific, limited frequency range. Robust control design with an appropriate additive uncertainty yields a controller which stabilizes and performs also with the actual plant which exhibits additional dynamics.

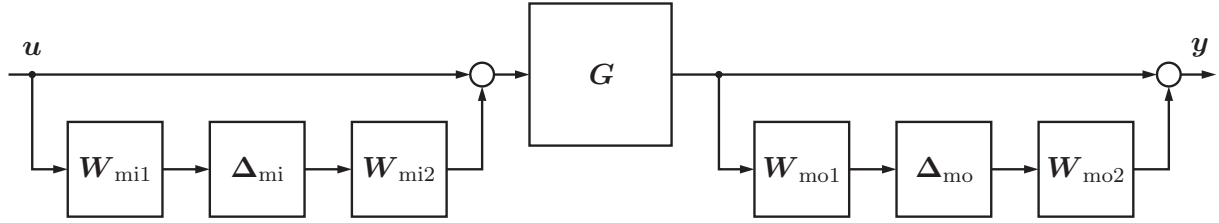


Figure 2.3: Multiplicative uncertainty model (multiplicative input uncertainty $\mathbf{W}_{mi2}\Delta_{mi}\mathbf{W}_{mi1}$, multiplicative output uncertainty $\mathbf{W}_{mo2}\Delta_{mo}\mathbf{W}_{mo1}$)

Multiplicative Uncertainty

Fig. 2.3 depicts two important variants of multiplicative (relative) uncertainty, one located at the plant inputs and one at the plant outputs. The nominal ($s \times r$) plant \mathbf{G}_0 has an ($r \times 1$) input vector \mathbf{u} and an ($s \times 1$) output vector \mathbf{y} . Both multiplicative uncertain paths are square, and the overall transfer function reads

$$\mathbf{G}_p = (\mathbf{I} + \mathbf{W}_{mo2}\Delta_{mo}\mathbf{W}_{mo1})\mathbf{G}_0(\mathbf{I} + \mathbf{W}_{mi2}\Delta_{mi}\mathbf{W}_{mi1}). \quad (2.45)$$

The configuration as shown here, utilizing two-sided weights, is rarely used – instead, it is often sufficient to choose one diagonal, square weight per Δ block. Contrary to additive uncertainty modeling, the uncertain blocks typically possess structure:

- They are assumed stable,
- linear time-invariant,
- norm-bounded by $\|\Delta_{mi}\|_\infty \leq 1$, $\|\Delta_{mo}\|_\infty \leq 1$,
- complex-valued (phase relations are unknown), and
- diagonally structured, i.e. no coupling between different input channels (or output channels, respectively) is assumed.

Consequently, this uncertainty is often called structured complex uncertainty or complex-diagonal uncertainty.

The multiplicative uncertainty model is widely used to model the following types of behaviour:

- Actuator uncertainty in gain and phase over frequency (multiplicative input uncertainty)
- Sensor / measurement uncertainty covering bounded drifts / gain / phase variations (multiplicative output uncertainty)
- Uncertain pole locii (time constants) and zero locii in the SISO case and uncertain delays in single channels, see [143].

As reasoned in [143], actuator and sensor uncertainty is always present in real plants and for control laws to work in practice, they always have to be robust against this type of uncertainty. Therefore, these uncertainties ought to be included in any robust design and validation. Note, however, that in case other uncertainty models are more important, it can be a better choice not to add this additional complexity in favor of a simpler design plant to facilitate control design and to reduce sources of design conservativeness.

Overview on Parametric Uncertainty Modeling

Consider a parametrized family of linearized plants $\mathbf{P}(\boldsymbol{\rho})$ as in Sec. 2.1.1, where the exact value of the parameter vector is unknown. Instead, only the bounds of the (real) parameter values are known.

Parametric uncertainties model the system variations due to changes in these plant parameters, whereby the parameters enter the system equations often non-linearly. Contrary to the types of uncertainty mentioned earlier, parametric uncertainties are modeled by a structured (diagonal) and real-valued uncertain block. Its diagonal elements are the normalized parameter values and possibly occur repetitively. Generic methods exist to transform many important classes of system parametrizations into a parametrized uncertain set in LFT form as long as the dependency of the state-space matrices on the parameters is at least polynomial or rational.

Tools to accomplish these transformations are subject of active research [7], [56] because of several reasons:

- Systems parametrized in LFT form are useful in robust control design, but also in the design and implementation of scheduled control laws (LFT scheduling, observer-based realization with an observer under LFT form) and in modeling and control design for Linear Parameter-Varying (LPV) systems.
- In all of these applications, the resulting high system complexity is the major obstacle. Efficiently modeling the system to the required accuracy with minimal complexity (that is, few parameter repetitions in $\boldsymbol{\Delta}$) is thus a key issue.

The book [7] lists several methods to generate a linear-fractional representation (LFR) from a given system parametrization. Morton's method [98], [99] can efficiently transform affine parametrizations. Extensions of this method can be applied to polynomial parametrization, but the size of the resulting $\boldsymbol{\Delta}$ -block quickly increases and is generally not minimal. Symbolic preprocessing and simplification is proposed in [57] to reduce the parametrization complexity. A recent approach covering optimal parametrization by approximation and effective simplification steps are documented in [56], [111], and [112]. A MATLAB[®] toolbox for LFT modeling is available [58].

Parametric Uncertainty: Pole locii of flexible modes

Parametric uncertainties for structural modes with low damping (uncertain in their frequency and damping in known intervals) can be modeled directly using LFTs (see [174]). The perturbation of one uncertain mode is modeled as:

$$\omega_p(\delta_1) = \omega(1 + W_1\delta_1) \quad (2.46)$$

$$\zeta_p(\delta_2) = \zeta(1 + W_2\delta_2), \quad (2.47)$$

where the index p indicates perturbed quantities, ω and ζ are the angular frequency and the damping coefficient of the oscillatory mode, respectively, and W_1 and W_2 are uncertainty scaling factors. The real-valued uncertain parameters are δ_1 and δ_2 ($\delta_1, \delta_2 \in \mathbb{R}$, $|\delta_1| \leq 1$, $|\delta_2| \leq 1$). The uncertainty scaling can be defined from the interval bounds $W_1 = 2 \cdot \frac{\omega_U - \omega_L}{\omega_U + \omega_L}$ (analogously for W_2).

A specifically derived Linear Fractional Representation of these uncertainties for systems in modal form follows. Complex flexible structure systems can be transformed to modal form as outlined in [39] and [158]. Their nominal and the perturbed mode's (2×2) system matrix block on the main diagonal of \mathbf{A} reads $\mathbf{A}_m = \begin{pmatrix} 0 & \omega \\ -\omega & -2\zeta\omega \end{pmatrix}$, respectively $\mathbf{A}_{m,p}(\delta_1, \delta_2) =$

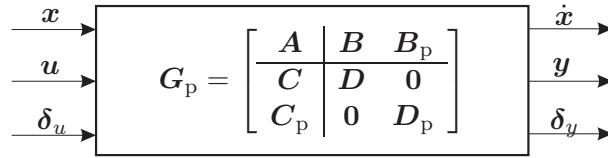


Figure 2.4: Parametric state space uncertainty

$\begin{pmatrix} 0 & \omega_p(\delta_1) \\ -\omega_p(\delta_1) & -2\zeta_p(\delta_2)\omega_p(\delta_1) \end{pmatrix}$ where the polynomials ω_p and ζ_p in δ_1 and δ_2 can be realized by LFTs of the system in Fig. 2.4 with

$$\left(\begin{array}{c|c} \mathbf{A}_m & \mathbf{B}_p \\ \mathbf{C}_p & \mathbf{D}_p \end{array} \right) = \left(\begin{array}{cc|ccc} 0 & \omega & \omega W_1 & 0 & 0 & 0 \\ -\omega & -2\zeta\omega & 0 & -\omega W_1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2\zeta\omega W_1 & 0 & 0 & 0 & W_1 \\ 0 & -2\zeta\omega W_2 & 0 & 0 & 0 & 0 \end{array} \right) \quad (2.48)$$

and with the diagonal real-valued uncertainty block

$$\Delta_p = \begin{bmatrix} \delta_1 & & & & & \\ & \delta_1 & & & & \\ & & \delta_1 & & & \\ & & & \delta_2 & & \end{bmatrix}. \quad (2.49)$$

Note that the parameter δ_1 , associated with the angular frequency ω , is repeated three times in the Δ_p -block because it enters the \mathbf{A} matrix at three positions. In contrast, the parameter δ_2 (modeling the variation of the damping ζ) occurs only once as its influence on \mathbf{A} is only linear in a single matrix element. This highlights the importance of LFR construction algorithms that can find and exploit common dependencies of matrix elements on the same parameter.

The perturbed state space description matrix Γ_p is computed from the LFT $\mathcal{F}_1(\mathbf{G}_p, \Delta_p)$.

Example 2.3.1: Second-order system with an uncertain oscillatory mode

Consider the SISO state space system

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & \omega \\ -\omega & -2\zeta\omega \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (2.50)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x} \quad (2.51)$$

with $\mathbf{x} = \begin{bmatrix} q & \dot{q} \end{bmatrix}^T$ and uncertain but interval-bounded $\omega \in [10, 20]$, $\zeta \in [0.1, 0.2]$. These parameters can be written as functions of unit-norm-bounded uncertainties δ_1, δ_2 ($|\delta_1| \leq 1, |\delta_2| \leq 1$):

$$\omega = 15(1 + 5\delta_1) \quad (2.52)$$

$$\zeta = 0.15(1 + 0.05\delta_2) \quad (2.53)$$

The parametric uncertainty description for independent ω and ζ is thus

$$\left(\begin{array}{c|c} \mathbf{A}_m & \mathbf{B}_p \\ \mathbf{C}_p & \mathbf{D}_p \end{array} \right) = \left(\begin{array}{cc|ccc} 0 & 15 & 75 & 0 & 0 & 0 \\ -15 & -4.5 & 0 & -75 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -22.5 & 0 & 0 & 0 & 5 \\ 0 & -0.225 & 0 & 0 & 0 & 0 \end{array} \right). \quad (2.54)$$

The system interconnection structure to realize this parametrization is shown in Fig. 2.4. Formally, the parametrized system can be written as

$$G = \mathcal{F}_1 \left(\left(\begin{array}{c|cc} \mathbf{A}_m & \mathbf{B} & \mathbf{B}_p \\ \hline \mathbf{C} & \mathbf{D} & \mathbf{0} \\ \mathbf{C}_p & \mathbf{0} & \mathbf{D}_p \end{array} \right), \Delta_p \right) \quad (2.55)$$

2.3.3 Robust Control Design Problem Statement

For a set \mathcal{P} of linearized dynamic systems (validation set), the task is to design a controller \mathbf{K} which obtains for each validation system $\mathbf{P} \in \mathcal{P}$ stable closed-loop dynamics that fulfills predefined performance objectives (quantified by signal or system norms). Commonly the controller \mathbf{K} is designed on a simplified design plant $\tilde{\mathbf{P}}$ (extracted from \mathcal{P}) and validated thereafter. Two validation cases are distinguished.

Continuous Set of Plants Let a continuous set \mathcal{P} of perturbed linear dynamic systems around a nominal system \mathbf{P}_0 and a structured complex uncertainty matrix Δ , $\|\Delta\|_\infty < 1$ (compare Fig. 2.1) such that $\mathcal{P} = \mathcal{F}_u(\mathbf{P}_0, \Delta)$. The control objectives are fulfilled if RS and a suitable formulation of RP are fulfilled for the closed loop for all possible perturbations Δ (verified by μ analysis). This validation strategy can lead to direct results using state-of-the-art robust control design and analysis tools, but it requires an exact a priori known uncertainty description which is hard to obtain in practice.

Discrete Set of Plants Let a discrete set $\mathcal{P} = \{\mathbf{P}_i : i = 1, \dots, m\}$ of m linear (or non-linear) dynamic systems at different fixed system parameter values. The control objectives are fulfilled if stability and suitable performance requirements are met for each closed loop system $\mathbf{M}_i = \mathcal{F}_1(\mathbf{P}_i, \mathbf{K})$. This approach is relevant in practice as it allows to consider non-linear aspects in validation. The uncertainties have to be modeled and tuned for design by the control engineer to cover the arising system parameter variations.

2.4 Optimization Fundamentals

2.4.1 Generic Optimization Problem

The standard formulation of a finite-dimensional (static) optimization problem is

$$\text{minimize } f = f(\mathbf{x}) \quad (2.56)$$

subject to constraints of the form

$$g_i(\mathbf{x}) = 0, \quad i = 1, \dots, p, \quad (2.57)$$

$$h_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \quad (2.58)$$

Thereby, $\mathbf{x} \in \mathcal{X}$ is the vector of decision variables (or free variables), $f : \mathcal{X} \mapsto \mathbb{R}$ is called the objective function and formulates costs of each possible decision, $\mathbf{g} = [g_1, \dots, g_p]^T : \mathcal{X} \mapsto \mathbb{R}^p$ are the equality constraints, and $\mathbf{h} = [h_1, \dots, h_q]^T : \mathcal{X} \mapsto \mathbb{R}^q$ are the inequality constraints.

For simplicity of exposition, the decision variable vector is taken from the real vector space \mathbb{R}^n , and the feasible set $\mathcal{X} \subseteq \mathbb{R}^n$ is defined as the set of all \mathbf{x} which fulfill the constraints:

$$\mathcal{X} := \{\mathbf{x} : \mathbf{g}(\mathbf{x}) = \mathbf{0}, \mathbf{h}(\mathbf{x}) \leq \mathbf{0}\} \quad (2.59)$$

2.4.2 Convexity

Optimization problems can be characterized based on the classes of considered functions for f , g , and h . In particular, if f is convex and if the set \mathcal{X} is a convex set, then the optimization problem is convex [12].

Definition 2.4.1: Convex set

A set $\mathcal{X} \subseteq \mathbb{R}^n$ is a convex set if the line segment between any two points $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ lies in \mathcal{X} . Formally \mathcal{X} is convex if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}, \alpha \in [0; 1]$ all point $\mathbf{z} = \alpha \mathbf{x} + (1 - \alpha) \mathbf{y}$ are also in the set, $\mathbf{z} \in \mathcal{X}$. [12]

Definition 2.4.2: Convex function

A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex if its domain $\text{dom} f$ is a convex set and if for all $\mathbf{x}, \mathbf{y} \in \text{dom} f, \alpha \in [0; 1]$ the relation

$$f(\alpha \mathbf{x} + (1 - \alpha) \mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha) f(\mathbf{y}) \quad (2.60)$$

holds. [12]

If an optimization problem is convex, then it has an extremely important property: If it is feasible and if a locally optimal solution could be found, then the solution is globally optimal. For this reason, knowing or proving that a given problem is convex is an important task. Many important classes of optimization problems are convex and efficient, polynomial-time solution algorithms are available. These problems include the widely known linear programming (LP) problems, quadratic programming problems with a convex quadratic form, and LMI problems which are detailed below.

2.4.3 Linear Programming

A linear program (LP) is an optimization problem

$$\inf_{\mathbf{x}} \mathbf{c}^T \mathbf{x} \quad (2.61)$$

subject to

$$\mathbf{a}_i^T \mathbf{x} \leq b_i, \quad i = 1, \dots, k \quad (2.62)$$

$$\mathbf{a}_{\text{eq},j}^T \mathbf{x} \leq b_{\text{eq},j}, \quad j = 1, \dots, k_{\text{eq}}. \quad (2.63)$$

Thereby, $\mathbf{x} \in \mathbb{R}^n$ are the free *decision variables* and $\mathbf{c} \in \mathbb{R}^n$ is the cost vector. For correctness, the infimum (inf) is used instead of the minimum (min) here because the minimum might not be actually reached in an unbounded feasible set. The coefficient vectors $\mathbf{a}_i \in \mathbb{R}^n$ and the associated right-hand sides b_i define k inequality constraints. Similarly, the coefficient vectors $\mathbf{a}_{\text{eq},j} \in \mathbb{R}^n$ and the associated right-hand sides $b_{\text{eq},j}$ define k_{eq} equality constraints. Often, these constraints are denoted in the compact form as

$$\mathbf{A} \mathbf{x} \leq \mathbf{b} \quad (2.64)$$

$$\mathbf{A}_{\text{eq}} \mathbf{x} = \mathbf{b}_{\text{eq}} \quad (2.65)$$

with $\mathbf{A} = [\mathbf{a}_1 \ \dots \ \mathbf{a}_k]^\top$ and $\mathbf{A}_{\text{eq}} = [\mathbf{a}_{\text{eq},1} \ \dots \ \mathbf{a}_{\text{eq},k_{\text{eq}}}]^\top$. Note, however, that this notation indicates row-wise scalar inequalities and equalities. Note also that linearly independent equality constraints reduce the number of free variables to $n - k_{\text{eq}}$, so some standard LP notations do not include explicit equality constraints.

A wide variety of problems can be cast into or approximated by an LP formulation. The reference algorithm to efficiently solve LPs is the well-known Simplex algorithm, developed by George Dantzig in the year 1947 [17]. Later in history, efficient Interior-Point (IP) methods have been developed [71]. The main advantage in the use of LPs lies in their efficient solution — problems with millions of variables and constraints can be solved today. Since this problem type is a convex problem a locally optimal solution is also globally optimal.

2.4.4 Quadratic Programming

A quadratic program (QP) is an optimization problem with a quadratic objective function and linear inequality (and/or equality) constraints:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{c}^\top \mathbf{x} \quad (2.66)$$

subject to

$$\mathbf{A} \mathbf{x} \leq \mathbf{b}, \quad (2.67)$$

$$\mathbf{A}_{\text{eq}} \mathbf{x} = \mathbf{b}_{\text{eq}}. \quad (2.68)$$

Thereby, (2.67)–(2.68) define inequality and equality constraints as in (2.62)–(2.63). If $\mathbf{Q} \succeq 0$ holds (that is, \mathbf{Q} is positive semi-definite), the optimization problem is convex and efficiently solvable. Common algorithms are interior point-, active set- or Simplex-based methods. However, if \mathbf{Q} is indefinite, no general computationally tractable algorithm is known. Linear programs are special cases of quadratic programs with $\mathbf{Q} = \mathbf{0}$.

2.4.5 Linear Matrix Inequalities (LMIs)

A LMI is an affine matrix constraint of the form:

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \sum_{i=1}^n x_i \mathbf{F}_i \succeq 0. \quad (2.69)$$

The matrices $\mathbf{F}_i = \mathbf{F}_i^\top \in \mathbb{R}^{n \times n}$ are symmetric and fixed, and $\mathbf{x} = [x_1, \dots, x_n]^\top$ are the free decision variables. The constraint $\mathbf{F}(\mathbf{x}) \succeq 0$ means that the matrix $\mathbf{F}(\mathbf{x})$ is required to be positive-semidefinite, that is, that it possesses only non-negative eigenvalues.

In mathematical programming, the term Semidefinite Program (SDP) is used to denote an optimization problem of the form

$$\min_{\mathbf{x}} \mathbf{c}^\top \mathbf{x} \quad (2.70)$$

subject to

$$\mathbf{A} \mathbf{x} = \mathbf{B}, \quad (2.71)$$

$$\mathbf{x} \in \mathbf{K}, \quad (2.72)$$

where the linear objective function shall be minimized. The first set of constraints are linear equality constraints, the set of inequality constraints is represented implicitly where \mathbf{K} is a cone (see Def. 2.4.3 below). This problem formulation is called the primal SDP form.

Definition 2.4.3: Cone

A (pointed) cone \mathbf{K} is the subset of a vector space $\mathbf{V} \subset \mathbb{R}^n$ that is closed under positive scaling:

$$\mathbf{x} \in \mathbf{K} \Leftrightarrow \lambda \mathbf{x} \in \mathbf{K} \quad \forall \lambda \geq 0. \quad (2.73)$$

Important cones are the positive orthant (LP cone) $\mathbf{K}_{\text{LP}} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq \mathbf{0}\}$, the quadratic (second-order) Lorentz cone (which arises in quadratic programming), and the positive semidefinite cone (SDP cone) defined for example by an LMI constraint.

Remark: In mathematical programming, an SDP primal problem is defined as in (2.70)–(2.72), while in the control community the “primal LMI problem” is usually favored:

$$\inf_{\mathbf{x}} \mathbf{c}_{\text{LMI}}^T \mathbf{x} \quad (2.74)$$

subject to

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \sum_{i=1}^n x_i \mathbf{F}_i \succeq 0. \quad (2.75)$$

The primal LMI problem is in dual form to an semidefinite program/ programming (SDP) problem. Forming the dual to (2.74)–(2.75), a primal SDP problem is obtained:

$$\sup_{\mathbf{z}} \text{trace}(\mathbf{F}_0 \mathbf{z}) \quad (2.76)$$

subject to

$$\text{trace}(\mathbf{F}_i \mathbf{z}) = c_i \quad i = 1, \dots, n \quad (2.77)$$

$$\mathbf{z} \geq 0 \quad (2.78)$$

In this work, the term LMI problem is used for problems stated as in (2.74)–(2.75). The concept of duality in optimization problems is not detailed here, for reference see [11] or [103].

Today, numerous solvers and numerical tools exist to efficiently solve LMI problems, for example MATLAB[®]'s LMILAB (see [38], [6]) as well as open-source alternatives such as SeDuMi (see [149]). Also, auxiliary programs, such as YALMIP is a modeling tool and provides interfacing between MATLAB[®] and many solvers, see [86] and the website [169] which also lists supported solvers. The tool GlptiPoly [64] implements an automated approximate solving of higher-order problems by LMI approximations, also refer to the survey [16].

Feasibility Problem and Cost-Optimization Problem

The obtained LMI problem can be solved in various ways: Firstly it may be of interest whether the problem is feasible at all (“feasibility problem”). Secondly, if it is feasible, an optimal solution should be obtained (“cost minimization problem”). Both tasks can be performed by today’s LMI solvers.

Control Applications

The book [11] gives a detailed account on the use of LMI formalisms in control problems. Some specific formulations are given in the following that are used in later chapters of this work.

Quadratic Stability Test: Given a set of k vertex system matrices $\mathbf{A}_i \in \mathbb{R}^{n \times n}$, $i = 1, \dots, k$, the problem is to decide whether all systems in the polytope spanned by the vertex systems

$$\mathbf{A}(\boldsymbol{\lambda}) = \sum_{i=1}^k \lambda_i \mathbf{A}_i \quad (2.79)$$

$$\sum_{i=1}^k \lambda_i = 1 \quad (2.80)$$

$$\lambda_i \geq 0 \quad (2.81)$$

are stable.

It can be shown that if a common, fixed, symmetric positive-definite ($n \times n$) matrix $\mathbf{P} = \mathbf{P}^T > 0$ (called Lyapunov matrix) is found that fulfills the following LMI constraints,

$$\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i \prec 0 \quad i = 1, \dots, k \quad (2.82)$$

$$\mathbf{P} \succ 0, \quad (2.83)$$

then all plants in the polytope are stable. Thereby, the matrix variable \mathbf{P} has $\frac{n(n+1)}{2}$ independent entries. Criterion (2.82)–(2.83) is called quadratic (Lyapunov) stability and is sufficient, but usually not necessary: if no corresponding matrix \mathbf{P} can be found (that is, if the LMI constraints render the problem infeasible), the polytope may still be stable everywhere.

Parameter-Varying Lyapunov Stability: For the polytope of systems as above (equations (2.79)–(2.81)), a less conservative LMI formulation can be given using a parameter-dependent Lyapunov matrix $\mathbf{P} = \mathbf{P}(\boldsymbol{\lambda})$. For a set of constant state space system matrices, the LMI feasibility constraint becomes [63]:

$$\begin{bmatrix} \mathbf{A}_i^T \mathbf{F}^T + \mathbf{F} \mathbf{A}_i & \mathbf{P}_i + \mathbf{A}_i^T \mathbf{G}^T - \mathbf{F} \\ \text{symm.} & -\mathbf{G} - \mathbf{G}^T \end{bmatrix} \prec 0 \quad i = 1, \dots, k \quad (2.84)$$

$$\mathbf{P}_i \succ 0 \quad i = 1, \dots, k, \quad (2.85)$$

where \mathbf{G} , \mathbf{F} are additional free ($n \times n$) matrices. Note that the variable count is strongly increased as compared to quadratic stability: $N_P = k \frac{n(n-1)}{2}$, $N_F = n^2$, $N_G = n^2$ where N_P , N_F , N_G are the number of free variables in \mathbf{P} , \mathbf{F} , and \mathbf{G} , respectively. However, this stability test shows much less conservativeness than the quadratic stability test.

2.4.6 Genetic Algorithms

A genetic algorithm is an evolutionary, population-based, global search and optimization method (see also [35]) that attempts to find a cost-minimal solution $\mathbf{x}(\mathbf{p})$,

$$f_{\text{cost}}(\mathbf{x}(\mathbf{p})) \rightarrow \min, \quad (2.86)$$

encoded by a valued set of m decision variables $\mathbf{p} : \mathcal{P} \mapsto \mathbb{R}^m$. These free parameters constitute the „genome” which encodes a solution. Numerous solutions are collected in the solution population and are subject to evolution over generations using genetic operators, such as crossover (creating offspring solutions from parent solutions) and random mutation.

Particular advantages of genetic algorithms are the fact that they do neither require gradient information of the objective function nor smoothness of the problem landscape. However, genetic algorithms themselves cannot guarantee the convergence to a global or even local optima. If structure information on the problem is available, hybrid schemes of a genetic algorithm with local solution improvement through local, structured optimization is often utilized. This variant is often called memetic algorithm [76].

The main steps of a genetic algorithm are summarized in Tab. 2.1.

Table 2.1: Genetic Algorithm pseudo code

```

initialize population, evaluate

while termination criterion not fulfilled

    select solutions for reproduction
    create offspring via crossover
    mutate solutions randomly
    evaluate fitness of new population members
    store overall best solution

end

```

2.4.7 Evaluation Methodology “Goal Attainment”

One possible method to validate a resulting control law and to evaluate its performance on a set of validation plants \mathcal{S}_{val} is the Goal Attainment evaluation method (see [47]). It yields a global, aggregated scalar cost value for a set of k goals, defined by:

- $[x_1, x_2, \dots, x_k]$: Actual cost values of each goal of the closed-loop system
- $[t_1, t_2, \dots, t_k]$: Target values (as upper bound for acceptable goal cost), for example rise time, overshoot, or other quantifications of control performance or stability

The global value is defined as:

$$\text{cost function } f = \max_{j \in \{1, \dots, k\}} g_j, \quad g_j = \frac{x_j}{t_j}, \quad j = 1, \dots, m \quad (2.87)$$

which yields the worst-case goal attainment ratio as global cost. Only if all goals are fulfilled, the cost function value is less or equal to 1. Additionally, due to the max operator in (2.87), the search is directed towards improving the least fulfilled goal.

The max function in the evaluation methodology above only conveys information on the least-fulfilled goal to the solver algorithm. Moreover, the objective function is not continuously differentiable at points where two or more goals are of the same maximum value. To avoid related convergence or singularity problems in the solution process, a different variant of the optimization problem can be formulated. Thereby, the max function is replaced by a continuously differentiable exponential expression (“soft max”) with similar properties:

$$\text{cost function } f = \log \left(\sum_{j \in \{1, \dots, k\}} \exp(g_j) \right), \quad g_j = \frac{x_j}{t_j}, \quad j = 1, \dots, m. \quad (2.88)$$

This function is dominated by the least-fulfilled goals and additionally conveys information on all other goals. It is well-defined everywhere (except at $\mathbf{0}$) and thus can improve solver efficiency. However, no fixed value guarantees the fulfillment of all goals (as was the case before), so this simple interpretability is lost and additional post-processing and solution analysis is necessary.

2.5 Flexible-Structure Modeling

Fundamental properties of MIMO control systems and state space system calculus are presented in [143]. When working with flexible-structure systems with low-damped oscillatory modes, these can be transformed into a decoupled form with so-called modal coordinates which enables efficient computations. The textbook [39] gives detailed account on the related techniques in modeling, system analysis, and control design. A concise summary of modal state-space representations based on [39] is given in the following.

Assume the dynamics of a flexible structure is described by n linear (or linearized) equations of motion. The commonly used mathematical model can be represented by a set of n second-order linear ordinary differential equations (ODEs), whereby a crucial design decision is the choice of their coordinates. In the textbook of Gawronski [39], model representations of flexible structures are discussed in detail.

2.5.1 Nodal Models

So-called nodal models can be retrieved in nodal coordinates (displacements and velocities) for example by FE modeling. The linear equations of motion are commonly written in the following second-order nodal form:

$$\begin{aligned} M\ddot{\mathbf{q}}_n(t) + D\dot{\mathbf{q}}_n(t) + K\mathbf{q}_n(t) &= B\mathbf{u}(t) \\ \mathbf{y} &= C_{oq}\mathbf{q}_n(t) + C_{ov}\dot{\mathbf{q}}_n(t), \end{aligned} \quad (2.89)$$

where $M \succ 0$, $D \succeq 0$, and $K \succeq 0$ are the $(n \times n)$ mass-, damping-, and stiffness matrices of the system with n degrees of freedom. The $(n \times 1)$ nodal coordinate vector $\mathbf{q}_n(t)$ and its time derivatives contain the nodal displacements, velocities, and accelerations, respectively. The $(n \times s)$ nodal input matrix B maps the action of the s (generalized force) inputs \mathbf{u} on the system. Likewise, the $(r \times n)$ output matrices C_{oq} and C_{ov} define the r outputs $((r \times 1)$ output vector \mathbf{y}) as linear combination of the nodal displacements and velocities. In the following, the notion of explicit time dependence is omitted to improve readability.

This system of n coupled ODEs of order 2 can be rewritten into a system of $2n$ first-order ODEs in state space form by defining the state space vector as a combination of the structural displacements \mathbf{q}_n and velocities $\dot{\mathbf{q}}_n$:

$$\mathbf{x}_n = \begin{bmatrix} \mathbf{q}_n \\ \dot{\mathbf{q}}_n \end{bmatrix} \quad (2.90)$$

The state equation reads:

$$\dot{\mathbf{x}}_n = \begin{bmatrix} \mathbf{0}_{[n \times n]} & \mathbf{I}_{[n \times n]} \\ -M^{-1}C & -M^{-1}K \end{bmatrix} \mathbf{x}_n + \begin{bmatrix} \mathbf{0}_{[n \times r]} \\ M^{-1}B \end{bmatrix} \mathbf{u} \quad (2.91)$$

with $\mathbf{0}$ and \mathbf{I} as zero and unity matrices with the indicated dimensions. The system output can be written as:

$$\mathbf{y} = \begin{bmatrix} C_{oq} & C_{ov} \end{bmatrix} \mathbf{x}_n \quad (2.92)$$

Thus, the common state space form is obtained:

$$\begin{aligned} \dot{\mathbf{x}}_n &= A_n \mathbf{x}_n + B_n \mathbf{u} \\ \mathbf{y} &= C_n \mathbf{x}_n \end{aligned} \quad (2.93)$$

2.5.2 Modal Form

A different choice of coordinates, especially advantageous for flexible mechanical structures, leads to so-called *modal models* of structures. They are described in modal coordinates and exhibit favorable properties. By applying the similarity transformation $\mathbf{T} = \mathbf{\Phi}$ with the eigenvector matrix $\mathbf{\Phi} = [\phi_1 \dots \phi_n]$ (eigenvectors listed column-wise) of the undamped eigenvector/eigenvalue problem $\mathbf{K}\phi_i = \omega_i^2 \mathbf{M}\phi_i$ to (2.89), the second-order system of ODEs is diagonalized as follows:

$$\begin{aligned} \mathbf{M}_m \ddot{\mathbf{q}}_m + \mathbf{D}_m \dot{\mathbf{q}}_m + \mathbf{K}_m \mathbf{q}_m &= \mathbf{\Phi}^T \mathbf{B}_o \mathbf{u} \\ \mathbf{y} &= \mathbf{C}_{oq} \mathbf{\Phi} \mathbf{q}_m + \mathbf{C}_{ov} \mathbf{\Phi} \dot{\mathbf{q}}_m \end{aligned} \quad (2.94)$$

where $\mathbf{M}_m = \mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi}$, $\mathbf{D}_m = \mathbf{\Phi}^T \mathbf{D} \mathbf{\Phi}$, and $\mathbf{K}_m = \mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi}$ are the modal mass-, damping-, and stiffness matrices. Due to the similarity transformation, \mathbf{M}_m and \mathbf{K}_m are diagonal, and also the modal damping matrix \mathbf{D}_m becomes diagonal in the commonly assumed case of Rayleigh-damping ($\mathbf{D} = \alpha_1 \mathbf{M} + \alpha_2 \mathbf{K}$ with $\alpha_1, \alpha_2 \geq 0$), see [39]. This formulation is obtained by the coordinate transformation $\mathbf{q}_m = \mathbf{\Phi} \mathbf{q}_m$, where \mathbf{q}_m are the modal displacements or coordinates.

For systems with well-separated oscillatory eigenmodes with low damping, the representation in modal coordinates exhibits advantageous properties for dynamics analysis and system identification, since they can be written in mode-wise decoupled form. This representation is thus well-suited to treat typical flexible structures in lightweight engineering. Numerous tools, for example state-of-the-art actuator/sensor placement procedures [39], [62], [60], [61], [82], [136], [158], exploit the distinct features of modal models.

As for the nodal model formulation, the system of second-order ODEs (2.94) can be written into a state-space system of first order. The following choice of the modal state vector is numerically well-conditioned:

$$\mathbf{x}_m = \begin{bmatrix} \mathbf{x}_{m1} \\ \vdots \\ \mathbf{x}_{mi} \\ \vdots \\ \mathbf{x}_{mn} \end{bmatrix} \quad \mathbf{x}_{mi} = \begin{bmatrix} \omega_i q_{mi} \\ \dot{q}_{mi} \end{bmatrix} \quad (2.95)$$

With this choice of states, the modal state space model can be written as:

$$\dot{\mathbf{x}}_m = \mathbf{A}_m \mathbf{x}_m + \mathbf{B}_m \mathbf{u} \quad (2.96)$$

$$\mathbf{y} = \mathbf{C}_m \mathbf{x}_m \quad (2.97)$$

with the following (2×2) block-diagonal structure for the modes $i = 1, \dots, n$ (usually sorted by their natural frequency ω_i):

$$\begin{aligned} \mathbf{A}_{mi} &= \begin{bmatrix} 0 & \omega_i \\ -\omega_i & -2\zeta_i \omega_i \end{bmatrix} & \mathbf{B}_{mi} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{b}_{mi}^T \end{bmatrix} \\ \mathbf{C}_{mi} &= \begin{bmatrix} \frac{1}{\omega_i} \mathbf{c}_{mqi} & \mathbf{0} \end{bmatrix} \end{aligned} \quad (2.98)$$

$$\begin{aligned} \mathbf{A}_m &= \text{diag}(\mathbf{A}_{mi}) & \mathbf{B}_m &= \begin{bmatrix} \mathbf{B}_{m1} \\ \mathbf{B}_{m2} \\ \vdots \\ \mathbf{B}_{mn} \end{bmatrix} \\ \mathbf{C}_m &= \begin{bmatrix} \mathbf{C}_{m1} & \mathbf{C}_{m2} & \dots & \mathbf{C}_{mn} \end{bmatrix} \end{aligned} \quad (2.99)$$

One of the main reasons to use the modal state space representation is that in case of high order nodal models (as obtained via FE modeling), the order can easily be reduced in the modal representation by truncating higher modes without significantly changing system behavior. Also, the obtained set of modes is orthogonal (in the weakly damped case), which allows to break down the complexity of operations and system treatment considerably.

The modal state space representation (2.96)–(2.99) can also be determined by transformation of a system model given in an arbitrary state space representation. This is required for example for models established by system identification or reduced order models from a balanced model reduction where no physical interpretation of the state vector exists. This is accomplished through four consecutive transformation steps that are presented in detail in [158]:

- Diagonalization by similarity transformation: The system is transformed to its complex-diagonal form by state transformation with the eigenvector matrix.
- Transformation to modal form: By subsequent transformation steps, the system matrix can be brought to modal form (2.98).
- Create correct input and output matrix structure: Further transformation is necessary to obtain the necessary zero entries in the input and output matrices.
- Perform gain shift between input and output matrix: Finally the correct gain distribution is computed by matching the gains of (nearly-)collocated input and output channels (observing the factor ω_i in (2.98)).

2.5.3 Acceleration Sensors

Acceleration quantities are often important outputs of a structural control system, for example as to estimate passenger ride comfort. However, as evident in (2.96), acceleration outputs in the form of

$$\ddot{\mathbf{y}} = \mathbf{C}_{ma}\ddot{\mathbf{q}}_m, \quad (2.100)$$

are not directly available as linear combination of the states. Using the lower half of (2.96), the vector of modal accelerations can be formulated.

Given the displacement outputs for mode i

$$\mathbf{y} = \begin{bmatrix} \frac{\mathbf{c}_{mqi}}{\omega_i} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \omega_i q_{mi} \\ \dot{q}_{mi} \end{bmatrix} = \mathbf{c}_{mqi} q_{mi}, \quad (2.101)$$

it is possible to formulate the corresponding acceleration outputs by eliminating \ddot{q}_{mi} via equation (2.96):

$$\ddot{\mathbf{y}} = \mathbf{c}_{mqi}\ddot{q}_{mi} \quad (2.102)$$

$$= \begin{bmatrix} -\mathbf{c}_{mqi} & -2\zeta_i\omega_i\mathbf{c}_{mqi} \end{bmatrix} \begin{bmatrix} \omega_i q_{mi} \\ \dot{q}_{mi} \end{bmatrix} + \mathbf{c}_{mqi}\mathbf{b}_{mi}^T\mathbf{u}. \quad (2.103)$$

2.6 Input / Output Selection

In control design, the choice of inputs (actuator signals) and outputs (measurements) that are accessible by a controller has a significant impact on the properties and the faced limitations of the control system. As discussed in [143], fundamental qualitative limitations can be faced in control design for a certain input/output (I/O) choice which restrict achievable stability, performance, and robustness. Moreover, note that in practice actuation capabilities are limited,

measurements are corrupted by noise, and design models differ from reality. Thus the I/O-selection task must also consider I/O efficiency and the sensitivity of I/O-choices to model perturbations which all affects attainable control performance quantitatively. For these reasons it is vital to make this choice in an optimized manner before carrying out the actual controller design. Many corresponding criteria for control system input/output evaluation and selection exist nowadays, and an excellent survey is given by [152].

Three important groups of I/O selection criteria are

energy-based methods which aim to find I/Os which maximize controllability/observability-related quantities (see [39], [51], [82] and the recently proposed criterion in [158]),

information-based methods which aim to maximize information/observation independence in sensor selection, and

closed-loop criteria which directly evaluate attainable closed-loop performance for each I/O-candidate (see [60] for a recently proposed LQG criterion).

To select suitable candidates from a large set of potential I/O-combinations for robust control design, a new computationally efficient selection method has been proposed in [61] and [62]. The doctorate thesis [59] addresses these developments in detail and studies their application at a flexible BWB aircraft model. A comparison of the criteria from [39], [158], and [60] is carried out in [136] at the flexible BWB aircraft model which is also utilized in this work. I/O-selection results for various other applications are presented in [8] and [127] (piezo actuator positioning for flexible structure control of a railcar body), [27] and [82] (piezo actuators/sensors on flexible structures).

Considering the effort required by the I/O-selection task and on the degree and quality of available information, the following advantages and limitations arise (see also [136]):

Advantages of energy- and information-based onsets over closed-loop criteria are that

- they provide qualitative statements over a wide range of control system architectures,
- their evaluation is typically less computationally demanding, and
- physical interpretation is possible in the case of structure mechanics.

Their limitations are that

- quantitative statements are often dependent on artificial weighting factors, thus results cannot be easily verified beforehand,
- robustness with respect to parameter variations is not addressed explicitly, and that
- accuracy of the results depends on the ability of introducing relevant expert knowledge (for example, weighting to reflect control design specifications).

In turn, if the actual control design process can be predicted well, if the design model is of sufficient accuracy, and if the final control specifications are known, closed-loop I/O-selection criteria could be carried out such that they actually mimic the later control design and provide accurate, specialized results.

However, the following limitations are evident:

- This approach requires high computational and modeling effort.
- Verification is difficult or not feasible until the final design has been carried out.

As a consequence, in an early design phase of the control system, having no or only uncertain information about the entire control design process, the latter approach is not expected to yield results superior to more general approaches, such as energy-based criteria. For this reason it is of interest to have generic methods which can be augmented by simple weighting functions to incorporate existing or derived expert knowledge on typical control system characteristics, in order to improve the reliability and quality of the decision basis for I/O-selection.

Remarks on robustness: Especially for systems which are operated in a wide range of physical parameters (such as aircraft), an I/O-selection which yields good control performance for all occurring perturbations is vital. However, most I/O-selection methods do not assess robustness directly. Two possibilities to obtain robust I/O choices with respect to plant model perturbation are to evaluate each member of a sufficiently fine-gridded multi-model of the system by the I/O-selection methods and to aggregate the resulting selection indicators suitably [136], [62], or — for closed-loop criteria — to resort to true robust control design and robust analysis for performance evaluation (however at drastically increased effort needed in the I/O-selection task).

2.6.1 Information-based Criteria

Most information-based selection criteria — typically for sensor selection — are based on properties of the so-called Fisher Information matrix [33] which expresses the innovation introduced by a particular sensor (in terms of the relevance and independence of its information compared to the information provided by the remaining sensors) [124]. A constructive method to extract a set of relevant sensors is the Effective Independence (EFI) method, see [116], [95], [85].

In [54], the EFI method is modified to additionally avoid spillover with respect to undesired system modes. Comparisons of information-based approaches to other onsets are given in [85] and [161] and show the close relationship of various actuator / sensor selection approaches.

Sensor selection or placement based on information approaches has been carried out for various applications such as engine health monitoring [10], bridge structures [95], [85], space structures [170], and aircraft structures [55].

2.6.2 Energy Criteria

Three existing energy-based actuator and sensor positioning criteria are detailed in the following: one is proposed by Gawronski [39], the second is given by Hac & Liu [52], and the third is a combination and extension to the criteria in [39], [52], and [82] and is proposed in [158] and [136].

Gawronski's criteria [39]

A mode-based aggregate index that quantifies total energy transfer into or out of the structure has been proposed in the textbook [39]. For a system in modal form (2.96)–(2.97) and the j th actuator, the actuator index $\sigma_{\text{act},j}$ is defined as

$$\sigma_{\text{act},j} = \sqrt{\sum_{i=1}^n \sigma_{ij}^2}, \quad \sigma_{ij} = w_{ij} \frac{\|\mathbf{G}_{ij}\|_2}{\|\mathbf{G}\|_2} \quad (2.104)$$

where

$$\|\mathbf{G}_{ij}(j\omega)\|_2 \cong \frac{\|\mathbf{B}_{mij}\|_2 \|\mathbf{C}_{mi}\|_2}{2\sqrt{\zeta_i \omega_i}}, \quad (2.105)$$

is the \mathcal{H}_2 norm of the i th mode and j th actuator, $\|\mathbf{G}\|_2$ is the \mathcal{H}_2 norm of the system according to Def. A.4.7, ζ_i and ω_i are the modal damping and frequency of mode i , and w_{ij} is a weighting factor for mode and actuator weighting.

Analogously, sensor placement indices are defined by using the quantities related to the k th sensor:

$$\sigma_{\text{sens},k} = \sqrt{\sum_{i=1}^n \sigma_{ik}^2}, \quad \sigma_{ik} = w_{ik} \frac{\|\mathbf{G}_{ik}\|_2}{\|\mathbf{G}\|_2} \quad (2.106)$$

where

$$\|\mathbf{G}_{ik}(j\omega)\|_2 \simeq \frac{\|\mathbf{B}_{mi}\|_2 \|\mathbf{C}_{mik}\|_2}{2\sqrt{\zeta_i \omega_i}}. \quad (2.107)$$

Note that the criterion seeks to maximize the total energy throughput, but does not prevent the placement of actuators at nodes of higher-order modes due to its aggregating onset.

Criterion of Hac / Liu [52]

The performance index ρ for the j th actuator as defined in [52] is given by

$$\rho_j = \left(\sum_{i=1}^{2n} \lambda_i \right)^{2n} \sqrt[2n]{\prod_{i=1}^{2n} (\lambda_i)}, \quad (2.108)$$

where λ_i is the i th eigenvalue of the controllability Gramian of the pair $(\mathbf{A}, \mathbf{b}_j)$ (see [39]) and \mathbf{b}_j is the column vector of \mathbf{B} corresponding to the j th actuator.

This criterion incorporates the geometric mean of the Gramian eigenvalues which acts as a penalty term to ensure controllability of all considered modes, however, the performance index value is dependent on the representation (state choice) and the gain distribution between input and output matrices which cannot be reconstructed uniquely if no input/output collocation exists.

Combined criterion as proposed in [136] and [158]

An extended positioning criterion has been proposed by the authors in [158] that combines features of criteria proposed in [39], [52], and [82]. It adds the requirement that a set of modes be controllable or observable to result in a non-zero criterion value. For actuator placement the newly proposed performance index is defined as

$$\pi_{\text{act},j} = \sqrt{\sum_{i=1}^n \left(\frac{\|\mathbf{B}_{mij}\|_2 \|\mathbf{C}_{mi}\|_2}{2\sqrt{\zeta_i \omega_i}} \right)^2} \cdot \sqrt[2n]{\det(\mathbf{W}_{cj})}, \quad (2.109)$$

with \mathbf{B}_{mij} as the input matrix for the j th actuator under study, \mathbf{C}_{mi} as the performance output matrix, and n as the number of modeled modes. The controllability Gramian \mathbf{W}_{cj} is computed for the system with single input j and the set of modes that are required to be controllable. Note that this requirement prevents placement in and near modal nodes which improves robustness against small changes of the modal node locations. Those actuator positions with highest placement index are best suitable both in terms of actuation efficiency and from a robust point of view (all modes best controllable).

For sensor placement the performance index is defined as

$$\pi_{\text{sens},k} = \sqrt{\sum_{i=1}^n \left(\frac{\|\mathbf{B}_{mi}\|_2 \|\mathbf{C}_{mik}\|_2}{2\sqrt{\zeta_i \omega_i}} \right)^2} \cdot \sqrt[2n]{\det(\mathbf{W}_{ok})}, \quad (2.110)$$

with \mathbf{B}_{mi} as the disturbance input matrix, \mathbf{C}_{mik} as the output matrix for the k th sensor under study, and \mathbf{W}_{ok} as the observability Gramian of the system with single output k and the modes required to be observable. Those sensor positions with highest placement index are best suitable both in terms of sensor efficiency and from a robust point of view (all modes best observable).

2.6.3 Incorporation of Design Constraints by System Weighting

A general, yet simple solution to incorporate design constraints and conditions into system analysis is to apply a physically motivated input/output weighting to the system model before the actual system evaluation. Modal system representations allow simplified consideration of mode-wise weights if the weighting functions fulfil mild regularity conditions (see [39]).

Let $(\mathbf{A}_m, \mathbf{B}_m, \mathbf{C}_m)$ be a modal MIMO state-space system description (2.96)–(2.97), such as a model of the elastic modes of a flexible structure. Consider a sufficiently smooth scalar transfer function $W_o(j\omega)$ describing a frequency weighting that should be applied at one considered system output, such as a dynamic performance weighting of an exogenous system output (compare [39]). The output-weighted system $(\mathbf{A}_m, \mathbf{B}_m, \mathbf{C}_{m,w})$ is obtained by mode-wise output scaling by the values $|W_o(j\omega_i)|$, where ω_i is the i th mode's frequency:

$$\mathbf{C}_{m,w} = [|W_o(j\omega_1)|\mathbf{C}_{m1} \quad \dots \quad |W_o(j\omega_n)|\mathbf{C}_{mn}] \quad (2.111)$$

Analogously, frequency input weighting translates to mode-wise scaling of the input matrix with weighting function $W_i(j\omega)$, for example to incorporate a known excitation spectrum at performance inputs:

$$\mathbf{B}_{m,w} = \begin{bmatrix} \mathbf{B}_{m1}|W_i(j\omega_1)| \\ \mathbf{B}_{m2}|W_i(j\omega_2)| \\ \vdots \\ \mathbf{B}_{mn}|W_i(j\omega_n)| \end{bmatrix} \quad (2.112)$$

Finally, weightings on actuators or sensors are usually applied by diagonal scaling matrices in series connection to the plant. The I/O-weighted system $(\mathbf{A}_m, \mathbf{B}_{m,w}, \mathbf{C}_{m,w})$ has the same modal properties as the original system (natural frequencies, damping); only the mode gains are affected by the weighting process.

Remark: In system design and related analysis and evaluation tasks it is crucial to correctly formulate and incorporate relevant design constraints or conditions. These can be for example the expected excitation spectra for vibration control design, performance quantity weighting (weighted accelerations are used to estimate passenger ride comfort), or an actuator efficiency quantity.

Chapter 3

Optimal & Robust Control Design Methods

In this chapter, state-of-the-art feedback control design methods from optimal and robust control are collected and reviewed. This compact exposition of the state of the art should serve as a basis for the novel developments later in this thesis.

The chapter is structured as follows: The standard problem formulations for \mathcal{H}_2 and \mathcal{H}_∞ (sub-)optimal feedback control designs are given first. A short overview on design plant formulation, modeling, and weighting follows which shows the variety of covered problems and points out important aspects in the control engineering process. Then, the core state-of-the-art algorithms to design the sought (sub-)optimal feedback controllers are summarized from the literature. The exposition concentrates on the basic underlying solution structure and the similarity between \mathcal{H}_∞ and \mathcal{H}_2 optimization approaches, based on the reference papers [42] and [24], as well as on the exposition in the book [143]. Subsequently, some well-known LQ-based optimal control formulations (LQG, output-weighted LQG, and Linear-Quadratic Integral (LQI) designs) are outlined and embedded in the \mathcal{H}_2 framework developed before. Finally, two flavors of μ synthesis algorithms for robust controller design, the DK-iteration and the DGK-iteration, are outlined and discussed.

3.1 Standard problem formulation for \mathcal{H}_2 and \mathcal{H}_∞ feedback control design

The goal is to design an internally stabilizing LTI dynamic feedback controller $\mathbf{K}(s)$ for a given design plant $\mathbf{P}(s)$ that minimizes the \mathcal{H}_2 or the \mathcal{H}_∞ norm of the closed-loop performance transfer function \mathbf{T}_{zw} (see Fig. 3.1):

$$\mathbf{T}_{zw} = \mathcal{F}_1(\mathbf{P}, \mathbf{K}) = \mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(\mathbf{I} - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21} \quad (3.1)$$

The general solutions of the \mathcal{H}_2 and of the \mathcal{H}_∞ design problems in state-space form are given in [24]. Basic assumptions on \mathbf{P} and some important results are summarized in the following from [24], [143].

3.1.1 \mathcal{H}_∞ weighting, stacking, and \mathcal{H}_∞ problem classes in standard form

The textbooks [143] and [174] treat many design problems and their formulations in various optimal and robust control frameworks in detail. This chapter names a few of the important problems that typically arise and provides an exemplary formulation of a design plant in standard form for \mathcal{H}_∞ feedback controller design as shown in Fig. 3.1.

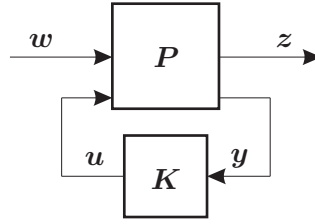


Figure 3.1: Standard form of the feedback control design problem

Weighting function choice

Some fundamental considerations are noted with respect to the design of frequency-domain weighting functions for \mathcal{H}_∞ control design. Given a *scalar* stable transfer function (weighting function) $W(s)$ and a stable MIMO or SISO transfer function $\mathbf{G}(s)$, note that at each frequency $s = j\omega$ the multiplicative property (A.10) of the spectral matrix norm is fulfilled with equality because the product is a scalar product:

$$\bar{\sigma}(W(j\omega)\mathbf{G}(j\omega)) = \bar{\sigma}(W(j\omega))\bar{\sigma}(\mathbf{G}(j\omega)) = |W(j\omega)|\bar{\sigma}(\mathbf{G}(j\omega)) \quad \forall \omega. \quad (3.2)$$

Consider that as a result of a control design, some weighted (closed-loop) transfer function $W\mathbf{G}$ fulfills

$$\|W\mathbf{G}\|_\infty < 1 \Leftrightarrow |W(j\omega)|\bar{\sigma}(\mathbf{G}(j\omega)) < 1 \quad \forall \omega. \quad (3.3)$$

Then, this is equivalent to

$$\bar{\sigma}(\mathbf{G}(j\omega)) < \frac{1}{|W(j\omega)|} = |W^{-1}(j\omega)| \quad \forall \omega, \quad (3.4)$$

so the inverse weight magnitude is a valid upper bound for the singular values of \mathbf{G} at all frequencies. This sketches the following weight selection procedure:

- Define the desired upper bounds on $\bar{\sigma}(\mathbf{G}(j\omega)) \forall \omega$.
- Shape a suitable scalar, minimum-phase, stable, invertible, and bi-proper transfer function $V(s)$ as an approximation from above to these upper bounds.
- Obtain the design weighting function by inversion: $W(s) = V^{-1}(s)$.
- Perform the corresponding controller design and try to achieve $\|W\mathbf{G}\|_\infty < 1$. If successful, it is guaranteed that the desired upper bounds are fulfilled.

The most common weighting function shapes are stable, minimum-phase, bi-proper high- and low-pass functions given by

$$V(s) = K \left(\frac{\frac{s}{\omega_1} + 1}{\frac{s}{\omega_2} + 1} \right)^n \quad (3.5)$$

with DC gain K , order n , and corner frequencies ω_1, ω_2 . If $\omega_1 < \omega_2$ holds, then $V(s)$ is a high-pass filter, whereas for $\omega_1 > \omega_2$ it is of low-pass behavior. The asymptotic magnitudes at low and high frequencies, respectively, are:

$$\lim_{\omega \rightarrow 0} V(j\omega) = K, \quad (3.6)$$

$$\lim_{\omega \rightarrow \infty} V(j\omega) = K \left(\frac{\omega_2}{\omega_1} \right)^n. \quad (3.7)$$

Caution: Note that these relations only hold if the weighting function is scalar, because only then the multiplicative property is fulfilled with equality. If $\mathbf{W}(s)$ is MIMO, even if it is only chosen diagonally with different entries along the diagonal, the above considerations do not hold in general and the inverse weight cannot be interpreted as an upper bound on $\bar{\sigma}(\mathbf{G}(j\omega))$ any more!

Restrictions on weighting functions: To formulate and solve standard \mathcal{H}_2 and \mathcal{H}_∞ design problems, a set of prerequisites or assumptions on the augmented design plant has to be fulfilled, see Sec. 3.1.2. This implies that the weights must be strictly stable, otherwise the design plant becomes non-stabilizable. This disqualifies pure integrators as weights which are usually replaced by “quasi-integrators” with a slightly stable, slow real pole and high DC gain. Analogously, the weights have to be proper, so ideal differentiators need to be approximated accordingly. The use of unstable and non-proper weights in \mathcal{H}_∞ design problems, related solutions and approximations are discussed in [94].

Stacking \mathcal{H}_∞ expressions

The following property is of central importance in \mathcal{H}_∞ control and allows to stack several objectives expressed as \mathcal{H}_∞ norm quantities (see [143, eq. (A.46)]):

$$\max\{\bar{\sigma}(\mathbf{A}), \bar{\sigma}(\mathbf{B})\} \leq \bar{\sigma}\left(\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}\right) \leq \sqrt{2} \max\{\bar{\sigma}(\mathbf{A}), \bar{\sigma}(\mathbf{B})\}. \quad (3.8)$$

Now, given two stable transfer functions $\mathbf{G}_1(s)$, $\mathbf{G}_2(s)$ with the same number of inputs (so that the transfer functions can be stacked), the fulfillment of

$$\left\| \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix} \right\|_\infty < 1 \quad (3.9)$$

implies (via (3.8) for each frequency $s = j\omega$) that

$$\|\mathbf{G}_1\|_\infty < 1 \text{ and } \|\mathbf{G}_2\|_\infty < 1 \quad (3.10)$$

both hold.

This fact allows one to formulate multiple \mathcal{H}_∞ objectives, to stack them, solve the stacked problem, and recover guarantees on the individual objectives again.

Note that the converse implication, however, is not true, but from (3.8) it is evident that the conservativeness is at most $\sqrt{2}$:

$$\|\mathbf{G}_1\|_\infty < 1 \text{ and } \|\mathbf{G}_2\|_\infty < 1 \Rightarrow \left\| \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix} \right\|_\infty < \sqrt{2}. \quad (3.11)$$

Mixed-sensitivity \mathcal{H}_∞ control

Disturbance rejection and tracking requirements give rise to so-called mixed-sensitivity designs. Two such problem classes are the $\mathbf{S}/\mathbf{K}\mathbf{S}$ and the \mathbf{S}/\mathbf{T} mixed-sensitivity designs. Thereby, \mathbf{S} refers to the closed-loop sensitivity function $\mathbf{S} = (\mathbf{I} + \mathbf{G}\mathbf{K})^{-1}$ and \mathbf{T} refers to the closed-loop complementary sensitivity function $\mathbf{T} = \mathbf{I} - \mathbf{S}$, see [143] for detailed interpretations.

Consider the three problems in Fig. 3.2: A regulation problem (a), a tracking problem (b), and a combined problem (c), in which both tracking and disturbance rejection are addressed. Formulating these problems with frequency-domain considerations leads to corresponding stacked \mathcal{H}_∞ formulations. These are solvable in the standard framework of \mathcal{H}_∞ -(sub-)optimal control design.

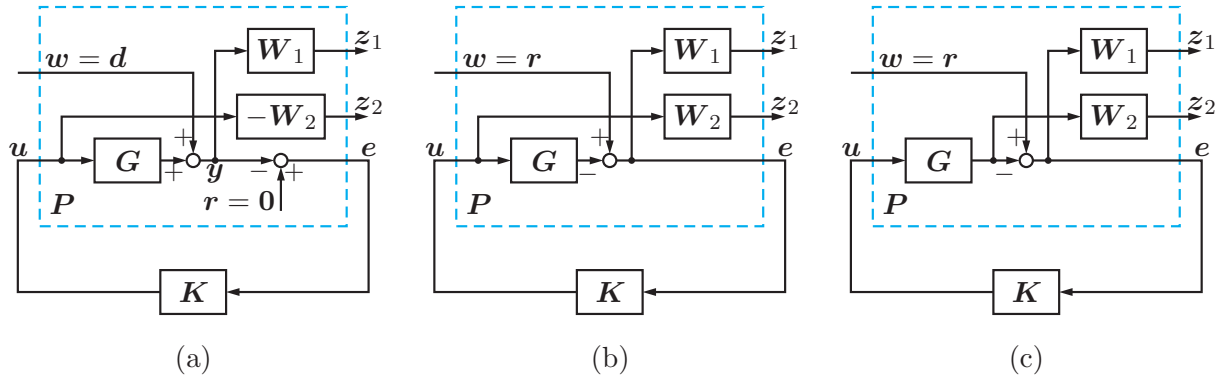


Figure 3.2: Mixed-sensitivity problems in standard form [143]: (a) regulation problem (S/KS -problem), (b) tracking problem (S/KS -problem), and (c) combined problem (S/T -problem)

The architecture in case (a) is obtained for a regulation (disturbance rejection) problem in which typically low-frequency disturbances $w = d$ should be rejected, that is, the magnitude of S should be small in that frequency range. This is obtained by a low-pass filter W_1 (typically chosen scalar or diagonal). Another typical constraint is limited actuation or control energy, which can be modeled in the design via limiting the weighted KS path, that is, the weighted transfer from $w = d$ to u . A high-pass W_2 penalizes large high-frequency magnitudes in u while allowing low-frequency control action. The augmented plant P in standard form is obtained from the block diagram as

$$\begin{bmatrix} z_1 \\ z_2 \\ e \end{bmatrix} = \underbrace{\begin{bmatrix} W_1 & W_1 G \\ \mathbf{0} & -W_2 \\ -\mathbf{I} & -G \end{bmatrix}}_P \begin{bmatrix} w \\ u \end{bmatrix}, \quad (3.12)$$

and the closed-loop performance transfer is

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \underbrace{\begin{bmatrix} W_1 S \\ W_2 K S \end{bmatrix}}_{T_{zw}} w. \quad (3.13)$$

The latter form shows that in fact a stacked \mathcal{H}_∞ criterion should be minimized:

$$\min_K \|T_{zw}\|_\infty = \min_K \left\| \begin{bmatrix} W_1 S \\ W_2 K S \end{bmatrix} \right\|_\infty. \quad (3.14)$$

When formulating a tracking problem as in case (b), the following requirements can be outlined: Firstly, the magnitude of the tracking error $e = r - y$ should be small (or zero) at low frequencies. This is accomplished by assigning a large weight at low frequencies (W_1 low-pass). Secondly, the control signal magnitude (especially at high frequencies, outside the tracking bandwidth) must be limited to small values. In the present approach, this is obtained by assigning a high-pass weight W_2 . With the augmented plant structured as

$$\begin{bmatrix} z_1 \\ z_2 \\ e \end{bmatrix} = \underbrace{\begin{bmatrix} W_1 & -W_1 G \\ \mathbf{0} & W_2 \\ \mathbf{I} & -G \end{bmatrix}}_P \begin{bmatrix} w \\ u \end{bmatrix}, \quad (3.15)$$

the same closed-loop performance transfer function \mathbf{T}_{zw} as in (3.13) and the same stacked \mathcal{H}_∞ criterion as in (3.14) is obtained.

The combined problem in case (c) is posed such that both, disturbance rejection (weighted sensitivity $\mathbf{W}_1\mathbf{S}$) and tracking (weighted complementary sensitivity $\mathbf{W}_2\mathbf{T}$) are considered simultaneously. As in (a), \mathbf{W}_1 is typically low-pass to reduce disturbance sensitivity at low frequencies, while \mathbf{W}_2 is typically high-pass to obtain good tracking and high-frequency noise attenuation: note that \mathbf{T} is the closed-loop transfer $\mathbf{y} = \mathbf{T}\mathbf{r}$, so a high-pass weight \mathbf{W}_2 avoids high-frequency tracking of noise, that is, performs noise attenuation. For tracking, \mathbf{T} should be close to \mathbf{I} at low frequencies, which is also obtained by making \mathbf{S} small there, because $\mathbf{T} = \mathbf{I} - \mathbf{S}$ holds. The augmented plant is

$$\begin{bmatrix} z_1 \\ z_2 \\ e \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{W}_1 & | & -\mathbf{W}_1\mathbf{G} \\ \mathbf{0} & | & \mathbf{W}_2\mathbf{G} \\ \mathbf{I} & | & -\mathbf{G} \end{bmatrix}}_P \begin{bmatrix} w \\ u \end{bmatrix}, \quad (3.16)$$

and the stacked \mathcal{H}_∞ criterion becomes

$$\min_{\mathbf{K}} \|\mathbf{T}_{zw}\|_\infty = \min_{\mathbf{K}} \left\| \begin{bmatrix} \mathbf{W}_1\mathbf{S} \\ \mathbf{W}_2\mathbf{T} \end{bmatrix} \right\|_\infty. \quad (3.17)$$

Sometimes, a combination of all three weighted terms \mathbf{S} (disturbance rejection), \mathbf{T} (tracking / noise attenuation), and $\mathbf{K}\mathbf{S}$ (control input magnitude limitation) is formulated in the same fashion, see [143].

The \mathcal{H}_∞ objectives also have clear interpretations in terms of robust stability:

- Small $\bar{\sigma}(\mathbf{K}(j\omega)\mathbf{S}(j\omega))$ corresponds to a large robustness margin against additive uncertainty at ω , and
- small $\bar{\sigma}(\mathbf{T}(j\omega))$ corresponds to a large robustness margin against (full complex) multiplicative output uncertainty.

Signal-based \mathcal{H}_∞ control, μ synthesis

The signal-based \mathcal{H}_∞ approach is a general onset suitable if several simultaneous \mathcal{H}_∞ MIMO objectives need to be accounted for, possibly including uncertain components. The expected frequency content of the exogenous input signals is modeled via weighting functions (shaping filters). Likewise, the exogenous output signals are defined as weighted error signals. These typically include frequency-weighted plant output signals, control input signals, and possibly tracking errors with respect to a reference model response. Moreover, unstructured or structured uncertainty models can be incorporated in the interconnection structure. Fig.3.3 shows an exemplary interconnection structure taken from [143]. The minimization of the performance path \mathcal{H}_∞ norm for all uncertainties $\|\Delta\|_\infty \leq 1$ leads to a robust performance problem: A controller \mathbf{K} is sought so that

$$\mu_{\hat{\Delta}_p}(\mathbf{M}(j\omega)) < 1 \quad \forall \omega \quad (3.18)$$

holds. The corresponding control design approach is called μ synthesis which attempts to find among all nominally internally stabilizing controller \mathbf{K} that which minimizes μ :

$$\mathbf{K}^* := \arg \min_{\mathbf{K}} \sup_{\omega} \mu_{\hat{\Delta}_p}(\mathbf{M}(j\omega)), \quad \mathbf{M} = \mathcal{F}_1(\mathbf{P}, \mathbf{K}) \quad (3.19)$$

This optimization can generally not be solved in closed form (that is, for a general uncertainty structure), but some heuristic approaches are widely recognized in practice. The most well-known algorithm is probably the DK-iteration or its extension, DGK-iteration which is detailed below in Sec. 3.5. Another approach, the so-called $\mathbf{Q} - \mu$ -synthesis (see Sec. 3.6.6) utilizes the Youla parametrization of the closed loop and additionally enables the use of nominal time-domain and \mathcal{H}_2 constraints simultaneously. Additionally, various LMI approaches have also been formulated to address specific uncertainty structures (see [11]).

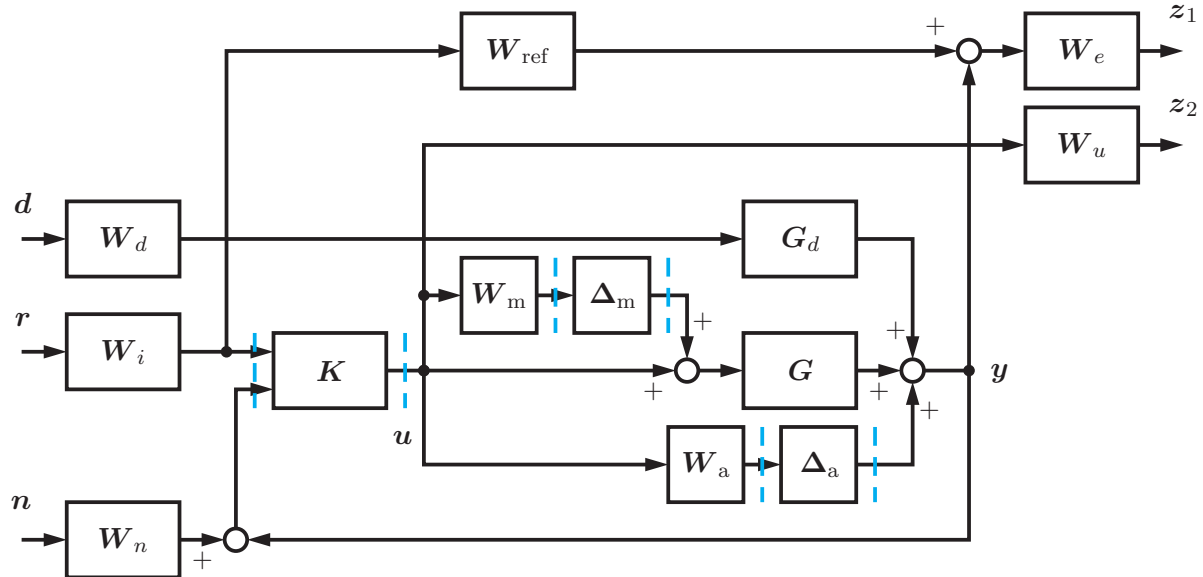


Figure 3.3: Typical augmented plant interconnection to design a two-degree-of-freedom controller for tracking under input constraints, noise, and uncertainties based on [143]. The dashed lines indicate the cuts necessary to reshape the interconnection into $\mathbf{P} - \mathbf{K} - \mathbf{\Delta}$ form.

3.1.2 Prerequisites for standard \mathcal{H}_2 and \mathcal{H}_∞ design

The central prerequisites to apply the standard methods for \mathcal{H}_2 and \mathcal{H}_∞ (sub-)optimal control design are outlined in the following. Note that current implementations of the algorithms as for example in the MATLAB[®] Robust Control Toolbox contain yet more general algorithms than those reviewed in the next sections to obtain sensible result when some of the posed assumptions are violated. These extensions are not covered here, the interested reader is referred to the MATLAB[®] documentation, the tools' source code and literature references therein for more details.

Definition 3.1.1: Assumptions for \mathcal{H}_2 and \mathcal{H}_∞ standard design problems

The following set of assumptions is posed on the standard feedback design problem statement for \mathcal{H}_2 and \mathcal{H}_∞ optimization, see [143]. Of these, A1-A4 are required and A5-A8 can (partially) be relaxed (in exchange for more complex solutions which are available in [24] but which are omitted here).

Assumption	Comment
A1.: $(\mathbf{A}, \mathbf{B}_2, \mathbf{C}_2)$ is stabilizable and detectable	required for the existence of a stabilizing \mathbf{K}
A2.: \mathbf{D}_{12} and \mathbf{D}_{21} have full rank	sufficient to ensure that \mathbf{K} is proper (i.e., realizable)
A3.: $\begin{bmatrix} \mathbf{A} - j\omega\mathbf{I} & \mathbf{B}_2 \\ \mathbf{C}_1 & \mathbf{D}_{12} \end{bmatrix}$ has full column rank for all ω	ensure that \mathbf{K} does not try to cancel poles or zeros on the imaginary axis which would render the closed loop unstable
A4.: $\begin{bmatrix} \mathbf{A} - j\omega\mathbf{I} & \mathbf{B}_1 \\ \mathbf{C}_2 & \mathbf{D}_{21} \end{bmatrix}$ has full row rank for all ω	
A5.: $\mathbf{D}_{11} = \mathbf{0}, \mathbf{D}_{22} = \mathbf{0}$	necessary for \mathcal{H}_2 case (strictly proper plants), simplifying the solution in the \mathcal{H}_∞ case (if not fulfilled, an equivalent problem in which A5 holds can be stated, see [143], [123])
A6.: $\mathbf{D}_{12} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}, \mathbf{D}_{21} = [\mathbf{0} \ \mathbf{I}]$	simplifying the solution, by scaling of \mathbf{u} and \mathbf{y} and a unitary transformation of \mathbf{w} and \mathbf{z} , this can always be obtained
A7.: $\mathbf{D}_{12}^T \mathbf{C}_1 = \mathbf{0}$ and $\mathbf{B}_1 \mathbf{D}_{21}^T = \mathbf{0}$	common for LQG control (no cross-coupling)
A8.: $(\mathbf{A}, \mathbf{B}_1)$ is stabilizable and $(\mathbf{A}, \mathbf{C}_1)$ is detectable	If A7 is true, A8 replaces A3 and A4.

3.2 \mathcal{H}_∞ suboptimal controller design

The standard solution for the \mathcal{H}_∞ suboptimal controller design problem is given in the following. This includes the central controller and the parametrization of all stabilizing suboptimal controllers and is summarized from the sources [24], [42], and [143].

Given a design plant $\mathbf{P}(s)$ configuration as in Fig. 3.1 which satisfies assumptions A1-A8 in Def. 3.1.1, all stabilizing controllers $\mathbf{K}(s)$ that satisfy

$$\|\mathcal{F}_1(\mathbf{P}, \mathbf{K})\|_\infty < \gamma \quad (3.20)$$

for a suboptimal bound $\gamma > \gamma_{\min}$ are sought.

They are obtained as follows [24, 42]:

1. Let $\mathbf{X}_\infty = \mathbf{X}_\infty^T \geq 0$ be a positive-semidefinite solution of the algebraic Riccati equation (ARE)

$$\mathbf{A}^T \mathbf{X}_\infty + \mathbf{X}_\infty \mathbf{A} + \mathbf{C}_1^T \mathbf{C}_1 + \mathbf{X}_\infty (\gamma^{-2} \mathbf{B}_1 \mathbf{B}_1^T - \mathbf{B}_2 \mathbf{B}_2^T) \mathbf{X}_\infty = \mathbf{0} \quad (3.21)$$

such that $\Re \{ \lambda_i [\mathbf{A} + (\gamma^{-2} \mathbf{B}_1 \mathbf{B}_1^T - \mathbf{B}_2 \mathbf{B}_2^T) \mathbf{X}_\infty] \} < 0 \ \forall i$ (called “stabilizing solution” of (3.21));

2. let $\mathbf{Y}_\infty = \mathbf{Y}_\infty^\top \geq 0$ be a positive-semidefinite solution of the ARE

$$\mathbf{A}\mathbf{Y}_\infty + \mathbf{Y}_\infty\mathbf{A}^\top + \mathbf{B}_1\mathbf{B}_1^\top + \mathbf{Y}_\infty(\gamma^{-2}\mathbf{C}_1^\top\mathbf{C}_1 - \mathbf{C}_2^\top\mathbf{C}_2)\mathbf{Y}_\infty = \mathbf{0} \quad (3.22)$$

such that $\Re\{\lambda_i[\mathbf{A} + \mathbf{Y}_\infty(\gamma^{-2}\mathbf{C}_1^\top\mathbf{C}_1 - \mathbf{C}_2^\top\mathbf{C}_2)]\} < 0 \quad \forall i$ (stabilizing solution of (3.22); and

3. let $\rho(\mathbf{X}_\infty\mathbf{Y}_\infty) < \gamma^2$ be fulfilled.

All controllers $\mathbf{K}(s)$ are given by $\mathbf{K} = \mathcal{F}_1(\mathbf{K}_{\text{cent}}, \mathbf{Q})$ where the ‘‘central controller’’ \mathbf{K}_{cent} is of the same order as \mathbf{P} and is given by

$$\mathbf{K}_{\text{cent}}(s) = \left[\begin{array}{c|cc} \mathbf{A}_\infty & -\mathbf{Z}_\infty\mathbf{L}_\infty & \mathbf{Z}_\infty\mathbf{B}_2 \\ \mathbf{F}_\infty & \mathbf{0} & \mathbf{I} \\ \hline -\mathbf{C}_2 & \mathbf{I} & \mathbf{0} \end{array} \right], \quad (3.23)$$

$$\mathbf{A}_\infty = \mathbf{A} + \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^\top\mathbf{X}_\infty + \mathbf{B}_2\mathbf{F}_\infty + \mathbf{Z}_\infty\mathbf{L}_\infty\mathbf{C}_2 \quad (3.24)$$

$$\mathbf{Z}_\infty = (\mathbf{I} - \gamma^{-2}\mathbf{Y}_\infty\mathbf{X}_\infty)^{-1} \quad (3.25)$$

$$\mathbf{L}_\infty = -\mathbf{Y}_\infty\mathbf{C}_2^\top \quad (3.26)$$

$$\mathbf{F}_\infty = -\mathbf{B}_2^\top\mathbf{X}_\infty \quad (3.27)$$

and $\mathbf{Q}(s)$ is any stable proper transfer function such that $\|\mathbf{Q}\|_\infty < \gamma$ holds.

If the stated conditions on the Riccati equations listed above (items 1–3) are not fulfilled, the tested value of γ is too small and hence infeasible. This is utilized to formulate a bisection algorithm, called γ -iteration, to search for a feasible value $\gamma \geq \gamma_{\min}$ close to the optimum to within a tolerance ε such that $\gamma - \gamma_{\min} < \varepsilon$ holds.

The MATLAB[®] Robust Control Toolbox algorithm `hinfscn` implements this procedure (γ -iteration and the solution to the suboptimal \mathcal{H}_∞ design problem with relaxed conditions). Therein, several methods to compute the solutions $\mathbf{X}_\infty, \mathbf{Y}_\infty$ are implemented. Also, assumption A2 in Def. 3.1.1 may be violated and a realizable controller for slightly perturbed entries in \mathbf{D} is computed in this case.

Remark: Note that the parametrization of all \mathcal{H}_∞ -suboptimal stabilizing controllers with $\mathbf{Q}(s)$ is in fact a Youla-parametrization of the closed-loop transfer function $\mathbf{T}_{zw}(s)$ which is affine in $\mathbf{Q}(s)$.

3.3 \mathcal{H}_2 optimal and suboptimal controller design

This section presents the standard solution for the \mathcal{H}_2 -optimal controller design problem from [24]. The optimal controller can also be viewed as central controller to a parametrization of a set of suboptimal controllers, analogous to the \mathcal{H}_∞ case.

Given a design plant $\mathbf{P}(s)$ configuration as in Fig. 3.1 which satisfies assumptions A1-A8 in Def. 3.1.1, the \mathcal{H}_2 -optimal controller $\mathbf{K}(s)$ is sought that leads to minimization of the \mathcal{H}_2 norm

$$\gamma_{2,\min} := \min_{\mathbf{K}} \|\mathbf{T}_{zw}\|_2 = \min_{\mathbf{K}} \|\mathcal{F}_1(\mathbf{P}, \mathbf{K})\|_2. \quad (3.28)$$

A unique optimal solution to this problem can directly be computed by solving two AREs, contrary to the \mathcal{H}_∞ controller design case.

Analogous to the \mathcal{H}_∞ design case, the suboptimal \mathcal{H}_2 design problem is defined as: Find all internally stabilizing controllers that achieve

$$\|\mathbf{T}_{zw}\|_2 = \|\mathcal{F}_1(\mathbf{P}, \mathbf{K})\|_2 < \gamma \quad (3.29)$$

where $\gamma > \gamma_{2,\min}$.

It turns out that the solution to the optimal \mathcal{H}_2 design problem (3.28) can be utilized to formulate a central controller to parameterize all stabilizing controllers that fulfill (3.29) for a suboptimal γ , see [24].

The unique optimal controller is obtained as follows:

1. Let $\mathbf{X}_2 = \mathbf{X}_2^T \geq 0$ be a positive-semidefinite solution of the ARE

$$\mathbf{A}^T \mathbf{X}_2 + \mathbf{X}_2 \mathbf{A} + \mathbf{C}_1^T \mathbf{C}_1 - \mathbf{X}_2 \mathbf{B}_2 \mathbf{B}_2^T \mathbf{X}_2 = \mathbf{0} \quad (3.30)$$

such that $\Re \{ \lambda_i [\mathbf{A} - \mathbf{B}_2 \mathbf{B}_2^T \mathbf{X}_2] \} < 0 \quad \forall i$ (stabilizing solution);

2. let $\mathbf{Y}_2 = \mathbf{Y}_2^T \geq 0$ be a positive-semidefinite solution of the ARE

$$\mathbf{A} \mathbf{Y}_2 + \mathbf{Y}_2 \mathbf{A}^T + \mathbf{B}_1 \mathbf{B}_1^T - \mathbf{Y}_2 \mathbf{C}_2^T \mathbf{C}_2 \mathbf{Y}_2 = \mathbf{0} \quad (3.31)$$

such that $\Re \{ \lambda_i [\mathbf{A} - \mathbf{Y}_2 \mathbf{C}_2^T \mathbf{C}_2] \} < 0 \quad \forall i$ (stabilizing solution).

Then, with

$$\mathbf{F}_2 = -\mathbf{B}_2^T \mathbf{X}_2 \quad (3.32)$$

$$\mathbf{L}_2 = -\mathbf{Y}_2 \mathbf{C}_2^T, \text{ and} \quad (3.33)$$

$$\widehat{\mathbf{A}}_2 = \mathbf{A} + \mathbf{B}_2 \mathbf{F}_2 + \mathbf{L}_2 \mathbf{C}_2, \quad (3.34)$$

the central controller is

$$\mathbf{K}_{2\text{cent}}(s) = \left[\begin{array}{c|cc} \widehat{\mathbf{A}}_2 & -\mathbf{L}_2 & \mathbf{B}_2 \\ \mathbf{F}_2 & \mathbf{0} & \mathbf{I} \\ \hline -\mathbf{C}_2 & \mathbf{I} & \mathbf{0} \end{array} \right], \quad (3.35)$$

and the family of all stabilizing controllers such that $\|\mathbf{T}_{zw}\|_2 \leq \gamma$ holds is obtained by

$$\mathbf{K}_2(s) = \mathcal{F}_1(\mathbf{K}_{2\text{cent}}, \mathbf{Q}) \quad (3.36)$$

where $\mathbf{Q}(s)$ is stable, strictly proper, and fulfills $\|\mathbf{Q}\|_2^2 < \gamma^2 - \gamma_{2,\min}^2$. The optimal controller is obtained from (3.36) by setting $\mathbf{Q}(s) = \mathbf{0}$.

The MATLAB[®] Robust Control Toolbox algorithm `h2syn` implements the optimal \mathcal{H}_2 design procedure. It also treats non-strictly-proper plants (with $\mathbf{D}_{11} \neq \mathbf{0}$) by ignoring this direct feedthrough term and returning the resulting optimal controller. The AREs are solved via MATLAB[®]'s `care` and `dare` commands for the continuous-time and for the discrete-time cases, respectively.

Remark: Note the similarities between the \mathcal{H}_∞ and the \mathcal{H}_2 suboptimal control design procedures, in particular the AREs which are equivalent except for the γ^{-2} -term in the \mathcal{H}_∞ -case. This indicates that if relaxing the γ -bound in \mathcal{H}_∞ design to $\gamma \rightarrow \infty$, the design becomes an \mathcal{H}_2 -optimal design (if all assumptions are fulfilled and an \mathcal{H}_2 design is feasible). Loosely speaking, an \mathcal{H}_∞ design with a γ -bound which is not close to γ_{\min} can be interpreted as a mixture of \mathcal{H}_∞ and \mathcal{H}_2 optimality. The MATLAB[®] Robust Control Toolbox provides the algorithm `h2hinfsyn` which synthesizes a controller for specified \mathcal{H}_2 , \mathcal{H}_∞ , and pole region constraints (although with a different approach via an LMI formulation).

3.4 LQ-based Optimal Control

3.4.1 Linear Quadratic Gaussian (LQG) Control

Linear Quadratic Gaussian (LQG) control became popular in the 1960s, especially for aerospace applications, and was one of the main drivers of the rise of optimal control [143, pp. 344–352], [40], [84]. Let a linear dynamic plant be given with stochastic noise excitation of known statistical properties,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \quad (3.37)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{v}, \quad (3.38)$$

where the noise signals \mathbf{w} , \mathbf{v} are in its basic formulation assumed to be uncorrelated zero-mean Gaussian stochastic processes with constant power spectral density matrices \mathbf{W} and \mathbf{V} , respectively. For compactness and for direct interpretation of the LQG design in terms of \mathcal{H}_2 -optimal design, only the strictly-proper case $\mathbf{D} = \mathbf{0}$ is considered, however LQG control is also applicable to bi-proper plants with $\mathbf{D} \neq \mathbf{0}$.

LQG stands for (the derivation of) a control law that is optimal in the following sense:

Theorem 3.4.1: LQG control problem statement

Given the system in (3.37)–(3.38), find the optimal control $\mathbf{u}(t)$ which minimizes

$$J = \mathbb{E} \left\{ \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)] dt \right\}, \quad (3.39)$$

where $\mathbb{E}\{\cdot\}$ is the expected value operator, \mathbf{Q} and \mathbf{R} are constant real matrices (design parameters) where $\mathbf{Q} = \mathbf{Q}^T \geq 0$ (\mathbf{Q} symmetric and positive semi-definite) and $\mathbf{R} = \mathbf{R}^T > 0$ (\mathbf{R} symmetric and positive definite) hold. The signals' time dependency is denoted explicitly here for a correct formulation, but dropped in the following in favor of a clear exposition.

It turns out that the optimal solution decomposes into finding a constant state vector feedback gain \mathbf{K}_c via the associated deterministic LQR problem and computing a dynamic Kalman state estimator (observer) of plant order n , see Fig. 3.4 where \mathbf{K}_{est} is the Kalman estimator. Its state estimate $\hat{\mathbf{x}}$ is utilized in place of the real (non-measurable) state vector \mathbf{x} . [143]

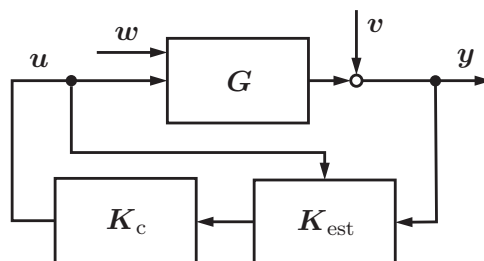


Figure 3.4: LQG architecture

LQR Problem and Solution

The LQR (Linear Quadratic Regulator) problem is to find the input signal $\mathbf{u} = \mathbf{u}(t)$ to minimize the following objective function in \mathbf{u} and $\mathbf{x} = \mathbf{x}(t)$:

$$J_{\text{LQR}} = \int_0^{\infty} [\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)] dt. \quad (3.40)$$

The optimal solution for any initial state $\mathbf{x}(0)$ is

$$\mathbf{u}(t) = -\mathbf{K}_c\mathbf{x}(t) \quad (3.41)$$

$$\mathbf{K}_c = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{X}, \quad (3.42)$$

where $\mathbf{X} = \mathbf{X}^T \geq 0$ is the unique symmetric and positive semi-definite solution of the controller algebraic Riccati equation (CARE)

$$\mathbf{A}^T\mathbf{X} + \mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{X} + \mathbf{Q} = \mathbf{0}. \quad (3.43)$$

Kalman Filter

The Kalman filter or estimator \mathbf{K}_{est} , also called Kalman-Bucy filter, has the same structure as an ordinary Luenberger observer (see Sec. B.1) with the dynamics

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}), \quad (3.44)$$

where the optimal filter gain

$$\mathbf{L} = \mathbf{Y}\mathbf{C}^T\mathbf{V}^{-1} \quad (3.45)$$

minimizes the expected observation error covariance $\text{E}\{[\mathbf{x} - \hat{\mathbf{x}}]^T[\mathbf{x} - \hat{\mathbf{x}}]\}$, see [45]. The matrix $\mathbf{Y} = \mathbf{Y}^T \geq 0$ in (3.45) is the unique symmetric and positive semi-definite solution of the associated filter algebraic Riccati equation (FARE):

$$\mathbf{Y}\mathbf{A}^T + \mathbf{A}\mathbf{Y} - \mathbf{Y}\mathbf{C}^T\mathbf{V}^{-1}\mathbf{C}\mathbf{Y} + \mathbf{E}\mathbf{W}\mathbf{E}^T = \mathbf{0}. \quad (3.46)$$

Existence and Computation

The solutions to the CARE (3.43) and FARE (3.46) and thus the LQG controller exist if the state-space systems $(\mathbf{A}, \mathbf{B}, \mathbf{Q}^{\frac{1}{2}})$ and $(\mathbf{A}, \mathbf{W}^{\frac{1}{2}}, \mathbf{C})$ are stabilizable and detectable.

Note that in obtaining the LQR feedback gain \mathbf{K}_c and the Kalman filter gain \mathbf{L} the algorithm is equivalent, and only the inserted quantities differ. The MATLAB[®] command¹ `care` can be utilized to design \mathbf{K}_c and \mathbf{L} in the following way:

```
[X, lambda, Kc] = care(A, B, Q, R)
[Y_t, lambda, L_t] = care(A', C', W', V')
L = L_t'
```

¹Note that the MATLAB[®] command `care` abbreviates *continuous-time algebraic Riccati equation* versus the command `dare` for the discrete-time case.

Relation to \mathcal{H}_2 -optimal control design

The LQG problem with a strictly proper system (3.37)–(3.38) with $\mathbf{D} = \mathbf{0}$ and objective function (3.39) is a special case of the \mathcal{H}_2 -optimal control design problem. Consider for simplicity that $\mathbf{E} = \mathbf{I}$ holds. Then, the exogenous output signal

$$\mathbf{z} = \begin{bmatrix} \mathbf{Q}^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \quad (3.47)$$

is defined. An $((n + n_y) \times 1)$ exogenous input vector $\tilde{\mathbf{w}}$ is considered, which is a unit intensity white noise process (n_y is the number of outputs). Then the LQG problem's noise variables are:

$$\begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{W}^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^{\frac{1}{2}} \end{bmatrix} \tilde{\mathbf{w}}. \quad (3.48)$$

With these transformations, the expected value of $\mathbf{z}^T \mathbf{z}$ is equal to the LQG cost function and at the same time the squared \mathcal{H}_2 -norm of the closed-loop system. Thus the (unique, optimal) LQG controller is equal to the unique \mathcal{H}_2 -optimal controller of the associated \mathcal{H}_2 design problem. For the structure of the augmented plant in this case see [143].

3.4.2 Output-weighted LQG Control Design

The output-weighted LQG design variant minimizes an integral quadratic function of the output and the control input variables (instead of states and control inputs) of the form

$$J = \mathbb{E} \left\{ \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \left[\mathbf{y}^T(t) \tilde{\mathbf{Q}} \mathbf{y}(t) + \mathbf{u}^T(t) \tilde{\mathbf{R}} \mathbf{u}(t) \right] dt \right\}, \quad (3.49)$$

which again decomposes into the state-feedback and the Kalman state estimation problems. While the Kalman estimator is unchanged because it only depends on the noise properties, the LQR gain \mathbf{K}_c has to minimize

$$J_{\text{LQR},y} = \int_0^\infty \left(\mathbf{y}^T(t) \tilde{\mathbf{Q}} \mathbf{y}(t) + \mathbf{u}^T(t) \tilde{\mathbf{R}} \mathbf{u}(t) \right) dt. \quad (3.50)$$

The solution can be found by eliminating $\mathbf{y}(t)$ in (3.50) with the deterministic output equation (3.38) (without noise). Then, the following relations,

$$\mathbf{Q} = \mathbf{C}^T \tilde{\mathbf{Q}} \mathbf{C}, \quad (3.51)$$

$$\mathbf{R} = \mathbf{D}^T \tilde{\mathbf{Q}} \mathbf{D} + \tilde{\mathbf{R}}, \quad (3.52)$$

hold for the equivalent formulation of (3.40) and the standard LQR solution procedure can be used.

3.4.3 LQI Tracking Control Design

The LQG control design can be extended by zero position error properties when using the LQI architecture as shown in Fig. 3.5. The design plant is thereby augmented by states which integrate the control error(s) $\mathbf{r} - \mathbf{y}$, where \mathbf{r} is the reference signal that should be tracked:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_i \end{bmatrix}, \quad (3.53)$$

with $\dot{\mathbf{x}}_i = \mathbf{r} - \mathbf{y}$. This formulation yields asymptotic tracking of all outputs \mathbf{y} . If only a subset of the output signals should be tracked, the architecture has to be adapted accordingly.

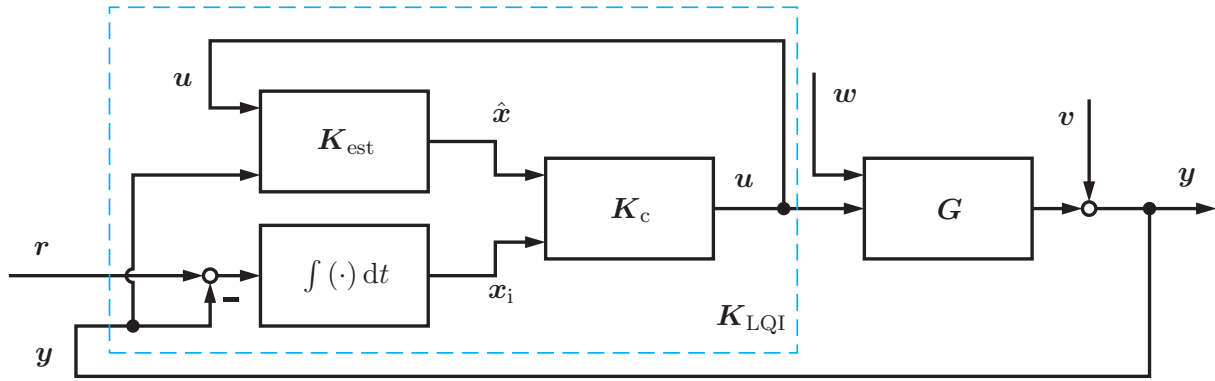


Figure 3.5: LQI architecture

3.5 μ -Synthesis: DK-Iteration and DGK-Iteration Design

As outlined in Sec. 3.1.1, the problem of μ synthesis is to design a feedback controller \mathbf{K} that, interconnected to an uncertain augmented plant \mathbf{P} , attains a minimal, or at least a small value of a specified μ condition.

For general uncertainty structures, no computationally tractable direct method is known to obtain a μ -optimal controller. However, often good results are obtained by splitting the μ synthesis problems into μ analysis and controller synthesis subproblems which are solved in an alternating, iterative fashion. For complex perturbations, the DK-iteration algorithm [6] produces good results. Its extension, named DGK-iteration (also referred to as mixed- μ synthesis) addresses problems with mixed (i.e., real and complex) structured perturbations. The basic structure of these algorithms is outlined below and consists of alternate steps of μ analysis, problem scaling, and \mathcal{H}_∞ (sub-)optimal controller design of the scaled problem.

3.5.1 Complex perturbations & DK-Iteration

For complex perturbations, an upper bound on μ is

$$\mu(N(j\omega)) \leq \min_{\mathbf{D}(j\omega) \in \mathcal{D}} \bar{\sigma}(\mathbf{D}(j\omega)N(j\omega)\mathbf{D}^{-1}(j\omega)), \quad (3.54)$$

where $\mathcal{D} = \{\mathbf{D} : \mathbf{D}\mathbf{\Delta} = \mathbf{\Delta}\mathbf{D}\}$ is the set of all matrices that commute with the uncertainty block $\mathbf{\Delta}$.

This gives rise to the basic idea of DK-iteration, namely to find the controller \mathbf{K} that minimizes the peak of this upper bound over all frequencies by alternately optimizing over the set of stabilizing \mathcal{H}_∞ sub-optimal controllers $\mathbf{K}(s)$ and over the set of (approximated dynamic) scalings $\mathbf{D}(s)$, expressed as:

$$\min_{\mathbf{K}} \min_{\mathbf{D}(s) \in \mathcal{D}(s)} \|\mathbf{D}\mathbf{N}\mathbf{D}^{-1}\|_\infty. \quad (3.55)$$

The algorithm iterates as follows:

1. Initialize $\mathbf{D}(s)$, typically with \mathbf{I} (provided that the problem is reasonably scaled)
2. K-step: Design an internally stabilizing controller $\mathbf{K}(s)$ for the scaled plant by solving the \mathcal{H}_∞ (sub-)optimal controller design problem $\min_{\mathbf{K}} \|\mathbf{D}\mathbf{N}(\mathbf{K})\mathbf{D}^{-1}\|_\infty$.

3. D-step: Find $\widehat{\mathbf{D}}(j\omega)$ that minimizes $\bar{\sigma}\left(\widehat{\mathbf{D}}(j\omega)\mathbf{N}(j\omega)\widehat{\mathbf{D}}^{-1}(j\omega)\right)$ for each frequency in a predefined frequency gridding, where $\mathbf{N} = \mathcal{F}_1(\mathbf{P}, \mathbf{K})$ is kept fixed. If μ -based termination criterion is fulfilled, stop.
4. Approximate the magnitude of each element in $\widehat{\mathbf{D}}(j\omega)$ by a stable and minimum-phase transfer function to obtain the dynamic scaling transfer function $\mathbf{D}(s)$. Go to step 2.

While this algorithm has the potential to solve the important μ synthesis problem in practice well enough, it exhibits some limitations and disadvantages that should be noted. One fundamental problem is that the combined optimization problem (K-step and D-step) is not convex although the individual steps are. This results in two issues: Firstly, the combined problem does not necessarily have a unique global optimum but may have several local optima. Secondly, convergence is not guaranteed, so an actual increase in the μ value from one iteration to the next can occur. Another issue is the (high) dynamic order of the resulting controller. Because the \mathcal{H}_∞ controller is designed upon the dynamically scaled plant, its order is $n_K = n_P + 2n_D$ where n_P is the order of the augmented plant (that is, the order of the plant and the order of all dynamic weightings in the augmented plant interconnection) and n_D is the total dynamic order of the scaling transfer function $\mathbf{D}(s)$. Thus, the controller order can quickly inflate to the order of hundreds which poses problems both numerically in the design as well as in controller implementation. Wise, low-order modeling of the relevant plant characteristics is therefore necessary to obtain a well-posed design problem and to obtain a reasonably-sized controller. Usually, controller order reduction methods are applied a posteriori to vastly reduce its complexity, see Chap. 7.

Despite these drawbacks, DK-iteration is usually a well-performing algorithm if the problem is carefully formulated and well-scaled.

3.5.2 Real and complex perturbations & DGK-Iteration

The DGK-iteration aims to solve the μ synthesis problem in presence of mixed real and complex perturbations. It utilizes a generalized upper bound [143, 172] for mixed μ , formulated as follows: Let \mathbf{M} and \mathbf{D} be the complex transfer matrices of $\mathbf{M}(s)$ and $\mathbf{D}(s)$ at frequency $s = j\omega$, respectively. If there exist a real scalar $\beta > 0$ and complex \mathbf{D} and real \mathbf{G} matrices of specific block-diagonal structure (see below), so that

$$\bar{\sigma}\left(\left(\mathbf{I} + \mathbf{G}^2\right)^{-\frac{1}{4}}\left(\frac{1}{\beta}\mathbf{D}\mathbf{M}\mathbf{D}^{-1} - \mathbf{j}\mathbf{G}\right)\left(\mathbf{I} + \mathbf{G}^2\right)^{-\frac{1}{4}}\right) \leq 1 \quad (3.56)$$

is fulfilled, then

$$\mu(\mathbf{M}) \leq \beta \quad (3.57)$$

holds. Thereby, \mathbf{D} are scalings exploiting the complex uncertainty structure (as with the DK-iteration), and \mathbf{G} are real-valued diagonal scaling matrices which exploit real perturbations with real nonzero entries at the locations of the real parameters in the uncertain block $\mathbf{\Delta}$.

The DGK-iteration algorithm (presented in [172]) contains the same basic steps as the DK-iteration, albeit extended by evaluating the mixed μ bound above and approximating additional dynamic scalings $\mathbf{G}(s)$ in the scaling step. The algorithm is by now available as part of the MATLAB[®] Robust Control Toolbox and can in cases with significant real structure in the uncertainties obtain superior results. However, note that the increased number of degrees of freedom in scaling further inflates the controller size which aggravates numeric issues.

3.6 Convex Controller Synthesis

This section introduces the concept of convex controller synthesis for feedback and feed-forward control configurations. The feedback case is addressed by the Youla parametrization which represents the closed loop affine in the designed Youla parameter, while the feed-forward case is directly affine in the controller transfer. The affine representation enables the formulation and efficient solution of convex optimization problems to directly optimize and constrain the controlled system's time- and frequency-domain responses.

After an overview, motivation and critical discussion of the onset, a detailed account on the literature and state of the art in convex control design methods is given. This is organized in the common sequence of design tasks in the feedback and, subsequently, the feed-forward control design cases:

- First, strategies and relevant considerations in the design on initially stabilizing controllers are discussed.
- Second, the Youla parametrization of the closed loop is explained.
- Next, the formulation of the most commonly used \mathcal{H}_∞ , \mathcal{H}_2 , and time-domain objectives and constraints is shown, leading to a convex optimization problem.
- Finally, some remarks on available numeric solvers, on the choice of basis functions, and on extensions to robust control design are given.

3.6.1 Motivation & Overview

The central challenge for a control engineer is to translate the given specifications efficiently into design parameters for the utilized synthesis methods (usually from the optimal or robust control domains). Typically, these constraints are either stated as weighting functions in the frequency domain (\mathcal{H}_∞ / \mathcal{H}_2 control, DK-iterations) or as objective function weightings (as in LQ control). It turns out that many important time- and frequency-domain requirements can be formulated as (quasi-)convex constraints or objectives of an optimization problem [11].

The term *Convex Synthesis* spans techniques to synthesize control laws (feed-forward and / or feedback) in which the problem is cast into and solved as convex optimization problem over a finite set of design parameters. This typically leads to a computationally tractable design algorithm, and many such design problems yield linear program, linear programming (LP), quadratic program, quadratic programming (QP), or LMI problems which can be solved efficiently with available tools.

Convex design for the control of conventional flexible aircraft has been studied, among others, in the PhD thesis [18] as well as in [118] (with subsequent controller order reduction) and in [151] (a self-scheduling approach). The authors of ref. [32] consider robust convex feed-forward design when an analytic uncertainty description is available.

To efficiently solve control design problems it is often of great help if these can be formulated as convex optimization problems. Firstly, this ensures the existence of one global optimum, and secondly, highly efficient algorithms are often available to solve even large problems quickly. As stated in Sec. 2.4.2, a convex optimization problem is comprised of convex objective and constraint functions as well as a convex admissible set. One class of convex functions are affine functions². These often arise in the modeling of control design problems: Consider, for example, a feed-forward control design problem where the controller is generated as the weighted sum of dynamic basis functions. These weights constitute the decision variables of the design problem.

²Affine functions are of the form $f(x) = a + bx$. In contrast, linear functions in optimization literature do not contain the affine term a : $f(x) = bx$.

The controlled system transfer function, its step responses, and even some system norms are affine functions of these weights. A versatile multi-model approach to feed-forward control design is presented in Sec. 5.4.

For feedback control design, such direct parametrization of the controller unfortunately does not yield affine closed-loop expressions. Many results in control theory address this central difficulty brought about by the closure of the feedback loop: the \mathcal{H}_∞ suboptimal design in Sec. 3.2, for example, utilizes a different representation (the parametrization of all internally stabilizing controllers) which allows to express the suboptimal \mathcal{H}_∞ design problem as convex problem. Here, the closely-related so-called Youla parametrization is utilized to obtain an affine, convex representation of the closed-loop transfer paths, see Sec. 3.6.3.

In robust control applications, robust stability of the closed-loop, which means internal stability in the presence of the modeled, bounded uncertainties, is usually the most fundamental requirement. However, one additional, important requirement for reliable control is the stability of the controller itself (referred to as strong stabilization, see for example [69] and [154]), which is not guaranteed by standard optimal and robust design methods. This is however imperative in the case of potential actuator or sensor faults and in cases when the actuators saturate. Some solutions to strong stabilization control design are available in the literature, but to the author's knowledge no approach exists that preserves affinity of the closed loop in the problem representation. This, however, is necessary to incorporate time- and frequency-domain constraints and thus highly beneficial for the control design task. A non-convex strong stabilization constraint is presented in Sec. 5.2, along with a convex embedding approximation method. Numeric results are given and discussed.

Comparison to standard \mathcal{H}_∞ suboptimal control design

When comparing convex control design to the \mathcal{H}_∞ suboptimal control design approach, some notable differences arise. In the following, a listing of main advantages and disadvantages of convex synthesis methods over standard \mathcal{H}_∞ methods, as listed in [20], is given from the control engineer's perspective. The advantages are:

- The optimization problem is convex and can thus be solved efficiently also for large problems (however the robust feedback control law synthesis is non-convex, just as DK-iteration is not). The number of free variables of the optimization problem is typically not dependent on the order of the plant, but on the chosen number of free parameters or basis functions of the designed controller.
- Frequency- *and* time-domain specifications can directly be incorporated into the design. However, this strongly impacts the number of optimization problem constraints, see below.
- These specifications can be given in terms of non-parametric time-domain or frequency-domain templates (no weighting functions need to be found, and thus no dynamic performance weightings need to be included in the augmented plant). Consequently, the controller order is independent of the type and shape of the defined constraints and objectives.
- The objectives and constraints can be defined individually per I/O-channel, whereas for \mathcal{H}_∞ design only the entire (weighted) closed-loop transfer norm from exogenous inputs to exogenous outputs can be bounded. This eliminates one source of conservativeness.
- If a solution is found, it is a global optimizer in the utilized search space (i.e. over the span of the chosen filter basis).

However, disadvantages and caveats of the convex synthesis methods also arise:

- The presented approach evaluates time- and frequency-domain objectives and constraints on a (finite) frequency gridding. Each instance of each constraint or objective at a considered gridpoint yields a constraint in the optimization problem which produces a conflict between optimization problem constraint count and time-/frequency resolution of the actual constraints and objectives. An efficient solution followed here, starting at a relaxed formulation with few defined optimization problem constraints, is to adaptively add necessary constraints and re-solve the problem until the solution fulfills all constraints and objectives sufficiently well on fine validation grids. This is realized as a core feature of the convex optimization framework developed in Sec. 5.1.
- When utilizing a large set of basis functions, the resulting controller order can be very large. It is, however, fixed and known a priori, in contrast to controllers obtained by DK-iteration with dynamic scalings.
- The achievable performance depends strongly on the choice of the Youla parameter filter basis. The finite-dimensional subset of proper stable transfer functions, parametrized by the design parameters, is an inherent limitation or constraint to the algorithm. Methods to choose suitable basis functions are discussed in Sec. 3.6.5.

3.6.2 Initial controller design

Before convex control design can be applied, the plant must be stabilized and, in many cases, its dynamics must be pre-shaped by a suitable initial controller because of two reasons: First, the the poles of the initial closed loop (that is, the plant interconnected with an initial feedback controller) **cannot** be shifted by the convex control design any more. This necessitates initial internal stabilization. Second, correct pole placement by the initial controller is efficient and thus important. Convex control design cannot shift the poles, resulting in a large effort in terms of controller complexity and authority to obtain similar effects in the closed-loop response by convex design alone.

Suitable initial controllers can be designed by any method, such as pole placement respectively eigenstructure assignment or other types of feedback designs (LQ-based, \mathcal{H}_∞ -or \mathcal{H}_2 -designs, for example). Two general concepts of designing an initial controller are of particular interest with respect to convex control design:

1. One can design a low-order controller which robustly shapes the relevant dynamics and absorb it into the plant. Then, this pre-shaped plant is considered as open-loop stable system and the Youla parametrization can directly be constructed. This approach is possible for all internally stabilizing controllers, but the controller resulting of the convex control design may be of high order.
2. If, in turn, a high-order initial controller (of full or augmented order, that is, of at least the order of the design plant) can be obtained and transformed to an observer-based realization (see Sec. B.3), an efficient Youla parametrization can be built.

Remark: Convex control design has its clear advantages in the incorporation of time-domain constraints and can be utilized to improve or extend a given control law. In a robust design, however, convex control design can also address robust stability because only the nominal plant poles are immobile; this is not true for perturbed plants. One such methodology, realizing a μ synthesis approach in the convex control design approach is the $Q - \mu$ -synthesis, see Sec. 3.6.6.

Robust Partial Eigenstructure Assignment

One particularly suitable way of designing a robust low-order initial controller for robust control design is by performing robust partial eigenstructure assignment. Static and low-order dynamic output feedback laws can be designed by using the techniques and tools given in [93]. This type of control design shows the following advantages with respect to subsequent convex control design:

- The complementary task to convex control design, namely eigenstructure (pole and eigenvector) assignment, is directly addressed.
- The resulting control laws are typically of low dynamic order (if only few poles need to be shifted) and have a well-defined structure (typically consisting of a static feedback gain, elementary/modal observers, and low-order I/O-filters).
- The design produces robustified (more exact: insensitive) eigenstructure assignment with good results in many applications, see [93].

However, note also the following points of caution:

- Internal stability or robust stability is not guaranteed by the design. This requires repetitive design and validation cycles, some tuning, as well as some amount of trial-and-error.
- Observer design and optimization as carried out in the toolbox is randomized. This can yield unreliable or non-reproducible design results for sensitive plants (which has been observed in the works of [157]).

The basic eigenstructure assignment approach works as follows, refer to [93] for details and various extensions: Consider a state-space system \mathbf{P} of order n with n_y outputs in state space representation,

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}.\end{aligned}\tag{3.58}$$

The goal is to assign q triplets $(\lambda_i, \mathbf{v}_i, \mathbf{w}_i)$ (eigenvalue, input-, and output direction, respectively) in closed loop, where $q \leq n_y$. Denote $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_q]$, $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_q]$ and $\mathbf{X} = \mathbf{C}\mathbf{V} + \mathbf{D}\mathbf{W}$.

Then, the partial eigenstructure assignment is performed by the static output feedback law

$$\mathbf{u} = \mathbf{K}\mathbf{y}\tag{3.59}$$

with

$$\mathbf{K} = \mathbf{W}\mathbf{X}^\dagger\tag{3.60}$$

where \mathbf{X}^\dagger denotes the Moore-Penrose matrix pseudo-inverse ($\mathbf{X}^\dagger = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$) which is equivalent to the matrix inverse for the case $q = n_y$. The choice of the Moore-Penrose matrix pseudo-inverse is taken to obtain the minimum-norm gain \mathbf{K} , but other choices are possible, see [93].

3.6.3 Youla parametrization

Consider a plant \mathbf{P} and a stabilizing controller \mathbf{K} in standard form as in Fig. 2.1, such that

$$\mathbf{T} = \mathcal{F}_1(\mathbf{P}, \mathbf{K}) = \mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(\mathbf{I} - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}\tag{3.61}$$

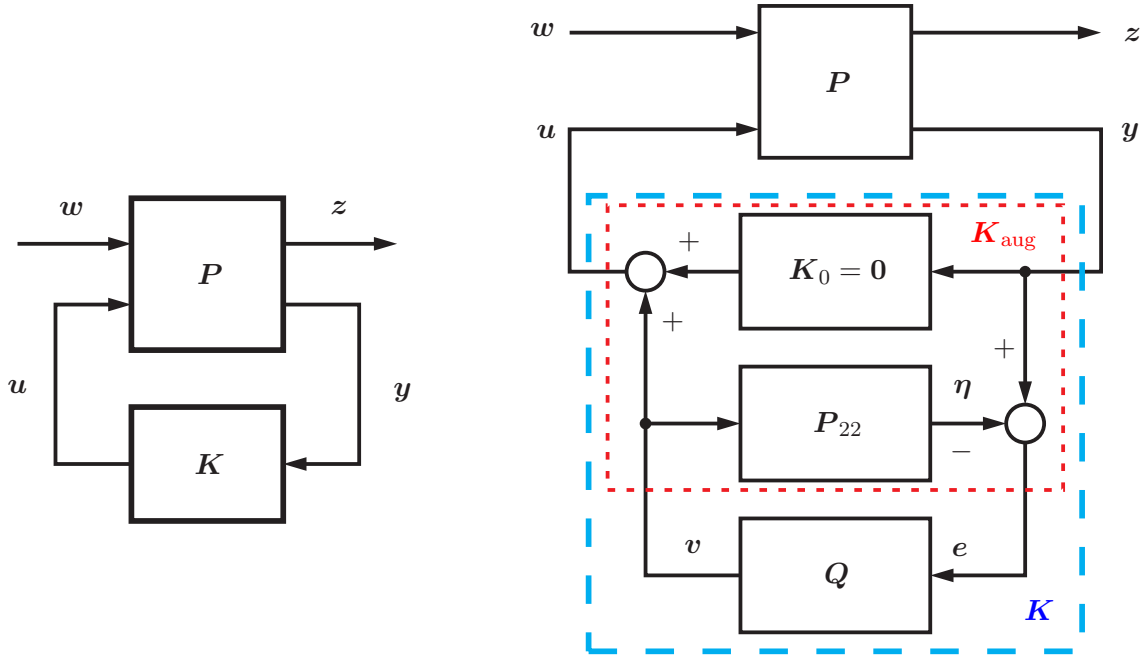


Figure 3.6: Standard feedback interconnection (left), Youla parametrization (right) for a stable plant P and, formally, a zero feedback $K_0 = 0$

with all terms P_{11} , P_{12} , K , P_{21} , and $(I - P_{22}K)^{-1}$ stable.

The Youla parametrization adds auxiliary controller inputs v and outputs e (yielding an augmented controller K_{aug}), such that the *closed-loop transfer* T_{ev} is zero. Connecting an arbitrary stable and compatible transfer function $Q(s)$ (Youla parameter) with free order between e and v parameterizes all internally stabilizing controllers K affinely in $Q(s)$. In the case of an initially stable plant (or a stabilized plant with an absorbed initial controller), one possible extension with these properties is depicted in Fig. 3.6 (right), see [11]. Equation (3.61) becomes:

$$T = T_1 + T_2 Q T_3, \quad (3.62)$$

$$T_1 = P_{11}, \quad T_2 = P_{12}, \quad T_3 = P_{21}, \quad (3.63)$$

$$Q = K(I - P_{22}K)^{-1}, \quad (3.64)$$

$$K = (I + QP_{22})^{-1}Q. \quad (3.65)$$

In implementations $Q(s)$ has to be restricted to a finite, weighted sum of n basis functions. A parameter-affine state-space realization

$$Q = \sum_{i=1}^{n_Q} Q_i(s) \theta_i \quad Q_i = \left[\begin{array}{c|c} A_{Q_i} & B_{Q_i} \\ \hline C_{Q_i} & D_{Q_i} \end{array} \right] \quad \theta = [\theta_1, \dots, \theta_{n_Q}]^T \quad (3.66)$$

can be formed with input- or output-parametrization:

$$Q_i(s) \theta_i = \left[\begin{array}{c|c} A_{Q_i} & B_{Q_i} \theta_i \\ \hline C_{Q_i} & D_{Q_i} \theta_i \end{array} \right] \quad \text{or} \quad \left[\begin{array}{c|c} A_{Q_i} & B_{Q_i} \\ \hline \theta_i C_{Q_i} & \theta_i D_{Q_i} \end{array} \right]. \quad (3.67)$$

With P_{22} (strictly proper, which ensures the existence of a proper controller K) and Q (both stable) in state space representation,

$$P_{22} = \left[\begin{array}{c|c} A_P & B_P \\ \hline C_P & 0 \end{array} \right], \quad Q = \left[\begin{array}{c|c} A_Q & B_Q \\ \hline C_Q & D_Q \end{array} \right], \quad (3.68)$$

the state space representation of \mathbf{K} is:

$$\begin{aligned} \mathbf{x}_K &= \begin{bmatrix} \mathbf{x}_P \\ \mathbf{x}_Q \end{bmatrix} \\ \dot{\mathbf{x}}_K &= \underbrace{\begin{bmatrix} \mathbf{A}_P - \mathbf{B}_P \mathbf{D}_Q \mathbf{C}_P & \mathbf{B}_P \mathbf{C}_Q \\ -\mathbf{B}_Q \mathbf{C}_P & \mathbf{A}_Q \end{bmatrix}}_{\mathbf{A}_K} \mathbf{x}_K + \begin{bmatrix} \mathbf{B}_P \mathbf{D}_Q \\ \mathbf{B}_Q \end{bmatrix} \mathbf{y} \\ \mathbf{u} &= \begin{bmatrix} -\mathbf{D}_Q \mathbf{C}_P & \mathbf{C}_Q \end{bmatrix} \mathbf{x}_K + \mathbf{D}_Q \mathbf{y}. \end{aligned} \quad (3.69)$$

Remarks:

- The same convex optimization (reported in the following) can directly be employed for a feed-forward design which is already affine in the parameters.
- Various choices of basis functions $\mathbf{Q}_i(s)$ in (3.66) have been reviewed in [18]. Essentially, large bases are beneficial for determining feasibility of given specifications, while a low-dimensional basis of low dynamic order is advantageous for controller implementation.

Closed Loop with Observer-based Realization of the Compensator

If a stabilizing initial controller can be transformed into a state observer – feedback gain form (observer-based representation (OBR), see Chap.B), the closed-loop transfer can be efficiently represented in a Youla-parametrized form. Note the following theorem which is based on [174, Theorem 12.16 on p. 323]:

Theorem 3.6.2: Closed-Loop Transfer Matrix with OBR Compensator

Consider the OBR of a compensator \mathbf{K} with state feedback gain \mathbf{K}_c , observer gain \mathbf{K}_f , the partitioned plant model

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} = \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{array} \right] \quad (3.70)$$

as in Sec. 2.3.1 and the stable Youla parameter \mathbf{Q}_0 such that $\mathbf{I} + \mathbf{D}_{22}\mathbf{Q}_0(\infty)$ is invertible and $\mathbf{A} + \mathbf{B}_2\mathbf{K}_c$ and $\mathbf{A} + \mathbf{K}_f\mathbf{C}_2$ are stable.

Then the set of all closed-loop transfer matrices from \mathbf{w} to \mathbf{z} achievable by an internally stabilizing proper controller is equal to

$$\mathcal{F}_1(\mathbf{T}, \mathbf{Q}) = \{\mathbf{T}_{11} + \mathbf{T}_{12}\mathbf{Q}\mathbf{T}_{21} : \mathbf{Q} \in \mathcal{RH}_\infty, \mathbf{I} + \mathbf{D}_{22}\mathbf{Q}(\infty) \text{ invertible}\} \quad (3.71)$$

where \mathbf{Q} is of free order and \mathbf{T} is given by

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} \mathbf{A} + \mathbf{B}_2\mathbf{K}_c & -\mathbf{B}_2\mathbf{K}_c & \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{0} & \mathbf{A} + \mathbf{K}_f\mathbf{C}_2 & \mathbf{B}_1 + \mathbf{K}_f\mathbf{D}_{21} & \mathbf{0} \\ \hline \mathbf{C}_1 + \mathbf{D}_{12}\mathbf{K}_c & -\mathbf{D}_{12}\mathbf{K}_c & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{0} & \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{0} \end{array} \right]. \quad (3.72)$$

Particularly, the closed-loop with the original \mathbf{K} is obtained by setting $\mathbf{Q} = \mathbf{Q}_0$.

3.6.4 Convex formulation of control design constraints and objectives

With a plant parametrized in an affine form as in (3.62), important time- and frequency-domain requirements can be stated as LP, QP, or LMI constraints (convex in the parameters $\boldsymbol{\theta}$), see [18]. Similarly, by bounding a constraint by an additional free variable instead of a constant, any such constraint can be turned into an objective.

For the following formulations, let n_Q be the number of basis functions \mathbf{Q}_i (or, the number of parameters θ_i). The constraints are formulated as non-strict inequalities for practical reasons because then the admissible set of the optimization problem is closed. If strict inequalities are required, it will be explicitly noted in the text.

Time-domain l_∞ bounds

A central benefit of convex synthesis methods is the direct incorporation of time-domain constraints and objectives, which enables template-based step-response shaping. Closed-loop time-domain responses are affine in $\boldsymbol{\theta}$. This can be seen by linearity of the Laplace transform $\mathcal{L}\{\cdot\}$. Let $\mathbf{w}(t)$ and $\mathbf{W}(s) = \mathcal{L}\{\mathbf{w}(t)\}$ be a known, fixed exogenous input signal in time- and frequency-domain, respectively. Then, the closed-loop response $\mathbf{z}(t) = \mathcal{L}^{-1}\{\mathbf{Z}(s)\}$ is obtained as

$$\mathbf{Z}(s) = \mathbf{T}(s)\mathbf{W}(s) = \mathbf{T}_1(s)\mathbf{W}(s) + \mathbf{T}_2(s) \left(\sum_{i=1}^{n_Q} \mathbf{Q}_i(s)\theta_i \right) \mathbf{T}_3(s)\mathbf{W}(s) \quad (3.73)$$

$$\mathbf{z}(t) = \underbrace{\mathcal{L}^{-1}\{\mathbf{T}_1(s)\mathbf{W}(s)\}}_{\mathbf{z}_0(t)} + \sum_{i=1}^{n_Q} \underbrace{\mathcal{L}^{-1}\{\mathbf{T}_2(s)\mathbf{Q}_i(s)\mathbf{T}_3(s)\mathbf{W}(s)\}}_{\mathbf{z}_i(t)} \theta_i. \quad (3.74)$$

This relationship is valid both for SISO and MIMO cases, however the exposition is continued in the SISO case because these occur frequently in practice. To constrain a SISO response

$$z(t) = z_0(t) + \sum_{i=1}^{n_Q} z_i(t)\theta_i \quad (3.75)$$

by lower and upper bounds (templates) defined on a finite time grid,

$$z_L(t_k) \leq z(t_k) \leq z_U(t_k), \quad t_k \in \{t_1, \dots, t_{n_t}\}, \quad (3.76)$$

these inequalities are expanded by (3.75), yielding two LP-type constraints for each time instant t_k :

$$\begin{bmatrix} -z_1(t_k) & \dots & -z_{n_Q}(t_k) \end{bmatrix} \boldsymbol{\theta} \leq z_0(t_k) - z_L(t_k), \quad (3.77)$$

$$\begin{bmatrix} z_1(t_k) & \dots & z_{n_Q}(t_k) \end{bmatrix} \boldsymbol{\theta} \leq z_U(t_k) - z_0(t_k). \quad (3.78)$$

Note that (3.77)–(3.78) also represent two (scalar) LMI constraints and can directly be plugged into an LMI problem.

Frequency-domain: MIMO \mathcal{H}_∞ bound

The constraint $\|\mathbf{G}\|_\infty \leq \gamma$ can be discretized for a stable $\mathbf{G}(j\omega)$ at a frequency grid $\omega_k \in \{\omega_1, \dots, \omega_{n_\omega}\}$ via n_ω constraints $\bar{\sigma}(\mathbf{G}(j\omega_k)) \leq \gamma$. These can be translated into (real-valued) LMI constraints:

$$\left[\begin{array}{cccc} \gamma \mathbf{I} & \Re(\mathbf{G}) & \mathbf{0} & \Im(\mathbf{G}) \\ & \gamma \mathbf{I} & \Im(\mathbf{G}^H) & \mathbf{0} \\ & \star & \gamma \mathbf{I} & \Re(\mathbf{G}) \\ & & & \gamma \mathbf{I} \end{array} \right]_{j\omega_k} \succ 0, \quad (3.79)$$

where \star denotes the terms induced by symmetry, $\Re(\cdot)$, $\Im(\cdot)$, and $(\cdot)^H$ indicate the real part, imaginary part, and the Hermitian (complex-conjugate) transpose, respectively. This LMI is affine in the parameters (the expansion is omitted here for brevity, see [18]). Note that $\gamma = \gamma(\omega_k)$ can be utilized to realize a point-wise template for the maximum singular value over the frequency gridding. No parametric representation (in form of a transfer function $\gamma(s)$) is necessary, as it is in the case of weighting functions in \mathcal{H}_∞ design.

Frequency-domain: MIMO \mathcal{H}_2 bound

As given in Def. A.4.7, the \mathcal{H}_2 norm of a stable, strictly proper linear dynamic system $\mathbf{G}(j\omega)$ is

$$\|\mathbf{G}\|_2 = h = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace}[(\mathbf{G}(j\omega))^H (\mathbf{G}(j\omega))] d\omega}. \quad (3.80)$$

For a sufficiently fine and broad finite frequency gridding $\{\omega_1, \dots, \omega_k, \dots, \omega_{n_\omega+1}\}$, (3.80) can be approximated by the Riemann sum

$$h \cong \tilde{h} = \sqrt{\frac{1}{\pi} \sum_{k=1}^{n_\omega} \text{trace}[(\mathbf{G}(j\omega_k))^H (\mathbf{G}(j\omega_k))] \underbrace{(\omega_{k+1} - \omega_k)}_{\Delta\omega_k}}. \quad (3.81)$$

This \mathcal{H}_2 norm approximation can be expanded to a (convex) quadratic form for \mathbf{G} affine in $\boldsymbol{\theta}$ with coefficients β , $\boldsymbol{\gamma}$, and $\boldsymbol{\Gamma}$ (real scalar $\beta \in \mathbb{R}$, real vector $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_{n_Q}]^T \in \mathbb{R}^{n_Q}$, and real matrix $\boldsymbol{\Gamma} = [\Gamma_{ij}] \in \mathbb{R}^{n_Q \times n_Q}$):

$$\mathbf{G}(j\omega_k) =: {}_k\mathbf{G} = {}_k\mathbf{G}_0 + \sum_{i=1}^{n_Q} ({}_k\mathbf{G}_i) \theta_i, \quad (3.82)$$

$$\tilde{h}^2 = \frac{1}{\pi} (\beta + \boldsymbol{\gamma}^T \boldsymbol{\theta} + \boldsymbol{\theta}^T \boldsymbol{\Gamma} \boldsymbol{\theta}), \quad (3.83)$$

$$\beta = \sum_{k=1}^{n_\omega} \text{trace}[({}_k\mathbf{G}_0)^H ({}_k\mathbf{G}_0)] \Delta\omega_k, \quad \gamma_i = \sum_{k=1}^{n_\omega} 2 \text{trace}[({}_k\mathbf{G}_0)^H ({}_k\mathbf{G}_i)] \Delta\omega_k,$$

$$\Gamma_{ij} = \sum_{k=1}^{n_\omega} \text{trace}[({}_k\mathbf{G}_i)^H ({}_k\mathbf{G}_j)] \Delta\omega_k.$$

Note that only real terms remain due to the properties of the trace and the Hermitian transpose. If the matrix $\boldsymbol{\Gamma}$ is strictly positive definite, making the function $\tilde{h}^2(\boldsymbol{\theta})$ strictly convex, it can be decomposed into its Cholesky factors $\boldsymbol{\Gamma} = \mathbf{L}^T \mathbf{L}$.

Then the approximated \mathcal{H}_2 constraint

$$\|\mathbf{G}\|_2^2 \cong \tilde{h}^2 \leq \delta^2 \quad (3.84)$$

with $\delta^2 \in \mathbb{R}^+$ is equivalent to the LMI constraint

$$\left[\begin{array}{c|c} \mathbf{I}_{n_Q} & \mathbf{L}\boldsymbol{\theta} \\ \hline \boldsymbol{\theta}^T \mathbf{L}^T & \delta^2 - \beta - \boldsymbol{\gamma}^T \boldsymbol{\theta} \end{array} \right] \succeq 0. \quad (3.85)$$

This LMI form is obtained from (3.83) and (3.84) using the Schur complement [143, pp. 481–482], see also [142].

Remarks:

- If $\mathbf{\Gamma}$ is only positive semi-definite, some parameters θ_i typically do not affect the considered \mathcal{H}_2 norm. Then the Cholesky factorization cannot be computed. However, it can be approximated by considering a perturbed $\mathbf{\Gamma} + \varepsilon \mathbf{I}$ ($\varepsilon \ll \|\mathbf{\Gamma}\|$). Another onset is to reduce the set of considered parameters θ_i in the formulation of the constraint to those which do have relevant impact on the closed-loop \mathcal{H}_2 norm.
- Note that this constraint is of size $((n_Q + 1) \times (n_Q + 1))$, so it is not affected by the frequency grid size n_ω which allows to use high approximation precision by a fine grid in the precalculation of β , γ , and $\mathbf{\Gamma}$ without enlarging the optimization problem size.

3.6.5 Selection of the filter basis

The selection of basis functions $\mathbf{Q}_i(s)$ in (3.66) is no trivial problem and various types of bases and their characteristics have been reviewed in [18]. It is usually of benefit to introduce expert knowledge (known or expected system dynamics, pole locations) into the design of basis functions. The intended use and context further determine which properties are important:

- For feedback and feed-forward control design it is relevant to keep the dynamic order of the individual basis functions \mathbf{Q}_i low to limit the resulting controller order. To design MIMO Youla parameters, a first design could be carried out with static gains \mathbf{Q}_i only. Also, ad-hoc bases of first- and second-order dynamics within the bandwidth of the closed loop often yield good performance at limited controller order.
- To test the feasibility of defined design constraints, a maximally spanning basis is important. Contrary to the design case, the dynamic order of \mathbf{Q}_i does not matter. In turn, it may be beneficial to utilize orthonormal bases. These can improve numeric conditioning of the problem and help convergence which helps in problems with many parameters. Various types of orthonormal bases have been proposed:
 - The Laguerre basis is defined and constructed with respect to one real and dominant pole.
 - The Kautz basis is defined and constructed with respect to one complex-conjugate dominant pole pair.
 - Ninness & Gustafsson [105] proposed an orthonormal basis construction which considers an arbitrary number of real and complex poles. The Laguerre and Kautz bases are special cases of this construction.

Considering the discrete-time case, orthonormality of two basis functions $\mathcal{B}_n(z)$, $\mathcal{B}_m(z)$ is thereby understood with respect to the standard inner product on $\mathcal{H}_2(C)$ (see [105]), where C is the unit circle ($C = \{z : |z| = 1\}$):

$$\langle \mathcal{B}_n(z), \mathcal{B}_m(z) \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{B}_n(e^{j\omega}) \overline{\mathcal{B}_m(e^{j\omega})} d\omega. \quad (3.86)$$

3.6.6 $\mathbf{Q} - \mu$ -Synthesis for Robust Control Design

An extension to address the μ synthesis task with the convex control design approach is developed in the PhD thesis [18] and is called $\mathbf{Q} - \mu$ -synthesis. Similar to the DK- or DGK-iteration, the $\mathbf{Q} - \mu$ -synthesis consists of an alternating sequence of controller synthesis and μ analysis. The \mathbf{D} - and \mathbf{G} -scalings of the mixed μ analysis step are utilized to directly formulate a singular value objective for subsequent control design iterations.

Following the exposition in [18], the algorithm carries out the following steps. Consider the $\mathbf{P} - \mathbf{K} - \mathbf{\Delta}$ form as in Sec. 2.3.1 where

$$\mathbf{M} = \mathbf{M}(s, \boldsymbol{\theta}) = \mathbf{T}_1(s) + \sum_{i=1}^{n_Q} \mathbf{T}_2(s) \mathbf{Q}_i(s) \mathbf{T}_3(s) \theta_i. \quad (3.87)$$

1. First, a nominal control design is carried out and an optimal controller is obtained for the nominal plant. It is assumed that nominal performance is attained. The resulting optimal parameters $\boldsymbol{\theta}^*$ are fixed.
2. For each frequency ω_k in a given frequency grid, the mixed- μ upper bound analysis is carried out for $\mathbf{M} = \mathbf{M}(j\omega_k, \boldsymbol{\theta}^*)$ and the scalings $\mathbf{D} = \mathbf{D}(j\omega_k)$ and $\mathbf{G} = \mathbf{G}(j\omega_k)$ are obtained, see Sec. 2.3.1 and Sec. 3.5.2. If the condition

$$\bar{\sigma} \left((\mathbf{I} + \mathbf{G}^2)^{-\frac{1}{4}} (\mathbf{DMD}^{-1} - j\mathbf{G}) (\mathbf{I} + \mathbf{G}^2)^{-\frac{1}{4}} \right) < 1 \quad (3.88)$$

is fulfilled for all ω_k ,

$$\mu(\mathbf{M}) < 1 \quad (3.89)$$

holds, robust performance is obtained, and the $\mathbf{Q} - \mu$ -synthesis stops. From the obtained final parameter vector $\boldsymbol{\theta}^*$ the Youla parameter $\mathbf{Q}^* = \mathbf{Q}(\boldsymbol{\theta}^*)$ and, consequently, the final controller $\mathbf{K}^* = \mathbf{K}(\mathbf{P}_{22}, \mathbf{Q}^*)$ is computed.

3. If (3.89) is not fulfilled, the optimal scalings $\mathbf{G}(j\omega_k)$, $\mathbf{D}(j\omega_k)$ are fixed (denoted $\widehat{\mathbf{G}}(j\omega_k)$, $\widehat{\mathbf{D}}(j\omega_k)$, respectively). Then, a new convex control design problem in $\boldsymbol{\theta}$ is solved with the objective to minimize γ under the constraints

$$\bar{\sigma} \left((\mathbf{I} + \widehat{\mathbf{G}}_k^2)^{-\frac{1}{4}} \left(\widehat{\mathbf{D}}_k \mathbf{M}(j\omega_k, \boldsymbol{\theta}) \widehat{\mathbf{D}}_k^{-1} - j\widehat{\mathbf{G}}_k \right) (\mathbf{I} + \widehat{\mathbf{G}}_k^2)^{-\frac{1}{4}} \right) \leq \gamma \quad (3.90)$$

at each ω_k with $\widehat{\mathbf{G}}_k = \widehat{\mathbf{G}}(j\omega_k)$, $\widehat{\mathbf{D}}_k = \widehat{\mathbf{D}}(j\omega_k)$. If an optimal value $\gamma < 1$ is obtained, robust performance is obtained and the algorithm stops. If not, the optimal parameters $\boldsymbol{\theta}^*$ are fixed and the algorithm goes to step 2.

Chapter 4

Flexible Aircraft Models

To demonstrate the tools and methodologies developed in the next part, several aspects in flight control design for large flexible aircraft will be considered. A large flexible BWB aircraft model as well as a large flexible conventional aircraft model will be utilized in these case studies. After a short introduction to some fundamental considerations and challenges in flight control design, these aircraft models are introduced in Sec. 4.2 and Sec. 4.3.

4.1 Introduction: Flight Dynamics and Control Design

The control of flexible aircraft has become a research topic of high interest over the last decade [53], [70], [78], [162], [164] due to the potential reduction in structural weight. Large lightweight aircraft structures and novel concepts, such as BWB aircraft configurations, can lead to higher fuel efficiency and reduced emissions [96]. These large, light-weight flexible structures exhibit low-frequency elastic vibration modes, and coupling of those with the flight mechanic modes may occur. Moreover, the aircraft dynamics is significantly dependent on the flight parameters. Hence the task of developing robust and well-performing flight control laws faces significant challenges.

Traditional methods for flight control design typically use nested SISO control loops and strongly structured control architectures [145]. These methods are based on detailed aircraft system analysis and exploit paths with weak coupling to obtain good results for conventional flight control design. However, multivariate methods, such as optimal control and particularly robust control design methods are state of the art for more complex flight control tasks under coupled and/or uncertain system dynamics. The two large groups of control design methodologies, optimal as well as robust control designs (see Chap. 3 and [7] for an aerospace-specific overview) are widely applied in the aerospace domain. For conventional aircraft, LQ-based control designs have been employed for longitudinal flight control design [79], [80], as well as for lateral flight control design [141], [23], often combined with partial eigenstructure assignment and optimization techniques.

Two models of flexible aircraft are introduced here which are utilized in specific control design examples in the following chapters. These models are:

- an integrated lateral dynamic model of a large, flexible BWB transport aircraft in the predesign stage (see Sec. 4.2), and
- an integrated longitudinal dynamic model of a large, flexible conventional transport aircraft (see Sec. 4.3).

In both aircraft models, flight mechanics coupled with aeroservoelastic effects for varying flight conditions and a grid of fuel mass and payload cases are considered. The BWB aircraft model

represents one of the considered aircraft configurations in the ACFA 2020 EU FP7 project [1] (and were partially also based on earlier research projects).

4.2 Lateral Blended Wing Body (BWB) Aircraft Model

Longitudinal and lateral flight mechanics and aeroelastic effects of a large BWB aircraft pre-design and their coupling were modeled in an integrated fashion by the authors' project partners [147], [148]. These models have been utilized in the related publications [135], [138], and [137].

In this work, the lateral dynamics as well as the flexible structure modes and aerodynamic lag states are considered to design and validate lateral control laws. A set of $k = 18$ linearized state space systems for various parameter values of fuel and payload mass (at fixed cruise altitude and airspeed) is available ($i = 1, \dots, k$):

$$\dot{\mathbf{x}} = \mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} \quad (4.1)$$

$$\mathbf{y} = \mathbf{C}_i \mathbf{x} + \mathbf{D}_i \mathbf{u}, \quad (4.2)$$

where the state vector \mathbf{x} is composed of 4 flight-mechanic states (side slip angle β , roll rate p , yaw rate r , roll angle ϕ), 24 elastic states (12 structural modes), as well as 10 aerodynamic lag states. The integrator states ψ (yaw angle) and y (horizontal displacement) are neglected in this work. This system is augmented as shown in Fig. 4.1 by actuator and sensor dynamics.

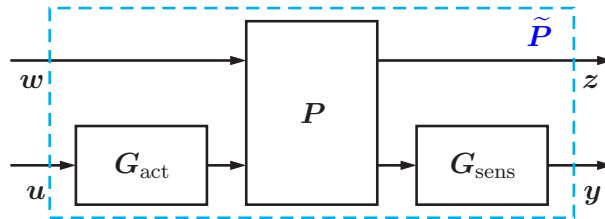


Figure 4.1: Aircraft system model with actuator and sensor dynamics

Utilized inputs \mathbf{u} for control design are:

- Symmetric rudder deflection and rate u_{RU}, \dot{u}_{RU}
- One collected antisymmetric aileron deflection and rate: outer, middle, and inner ailerons are deflected equally and antisymmetrically (u_{AIL}, \dot{u}_{AIL}), see also Fig. 4.2.

The actuator dynamics \mathbf{G}_{act} are modeled via second-order low-pass filters.

Measured outputs \mathbf{y} available for control design are:

1. Roll angle ϕ
2. Side-slip angle β (near the CG position)
3. Roll rate p
4. Yaw rate r
5. (optional) Lateral acceleration N_{yCG} (near the CG position)
6. (optional) Antisymmetric wing bending modal sensor $N_{z_{lat.law}}$

where signals 1–5 are subject to a time delay of 160 ms (to model signal processing latency, modeled via a second-order Padé approximations) and additionally low-pass filtered via Butterworth filters of second order. Signal 6 is delayed by 60 ms (without Butterworth-filter). The sensor filters are collected in \mathbf{G}_{sens} in Fig. 4.1. The augmented system $\tilde{\mathbf{P}}$ with only outputs 1–4 (and thus only their sensor dynamics) is of order 58.

Additional exogenous input and output signals for validation are considered — a wind gust disturbance input (lateral wind speed $w = v_{\text{lat}}$) as well as two structure loads outputs Mx_{wing} and Mx_{fin} (inner moments about the x -axis at the wing root and at the fin root) and the lateral acceleration Ny_{CG} (near the CG position).

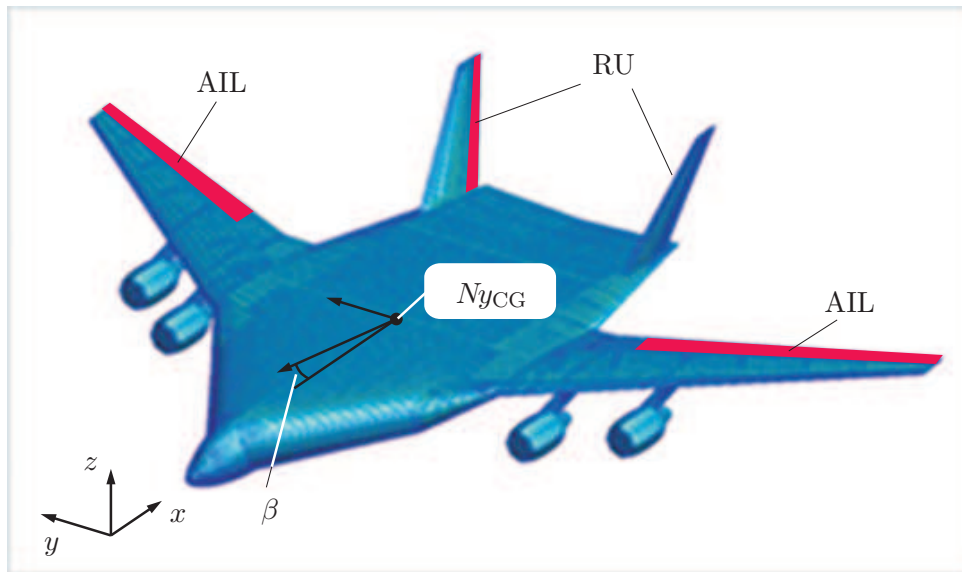


Figure 4.2: BWB aircraft schematics (RU: Rudder, AIL: Ailerons)

4.2.1 Open-Loop Analysis

The lateral aircraft dynamics is open-loop stable for all considered mass cases at cruise condition. The Bode magnitude plot of the rudder – yaw rate transfer in Fig. 4.3 shows a prominent, low-damped Dutch Roll mode (DR mode) between 0.5 and 0.75 rad/s with a damping between 0.1 and 0.25 . Step responses for rudder – yaw rate and vertical gust – roll rate are given in Fig. 4.4 and Fig. 4.5. At higher frequencies, the structural modes are visible. For this study, they are only of interest for validation as they are sufficiently well separated from the lateral flight mechanic modes.

Inspection of those SISO transfer functions commonly used for lateral control design shows a fundamental limitation for SISO control schemes here — a very slow right half plane (RHP) zero is present in the transfer functions rudder \rightarrow yaw rate, rudder $\rightarrow Ny_{\text{CG}}$, and rudder $\rightarrow \beta$ (smallest RHP zero around $+10^{-3} \text{ rad/s}$). This poses fundamental limitations conflicting with the control goals defined in Sec. 4.2.2. However, the respective transfer functions from the aileron input to these outputs show much faster RHP zeros, also all MIMO RHP zeros are faster (around 1 rad/s or above). This indicates that a SISO approach will encounter significant limitations, whereas a MIMO approach will not.

Moreover, as shown in Fig. 4.6, a number of low-damped poles affect the disturbance (lateral wind) – lateral acceleration path strongly, which is relevant for passenger comfort and possibly fatigue. The damping of the most critical mode around 20 rad/s should thus be increased robustly.

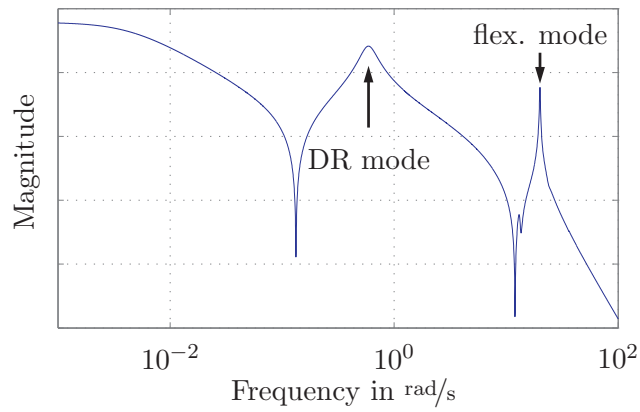


Figure 4.3: Open-loop Bode magnitude: Rudder \rightarrow yaw rate transfer

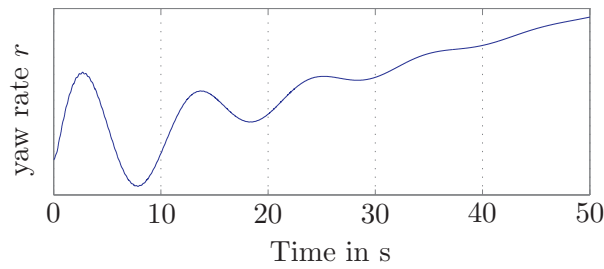


Figure 4.4: Open-loop: yaw rate response to a rudder step

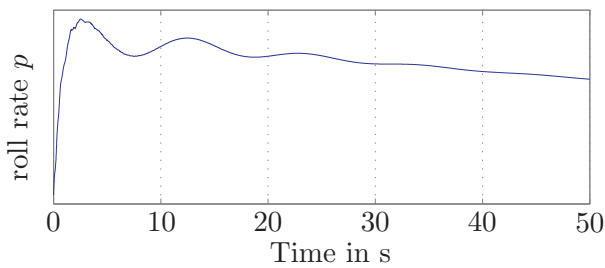


Figure 4.5: Open-loop: roll rate response to a wind disturbance step

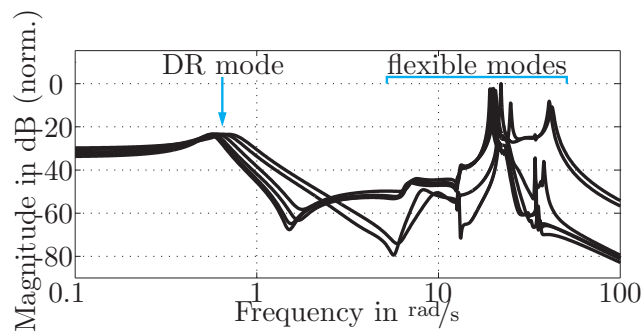


Figure 4.6: Bode magnitude plot of lateral wind v_{lat} – lateral acceleration Ny_{CG} for all mass cases

4.2.2 Control Goals

The major control goals addressed in the related case studies in this work are to robustly shape the lateral reference and disturbance responses, with special emphasis on maneuver loads alleviation and disturbance rejection. Tab.4.1 shows the set of relevant control specifications and whether they are exclusively related to feedback (FB) or feed-forward (FF) control design, or considered in both design cases.

Table 4.1: Lateral control goals for large flexible BWB aircraft

Specification	Design	
	FB	FF
General and tracking specifications		
DR mode	$\zeta_{\text{DR}} \geq 0.7$, ω_{DR} unchanged	•
Decoupling	Generate inputs with high coupling to roll and side slip angles and low cross-coupling	• •
Roll ϕ	static gain (DC gain) similar for all mass cases, rise time to 90% in $t_{\text{rise}} \leq 7$ s, max. 5% overshoot	• •
Side slip β	DC gain similar for all mass cases, $t_{\text{rise}} \leq 5$ s	• •
Robustness	Stable controller, Robust Performance for all mass cases	• •
Disturbance rejection specifications		
	Minimize the influence of lateral gust on roll, side slip, and lateral acceleration, while obeying the tracking specifications above. Moreover, the loads must not be increased.	•
Maneuver loads alleviation specifications		
	Robustly minimize Mx_{wing} and Mx_{fin} loads induced by roll- and side-slip step responses	•

4.3 Longitudinal Flexible Conventional Aircraft Model

The longitudinal motion of a large, flexible conventional airliner has been modeled and utilized for control design and validation, see [162] and [165]. The aircraft models (taken from these sources) are linearized state-space models and include the flight mechanic and aeroelastic aircraft behavior at several mass cases. For this study one design mass case (low fuel) and four validation mass cases (low to full fuel with two different center of gravity (CG) positions: A2, B1, C2, D1) are used. The state-space models include 42 states, 1 wind input, 5 control surface inputs and numerous sensor outputs (vertical accelerations, angles and angular rates at various positions in the aircraft). The models are augmented by second-order low-pass actuator dynamics for the used control surfaces and time-delays in sensor measurement channels to account for signal processing delays. Corresponding to the formal definition in Sec. 2.3.3, the problem is given with a discrete set of 4 validation plants.

4.3.1 Definition of Control Goals and Architecture

As detailed in [162] and [165], the control goals and constraints are simultaneously to:

- ensure robust stability (robustness with respect to the 4 validation plants),
- shape the flight dynamic modes such that time-domain specifications (overshoot, rise time) in the pilot command – pitch rate response are robustly fulfilled,

- robustly attenuate structural vibrations in the first wing bending and first hull bending modes, and
- fulfill other constraints (robustly attenuate other modes and performance position signal RMS values), see the denoted references.

In [162] a two-degree-of-freedom feedback and pilot command shaping control architecture has been developed. This work used the design parameter optimization framework developed in Chap. 6 already.

The controller inputs are reference and measurement signals of the vertical acceleration Nz_{CG} and pitch rate q_{CG} and a modal vertical acceleration sensor signal Nz_{law} capturing the first wing bending mode. Its outputs are the control surface commands for elevator, inner aileron and 3 direct-lift-control flaps:

$$\begin{bmatrix} \delta_{EL} \\ \delta_{IA} \\ \delta_{DLC1} \\ \delta_{DLC2} \\ \delta_{DLC3} \end{bmatrix} = \mathbf{K} \begin{bmatrix} Nz_{CG,ref} \\ q_{CG,ref} \\ Nz_{CG,meas} \\ q_{CG,meas} \\ Nz_{law} \end{bmatrix} \quad (4.3)$$

The following results highlight two important aspects of the methodology: system scaling and design parameter optimization of the referenced 2-degree-of-freedom design.

4.3.2 Scaling and Balanced Reduction

Figure 4.7 shows the importance of appropriate scaling before performing a balanced system reduction. The aircraft inputs and outputs can be grouped into the wind input and the control surface inputs as well as the angular rate outputs (e.g. pitch rate), vertical acceleration outputs, and load outputs (e.g. wing root bending moments). Here, the load output paths have about 100 dB higher magnitude than the other outputs. When reducing the full-order system from order 56 to order 12 without scaling, the high-gain paths are well-preserved over the reduction while the other paths' relevant states are eliminated, leading to unacceptable relative errors there. Performing the reduction after applying appropriate I/O-scaling to shift all relevant transfer paths to similar magnitudes yields an equally good system description in all transfer paths.

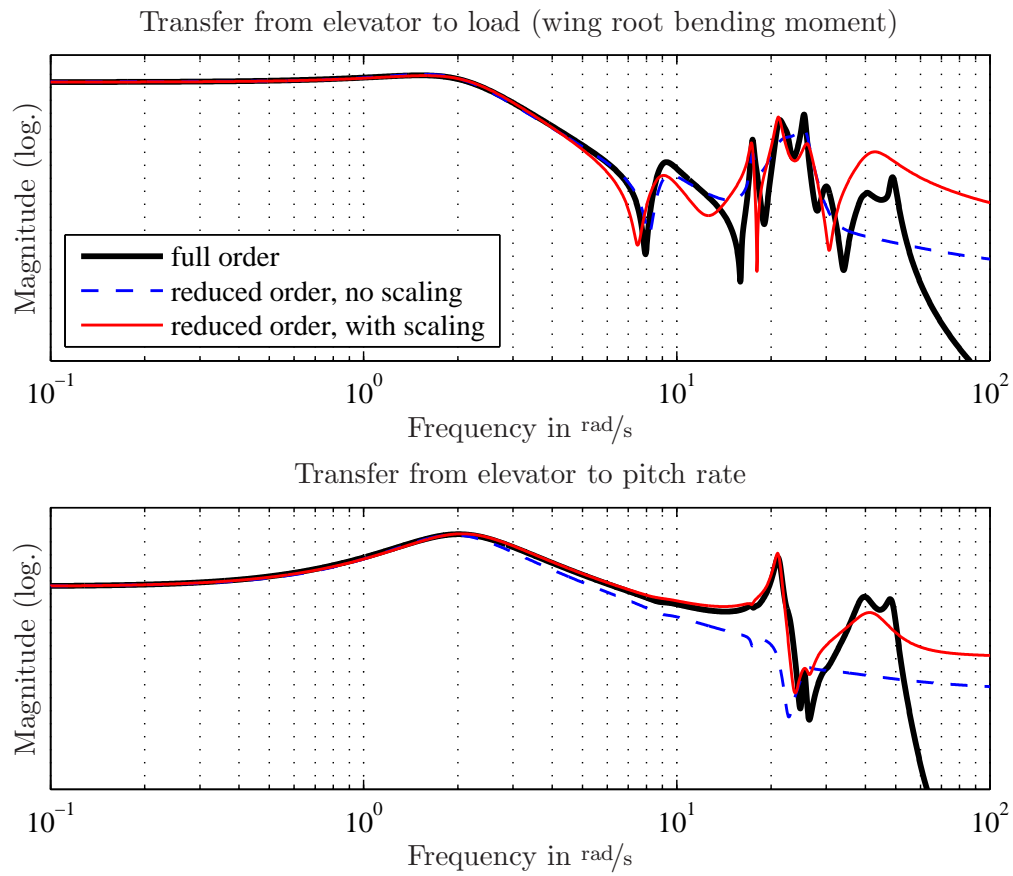


Figure 4.7: Benefit of system scaling for balanced system reduction (*top*: large magnitude load output path; *bottom*: small magnitude pitch rate output path)

Part II

Novel Contributions

Chapter 5

Extensions to the Convex Control Design Methodology

This chapter presents novel contributions made for the convex control design onset introduced in Sec. 3.6. To efficiently formulate, solve, and validate convex control design problems, suitable supporting tools have to be developed. In Sec. 5.1, such tool in form of a novel, efficient, and powerful framework methodology to formulate and solve highly complex and high-dimensional convex control design problems is developed and presented. A general optimization framework is established, and effective heuristic methods to adaptively refine problem objectives and constraints are presented. The framework's performance is demonstrated at several large-scale examples later in this chapter. Sec. 5.2 discusses methods and presents novel formulations to achieve strong stabilization (that is, internal stability of the closed loop with a stable controller) in the context of convex control design. A non-convex strong stabilization constraint is formulated and several convex approximation methods are provided and illustrated at academic examples and at a large-scale industrial application — the lateral feedback control design for a large flexible BWB aircraft. Finally, Sec. 5.4 presents an efficient and highly flexible multi-model scheduled feed-forward control design approach via convex optimization. In this formulation, both multi-model robustness and partial scheduling can be freely formulated and directly integrated into the optimization. The BWB aircraft example is used to demonstrate scheduled robust feed-forward design and to assess the attainable performance gain.

5.1 Convex control design framework methodology

To ease the complex design process in convex control design, including the formulation of the control architecture, Youla parametrization and the definition of its basis functions, constraints, and objectives, a flexible implementation of an optimization framework has been carried out. Besides supporting the engineer in problem formulation, concepts that reduce optimization problem size and integrate performance validation are realized, so that large-scale problems can be solved with high efficiency. Therefore, an adaptive constraint refinement procedure for problem size reduction is introduced. Then, the framework implementation is outlined in more detail; an overview on the implemented objects and methods is given in the Appendix, Chap. D. This framework serves as the basis for the further developments in this chapter.

5.1.1 Adaptive constraint refinement

The optimization problem size is primarily determined by the number of independent variables in θ , by the number of constraints, and by the number of grid points at which the constraints

(or objectives) are defined. An adaptive grid refinement procedure has thus been implemented for high computational efficiency:

1. Start with coarse design grids.
2. Formulate & solve the resulting LMI optimization problem (see Sec. 3.6.4).
3. Validate solution on fine validation grids.
4. Pick, per violated objective/constraint, at most n_{add} points that are most violated (and mutually sufficiently spaced) and add them to the design grids.
5. If no violations: done. Else: return to step (2).

For the computations, this algorithm has been implemented via an object-oriented approach in MATLAB[®]. The overall efficiency could be significantly increased by using data preprocessing, caching methods, and efficient updates of the LMI definitions.

5.1.2 Object-Oriented Implementation of Framework

The framework has been developed and implemented in MATLAB[®], and an outline and description of its main structural components is given in the following.

Main Requirements

The following list of main requirements already outlines the basic functionality seen in the framework and builds the basis of its development:

- A simple, logical, and lean syntax for problem statement, optimization, and evaluation should be provided to the MATLAB[®] user. For this main use case, the necessary commands should fit in a short m-file of 50 – 100 lines.
- All presented functionality should be covered and supported. A shortlist of core functionality requirements:
 - Management of Youla parametrization, basis function generation, system definition
 - Template-based definition of the time- and frequency-domain constraints and objectives from Sec. 3.6.4
 - Embedded LMI formulation and solution with MATLAB[®]'s LMILAB
 - Adaptive constraint refinement procedure as shown in Sec. 5.1.1
 - Solution evaluation, plotting, exporting
 - Streamlining of the design process: saving/loading of problems, grid definitions, caching of precalculations
- Maintainability, didactic value (code readability, logical structure), and simple extensibility for future functionality (for example, new constraint/objective definitions, additional interfacing of different solvers, and new problem configurations) should be enabled and facilitated by according framework design decisions and by the way of implementation.
- The framework should, for practical reasons, be embedded in the established standard MATLAB[®] environment, in particular its object-oriented functionality, syntax, and the LMILAB solver provided by the Robust Control Toolbox [6].

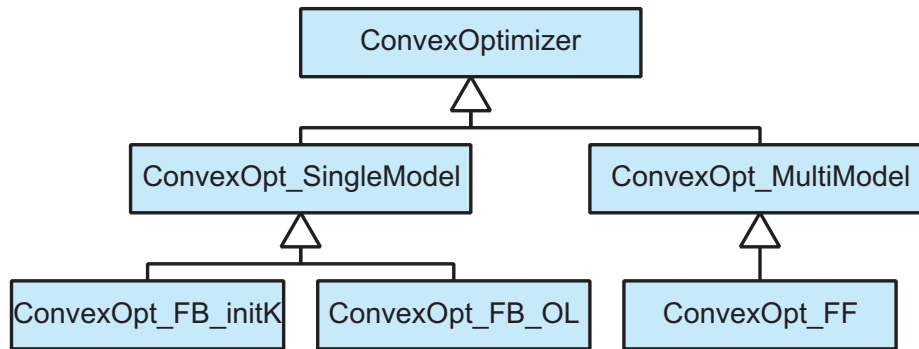


Figure 5.1: Class hierarchy of ConvexOptimizer and its subclasses

Object-Oriented Programming

To address the main requirements, an object-oriented onset to design and implement the framework is chosen. Object-oriented programming (OOP) provides many benefits that address the maintainability and extensibility requirements listed above. A short overview on main characteristics of OOP is given in the following, see [163] and the references therein for a more extensive discussion.

From one of many views on the subject, object-oriented programming considers a world (collection) of virtual objects which interact with each other. Each concrete object (also called instance or class instance) is derived from one or more classes. A class defines the structure of objects, much like a blueprint of the concrete object. This structure consists of properties (which can be thought of variables, memory, attributes, or data which is associated to one particular object) and so-called methods (which are functions or programmed behavior that act on the object and its properties). This very general and abstract concept is moreover hinged on some fundamental aspects:

- Information hiding, encapsulation, decoupling: Only well-defined interfaces of an object are exposed to the outside. Within one object, the functional relations and interactions are dense, whereas they are sparse between different objects. This decoupling facilitates code re-use and maintainability.
- Class inheritance: One class can inherit properties and methods of one or more other classes. This reduces redundancy in the code base, enables more logical structuring, and again facilitates code maintainability and extensibility.

Class Structure of the Convex Control Design Framework

The framework is organized into groups of classes with hierarchical structure where appropriate. A short description of the main structures are listed and some key aspects of the framework structure are explained. In the Appendix Chap. D, an outline of each class with short comments and their relevant properties and methods is given.

ConvexOptimizer hierarchy: This set of classes realizes the top-level problem solving tasks for the considered convex control design problems. The implementation allows to cast the various control design formulations (feedback and feed-forward design, zero or OBR initial controllers) into LMI form, solve them, and evaluate the results. The structure is depicted in Fig. 5.1.

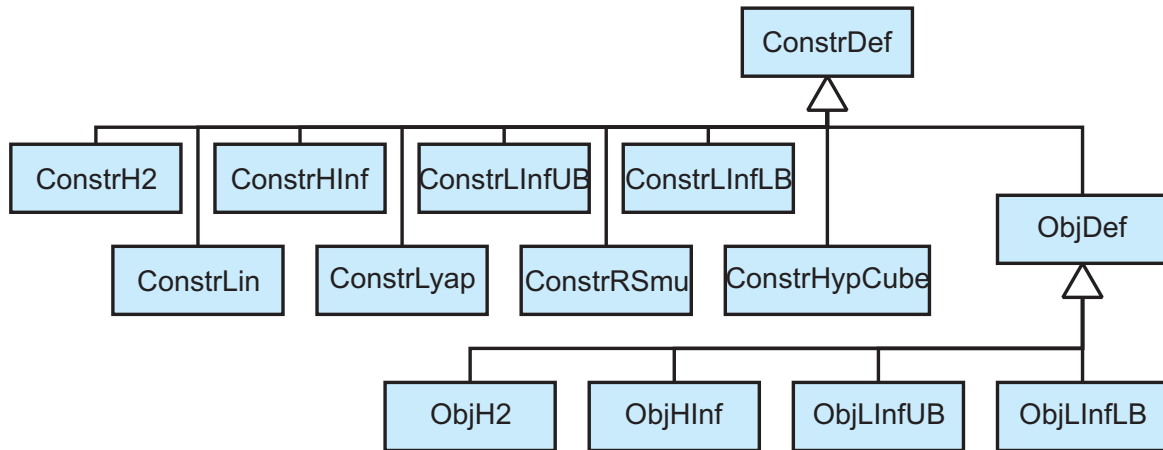


Figure 5.2: Constraint/objective class hierarchy

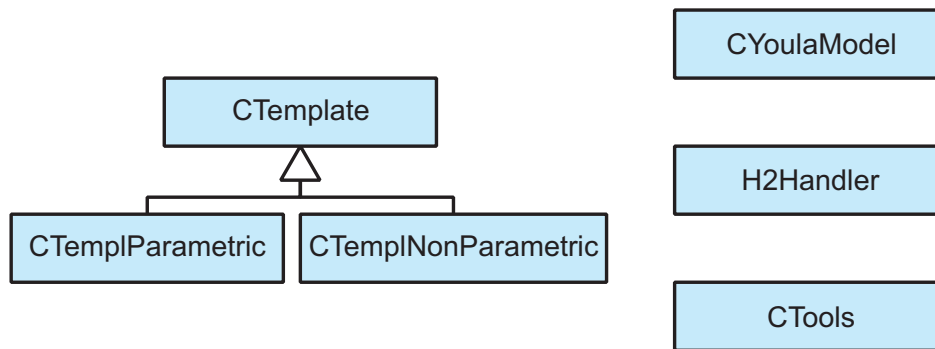


Figure 5.3: Auxiliary classes

Constraint/objective class hierarchy: This set of classes implements constraint and objective definitions and their behavior. They provide common interfacing for LMI formulation, design and validation grid handling, adaptive refinement procedures, and result plotting. The class hierarchy is depicted in Fig. 5.2 and short class descriptions are given in the following.

Auxiliary classes: Six auxiliary classes are realized which perform auxiliary tasks: three classes, `CTemplate`, `CTemplParametric`, and `CTemplNonParametric` implement parametric and non-parametric templates of signals over a time- or frequency-gridding. Youla parameter computations, basis function generation (see Sec. 3.6.5), and response caching is implemented in the class `CYoulaModel`. The class `H2Handler` realizes computation and caching of the precomputable quantities β , γ , Γ , and L when modeling \mathcal{H}_2 constraints and objectives (see Sec. 3.6.4). Other helper functions are collected in the class `CTools`. These classes are depicted in Fig. 5.3.

5.2 Strong Stabilization

A strongly stabilizing controller is a stable and internally stabilizing controller for a given plant [154]. Note that this is not guaranteed by usual optimal and robust control design methods: they yield internal stability, but the controller transfer function may be unstable [69].

5.2.1 Motivation & Introduction

Strong stabilization is relevant in control design if control input signal saturation occurs. Then stability of the controller is required to retain bounded internal signals. The feedback path is effectively opened in this situation and the internal signals' behavior is characterized by the plant's and controller's open-loop transfer functions. The marginal case of integrating behavior in the controller transfer function leads to the well-known integral wind-up phenomenon [143]. If the controller transfer function is unstable, the internal signals can grow exponentially fast, thus violating the practical requirement of bounded internal signals. Moreover, strong stabilization is closely connected to simultaneous stabilization, that is, the problem of stabilizing several plants with the same controller [154].

Additionally, two observations were made in many practical control design problems by the author:

1. The robustness against (unmodeled) plant model perturbations tends to be significantly worse when the controller itself is unstable than when it is stable.
2. Whether or not a resulting optimal controller is unstable strongly depends on the design weightings and the plant characteristics. Often, if plant transfer functions which contain RHP zeros are being shaped by control, a high performance requirement (formulated by the weighting functions) will enforce an unstable controller (see also [69] and [83]). If, however, the controller is given less authority with weaker performance demands, the same design can result in a stable controller.

These observations suggest that improved robustness of stable controllers in observation 1 could be caused by lower control authority. This is immediately plausible for the case of an initially stable (perturbed) plant. Even though strong stabilization may result in inferior closed-loop performance, means to enforce controller stability are relevant for the stated reasons.

The set of plants that can be stabilized by a stable controller (called the set of strongly stabilizable plants) is a subset of all stabilizable plants. The decisive property thereby is the so-called parity interlacing property which was firstly formulated in [171]. A parametrization of all strongly stabilizing controllers is found in [154]. Several solutions to strong stabilization control design are available, notably a (suboptimal) weighted sensitivity minimization with stable controllers [69].

An ad-hoc method to avoid the generation of unstable controllers is to introduce sufficient plant uncertainty and to synthesize a robustly stable and performing controller. However, this does not guarantee controller stability either, and sometimes it is difficult to tune these auxiliary uncertainties to obtain the desired effect.

However, to the best knowledge of the author, no approach exists that preserves affinity of the closed loop in the problem representation which is necessary to incorporate time- and frequency-domain constraints as shown in Sec.3.6.4 and thus highly beneficial for the control design task. Such strong stabilization constraint is formulated in the following, but it turns out to be non-convex in the design parameters. Illustrative examples for this fact are given next, followed by two onsets to approximate the strong stabilization constraint for inclusion in an LMI problem: a Lyapunov-iteration approach suitable for small problems, and a convex embedding approach targeted at high-dimensional control design tasks. Numeric results are given and discussed.

5.2.2 Illustrating Examples

Two academic examples are given in this section to illustrate the non-convexity of the strong stabilization constraint in the Youla parameter weights θ_i . A polynomial approach will be followed for these examples.

Consider the Youla parameter based on the zero-feedback parametrization of an initially stable plant \mathbf{P} as in Sec. 3.6.3. With

$$\mathbf{T}_{zw}(s) = \mathbf{P}_{11}(s) + \mathbf{P}_{12}(s) \underbrace{\mathbf{K}(\mathbf{I} - \mathbf{P}_{22}(s)\mathbf{K}(s))^{-1}}_{\mathbf{Q}(s)} \mathbf{P}_{21}(s) \quad (5.1)$$

$$= \mathbf{T}_1(s) + \mathbf{T}_2(s)\mathbf{Q}(s)\mathbf{T}_3(s), \quad (5.2)$$

the relations between $\mathbf{Q}(s)$ and $\mathbf{K}(s)$ are:

$$\mathbf{Q}(s) = \mathbf{K}(s) (\mathbf{I} - \mathbf{P}_{22}(s)\mathbf{K}(s))^{-1} \quad (5.3)$$

$$\mathbf{K}(s) = (\mathbf{I} + \mathbf{Q}(s)\mathbf{P}_{22}(s))^{-1} \mathbf{Q}(s). \quad (5.4)$$

It can be seen that with a stable $\mathbf{Q}(s)$, the controller can become unstable if $(\mathbf{I} + \mathbf{Q}(s)\mathbf{P}_{22}(s))^{-1}$ is unstable. Note that it is however guaranteed that the closed loop is internally stable!

The SISO case

Given a stable (or stabilized) SISO plant $P(s) = \frac{B_P(s)}{A_P(s)}$ and a stable SISO Youla parameter $Q(s) = \frac{B_Q(s)}{A_Q(s)}$ (where A, B are the denominator/numerator polynomials in s , respectively), the controller is given as

$$K(s) = (1 + Q(s)P(s))^{-1} Q(s) \quad (5.5)$$

$$= \frac{1}{1 + \frac{B_Q B_P}{A_Q A_P}} \frac{B_Q}{A_Q} \quad (5.6)$$

$$= \frac{A_Q A_P}{A_Q A_P + B_Q B_P} \frac{B_Q}{A_Q}. \quad (5.7)$$

While A_P and A_Q are stable (their roots are all in the left half plane (LHP)), the denominator part $A_Q A_P + B_Q B_P$ can have roots in the RHP. This can be tested, for example, via the Hurwitz stability test on the characteristic polynomial

$$p(s) = A_Q A_P + B_Q B_P = a_n s^n + \dots + a_1 s + a_0. \quad (5.8)$$

The Hurwitz conditions for stability (all roots of $P(s)$ in the LHP) are:

$$a_k > 0, \quad k = 1, \dots, n \quad (5.9)$$

$$\mathbf{H} = \begin{bmatrix} a_{n-1} & a_{n-3} & a_{n-5} & \dots \\ a_{n-0} & a_{n-2} & a_{n-4} & \dots \\ 0 & a_{n-1} & a_{n-3} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \succ 0, \quad (5.10)$$

where \mathbf{H} is the $(n \times n)$ Hurwitz matrix of $p(s)$. This matrix is required to be positive definite (which is commonly tested via the positive-definiteness of its principal minors).

Example 5.2.1: Strong stabilization constraint for SISO feedback design

Given the stable, proper, non-minimum-phase SISO plant

$$P(s) = \frac{s-1}{s+2} \quad (5.11)$$

and the Youla parameter

$$Q(s) = \frac{3}{s+1}\theta, \quad \theta \in \mathbb{R}, \quad (5.12)$$

formulate the constraint corresponding to a stable controller $K(s)$, where

$$K(s) = \frac{Q(s)}{1 + P(s)Q(s)} = \frac{A_Q A_P B_Q}{(A_Q A_P + B_Q B_P) A_Q} = \frac{A_P B_Q}{A_Q A_P + B_Q B_P}. \quad (5.13)$$

The characteristic polynomial of the controller transfer function thus is

$$p_K(s) = A_Q A_P + B_Q B_P = (s+1)(s+2) + 3\theta(s-1) = s^2 + (3+3\theta)s + (2-3\theta). \quad (5.14)$$

The corresponding Hurwitz conditions for stability are

$$s^1: 3+3\theta > 0 \quad (5.15)$$

$$s^0: 2-3\theta > 0 \quad (5.16)$$

$$\det(\mathbf{H}) = \left| \begin{bmatrix} 3+3\theta & 0 \\ 1 & 2-3\theta \end{bmatrix} \right| > 0. \quad (5.17)$$

For this simple example, the equations are fulfilled for the interval $-1 < \theta < \frac{2}{3}$ (and the positive definiteness of \mathbf{H} follows from the positivity of the coefficients).

The MIMO case: a polynomial formulation

For the MIMO case, the constraint takes the form

$$(\mathbf{I} + \mathbf{Q}(s)\mathbf{P}_{22}(s))^{-1} \text{ stable.} \quad (5.18)$$

The inverse of a regular square matrix \mathbf{X} can be written as

$$\mathbf{X}^{-1} = \frac{1}{\det(\mathbf{X})} \text{adj}(\mathbf{X}). \quad (5.19)$$

For a minimal realization of a linear dynamic MIMO system $\mathbf{X}(s)$, i.e. a rational transfer matrix in s , the numerator of $\det(\mathbf{X})$ represents the characteristic polynomial whose roots are the poles of the individual SISO transfer functions in $\mathbf{X}(s)$. This can be verified by observing that the adjoint of \mathbf{X} ,

$$\text{adj}(\mathbf{X}) = (\tilde{x}_{ji}) \quad (5.20)$$

$$\tilde{x}_{ji} = (-1)^{i+j} \det(\mathbf{M}_{ij}), \quad (5.21)$$

does not introduce new denominator factors (they are all contained in the denominator of $\det(\mathbf{X})$ and thus eliminated when forming the product in (5.19)).

This leads to the idea that the stability test can be carried out on the numerator of the determinant of $(\mathbf{I} + \mathbf{Q}\mathbf{P}_{22})$ to enforce strong stabilization. With the $(r \times s)$ Youla parameter \mathbf{Q} and the $(s \times r)$ plant \mathbf{P}_{22} of the form

$$\mathbf{Q} = \frac{1}{A_Q} \begin{bmatrix} \sum_{i=1}^k B_{11,i}^Q \theta_{11,i} & \cdots & \sum_{i=1}^k B_{1s,i}^Q \theta_{1s,i} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^k B_{r1,i}^Q \theta_{r1,i} & \cdots & \sum_{i=1}^k B_{rs,i}^Q \theta_{rs,i} \end{bmatrix}, \quad (5.22)$$

$$\mathbf{P}_{22} = \frac{1}{A_P} \begin{bmatrix} B_{11}^P & \cdots & B_{1r}^P \\ \vdots & \ddots & \vdots \\ B_{s1}^P & \cdots & B_{sr}^P \end{bmatrix}, \quad (5.23)$$

one can write

$$\mathbf{I} + \mathbf{Q}\mathbf{P}_{22} = \frac{1}{A_Q A_P} \begin{bmatrix} A_Q A_P + \beta_{11} & \dots & \beta_{1r} \\ \vdots & \ddots & \vdots \\ \beta_{r1} & \dots & A_Q A_P + \beta_{rr} \end{bmatrix} \quad (5.24)$$

with

$$\beta_{jk} = \sum_{m=1}^s B_{mk}^P \sum_{i=1}^k B_{jm,i}^Q \theta_{jm,i}. \quad (5.25)$$

Example 5.2.2: Strong stabilization constraint for MIMO feedback design

Given the stable, proper, non-minimum-phase MIMO plant

$$P(s) = \begin{bmatrix} \frac{s-1}{s+2} & 0 \\ 1 & \frac{s-2}{s+3} \end{bmatrix} \quad (5.26)$$

and the Youla parameter

$$Q(s) = \begin{bmatrix} \frac{3}{s+1}\theta_{11} & \frac{3}{s+1}\theta_{12} \\ \frac{3}{s+1}\theta_{21} & \frac{3}{s+1}\theta_{22} \end{bmatrix}, \quad (5.27)$$

formulate the constraint corresponding to a stable controller $\mathbf{K}(s)$, where

$$\mathbf{K}(s) = (\mathbf{I} + \mathbf{P}(s)\mathbf{Q}(s))^{-1} \mathbf{Q}(s). \quad (5.28)$$

Since $\mathbf{Q}(s)$ has stable poles (at -1 with multiplicity 2 in a minimal realization), only the transfer $(\mathbf{I} + \mathbf{P}(s)\mathbf{Q}(s))^{-1}$ needs to be tested for stability. With

$$A_P(s) = (s+2)(s+3), \quad (5.29)$$

$$\mathbf{B}_P(s) = \begin{bmatrix} B_{11}^P & B_{12}^P \\ B_{21}^P & B_{22}^P \end{bmatrix} = \begin{bmatrix} (s-1)(s+3) & 0 \\ (s+2)(s+3) & (s-2)(s+2) \end{bmatrix}, \quad (5.30)$$

$$A_Q(s) = (s+1), \quad (5.31)$$

$$\mathbf{B}_Q(s) = \begin{bmatrix} B_{11}^Q & B_{12}^Q \\ B_{21}^Q & B_{22}^Q \end{bmatrix} = \begin{bmatrix} 3\theta_{11} & 3\theta_{12} \\ 3\theta_{21} & 3\theta_{22} \end{bmatrix}, \quad (5.32)$$

the controller $\mathbf{K}(s)$ is stable iff the numerator polynomial $p(s)$ of

$$\det \left(\mathbf{I} + \frac{1}{A_Q(s)A_P(s)} \mathbf{B}_Q(s)\mathbf{B}_P(s) \right) = \left(\frac{1}{A_Q(s)A_P(s)} \right)^2 \det (A_Q(s)A_P(s)\mathbf{I} + \mathbf{B}_Q(s)\mathbf{B}_P(s)) \quad (5.33)$$

$$= \left(\frac{1}{A_Q(s)A_P(s)} \right)^2 p(s) \quad (5.34)$$

is Hurwitz (i.e. has all its roots in the LHP). Evaluating the polynomial $p(s)$ yields:

$$p(s) = \begin{vmatrix} A_Q(s)A_P(s) + \beta_{11}(s) & \beta_{12}(s) \\ \beta_{21}(s) & A_Q(s)A_P(s) + \beta_{22}(s) \end{vmatrix} \quad (5.35)$$

$$= (A_Q(s)A_P(s) + \beta_{11}(s))(A_Q(s)A_P(s) + \beta_{22}(s)) - \beta_{12}(s)\beta_{21}(s) \quad (5.36)$$

with

$$A_Q A_P = (s+1)(s+2)(s+3) \quad (5.37)$$

$$\beta_{11}(s) = \sum_{m=1}^2 B_{1m}^Q B_{m1}^P = B_{11}^Q B_{11}^P + B_{12}^Q B_{21}^P \quad (5.38)$$

$$= 3\theta_{11}(s-1)(s+3) + 3\theta_{12}(s+2)(s+3) \quad (5.39)$$

$$= 3(\theta_{11} + \theta_{12})s^2 + 3(2\theta_{11} + 5\theta_{12})s + 3(-3\theta_{11} + 6\theta_{12}) \quad (5.40)$$

$$\beta_{12}(s) = \sum_{m=1}^2 B_{1m}^Q B_{m2}^P = B_{11}^Q B_{12}^P + B_{12}^Q B_{22}^P \quad (5.41)$$

$$= 3\theta_{12}s^2 - 12\theta_{12} \quad (5.42)$$

$$\beta_{21}(s) = \sum_{m=1}^2 B_{2m}^Q B_{m1}^P = B_{21}^Q B_{11}^P + B_{22}^Q B_{21}^P \quad (5.43)$$

$$= 3(\theta_{21} + \theta_{22})s^2 + 3(2\theta_{21} + 5\theta_{22})s + 3(-3\theta_{21} + 6\theta_{22}) \quad (5.44)$$

$$\beta_{22}(s) = \sum_{m=1}^2 B_{2m}^Q B_{m2}^P = B_{21}^Q B_{12}^P + B_{22}^Q B_{22}^P \quad (5.45)$$

$$= 3\theta_{22}s^2 - 12\theta_{22}. \quad (5.46)$$

This leads to the following multivariate polynomial in s, θ_{jk} ($j, k = 1, 2$):

$$p(s) = s^6 + 3(\theta_{11} + \theta_{12} + \theta_{22} + 4)s^5 \quad (5.47)$$

$$+ (24\theta_{11} + 33\theta_{12} + 18\theta_{22} + 9\theta_{11}\theta_{22} - 9\theta_{12}\theta_{21} + 58)s^4 \quad (5.48)$$

$$+ (60\theta_{11} + 141\theta_{12} + 21\theta_{22} + 18\theta_{11}\theta_{22} - 18\theta_{12}\theta_{21} + 144)s^3 \quad (5.49)$$

$$+ (30\theta_{11} + 291\theta_{12} - 54\theta_{22} - 63\theta_{11}\theta_{22} + 63\theta_{12}\theta_{21} + 193)s^2 \quad (5.50)$$

$$+ (288\theta_{12} - 63\theta_{11} - 132\theta_{22} - 72\theta_{11}\theta_{22} + 72\theta_{12}\theta_{21} + 132)s \quad (5.51)$$

$$+ (108\theta_{12} - 54\theta_{11} - 72\theta_{22} + 108\theta_{11}\theta_{22} - 108\theta_{12}\theta_{21} + 36). \quad (5.52)$$

It can be seen that this polynomial contains second-order mixed monomials of θ_{jk} which makes the corresponding Hurwitz matrix definiteness constraint a Bi-Linear Matrix Inequality (BMI) or, generally, a Polynomial Matrix Inequality (PMI) constraint in θ_{jk} .

BMIs or PMIs are generally non-convex and thus cannot directly be solved using LMI methods; however LMI relaxations (approximations) are possible and, using additional lifting variables and constraints, a solution to a PMI problem can be approached asymptotically (with increasing lifting order, i.e. problem inflation) [106]. Tools such as *GloptiPoly 3* [64] can tackle such problems via successively higher approximation precision until either a certified optimal solution is found or other termination criteria are fulfilled. Also, BMI solvers start becoming available, see for example the PhD thesis of Stingl [146] and the PENBMI solver's manual [74].

Note that it is not possible any more to explicitly state the admissible region in the parameter (θ_{jk}) space that fulfill this constraint.

For illustration, Fig. 5.4 depicts the stability region in the $\theta_{11} - \theta_{22}$ -plane with $\theta_{12} = \theta_{21} = 0$, Fig. 5.5 and Fig. 5.6 depict the stability region in the $\theta_{12} - \theta_{21}$ -plane with fixed $\theta_{11} = \theta_{22} = 0$ respectively $\theta_{11} = -1, \theta_{22} = -0.5$. The latter two are clearly non-convex admissible sets in the parameter space.

As evident from Example 5.2.2, this formulation of the strong stabilization constraint is exact, but computationally not tractable with growing system order.

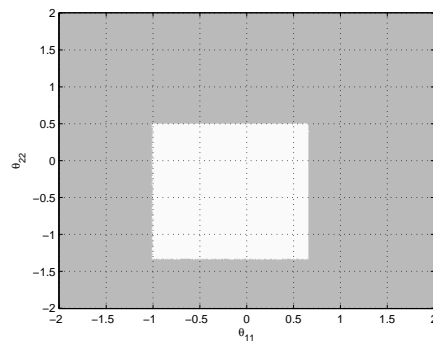


Figure 5.4: Stability region slice (white) for fixed $\theta_{12} = \theta_{21} = 0$

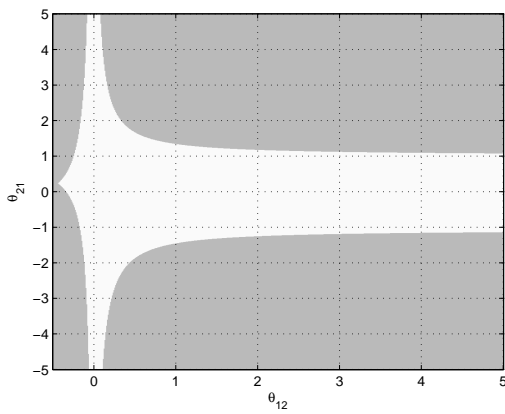


Figure 5.5: Stability region slice (white) for fixed $\theta_{11} = \theta_{22} = 0$

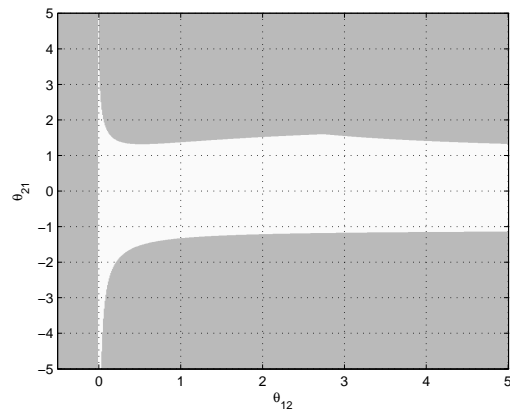


Figure 5.6: Stability region slice (white) for fixed $\theta_{11} = -1, \theta_{22} = -0.5$

5.2.3 Strong Stabilization Constraint Formulation

Consider the Youla parametrization of a stable plant \mathbf{P} and the feedback controller \mathbf{K} (which is to be designed) as in Sec. 3.6.3.

Note that the controller system matrix \mathbf{A}_K in (3.69) is affine in $\boldsymbol{\theta}$ which is seen by expanding (3.69) by either affine parametrization onset of \mathbf{Q} in (3.67). Exemplarily, with input parametrization, one obtains:

$$\mathbf{A}_K = \begin{bmatrix} \mathbf{A}_P - \sum_{i=1}^{n_Q} \mathbf{B}_P \mathbf{D}_{Q_i} \mathbf{C}_P \theta_i & \mathbf{B}_P \mathbf{C}_Q \\ - \sum_{i=1}^{n_Q} \mathbf{B}_{Q_i} \mathbf{C}_P \theta_i & \mathbf{A}_Q \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_P & \mathbf{B}_P \mathbf{C}_Q \\ \mathbf{0} & \mathbf{A}_Q \end{bmatrix}}_{\mathbf{A}_{K0}} + \sum_{i=1}^{n_Q} \underbrace{\begin{bmatrix} \mathbf{B}_P \mathbf{D}_{Q_i} \mathbf{C}_P & \mathbf{0} \\ \mathbf{B}_{Q_i} \mathbf{C}_P & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_{K_i}} \theta_i. \quad (5.53)$$

To obtain a stable controller with the convex design outlined above, a constraint to ensure stability of \mathbf{A}_K in (3.69) is necessary. Formally, the set of parameters that yield a stable controller is introduced:

$$\boldsymbol{\Theta}_{\text{sstab}} = \left\{ \boldsymbol{\theta} : \max_i [\Re(\lambda_i(\mathbf{A}_K(\boldsymbol{\theta}))) \leq 0] \right\} \quad (5.54)$$

Note that by requiring an initially stable plant, the strong stabilizability is fulfilled per definition. The zero feedback, obtained by $\boldsymbol{\theta} = \mathbf{0} \in \boldsymbol{\Theta}_{\text{sstab}}$, is strongly stabilizing.

5.2.4 Lyapunov Strong Stability Formulation

To test the stability of the controller, the Lyapunov stability criterion can be utilized. Its LMI-style formulation (with $\mathbf{P} = \mathbf{P}^T \succ 0$),

$$\mathbf{P} \mathbf{A}_K + \mathbf{A}_K^T \mathbf{P} \prec 0 \quad (5.55)$$

is however, although sufficient, not affine in the free LMI parameters (the free variables in $\mathbf{A}_K = \mathbf{A}_K(\boldsymbol{\theta})$ and \mathbf{P} arise in products) and is thus not applicable for synthesis. Moreover, a high (design) order n_K of \mathbf{A}_K yields a large problem as there are $\frac{1}{2}(n_K^2 - n_K)$ free variables only in \mathbf{P} .

However, in the spirit analogous to the DK-iteration, the problem could be addressed iteratively:

1. Initialize $M > 0$, $k = 1$, $\boldsymbol{\theta}_1 := \mathbf{0}$
2. Solve the LMI problem

$$(\mathbf{P}_k, \gamma_k) := \arg \min_{\mathbf{P}, \gamma} \gamma \quad (5.56)$$

$$\text{s.t. } \mathbf{P} \mathbf{A}_K(\boldsymbol{\theta}_k) + \mathbf{A}_K^T(\boldsymbol{\theta}_k) \mathbf{P} \leq \gamma \mathbf{I}, \quad (5.57)$$

$$\|\mathbf{P}\| \leq M. \quad (5.58)$$

The optimal value objective γ_k is strictly negative if $\mathbf{A}_K(\boldsymbol{\theta}_k)$ is stable. Note that the type of norm constraint on \mathbf{P} is arbitrary but necessary to bound the solution. Typically, LMI solvers can be parametrized to limit the decision variables in a ball which also could be utilized.

3. Formulate and compute the control design optimization problem

$$(\boldsymbol{\theta}_{k+1}, \boldsymbol{\alpha}_{k+1}) := \arg \min_{\boldsymbol{\theta}, \boldsymbol{\alpha}} \mathbf{c}^T \boldsymbol{\alpha} \quad (5.59)$$

with the constraints of the control design problem and with the additional constraint

$$\mathbf{P}_k \mathbf{A}_K(\boldsymbol{\theta}) + \mathbf{A}_K^T(\boldsymbol{\theta}) \mathbf{P}_k + (-\gamma_k) \varepsilon \mathbf{I} \preceq 0, \quad (5.60)$$

where $\varepsilon \in (0; 1)$ is a parameter to tune the allowed controller stability loss in optimizing $\boldsymbol{\theta}$.

4. If performance is satisfactory or no improvement is obtained any more, stop. Otherwise increase k by 1 and goto step 2.

Unfortunately, this most simple Lyapunov condition allows only for very small changes in $\boldsymbol{\theta}$ per iteration, so a high number of iterations are required to approach the stability limit. Moreover, the problem size quickly grows with the size of \mathbf{A}_K which renders the method impractical already for medium-order controllers, also restricting the dimension of the Youla parameter base and thus attainable performance.

One remedy is to approximate the feasible region by convex regions and formulate corresponding constraints. In the following, an algorithm that embeds such constraints is proposed in several variants.

5.2.5 Heuristic Embedded Convex Region Constraint

Hypercube Embedding: Box Constraint

This section reports on a novel heuristic method to obtain strong stabilization in high-dimensional control design by a convex region embedding procedure to formulate a corresponding convex constraint. The proposed algorithm approximately embeds a convex region in the non-convex strong stabilization feasibility set Θ_{sstab} as in (5.54). It is assumed here that the feasible set Θ_{sstab} features a particular structure:

1. $\mathbf{0} \in \Theta_{\text{sstab}}$: The origin is always feasible (which is fulfilled for the case of an open-loop stable plant; the respective controller (for zero $\boldsymbol{\theta}$ entries) is the (stable) zero transfer.
2. $\boldsymbol{\theta} \in \Theta_{\text{sstab}} \Rightarrow \lambda \boldsymbol{\theta} \in \Theta_{\text{sstab}} \forall \lambda \in [0, 1]$: The ray from the origin to each feasible $\boldsymbol{\theta}$ lies entirely in the feasible set. This property is a technical assumption to ease polytope embedding, but it has been observed that it holds in practice.

A simple and beneficial choice of the convex region shape that is embedded in Θ_{sstab} is a scaled hypercube, see Fig. 5.7. Other convex region shapes are discussed below. Exploiting the noted properties of Θ_{sstab} , the algorithm is given as follows:

1. Find $2n$ simplex polytope corners ($l = 1, \dots, n$, $\mathbf{e}_l = [0, \dots, 0, 1, 0, \dots, 0]^T$ is the l th unit vector)

$$\boldsymbol{\theta}_l^+ = \alpha \mathbf{e}_l : \alpha = \sup_{\alpha > 0} \alpha : \alpha \mathbf{e}_l \in \Theta_{\text{sstab}} \quad (5.61)$$

$$\boldsymbol{\theta}_l^- = \alpha \mathbf{e}_l : \alpha = \inf_{\alpha < 0} \alpha : \alpha \mathbf{e}_l \in \Theta_{\text{sstab}}. \quad (5.62)$$

2. Construct a hypercube $\Xi = (\boldsymbol{\theta}_1^- + \dots + \boldsymbol{\theta}_n^- \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_1^+ + \dots + \boldsymbol{\theta}_n^+)$.
3. Maximize a scalar $0 < \delta \leq 1$ such that a given number $M \gg 1$ of randomly sampled points $\boldsymbol{\theta} \in \delta \cdot \Xi$ pass the actual stability test $\max_i (\Re(\lambda_i(\mathbf{A}_K(\boldsymbol{\theta}))) < 0$.

From property 2, an extremal vertex for a given direction $\mathbf{e}_i \in \mathbb{R}^n$, $|\mathbf{e}_i| = 1$ can be found by a bisection algorithm using recursive binary search within a ray segment $\lambda \mathbf{e}_i$, $\lambda \in [\underline{\lambda}, \bar{\lambda}]$. With the stability testing function $\text{isstable}(\boldsymbol{\theta})$ and a given tolerance $\varepsilon > 0$, the algorithm can be stated as:

- Initialize e_i , $\underline{\lambda} \leftarrow 0$, $\bar{\lambda} \leftarrow M \in \mathbb{R}$, large
- If $\text{isstable}(\bar{\lambda}e_i)$, terminate.
- While $\Delta\lambda = \bar{\lambda} - \underline{\lambda} > \frac{\varepsilon}{M}$
 - $\lambda = \frac{1}{2}(\bar{\lambda} + \underline{\lambda})$
 - If $\text{isstable}(\lambda e_i)$: $\bar{\lambda} \leftarrow \lambda$
 - Else: $\underline{\lambda} \leftarrow \lambda$

The number of bisection iterations needed to obtain a fixed relative accuracy $1 \gg \frac{\varepsilon}{M} \geq \frac{\bar{\lambda} - \underline{\lambda}}{M} > 0$ is

$$k = \left\lceil \log_2 \frac{M}{\varepsilon} \right\rceil, \quad (5.63)$$

which takes the values 10, 20, and 30 for relative accuracies $\frac{\varepsilon}{M}$ of 10^{-3} , 10^{-6} , and 10^{-9} .

This construction is computationally efficient (the initial polytope has only $2n$ vertices, the hypercube has only $2n$ facets, and the line search over δ can be efficiently done via binary search). In some test scenarios and in the aircraft control design case, this algorithm yields better performing stable controllers as compared to merely downscaling the Youla parameter of an unstable controller. Finally, the inherent conservativeness can be reduced by alternately applying this algorithm and the actual LMI optimization, with the optimizer θ^* as new starting point (origin) for the hypercube construction.

Figure 5.7 illustrates the hypercube constraint in two parameter directions. The initial hypercube limits are found along the parameter axes and typically yield a too large parameter region \mathcal{E} . However, basic parameter scaling information with respect to controller stability is incorporated. Down-scaling by a suitable factor α embeds the contracted hypercube $\alpha\mathcal{E}$ sufficiently well in the non-convex actual parameter set Θ_{sstab} in which strong stabilization is attained. However, as can be seen, conservativeness of this approach can be large. In turn, this formulation is efficient in the number of half-spaces and yields a closed set, so the optimization problem is guaranteed to be bounded.

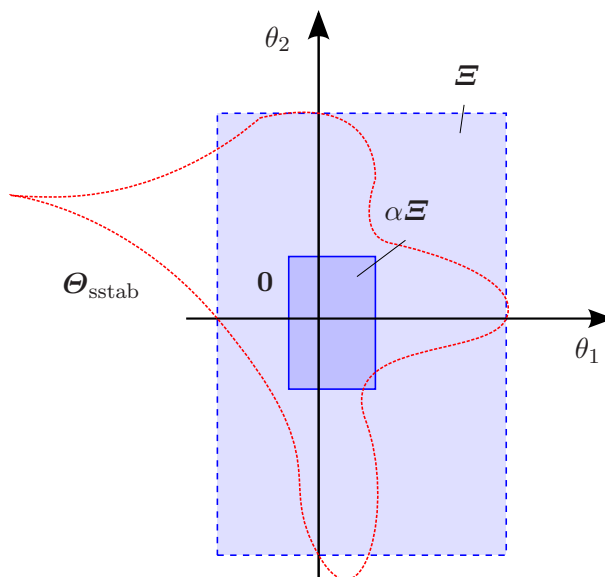


Figure 5.7: Hypercube constraint $\theta \in \alpha\mathcal{E}_{\text{sstab}}$ with downscaling factor α .

Extension: Polytope embedding

The shape of a hypercube as embedded region simplifies the constraints to parameter box constraints. However, any other polytopic region can be utilized. In the following, first some basic properties of polytopic / polyhedral sets are given. Then, strong stabilization LP constraints are formulated for given search directions in which the bounding hyperplanes are either orthogonal to the search directions or tangential to the boundary of the strongly stabilizing parameter set Θ_{sstab} .

First, some facts on convex polytopes are considered from [46]. These can typically be defined as the intersection of half-spaces (half-space representation) or as the convex hull (and its interior) of a set of points (vertex representation). In the half-space representation, all linear inequalities defining the polytope's facets can be combined to the constraint

$$\mathbf{A}\mathbf{x} \leq \mathbf{b} \quad (5.64)$$

which is an efficient representation for LP (or LMI) optimization problems if the number of facets (rows in \mathbf{A}) is low.

Note that the equation

$$\mathbf{a}^T \boldsymbol{\theta} = \mathbf{a}^T \mathbf{b} \quad (5.65)$$

defines all points $\boldsymbol{\theta}$ in a hyperplane which is orthogonal to the vector \mathbf{a} and goes through point \mathbf{b} . Furthermore, let k be the fixed, chosen number of facets of a polyhedron to construct (not necessarily bounded) and let $\boldsymbol{\theta}_j$, $j = 1, \dots, k$ be an associated set of sampling directions. These could be randomly sampled and/or chosen based on preferred directions towards an unconstrained optimal solution.

Variante 1: Orthogonal hyperplanes. A set of admissible half-spaces, each orthogonal to the associated search direction, is constructed analogously to the hypercube case as follows:

1. For each $j \in \{1, \dots, k\}$
 - (a) Obtain the critical scaling $\beta_j = \arg \max_{\beta} \beta \boldsymbol{\theta}_j \in \Theta_{\text{sstab}}$ that yields marginal strong stability in search direction $\boldsymbol{\theta}_j$ by binary search.
 - (b) The halfspace delimited by a hyperplane orthogonal to $\boldsymbol{\theta}_j$ through the point $\beta_j \boldsymbol{\theta}_j$ and including the origin is defined by

$$\boldsymbol{\theta}_j^T \boldsymbol{\theta} \leq \beta_j \boldsymbol{\theta}_j^T \boldsymbol{\theta}_j. \quad (5.66)$$

This onset does not utilize local information on the shape of the boundary of Θ_{sstab} , so it is suitable where gradient information is not available. Note that the hypercube constraint is obtained for $k = 2n_Q$ and $\boldsymbol{\theta}_j \in \{\pm \mathbf{e}_1, \dots, \pm \mathbf{e}_{n_Q}\}$. Figure 5.8 shows the orthogonal hyperplane constraint formulation with respect to three sampling directions.

Variante 2: Tangential hyperplanes. Let β_j be the critical scaling as above. Assume that the tangent space of the boundary of Θ_{sstab} at $\beta_j \boldsymbol{\theta}_j$ exists and is well-defined. This is the case if the function

$$f(\boldsymbol{\theta}) = \max_i \Re \{ \lambda_i(\mathbf{A}_K(\boldsymbol{\theta})) \} = \max_i \Re \left\{ \lambda_i \left(\mathbf{A}_{K0} + \sum_{j=1}^{n_Q} \mathbf{A}_{Kj} \boldsymbol{\theta}_j \right) \right\} \quad (5.67)$$

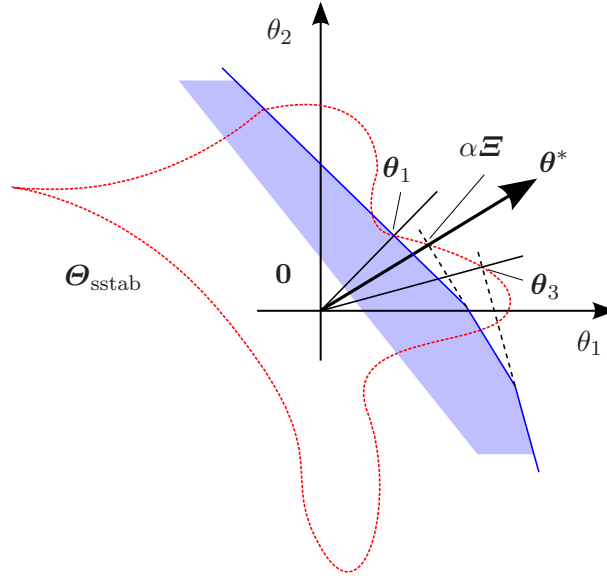


Figure 5.8: Polytope embedding, variant 1: Orthogonal hyperplanes

is continuously differentiable at $\beta_j \boldsymbol{\theta}_j$, which, in turn, is fulfilled if the maximum is attained uniquely (or by just one complex-conjugate eigenvalue pair). Thereby, $\lambda_i(\cdot)$ is the i th eigenvalue of the argument. Then the gradient of f ,

$$\nabla f(\boldsymbol{\theta}) = \left[\frac{\partial f}{\partial \theta_1} \quad \dots \quad \frac{\partial f}{\partial \theta_{n_Q}} \right] \quad (5.68)$$

exists and is orthogonal to the tangential hyperplane

$$(\nabla f(\beta_j \boldsymbol{\theta}_j))^T \boldsymbol{\theta} = (\nabla f(\beta_j \boldsymbol{\theta}_j))^T \beta_j \boldsymbol{\theta}_j. \quad (5.69)$$

Under the uniqueness conditions above, the gradient can be expressed analytically from the eigenvalue equation. Assume i is the critical eigenvalue index, λ_i , \mathbf{v}_i^T , and \mathbf{u}_i are the eigenvalue and its right and left eigenvectors normalized such that $\mathbf{v}_i^T \mathbf{u}_i = 1$ holds. With $\mathbf{X} = \mathbf{A}_K(\beta_j \boldsymbol{\theta}_j)$ and

$$\mathbf{X} \mathbf{u}_i = \lambda_i \mathbf{u}_i, \quad \mathbf{v}_i^T \mathbf{X} = \lambda_i \mathbf{v}_i^T, \quad (5.70)$$

the perturbed eigenvalue equation $(\mathbf{X} + d\mathbf{X})(\mathbf{u}_i + d\mathbf{u}_i) = (\lambda_i + d\lambda_i)(\mathbf{u}_i + d\mathbf{u}_i)$ can be simplified to

$$\mathbf{v}_i^T d\mathbf{X} \mathbf{u}_i + \mathbf{v}_i^T (\mathbf{X} + d\mathbf{X}) d\mathbf{u}_i = d\lambda_i + (\lambda_i + d\lambda_i) \mathbf{v}_i^T d\mathbf{u}_i. \quad (5.71)$$

To obtain a first-order approximation the eigenvector perturbation $d\mathbf{u}_i$ is neglected (which is admissible for small perturbations and well-separated eigenvalues [25]), arriving at

$$\mathbf{v}_i^T d\mathbf{X} \mathbf{u}_i \cong d\lambda_i. \quad (5.72)$$

To formulate the gradient, note that $d\mathbf{X} = \sum_j \mathbf{A}_{Kj} d\theta_j$, so

$$\frac{\partial \lambda_i}{\partial \theta_j} = \mathbf{v}_i^T \mathbf{A}_{Kj} \mathbf{u}_i \quad (5.73)$$

holds.

The constraint definition for the half-space which is delimited by the hyperplane tangential to Θ_{sstab} at $\beta_j \theta_j$ and which includes the origin $\mathbf{0}$ is given by

$$(\nabla f(\beta_j \theta_j)) \theta \leq (\nabla f(\beta_j \theta_j)) \beta_j \theta_j \quad \text{if } (\nabla f(\beta_j \theta_j)) \beta_j \theta_j > 0 \quad (5.74)$$

$$-(\nabla f(\beta_j \theta_j)) \theta \leq -(\nabla f(\beta_j \theta_j)) \beta_j \theta_j \quad \text{if } (\nabla f(\beta_j \theta_j)) \beta_j \theta_j < 0 \quad (5.75)$$

where $(\nabla f(\beta_j \theta_j)) = [\mathbf{v}_i^T \mathbf{A}_{K1} \mathbf{u}_i \quad \dots \quad \mathbf{v}_i^T \mathbf{A}_{Kn_Q} \mathbf{u}_i]$.

Figure 5.9 shows the tangential hyperplane constraints defined at three sampling directions. The local shape of the edge of Θ_{sstab} is approximated, so this constraint formulation is expected to give good results if the sampling directions are sufficiently close and cover the optimal parameter direction.

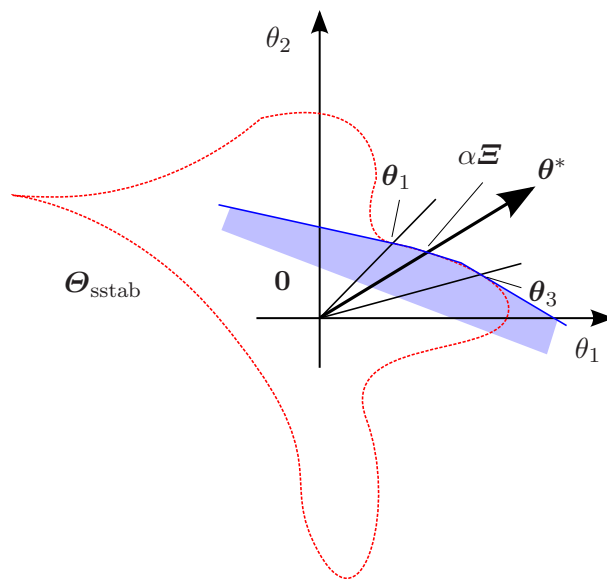


Figure 5.9: Polytope embedding, variant 2: Tangential hyperplanes

Remark on the curse of dimensionality: The reason for choosing half-space constraints as above, instead of formulating a convex-hull polytope of a set of critical vertices, lies in the dimensionality of the representation: note that the polytope which represents the convex hull and its interior of k vertices typically has an exponential number of facets (exponential in n_Q and k) [37] which renders the constraint (and the LMI problem) numerically non-tractable. The edge theorem [156] could be used to reduce the required stability tests to all one-dimensional bounding edges, but their number also grows exponentially with the dimension of the parameter space.

5.2.6 Test results using a high-order model

A high-order model is used to demonstrate the effect of the hypercube strong stabilization constraint in the following. A lateral BWB aircraft model is utilized, see Sec. 4.2. It is of order 42 with 8 outputs (4 performance channels, 4 measurement signals) and 3 inputs (1 exogenous disturbance input, 2 control signals). A static output feedback gain has been connected in closed loop for basic robust response shaping. The feedback Youla parametrization for this conditioned, open-loop stable plant is formed, using a basis of $\mathbf{Q}(s)$ of total dynamic order 56 and $n_Q = 40$ basis functions are utilized.

Table 5.1: Strong stabilization test results: l_∞ objective, 41 LMI variables.

Constr.	\mathbf{K} states	$f(\boldsymbol{\theta}^*)$, stability	objective	CPU time in s	constr.	iter.
-	56	4.69 (4 poles unstable)	0.0093	252	137	10
\mathcal{C}_{100}	55	0.120 (2 p. unstable)	0.0591	12	134	4
\mathcal{C}_{300}	54	0.334 (1 p. unstable)	0.0947	32	143	6
\mathcal{C}_{1000}	54	0.091 (1 p. unstable)	0.3283	6	116	5
\mathcal{C}_{10000}	51	< 0 (stable)	0.4379	4	113	4

As described in Sec. 5.2.5, a hypercube (box) constraint is embedded in the strong stabilization parameter set Θ_{sstab} such that a large number of random samples within the hypercube all yield a stable controller. These hypercubes are constructed using various choices of the maximal parameter magnitude $M = \{100, 300, 1000, 10000\}$ (which influences conservativeness due to the downscaling step) and are called \mathcal{C}_{100} , \mathcal{C}_{300} , \mathcal{C}_{1000} , and \mathcal{C}_{10000} in the following.

Table 5.1 shows the design results for the studied set of hypercube constraints obtained at a 3 GHz intel Pentium 4 workstation. Without strong stabilization constraint, the resulting controller is unstable and yields very good performance. For the hypercube constraints \mathcal{C}_{100} , \mathcal{C}_{300} , and \mathcal{C}_{1000} , the controller is still unstable. This is an indication that even by using intensive search in the interior of the tested parameter space region, the critical regions are not reached easily. Finally, the hypercube \mathcal{C}_{10000} constraint yields a stable controller. It is evident that the achieved performance is significantly lower than the non-strongly-stabilizing solutions.

5.3 Case Study 1: Strongly Stabilizing Lateral Feedback Controller for Flexible BWB Aircraft

This section presents convex feedback control design for the lateral dynamics of the considered BWB aircraft (see Sec. 4.2). A multitude of stringent constraints and goals are given in the time and frequency domain. An initial controller is designed using robust modal control design as in Sec. 3.6.2 (compare [93]) to achieve some of the goals most closely related to eigenstructure assignment. Based on the preshaped plant, a convex synthesis approach using the Youla parametrization and an LMI formulation is taken, optimizing directly for time- and frequency-domain goals. Moreover, a heuristic algorithm to achieve strong stabilization (i.e., a stable controller) in such high-dimensional problem is proposed. A stable feedback law is obtained that is validated successfully on all considered mass cases. High performance in loads alleviation and vibration reduction are achieved, and particularly the time-domain response specifications are obeyed, easing the task of subsequent command shaping (feed-forward design), see Sec. 5.5.

Previous, closely related studies include a general integrated methodology for multi-objective robust control design [133], LQ-based lateral control designs of the considered BWB aircraft [134], followed by the application of a genetic algorithm for parameter optimization of a multi-objective \mathcal{H}_∞ DK-iteration design [135]. The feedback controller designed in this section is incorporated in the subsequent feed-forward control design in [137]. Longitudinal BWB control using LPV control concepts are studied in [157].

This section presents the second step in a state-of-the-art flight control design for a novel application: feed-forward control design for the lateral dynamics of a large, flexible BWB transport aircraft predesign model. It is based on the feedback-controlled aircraft (the feedback design has been carried out by the authors in [138]). For given roll and side slip maneuvers, a scheduled feed-forward controller is designed by a convex multimodel approach. A set of stringent constraints and goals both in time and frequency domain are directly taken into account. Validation on various mass cases shows a significant, robust reduction in maneuver-induced loads and lateral accelerations while particularly the time-domain response specifications are obeyed and coupling is suppressed.

5.3.1 Initial Feedback Control Design

To achieve most of the general and tracking specifications in Tab. 4.1, an initial controller is designed by robust partial eigenstructure assignment (utilizing the MATLAB[®] Robust Modal Control Toolbox supplied with the book [93]). This is done in two steps:

1. Assign low-frequency (rigid-body) dynamics using low-pass output feedback,
2. Increase the damping of high-frequency flexible modes via a bandpass-filtered output feedback through eigenvector projection.

For step 1, a rigid-body model (consisting only of the states β , p (roll rate), r (yaw rate), ϕ (roll angle) and the rudder- and aileron second-order dynamics) was extracted from the full-order system by state truncation at a chosen mass case. It is augmented by two SISO third-order Butterworth low-pass filters with $\omega_c = 3 \text{ rad/s}$ at the actuator inputs (the dynamic order and corner frequency have been chosen to obtain sufficiently low spill-over in the validation). The relevant plant open-loop poles lie close to the respective poles of the full-order model at $p_{\text{rig},1} = -0.84 \cdot 10^{-3}$ (spiral mode), $p_{\text{rig},2,3} = -0.09 \pm 0.56i$ (DR mode), $p_{\text{rig},4} = -1.24$ (roll resilience). The desired DR pole location is at $p_{\text{rig,des},2,3} = \frac{|p_{\text{rig},2}|}{\sqrt{2}} (-1 \pm i) = -0.4 \pm 0.4i$. The DR damping requirement and the decoupling specifications (and partially the performance

specifications) are cast into eigenstructure constraints, see [93]:

$$\begin{aligned} p_{\text{rig,des},1} &= -0.6 & \mathbf{V}_1 &= [*, 0, *, 0, 0, 0, *, *]^T \\ p_{\text{rig,des},2,3} &= -0.4 \pm 0.4i & \mathbf{V}_{2,3} &= [0, *, 0, *, *, *, 0, 0]^T \\ p_{\text{rig,des},4} &= -1.24 & \mathbf{V}_4 &= [0, *, 0, *, *, *, 0, 0]^T, \end{aligned}$$

where the remaining eigenvector elements are unconstrained. The computed feedback gain robustly assigns a high DR mode damping, and the Butterworth low-pass filters are absorbed, yielding a preliminary controller $\mathbf{K}_{\text{prelim}}$ of order 6.

Design step 2 aims to increase damping of a flexible mode at $p_{\text{flex},1,2} = -0.1 \pm 19.6i$ that was found critical in earlier control designs. The previously neglected sensor delays must be considered now (and are absorbed into the full-order plant model). This full-order design plant is compensated with $\mathbf{K}_{\text{prelim}}$ and augmented by two SISO second-order Butterworth band-pass filters with pass band $14 \dots 25 \text{ rad/s}$ at the actuator inputs. The desired pole position $p_{\text{flex,des},1,2} = -0.3 \pm 19.6i$ is assigned, together with fixing another close flexible mode at its location $p_{\text{flex,des},3,4} = p_{\text{flex,des},3,4} = -0.6 \pm 20.2i$ using eigenvector projection. The final controller \mathbf{K}_{init} is composed of $\mathbf{K}_{\text{prelim}}$ and the band-pass filtered static output feedback gain resulting from step 2 with total dynamic order 10.

A comparison of open- and closed-loop validation step responses for selected, extremal mass cases is depicted in Fig. 5.10. It is evident that the controller yields mainly satisfactory results, however, the decoupling specifications are not entirely fulfilled (see roll reference (scaled aileron) to side slip). The initial control law \mathbf{K}_{init} is absorbed into the plants, and these preconditioned plants are used for convex dynamic feedback control design.

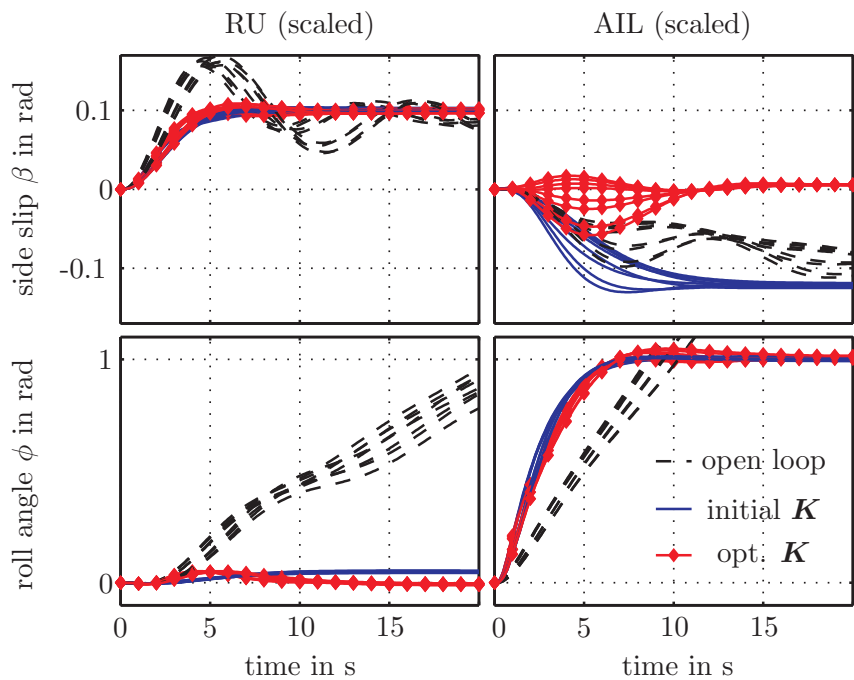


Figure 5.10: Step responses of selected, extremal mass cases without controller (open-loop), with initial controller, and with the convex optimized controller

Table 5.2: Objectives (O) and constraints (C) for feedback control optimization

O/C	Type & Path	Comment
O1	min. \mathcal{H}_∞ norm ($v_{\text{lat}} \rightarrow Ny_{\text{CG}}$)	
O2	min. \mathcal{H}_2 norm ($v_{\text{lat}} \rightarrow Ny_{\text{CG}}$)	
O3	min. l_∞ norm (AIL $\rightarrow Ny_{\text{CG}}$)	
C1	$ \theta_i $ bounds	stable controller
C2	l_∞ bounds (AIL \rightarrow roll)	retain response
C3	l_∞ bounds (RU \rightarrow side slip)	retain response
C4	l_∞ bounds (AIL \rightarrow side slip)	decoupling
C5	l_∞ bounds (RU \rightarrow roll)	decoupling
C6	l_∞ bounds (AIL $\rightarrow Mx_{\text{fin}}$)	avoid worsening

5.3.2 Convex Feedback Controller Optimization

The Youla parametrization is generated for a precompensated design plant (as outlined in Sec. 3.6.3), using an ad-hoc basis of $\mathbf{Q}(s)$ of total order 56. The basis functions are simple first- and second order SISO dynamics with their poles and zeros chosen within the expected control bandwidth, for each I/O-channel. A total of 40 scalar Youla parameters (4 inputs, 2 outputs, 5 SISO basis functions each) are utilized.

The convex optimization problem is formulated with specific constraints and objectives to retain the (satisfactory) tracking performance, while improving decoupling, disturbance rejection, and alleviating loads (see Tab. 5.2). The cost function is the sum of an \mathcal{H}_∞ , an \mathcal{H}_2 , and a time-weighted l_∞ objective, each normalized to 1 for the initial, precompensated plant. Constraint C1 restricts the parameters to the hypercube found to be strongly stabilizing (compare Sec. 5.2.5). The other constraints are l_∞ -bounds to retain the tracking performance and to avoid worsening of the peak values with respect to the precompensated plant.

The problem is solved to optimality and the constraint design grids are adaptively refined until all constraints are fulfilled on finely-spaced validation grids. Constraint C6 was found necessary in validation to limit fin loads in the roll maneuver, so it was added and the problem re-solved.

As an example, the AIL $\rightarrow Ny_{\text{CG}}$ step response in Fig. 5.11 (right) illustrates objective O3 with its bounds and the adapted definition grid. This, together with constraint C4 (whose effect is seen in Fig. 5.10), has the important consequence of achieving a coordinated turn and reasonably low lateral accelerations. All other responses are at least as good as those of the precompensated plant, with a number of notable improvements after optimization:

- Decoupling and coordinated turning is realized robustly (see Fig. 5.10).
- Disturbance attenuation is improved, particularly in the $v_{\text{lat}} \rightarrow Ny_{\text{CG}}$ (see Fig. 5.11 (left)) and $v_{\text{lat}} \rightarrow \beta$ paths.
- Wing and fin root loads are decreased in the RU and AIL step responses, both in terms of peak value as well as in DC gain.

5.3.3 Results, Robustness & Discussion

As evident from Fig. 5.10, the final controller is robustly stable and obtains good tracking and decoupling properties for all considered mass cases (low to full fuel, low to high payload) at cruise conditions. Detailed analysis of the entire validation set shows shortcomings of the control in low-fuel mass cases for the $v_{\text{lat}} \rightarrow Mx_{\text{wing}}$ and $v_{\text{lat}} \rightarrow Ny_{\text{CG}}$ transfers. High-frequency flexible

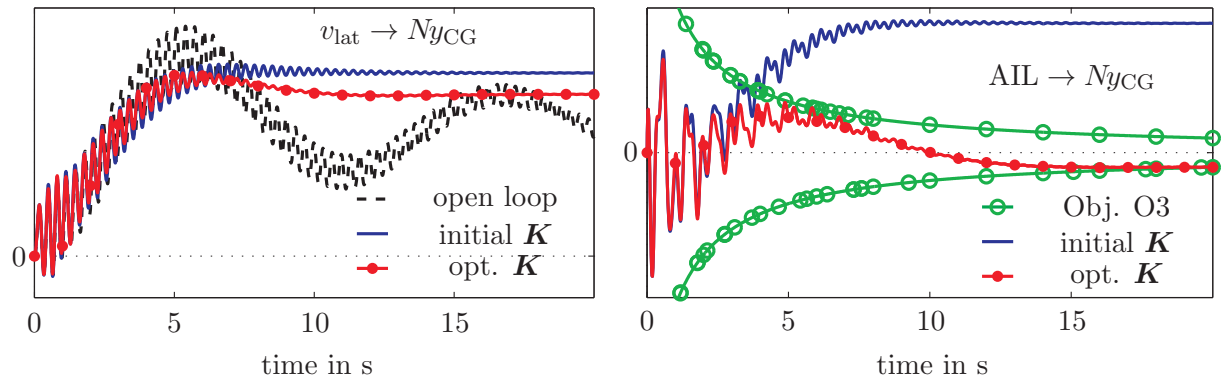


Figure 5.11: Optimization results: $v_{\text{lat}} \rightarrow Ny_{CG}$ is reduced via frequency-domain objectives O1,O2 (left); $AIL \rightarrow Ny_{CG}$ is reduced via a time-weighted l_{∞} objective O3 (right)

modes are excited in those mass cases, which requires further assessment. Potential solutions are a robust multimodel eigenstructure assignment (see [93]), utilizing a robustly performing initial controller (e.g. computed by DK-iterations), modeling of structured or unstructured uncertainty (and applying Q - μ -iterations in the convex synthesis framework, see [18]), or the synthesis of a scheduled control law (if the fuel level is available online for control) as in [151].

However, a high degree of disturbance rejection is achieved for $v_{\text{lat}} \rightarrow Mx_{\text{fin}}, \beta, \phi$ in all cases and in $v_{\text{lat}} \rightarrow Mx_{\text{wing}}, Ny_{CG}$ at medium and high fuel cases with typical reductions by 1...4 dB in \mathcal{H}_2 -norm and 5...8 dB in \mathcal{H}_{∞} -norm.

The final feedback controller is of dynamic order 84, which is clearly too high for implementation. For this, various robust controller order reduction methods (as discussed in [93], for example) are available.

Finally, the loads induced by the roll and side slip maneuvers are decreased as compared to the initial control design. In [137], the subsequent dynamic feed-forward design focuses on maneuver loads alleviation specifically.

5.3.4 Conclusions

A state-of-the-art flight control design for the lateral dynamics of a large, flexible BWB transport aircraft predesign model has been performed. A multitude of partially conflicting goals and constraints in the time, frequency, and eigenstructure domains have been directly considered. An initial controller has been designed using robust partial eigenstructure assignment via the Robust Modal Control MATLAB[®] Toolbox [93]. Then, a convex synthesis has been performed, optimizing the closed-loop for the remaining frequency- and time-domain objectives under time-domain and strong stabilization constraints. The resulting controller is validated at a set of aircraft mass cases with various fuel and payload levels at cruise conditions and performs well in the entire domain in terms of tracking, decoupling, and disturbance rejection. Problematic mass cases have been identified and discussed.

5.4 Convex design of a multi-model scheduled feed-forward control law

A convex formulation of the feed-forward control design problem is formulated in the following. The fact that the feed-forward-controlled plant transfer is directly affine in the control law allows to utilize a multi-model onset in the design. Furthermore, a method to parameterize (schedule)

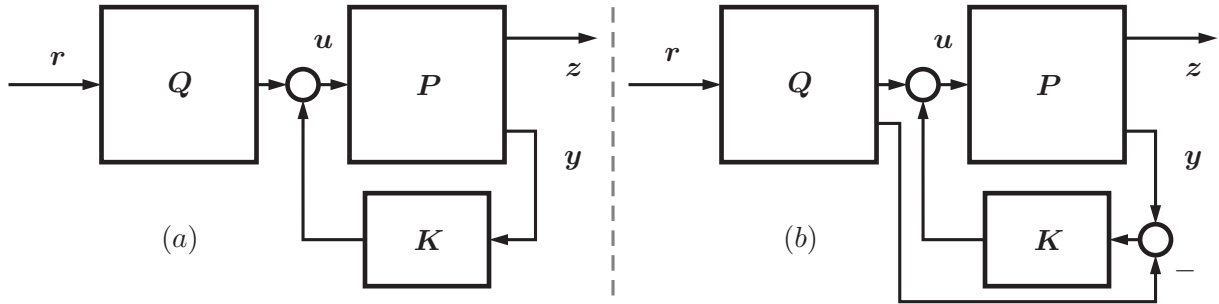


Figure 5.12: Two interconnection architectures of a (possibly scheduled) feed-forward controller Q with feedback-controlled plant P

the controller a priori is presented. These extensions enable a flexible design method to directly design robust and scheduled feed-forward controllers.

5.4.1 Problem statement

As introduced in Sec. 2.3.3, a discrete set of LTI validation plants $\mathcal{P} = \{P_j : j = 1, \dots, m\}$ is given in state-space form,

$$\begin{aligned} \dot{\mathbf{x}}_j &= \mathbf{A}_j \mathbf{x}_j + \mathbf{B}_j \mathbf{u}_j \\ \mathbf{y}_j &= \mathbf{C}_j \mathbf{x}_j + \mathbf{D}_j \mathbf{u}_j, \end{aligned} \quad (5.76)$$

describing a (possibly non-linear) parameter-dependent system at fixed parameter values. Let $\boldsymbol{\lambda} \in \mathbb{R}^a, \boldsymbol{\eta} \in \mathbb{R}^b$ be system parameters that are known and unknown (but bounded) during operation, respectively. Then, each LTI plant in \mathcal{P} can be written as $P_j(s) = P(s, \boldsymbol{\lambda}_j, \boldsymbol{\eta}_j)$.

The task is to design a feed-forward control law that provides optimal performance under given constraints and objective formulations for all (stable or stabilized) validation plants in \mathcal{P} . The plant is controlled by a robust LTI feedback compensator $K(s)$. Two possible interconnection architectures of the plant P , the feedback compensator K , and a parameter-dependent (scheduled) feed-forward controller Q are depicted in Fig. 5.12. Thereby, \mathbf{r} are reference signals, \mathbf{u} are the actual control inputs to the plant, \mathbf{z} are the exogenous output signals, and \mathbf{y} are the plant measurement signals used for feedback control. Noise inputs are not considered here as they are not relevant for feed-forward design.

For fixed parameters $\boldsymbol{\lambda}_j, \boldsymbol{\eta}_j$ the LTI plant $P_j(s) = P(s, \boldsymbol{\lambda}_j, \boldsymbol{\eta}_j)$ is obtained, and the feed-forward-controlled closed-loop transfer function $T_{zr,j}(s)$ from \mathbf{r} to \mathbf{z} is

$$\mathbf{T}_{zr} = \underbrace{P_{j,1} (\mathbf{I} - \mathbf{K} P_{j,2})^{-1}}_{\tilde{P}_{j,\text{cl}}} \mathbf{Q}_j, \quad (5.77)$$

for the architecture in Fig. 5.12, *left*, and

$$\mathbf{T}_{zr} = \underbrace{P_{j,1} (\mathbf{I} - \mathbf{K} P_{j,2})^{-1} [\mathbf{I} \mid -\mathbf{K}]}_{\tilde{P}_{j,\text{cl}}} \begin{bmatrix} \mathbf{Q}_{j,1} \\ \mathbf{Q}_{j,2} \end{bmatrix}, \quad (5.78)$$

for the architecture in Fig. 5.12, *right*, respectively. The plant is appropriately partitioned:

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} P_{j,1}(s) \\ P_{j,2}(s) \end{bmatrix} \mathbf{u}. \quad (5.79)$$

For the second architecture, the feed-forward controller is partitioned as well:

$$\mathbf{Q}_j(s) = \mathbf{Q}(s, \boldsymbol{\lambda}_j) = \begin{bmatrix} \mathbf{Q}_{j,1}(s) \\ \mathbf{Q}_{j,2}(s) \end{bmatrix}. \quad (5.80)$$

Note that \mathbf{T}_{zr} is affine in \mathbf{Q} , so all objectives and constraints that are convex in \mathbf{T} are also convex in \mathbf{Q} .

For implementations $\mathbf{Q}(s)$ must be restricted to a finite, weighted sum of n basis functions. For an LTI controller, the following parameter-affine form is commonly chosen:

$$\mathbf{Q}(s) = \sum_{i=1}^n \mathbf{Q}_i(s) \theta_i \quad \boldsymbol{\theta} = [\theta_1, \dots, \theta_n]^T \quad (5.81)$$

This can be extended to a scheduled parameter-affine control law by choosing h interpolation functions $\Psi_l(\boldsymbol{\lambda})$ ($l = 1, \dots, h$) and use these as weighting functions. For linear interpolation and a scalar parameter $\lambda \in [0, 1]$, the form

$$\begin{aligned} \mathbf{Q}(s, \lambda) = & \underbrace{(1 - \lambda)}_{\Psi_1(\lambda)} \sum_{i=1}^n \mathbf{Q}_{(1),i}(s) \theta_{(1),i} \\ & + \underbrace{\lambda}_{\Psi_2(\lambda)} \sum_{i=1}^n \mathbf{Q}_{(2),i}(s) \theta_{(2),i} \end{aligned} \quad (5.82)$$

can be chosen. This form is convenient because the extremal cases only depend on $\mathbf{Q}_{(1),i}$ or $\mathbf{Q}_{(2),i}$, respectively, which can be exploited to reduce the size of the subsequent optimization problem.

However, other choices of interpolation are directly possible (for example, a polynomial in λ , λ^2 , etc.) and provide a means to introduce expert knowledge about the nature of the plant's parameter dependency.

For the considered set of design or validation plants, the scheduling law relates each plant with a fixed value $\lambda = \lambda_k$, so that the weighted multi-model design remains affine in the stacked parameter vector $\boldsymbol{\theta} = [\boldsymbol{\theta}_{(1)}^T, \boldsymbol{\theta}_{(2)}^T]^T$.

Note that the exposition is restricted to quasistatic parameters; if the rate of parameter variation is non-negligible with respect to the system dynamics, LPV approaches are appropriate (see [159]).

5.4.2 Robustifying design algorithm

The design of a feed-forward control law which is scheduled on the known parameter $\boldsymbol{\lambda}$ and performs robustly with respect to changes in the unknown parameter $\boldsymbol{\eta} \in \boldsymbol{\Delta}$ can be realized in the following fashion for a set of validation plants $\mathcal{P} = \{\mathbf{P}_j, j = 1, \dots, k\}$:

1. Choose an initial set of design plants $\tilde{\mathcal{P}} = \{\tilde{\mathbf{P}}(s, \boldsymbol{\lambda}_1, \boldsymbol{\eta}_1), \dots, \tilde{\mathbf{P}}(s, \boldsymbol{\lambda}_k, \boldsymbol{\eta}_k)\}$ of (possibly reduced) plant descriptions $\tilde{\mathbf{P}}$.
2. Formulate the objectives and constraints with respect to all plants in $\tilde{\mathcal{P}}$ via the LMI formulation given in Sec. 3.6.4.
3. Solve the optimization problem.
4. Validate the objectives and constraints on all plants in the validation set \mathcal{P} .
5. If the results are not yet satisfactory, add the worst-case plant $\tilde{\mathbf{P}}_j$ to $\tilde{\mathcal{P}}$ and goto step (2).

This way, the design is made robust against the unknown parameter values $\boldsymbol{\eta}_j$ arising in the validation plant set.

5.5 Case Study 2: Robust Scheduled Lateral Feed-Forward Controller for Flexible BWB Aircraft

This paper follows on a series of related papers in which various studies on the same BWB aircraft models have been done: in [133], a general integrated methodology for multi-objective robust control design has been presented. LQ-based lateral control designs for the BWB aircraft have been studied in [134], and a longitudinal BWB control law using an LPV approach has been designed in [157]. These works were followed by the application of a genetic algorithm for parameter optimization of a multiobjective \mathcal{H}_∞ DK-iteration design [135]. The direct predecessor of this work is a robust convex feedback control design of a lateral feedback control law [138], which provides the shaped closed-loop plants for the present work.

5.5.1 Feed-Forward Control Goals and Control Architecture

A control law to robustly optimize the dynamic and static loads in predefined test maneuvers should be designed. The relevant control specifications are given in Tab. 4.1.

5.5.2 Scheduling Concepts: Feed-Forward Multi-Model Control Design

From the two plant parameters fuel and payload, the fuel level affects the plant dynamics much stronger than payload (in their respective bounds) does. This motivates the investigation of the benefit of using fuel as scheduling parameter.

The following control configurations will thus be considered:

1. a (2×2) feed-forward controller \mathbf{Q}_u acting only on \mathbf{u} as in Fig. 5.12 (a), and
2. a (6×2) feed-forward controller $\mathbf{Q}_{u,y} = [\mathbf{Q}_u^T, \mathbf{Q}_y^T]^T$ which is also given access to the feedback controller inputs (measurements) \mathbf{y} as in Fig. 5.12 (b).

Additionally, for these two configurations, a fixed (LTI) controller will be compared to a linearly scheduled controller (as in (5.82)).

For the purpose of this study, a common basis of $\mathbf{Q}_{(1)}(s) = \mathbf{Q}_{(2)}(s)$ of 8 different first- and second-order transfer functions, spread over all controller inputs and outputs ((2×2) or (6×2) , respectively) is utilized. Its poles are chosen ad-hoc within the expected dynamic range (real and low-damped oscillatory). This yields the following numbers n of free parameters, depending on the controller configuration: 32 ((2×2) LTI), 64 ((2×2) scheduled), 96 ((6×2) LTI), and 192 ((6×2) scheduled).

Four extremal plants (lowest and highest fuel filling, with lowest and highest payload) and a central plant are chosen for design. Tab. 5.3 lists all objectives and constraints that are defined on these plants. The objectives are scaled to 1 for the unity feed-forward to the plant input signals \mathbf{u} (and zero to the feedback controller inputs). No further weighting has been performed and all objective variables α enter the cost function with unit weight (compare (2.70)):

$$f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} = [\mathbf{0}^T \mid \mathbf{1}^T] \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\alpha} \end{bmatrix}. \quad (5.83)$$

Objective O1 is a MIMO \mathcal{H}_2 minimization objective for the transfer from $\{\phi_{\text{ref}}, \beta_{\text{ref}}\}$ to $\{Mx_{\text{wing}}, Mx_{\text{fin}}, Ny_{\text{CG}}\}$ with the aim to achieve broadband attenuation. Furthermore, O2–O7 explicitly reduce the peak loads $Mx_{\text{wing}}, Mx_{\text{fin}}$ and lateral accelerations Ny_{CG} in the maneuvers as being l_∞ minimization objectives. The defined constraints ensure the required tracking performance (l_∞ -templates of roll and side slip responses in C1–C2), decoupling (l_∞ -bounds C3–C4) and the observation of the control surface deflection and rate limits (C5–C12).

Table 5.3: Objectives (O) and constraints (C) for feed-forward control optimization

O/C	Type & Path	Comment
O1	min. \mathcal{H}_2 norm ($\mathbf{r} \rightarrow \{Mx_{\text{wing}}, Mx_{\text{fin}}, Ny_{\text{CG}}\}$)	
O2–7	min. l_∞ norms ($\mathbf{r} \rightarrow \{Mx_{\text{wing}}, Mx_{\text{fin}}, Ny_{\text{CG}}\}$)	
C1	l_∞ bounds ($\phi_{\text{ref}} \rightarrow \phi$)	retain response
C2	l_∞ bounds ($\beta_{\text{ref}} \rightarrow \beta$)	retain response
C3	l_∞ bounds ($\phi_{\text{ref}} \rightarrow \beta$)	decoupling
C4	l_∞ bounds ($\beta_{\text{ref}} \rightarrow \text{roll}$)	decoupling
C5–12	l_∞ bounds ($\mathbf{r} \rightarrow \{\mathbf{u}, \dot{\mathbf{u}}\}$)	input mag./rate limits

5.5.3 Scheduling Results, Robustness & Discussion

Optimization Results

The problem is formulated as an LMI problem and solved using MATLAB[®]'s LMILAB solver and the object-oriented environment developed in Sec. 5.1. After typically 5 to 10 iterations of adaptive constraint refinement (as in Sec. 5.1.1) the objectives and constraints are fulfilled everywhere on the validation grids. The preprocessing and computation times are some minutes to hours (for fairly large problems) on a Core i7 workstation.

Using the same basis for \mathbf{Q} and the same objective and constraint definitions, a scheduled and LTI controllers are designed. However, it turns out that only the scheduled control laws can satisfy the most stringent set of constraints, while the problem is infeasible for the LTI design cases. Thus, to enable controller comparison, these templates are relaxed (to allow 5% DC gain variation in $\beta_{\text{ref}} \rightarrow \beta$ and more cross-coupling) which leads to a feasible design for both LTI and scheduled controllers.

Performance of LTI and Scheduled Controllers

Fig. 5.13 shows the reference step responses for the (2×2) LTI and scheduled controllers, for all design cases and the defined (relaxed) time-domain template bounds. As an example, the responses $\phi_{\text{ref}} \rightarrow Mx_{\text{fin}}$ (objective O5) and $\phi_{\text{ref}} \rightarrow Ny_{\text{CG}}$ (objective O7) are depicted in Fig. 5.14 and Fig. 5.15, respectively, for the design mass cases, compared to an unfiltered step response. Strong reductions are evident especially for the scheduled controller, an observation which holds for all considered mass cases. Moreover, the high-frequency content is not excited in the maneuvers. Finally, the scheduled controller can exploit the optimization potential given by the reserve in rise time (so it decelerates the response as long as the template allows it), while at the same time obeying the low overshoot bound. The LTI control law clearly cannot utilize this potential.

Tab. 5.4 shows the relative cumulative cost values of the uncontrolled plant set and the cases of fixed and scheduled controllers, both for the (2×2) and the (6×2) configurations. While the simplest (2×2) LTI configuration only achieves moderate loads alleviation and vibration reduction, the additional degrees of freedom by accessing the feedback compensator inputs (the (6×2) LTI case) yield considerable objective function improvement. Finally, the scheduled controllers clearly dominate in terms of performance – the use of scheduling is thus a strong benefit for overall control performance in the studied application. This also demonstrates that already a simple linear interpolation onset can introduce vital information into the controller.

Robustness, Discussion, and Concluding Remarks

An analysis of the step response shapes and assessment of the other plants in the validation set shows a generally benign behaviour of the feed-forward controller. However, to strictly fulfill the

Table 5.4: Cumulative cost values on the design plant set, relaxed constraints

Controller	Relative cost
none ($\mathbf{Q} = \mathbf{I}$)	1.00
$\mathbf{Q}_u(s)$ ((2×2) LTI)	0.52
$\mathbf{Q}_{u,y}(s)$ ((6×2) LTI)	0.38
$\mathbf{Q}_u(s, \lambda)$ ((2×2) sched.)	0.37
$\mathbf{Q}_{u,y}(s, \lambda)$ ((6×2) sched.)	0.34

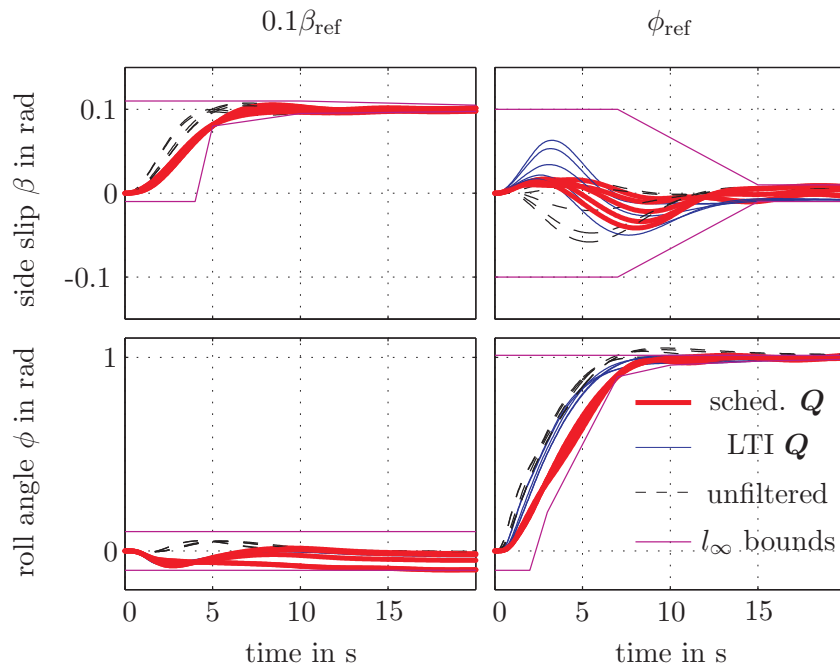


Figure 5.13: Step responses of design mass cases without (*dashed*), with fixed optimal (*thin, blue*), and scheduled optimal (*bold, red line*) feed-forward controllers

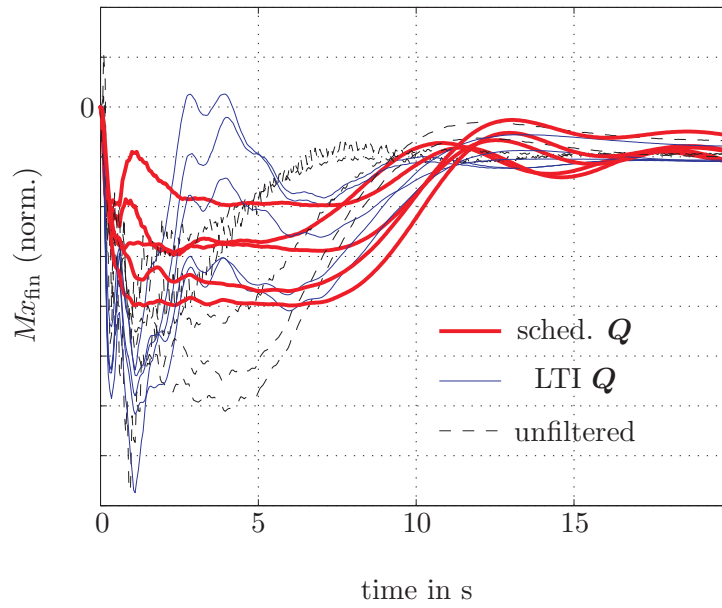


Figure 5.14: Shaped response $\phi_{\text{ref}} \rightarrow Mx_{\text{fin}}$ without (dashed), with fixed optimal (thin, blue), and scheduled optimal (bold, red line) feed-forward controllers

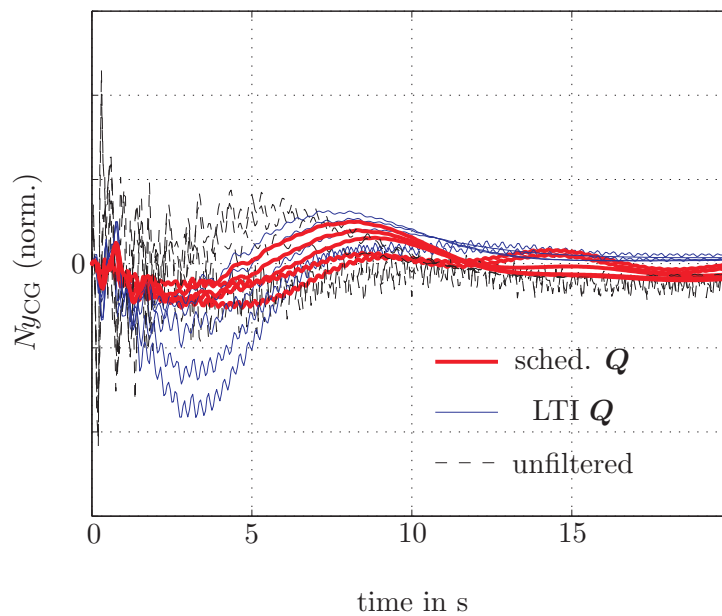


Figure 5.15: Shaped response $\phi_{\text{ref}} \rightarrow Ny_{\text{CG}}$ without (dashed), with fixed optimal (thin, blue), and scheduled optimal (bold, red line) feed-forward controllers

response templates for all mass cases, more plants have to be added to the design set (following the procedure proposed in Sec. 5.4.2). Although this further increases the size of the optimization problem, the solver runtime is expected to increase only moderately due to the simple nature of the constraints. Tests show that \mathcal{H}_∞ objectives (of which a convex formulation is given in [138]), on the other hand, have a high impact on runtime: no precalculations as in the case of \mathcal{H}_2 objectives are possible and the problem size quickly increases with refined gridding.

Minimal realizations of the final feed-forward controllers are of dynamic orders 22 to 26 which could further be reduced for implementation. One appropriate reduction method is stability preserving weighted balanced reduction (see [26]). Note that the choice of a common basis for each SISO channel in \mathbf{Q} reduces effectively the degrees of freedom. Reliable algorithms to obtain optimally chosen, well-conditioned, and low-order bases are thus highly relevant for convex control design.

Finally, the loads as well as the lateral acceleration peak values induced by the roll and side slip maneuvers are decreased significantly by installing the feed-forward controller. The linear scheduling shows further significant reduction, and tighter reference response templates can be obeyed.

A robust convex multi-model design of a scheduled feed-forward controller for a large, flexible BWB transport aircraft predesign model has been performed. A multitude of time- and frequency-domain constraints and objectives have been considered and optimal controllers of various architectures have been synthesized. Their evaluation unveils a significantly higher performance when the control law is scheduled with respect to fuel filling level, which strongly affects the system dynamics. As a result, strong reductions in maneuver-induced loads and lateral accelerations could be achieved robustly, while obeying reference response requirements.

Chapter 6

A Design Parameter Optimization Methodology for Control Design and Validation

6.1 Introduction

Today's robust control analysis and design tools enable the control engineers to analyze the effects of system perturbations or uncertainties and to design control laws which are robust with respect to these perturbations. Robust control design for complex flexible structures represents a difficult class of control problems. Challenging issues therein are strongly coupled or highly-damped dynamics and parameter-dependent or uncertain dynamics, often calling for robust scheduled controllers.

In this work, an integrated control design, validation, and optimization framework methodology is proposed that aims to improve closed-loop performance by appropriate adaption of frequency design weights for a pre-defined design architecture. It allows to incorporate expert knowledge in a clearly structured way at various levels of the design process while at the same time error-prone repetitive standard tasks are automated. This way the process of controller tuning, which represents the main effort in the synthesis procedure and which is often only insufficiently addressed, is essentially simplified and shortened for the engineer.

At a higher design abstraction level, approaches to control parameter optimization exist: a survey on evolutionary optimization of control design parameters is given in [34], a recent study using a combination of a genetic algorithm-based parameter search with fuzzy control performance evaluation is found in [119], and a formulation of the control design parameter optimization as a goal-attainment problem is done in [47].

The main contribution of this chapter is a methodology in form of a DPO framework which is suitable to consider the control design and validation problem *in its full detail*, encapsulated and viewed as a high-level optimization problem. The standard optimal and robust control design and analysis tools as in Chap. 2 and Chap. 3 are utilized in an automated way wherever beneficial. Essential control design tasks, such as system scaling, order reduction, uncertainty modeling, weighting, the μ calculus, design and analysis tools, and control architecture selection, are eased by this partially automated and modular support. In two case studies, this onset is applied to optimize control designs for the BWB aircraft in Sec. 6.4 as well as for a conventional flexible aircraft (based on the controller designed in [162]) in Sec. 6.5. The proposed methodology enables the engineer to efficiently test numerous design variants and it enables him or her to address specifications which otherwise cannot directly be accounted for in the utilized design methods, for example time-domain specifications in \mathcal{H}_∞ -based control design.

6.1.1 Motivation for an Integrated Methodology

Robust control of complex multivariate flexible structure systems is challenging. Main issues are robust controllability/observability and coupled system dynamics. Moreover, these systems often depend non-linearly on global system parameters. This can lead to mode-flipping in the linearized systems or to the loss of stability or non-minimum-phase behavior. These challenges and demanding control goals motivate a structured approach to translate the physical problem into the normalized $\mathbf{P} - \mathbf{K} - \mathbf{\Delta}$ - form, design the controller and validate it. Automating this process enables design tuning and optimization and leads to vital advantages:

- Efficient control design and tuning is possible by varying the design parameters.
- The modeling redundancy is reduced and design errors due to inconsistencies are avoided.
- The design can be clearly understood, maintained, and debugged.
- Well-posedness of the design problem is ensured by consistently using normalized systems.

One crucial requirement for such approach is to effectively incorporate the control engineer's expert knowledge — to formulate the objectives and to choose the control architecture and design weights.

6.2 Methodology

6.2.1 Basic Idea

The proposed control design, validation, and optimization methodology integrates model pre-processing, robust control designs, and automated validation and performance evaluation of the closed loop. A flow chart of this process is depicted in Fig. 6.1. This section details the essential design tasks a control engineer has to carry out for control design. A clean parameter-based implementation and automation of this design allows its reuse inside a design parameter optimization problem. The presented methodology and optimization scheme has been implemented in MATLAB[®] and successfully applied in practice.

6.2.2 Generating the Design Plant

Input/Output-Reduction & Additional Dynamics

To reduce computation demands and increase numeric stability, it is advisable to reduce the system I/Os to those necessary for the design or for the validation task. Also, I/O-weights and dynamics are incorporated into the model:

- Actuator dynamics, typically of low-pass behavior
- Excitation spectra to model stochastic disturbance (for example, wind turbulence in aircraft or track irregularities in rail vehicle systems)
- Sensor dynamics and delays (modeled for example via Padé approximations)

System Scaling

System scaling is critical in robust control design for various reasons:

- System reduction requires scaling to depict the reduction error correctly (compare [143]). As reduction indices often correlate with the I/O-magnitudes of the system, I/O-magnitude normalization is required to avoid reduction bias.

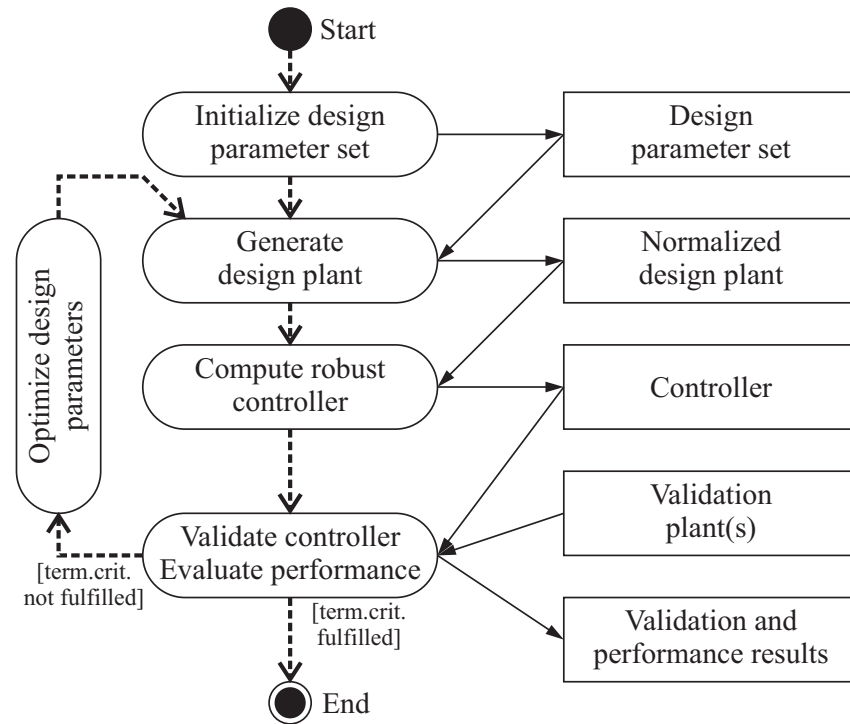


Figure 6.1: Automated control design, validation, and optimization: process and data flow

- Correct definition of the control goals with robust control analysis and design methods requires appropriate scaling and weighting (compare RS and RP in Sec. 2.3.1).

Static I/O-scaling (as used here) is defined using the diagonal scaling matrices \mathbf{S}_{in} and \mathbf{S}_{out} :

$$\mathbf{G}_{\text{scaled}}(s) = \mathbf{S}_{\text{out}} \mathbf{G}(s) \mathbf{S}_{\text{in}}. \quad (6.1)$$

The scaling factors can be obtained by

1. expert knowledge of signal amplitudes;
2. observing system norms of I/O-groups of the system; or
3. formally optimizing for quantities of numeric conditioning.

Balanced State Reduction

To obtain controllers of limited dynamic order, it is typically necessary to reduce the design plant order before control design. One widely applied method is the balanced state reduction method which is detailed in Chap. 2.2.

Uncertainty Modeling

Relevant uncertainties for robust control design are incorporated into the design plant by uncertainty modeling techniques, see Sec. 2.3.2.

6.2.3 Generalized Design Plant Interconnection

Having prepared the design plant \mathbf{G} , the control engineer needs to implement the control architecture by interconnecting the design plant, the weighting functions, and the uncertainty model

to the normalized and generalized plant \mathbf{P} such that the minimization of $\|\mathbf{N}\|_\infty$ (where \mathbf{N} is the perturbed closed-loop transfer function, see Fig. 2.1 on page 16) results in the fulfillment of the desired robust performance specifications. The parameters of the chosen weighting functions represent tuning knobs and are thus suitable as optimization variables.

6.2.4 Controller Synthesis

The normalized and generalized design plant and the uncertainty description are used to design a robust controller. Any design method using the standard $\mathbf{P} - \mathbf{K} - \mathbf{\Delta}$ form can be utilized in the present methodology. In particular, state-of-the-art robust controller synthesis tools, such as the DK-iteration algorithm (see Sec. 3.5), Convex Synthesis techniques (see Sec. 3.6), or robust fixed-order design tools (such as HIFOO [50]) can be chosen.

These methods can yield high-order controllers that have to be order-reduced before practical implementation, which is not a trivial task since robustness has to be maintained (see Chap. 7). Fixed-order control design methods circumvent this disadvantage by directly synthesizing a low-order controller, but computational effort can be too high.

6.2.5 Controller Validation and Performance Evaluation

The controller's closed-loop performance is validated with accurate plant models in the frequency-domain and often in detailed, non-linear time-domain simulations. To quantify the obtained control performance suitable for the optimization, the goal attainment procedure as in Sec. 2.4.7 can be utilized.

To enable efficient validation, it is necessary to structure the task into the following steps:

- define the architecture for validation (compatible to the controller),
- define the validation model set (for example an LTI model set associated to multiple physical plant parameter values, see below),
- construct the closed loop for a given plant and perform its evaluation, and
- evaluate control performance for each closed-loop system (that is, for each validation case).

Linear System Validation

The validation plants typically have to be prepared in a similar way as the design plant. After verifying the internal stability of the considered closed-loop systems, their performance is evaluated and compared to the open-loop systems' performance. Common closed-loop quantities of interest are:

- Achieved reduction in selected SISO transfer peak magnitudes at frequency ranges of relevant modes
- DC gain difference in selected SISO transfer functions

Moreover, an empirical marginal stability analysis can be performed. The MIMO gain margin is sought by closing the feedback path with $c\mathbf{K}$, where $c \in \mathbb{R}^+$ and finding c using a binary search algorithm such that the system is brought to the stability limit.

Non-Linear System Validation

If non-linear validation is required, time-domain simulations with a series of validation plants P_i are performed (see Sec. 2.3.3). This way, control input non-linearities such as saturation, rate limits, or hysteresis behavior and their effects on the closed-loop system can be examined. In contrast to linear validation usually no algebraic manipulations to obtain the closed-loop behavior exist.

The system behavior is typically examined in the time domain, simply because no direct frequency response representation (as in the linear case via the transfer function formulation) exists. Nevertheless, spectral analysis of concrete signals is possible via (Fast) Fourier transformation and power spectral density analysis tools.

6.3 Formulating and Solving the Optimization Problem

The presented design methodology involves a number of design parameters that define the weighting functions, uncertainty magnitudes, and performance formulations. For design optimization, these can be seen as free variables \mathbf{p} . Collecting the entire design and validation methodology above into one single functional block $f(\cdot)$ with the design parameters \mathbf{p} as inputs and the achieved validation performance $f(\mathbf{p})$ as fitness output to maximize, the generally non-convex and even non-smooth design optimization problem can be stated formally as:

$$\min_{\mathbf{p} \in \mathcal{P}} f(\mathbf{p}), \quad (6.2)$$

with the design parameter vector $\mathbf{p} = [p_1, \dots, p_m]^T \in \mathcal{P} \subseteq \mathbb{R}^m$ and a suitable cost function $f: \mathcal{P} \rightarrow \mathbb{R}$ which

- designs a controller \mathbf{K} from the given parameter values \mathbf{p} ,
- constructs the closed loop systems with a series of validation systems $S_i \in \mathcal{S}_{\text{val}}$,
- evaluates predefined criteria on these closed loop systems, and
- computes an aggregated cost value.

As this poses a difficult optimization problem, an optimal solution can only be approximated, for example using meta-heuristic optimization methods. However, local optimization in a limited parameter space can also already lead to practically well-performing results.

Focusing on one design parameter at a time, a gridding of its values over a chosen interval of interest is defined. The design methodology is carried out for each grid point and the achieved performance is evaluated in an automated fashion. Despite its simplicity, this approach produces quick remedies to performance problems and also conveys properties of the investigated parameter space to the engineer's understanding, such as smoothness or parameter sensitivity.

6.3.1 Using Genetic Algorithms for Design Parameter Optimization

The difficulty of the design process typically stems from the complexity of the considered system dynamics and the multitude of conflicting control goals. Formally defined design procedures enable a structured approach and make the task accessible to automation and optimization [133].

A survey on the use of evolutionary algorithms in the context of control design and optimization is given in [35], and the use of genetic algorithms in control is discussed in [34]. Specific use of genetic algorithms for multi-objective control problems has been studied in [36], a minimax approach to robustness in a multi-model, multi-objective control design setting has been taken in [113].

6.3.2 Implementation

The low-level control design methods, elementary system analysis tools, and the genetic algorithm are directly available in MATLAB[®]. The parametrization of the optimization problem, the control design problem setup, preprocessing, postprocessing, and the goal-attainment evaluation methodology have been implemented via object-oriented techniques in MATLAB[®]. This facilitates the design process and increases its flexibility and reliability because once the functionality is implemented it can directly be reused, see also the related discussion in Sec. 5.1.2. The class structure of the implementation can be seen in Fig. 6.2. The package `DesignPar` contains classes to realize a flexible parametrization of the design, while the package `GoalAttain` implements the Goal Attainment logics.

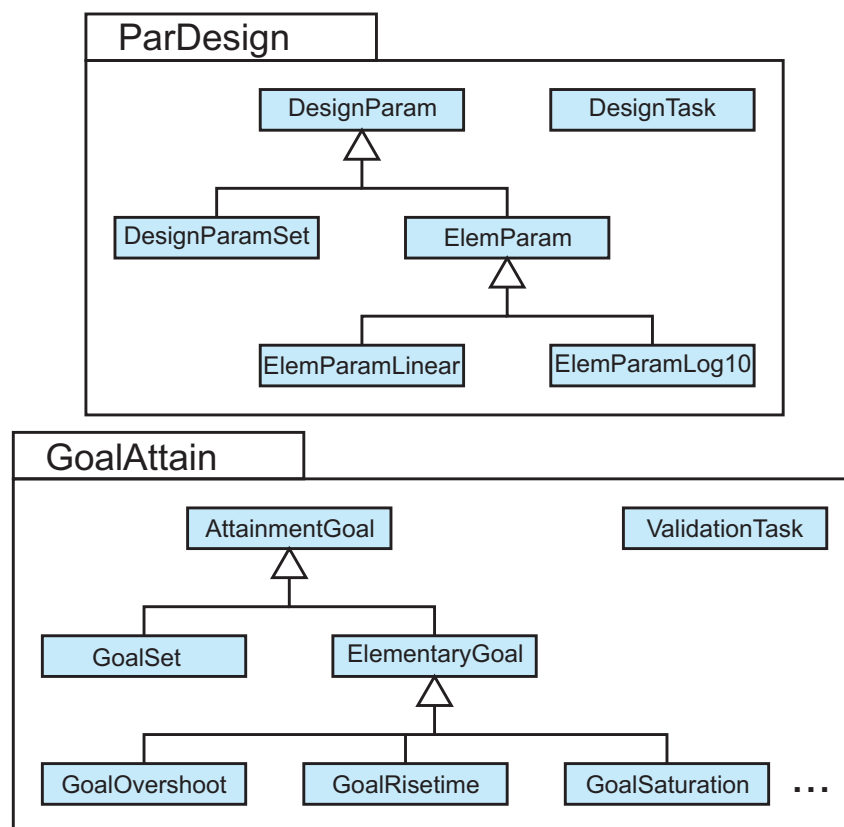


Figure 6.2: Class structure of the implementation

6.4 Case Study 3: DPO of Lateral Control of Flexible BWB Aircraft

The lateral control design case for the flexible BWB aircraft introduced in Sec. 4.2 is revisited. In this case study, a structured design onset for an integrated design of a two-degree-of-freedom controller is chosen with the aim to address all given control goals in one design.

Initial studies showed that augmenting the plant by integrators or quasi-integrators for asymptotic tracking renders it ill-conditioned for standard \mathcal{H}_∞ -based control design, despite taking common countermeasures (scaling, balancing, and relocating the integrator in the loop, compare [94] and [157]). For this reason, the chosen architecture consists of an inner-loop controller obtained by DK-iteration (see Sec. 3.5) which is a posteriori augmented by two PI SISO

feedback loops to obtain asymptotic tracking of the roll angle ϕ and the side slip angle β .

Supported by the established DPO framework, an initial design for the lateral motion of the aircraft is carried out and design parameters and their value ranges for optimization are defined. The specifications, given in both the time and in the frequency domain, are evaluated by the goal attainment approach shown in Sec. 2.4.7. The combined design and evaluation process is implemented as an objective function in the free design parameters and is thus accessible to formal parameter optimization which is addressed in this work by a genetic algorithm.

Related onsets have been investigated, for example in [23] where a hybrid optimization via ant systems and LQR design is carried out to obtain robust eigenstructure assignment for a conventional aircraft. The present work, however, utilizes frequency-domain μ synthesis and optimizes for loads and time-domain response characteristics for a BWB aircraft configuration.

6.4.1 Lateral Feedback Control Architecture and Initial Design

To address the control goals given in Sec. 4.2.2 are realized by a two-stage control concept consisting of a simple outer loop (autopilot) to ensure asymptotic tracking and an inner robust control law for basic response shaping, DR mode damping and loads alleviation. In this study the inner-loop controller is generated by DK-iteration design and the autopilot loops are realized afterwards by two SISO PI controllers for the roll angle ϕ and the sideslip angle β . The augmented design plant interconnection for the inner-loop control design is shown in Fig. 6.3. Thereby, \mathbf{G} denotes the order-reduced lateral aircraft model, \mathbf{G}_{ref} defines reference dynamics for desired roll and sideslip responses, \mathbf{W}_u weighs the control input magnitude over frequency, \mathbf{W}_{ref} shapes the admissible control errors, \mathbf{W}_p enables further shaping of the closed-loop dynamics by weighting exogenous outputs such as wing and fin root loads, and \mathbf{W}_a shapes an additive uncertainty Δ_a to account for differences between design and validation plant dynamics.

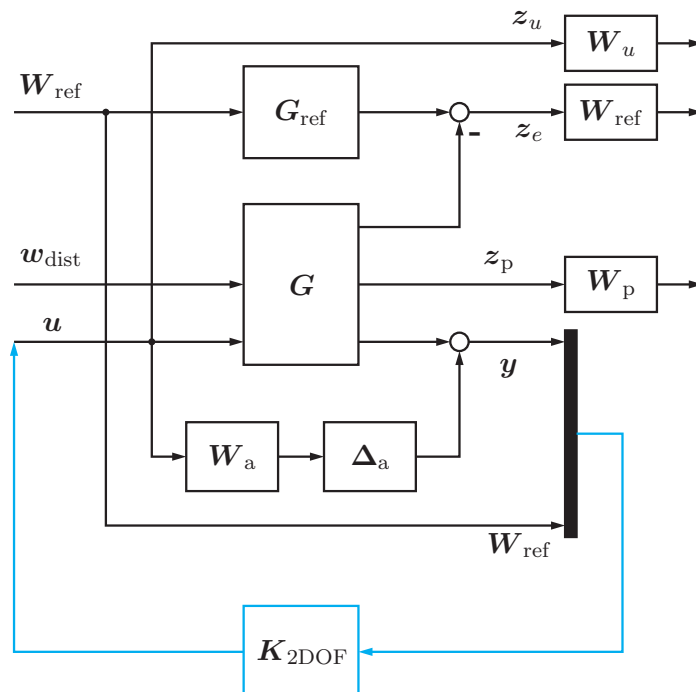


Figure 6.3: Two-degree-of-freedom (2DOF) control architecture

Weighting Basis Functions

For the underlying DK-iteration design method only stable, minimum-phase, and proper weighting functions are considered. Thus, to realize quasi-high- or quasi-low-pass filters, a parametrized PD/PT-structure is used:

$$W_{\text{PTPD}}(s, k_0, \omega_z, \omega_p) = k_0 \frac{\frac{1}{\omega_z} s + 1}{\frac{1}{\omega_p} s + 1}. \quad (6.3)$$

A quasi-low-pass is obtained with $\omega_z > \omega_p$, and ω_z can be chosen sufficiently fast (outside the control bandwidth) and fixed, thus reducing the function description to two free parameters.

In turn, a quasi-high-pass characteristics is obtained with $\omega_z < \omega_p$, for which ω_p can be fixed at a sufficiently low value. Here, the magnitude at infinite frequency k_∞ is of higher interest than the DC gain k_0 . These gains are related by $k_\infty = \frac{\omega_p}{\omega_z} k_0$.

Chosen Design Parameters

For an initial design, these weightings were selected by a step-by-step assembly of the design plant and recurring designs to test each added component. The following design weightings have been utilized:

- Performance weights for roll attitude and side slip angle error $W_{e,p\phi}$, $W_{e,p\beta}$ (quasi-low-pass, $\omega_z = 100 \text{ rad/s} = \text{const.}$)
- Control input magnitude weights for rudder and ailerons (quasi-high-pass, $\omega_p = 100 \text{ rad/s} = \text{const.}$)
- Performance weightings for wing root bending moment and fin root bending moment $W_{z,Mx,\text{wing}}$, $W_{z,Mx,\text{fin}}$, chosen of low-pass shape after validation of a preliminary design without these outputs.

$$W_{p,\phi}(s, k_{e,p\phi}, \omega_{e,p\phi}) = k_{e,p\phi} \frac{\frac{1}{100} s + 1}{\frac{1}{\omega_{e,p\phi}} s + 1} \quad (6.4)$$

$$W_{p,\beta}(s, k_{e,p\beta}, \omega_{e,p\beta}) = k_{e,p\beta} \frac{\frac{1}{100} s + 1}{\frac{1}{\omega_{e,p\beta}} s + 1} \quad (6.5)$$

6.4.2 DPO Formulation

Attainment Goal Set

For this study the control goals listed in Tab. 6.1 have been formulated. The target values were chosen from the given requirements and from an analysis of manual, initial designs. The goal set is evaluated at each of six given validation mass cases (fuel filling level from empty to full) of the BWB aircraft's lateral dynamics.

Control Design

Firstly, the control architecture, the control design process, and the free parameters as well as their ranges and their encoding in terms of the design parameters have to be defined. After analyzing the open-loop plant dynamics and relating it to the requirements, a manual initial design is carried out to collect information on the impact of the chosen design parameters on the closed-loop system dynamics.

Table 6.1: BWB lateral control goals

No.	Goal category	Quantity
1	DR damping	Bode magnitude $v_{\text{lat.gust}} \rightarrow \beta$
2	Loads	Bode mag. $\phi_{\text{ref}} \rightarrow M_{x,\text{wing}}$
3	Loads	Bode mag. $\phi_{\text{ref}} \rightarrow M_{x,\text{fin}}$
4	Roll	Overshoot $\phi_{\text{ref}} \rightarrow \phi$
5	Roll	Undershoot $\phi_{\text{ref}} \rightarrow \phi$
6	Roll	Rise time $\phi_{\text{ref}} \rightarrow \phi$
7	Roll	Time-weighted control error $\phi_{\text{ref}} \rightarrow \phi$
8	Roll	Saturation limit $\phi_{\text{ref}} \rightarrow \delta_{\text{RU,cmd}}$
9	Roll	Saturation limit $\phi_{\text{ref}} \rightarrow \delta_{\text{AIL,cmd}}$
10	Side slip	Overshoot $\beta_{\text{ref}} \rightarrow \beta$
11	Side slip	Rise time $\beta_{\text{ref}} \rightarrow \beta$
12	Side slip	Time-weighted control error $\beta_{\text{ref}} \rightarrow \beta$
13	Gust	Time-weighted control error $v_{\text{lat.gust}} \rightarrow \beta$
14	Gust	Max. peak $v_{\text{lat.gust}} \rightarrow \beta$
15	Gust	Time-weighted control error $v_{\text{lat.gust}} \rightarrow \phi$
16	Gust	Max. peak $v_{\text{lat.gust}} \rightarrow \phi$

The augmented design plant interconnection for DK-iteration is built according to the given parameters:

$$\mathbf{P} = \mathbf{P}(\mathbf{p}_{\text{DK}}), \quad (6.6)$$

where \mathbf{p}_{DK} contains the necessary design parameters to construct \mathbf{P} . These are gains and pole-/zero-locations of frequency-domain weighting functions (see Sec. 3.1.1) and physical parameters to select the utilized design model.

The controller $\mathbf{K}_{\text{DK}} = \mathbf{K}_{\text{DK}}(\mathbf{P})$ is computed via DK-iteration. The two SISO PI controllers are parametrized by

$$\mathbf{p}_{\text{PI}} = [k_{\text{PI},1} \quad z_{\text{PI},1} \quad k_{\text{PI},2} \quad z_{\text{PI},2}]^{\text{T}} \quad (6.7)$$

where $k_{\text{PI},i}$ and $z_{\text{PI},i}$ are the PI controllers' gains and zero locations, respectively:

$$K_{\text{PI},i}(s) = k_{\text{PI},i} \frac{s + z_{\text{PI},i}}{s}, \quad i = 1, 2 \quad (6.8)$$

The entire control law design is thus defined by the union of all subtask parameter sets $\mathbf{p} = [\mathbf{p}_{\text{DK}}^{\text{T}}, \mathbf{p}_{\text{PI}}^{\text{T}}]^{\text{T}}$.

Note that it is possible to mix design parameters of different types and subtasks arbitrarily. The initial values and admissible ranges of the design parameters are chosen based on the initial control design.

The validation and performance evaluation of the obtained controller is carried out as described in Sec. 6.2.5.

6.4.3 DPO Results

Initial Control Law Characteristics

The initial controller has been found by a step-by-step enlargement of the design architecture until all goals could be influenced by the introduced design parameters. The DK-iteration

controller is manually tuned to some extent to yield reasonable results, and the SISO PI reference tracking loops have been optimized via MATLAB[®]'s SISOTOOL and its response optimization functionality. A minimal realization of the initial controller is of order 43, which has to be reduced for implementation. Controller order reduction is treated in Chap. 7.

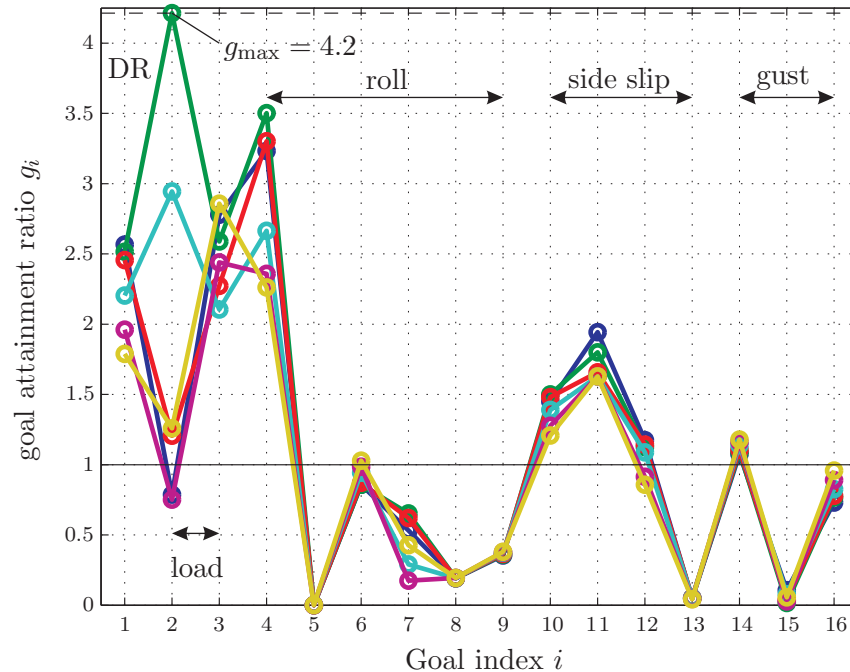


Figure 6.4: Goal attainment achieved by the initial controller

Fig. 6.4 shows the goal attainment ratios g_j for each goal in Tab. 6.1 as obtained by the initial controller. Note that the given multitude of goals define a complex and challenging design task for the given set of validation cases. It can be seen that the goals related to DR mode damping, maneuver loads, and the side slip reference response are not fulfilled. Moreover, the overshoot of the roll response is far too high. However, this design fulfills the remaining specifications on the roll response and on disturbance attenuation.

Optimized Control Law Performance

The genetic algorithm is seeded with the solution corresponding to the initial controller. The population size was set to 10 of which the 2 best solutions were taken into the next generation without crossover. Arithmetic crossover was utilized.

Fig. 6.5 shows the results obtained with the given control architecture in terms of the goal attainment ratios after optimization by the genetic algorithm (compare Tab. 6.1). It can be seen that the least fulfilled goal in the initial design has been considerably improved, but not all goals are yet fulfilled. The optimization lasted 50 generations at a total runtime of 16 h on an iPentium 4 HT desktop PC. Note that some goals have deteriorated as they do not influence the global cost function value.

Roll and Side-Slip Reference Step Responses

The closed-loop roll reference step responses of all validation mass cases is shown in Fig. 6.6 for both controllers. It is evident that the overshoot is strongly reduced by the design parameter

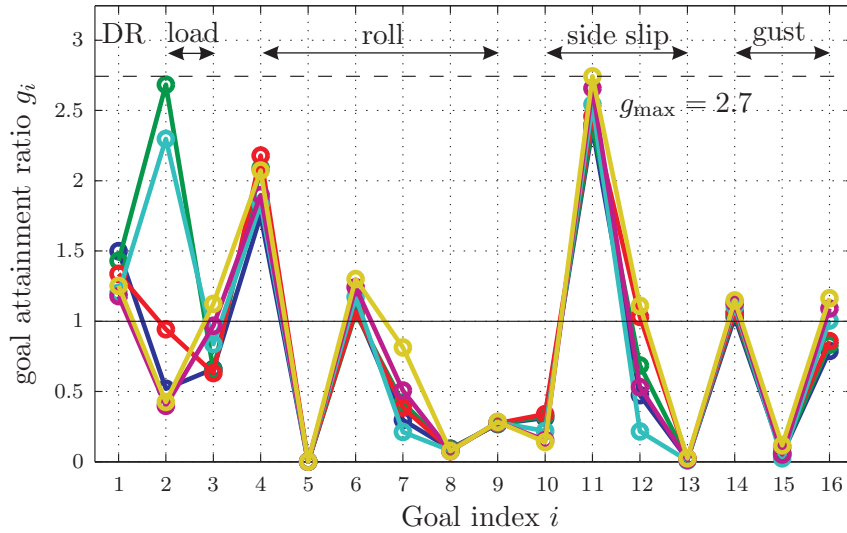


Figure 6.5: Goal attainment achieved by the optimized controller

optimization, at the cost of the rise time requirement. However, the rise time attainment ratio is still only 1.3.

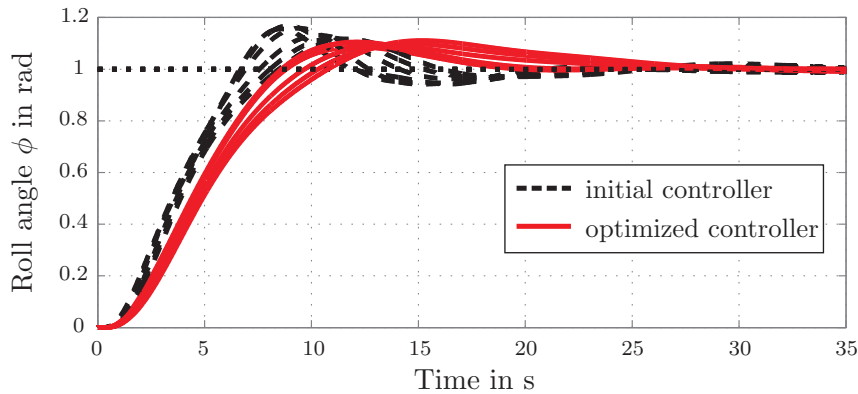


Figure 6.6: Roll reference step response for all mass cases

Fig. 6.7 shows the corresponding side slip reference step responses, and here strong improvements in tracking performance can be seen. Although the rise time requirement is still not fulfilled (and represents the least-fulfilled goal in the entire design with an attainment ratio of 2.7), the overshoot and the damping is considerably improved compared to the initial control law.

Wing and Fin Root Loads

Fig. 6.8 depicts the improvement in the wing root loads output from the initial to the optimized control law. Likewise, Fig. 6.9 shows the improvement in the fin root loads. Since the wing and fin root loads are heavily dependent on the roll reference tracking dynamics, the loads alleviation has to be attributed partly to the slower roll response. This indicates an inherent trade-off between tracking performance and induced loads.

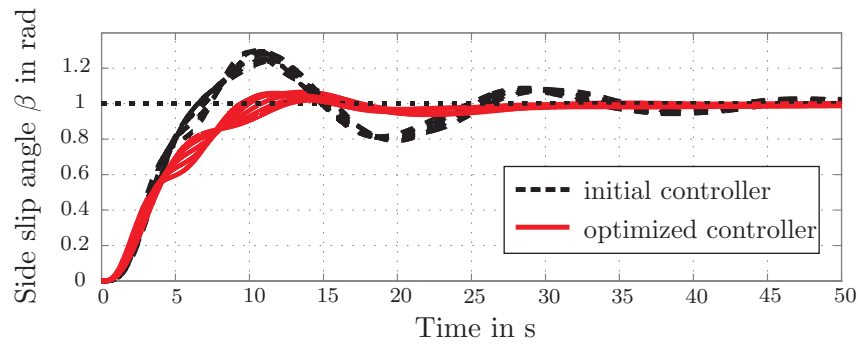


Figure 6.7: Side-slip reference step response for all mass cases

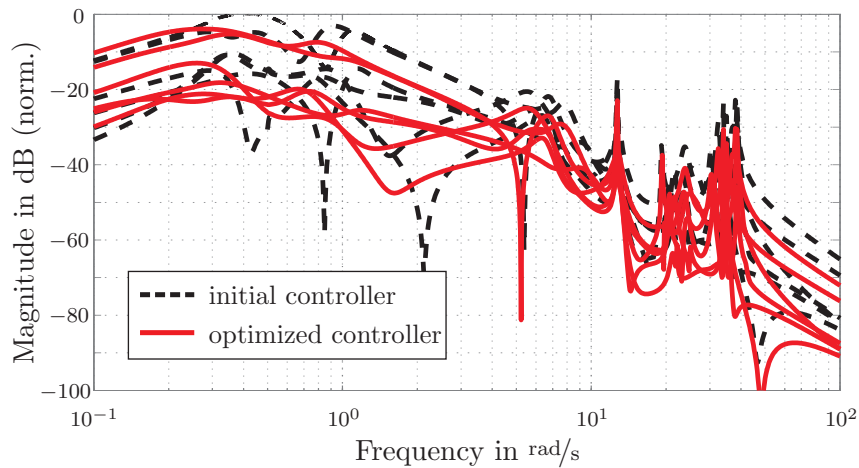


Figure 6.8: Bode magnitude plot for closed-loop roll reference to wing root load output

Concluding Remarks

The results of this case study clearly show the trade-off between the conflicting goals, especially rise time versus loads alleviation. Moreover, the design is in this formulation bound to a specific predefined control design architecture which on the one hand enables highly structured control design, but on the other hand can limit control performance.

The optimized design could not yet fulfill all goals to satisfaction, that is, the maximum goal attainment ratio $\max_{j=1,\dots,m} g_j > 1$ holds. Further optimization may yield a design that fulfills all goals. One approach to aid the optimization can be an iterative design in which the goals are relaxed so that an attainment ratio less than 1 is attained for all goals, after which the goals can be iteratively tightened.

6.5 Case Study 4: DPO of Longitudinal Control of Flexible Conventional Aircraft

In [162] the controller was obtained by DK-iteration (see Sec. 3.5), which is also applied in the following. The physical plant was reduced to 39 states, the augmented plant as well as the resulting controller are of order 83 with a (16×16) Δ -block (2 triple-repeated and 2 single real parameters, full-complex blocks of (3×3) and (5×5)). The validation results (see Fig. 6.11, taken from [162]) are based on simulated time-responses of the controlled aircraft (at 4 mass cases) to

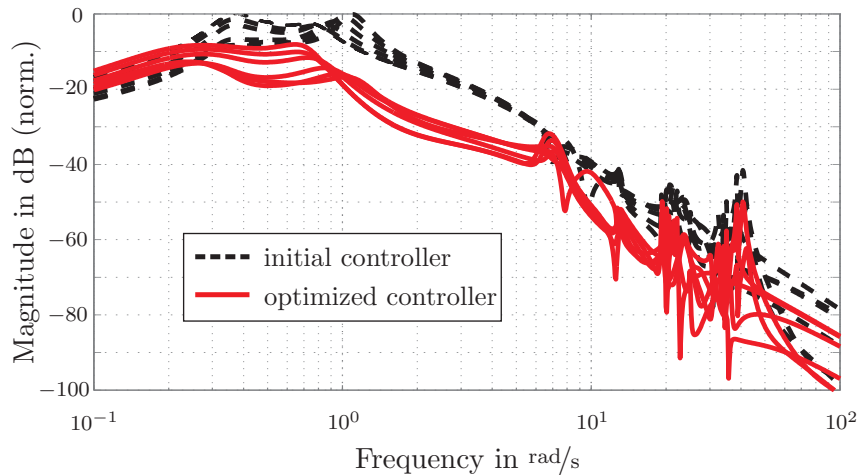


Figure 6.9: Bode magnitude plot for closed-loop roll reference to fin root load output

1. a vertical $1 - \cos$ gust with 100 m integral scale length and maximum vertical gust velocity of 25 m/s,
2. a pitch rate reference step, and
3. vertical turbulent wind following a von-Kármán turbulence spectrum[67].

The performance indices are computed as ratio of the closed-loop and the open-loop Mx_{HTP} (horizontal tail plane bending moment), Mx_{WR} (wing root bending moment), as well as Nz_{CG} and Nz_f (vertical accelerations at positions near the center of gravity as well as in the front near the cockpit), both in terms of RMS and time-domain peak values (\mathbf{L}_∞ -norms). These indices (see Fig. 6.11) show that all but one of the control goals have been met: only the vertical accelerations in the cabin front Nz_f were excited in two mass cases.

This controller has been optimized using the presented methodology to show the benefits of an integrated investigation and optimization. Analyzing the reason for the unmet performance index revealed that modes between 22 and 25 rad/s are excited by the controller. If the uncertainties of these modes are sufficiently well modeled, it may be possible to enforce damping by weighting these modes stronger for the controller design. To obtain this, a performance variable in which this mode is well observable has to be chosen. Introducing a performance weight that emphasizes those modes' frequency range leads to success. Note that if the mode variation is not described suitably by the plant uncertainty description, this solution has the opposite effect - it will improve the nominal case, but inevitably excite or destabilize other, critical validation cases. In such case, it is better to avoid control action by assigning a high uncertainty weight at that mode's frequency. The methodology was used within a one-dimensional parameter search, where the free parameter was the weight peak frequency. Figure 6.10 shows the change of the bode magnitude peak height of the critical structural mode in this frequency range of the Nz_f signal (which has been determined by the design parameter optimization procedure outlined above). It can be seen that a minimum is obtained near $\omega_{\text{peak}} = 24 \text{ rad/s}$. Moreover, it is evident that the obtained performance is smooth (continuous) around this minimum, which is an important feature of the found solution. The optimized controller also provides equivalent or better performance in all other performance quantities as seen in the performance chart in Fig. 6.12.

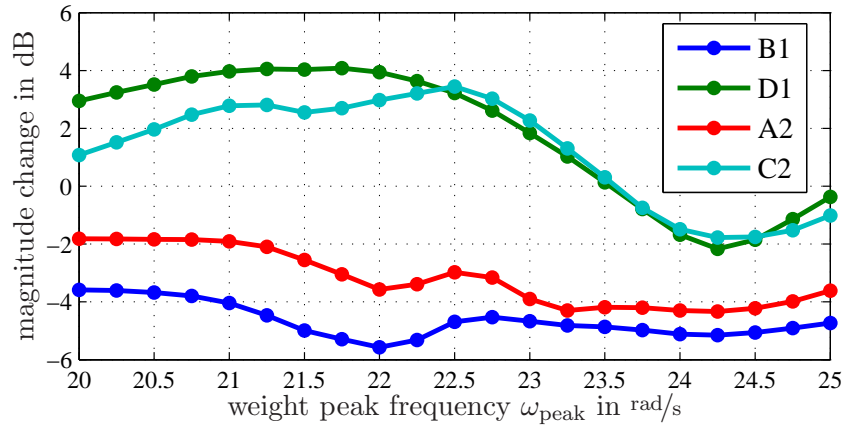


Figure 6.10: Obtained vibration reduction of the targeted mode when weight parameter ω_{peak} is varied

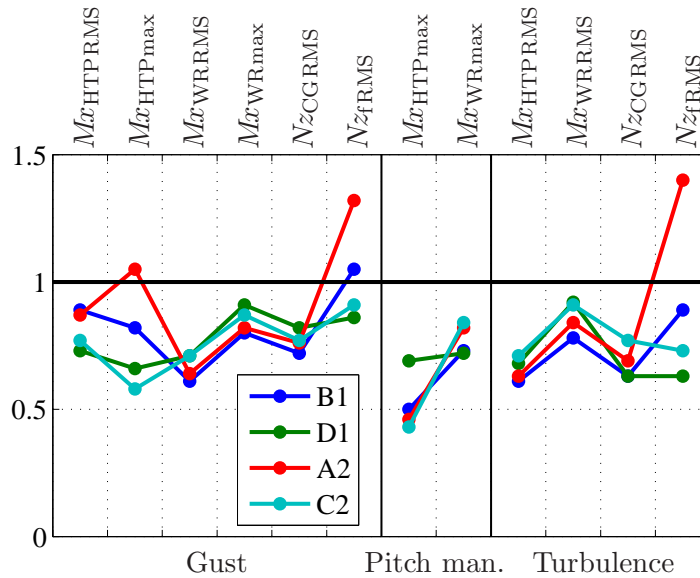


Figure 6.11: Control performance (controller taken from [162])

6.6 Discussion & Conclusions

A comprehensive methodology for robust control design, validation, and goal attainment evaluation for complex flexible-structure control has been presented. It allows quick optimization of design parameters to achieve the given control objectives. A genetic algorithm is employed to optimize the design (encoded by freely chosen control design parameters), aiming to fulfill all defined goals. These goals are arbitrary in nature, for example frequency- or time-domain quantities of the closed loop. It is shown that a consistent, comprehensive implementation of such framework methodology creates strong design advantages, while relieving the control engineer from time-consuming standard design tasks. The uncertainty description can be tuned to fit a set of validation plants to obtain robust high-performance controllers. Already simple optimization procedures lead to remarkable results.

The methodology has been implemented in MATLAB[®] in an object-oriented fashion and has been demonstrated at a difficult lateral control design task of a loads alleviation and tracking

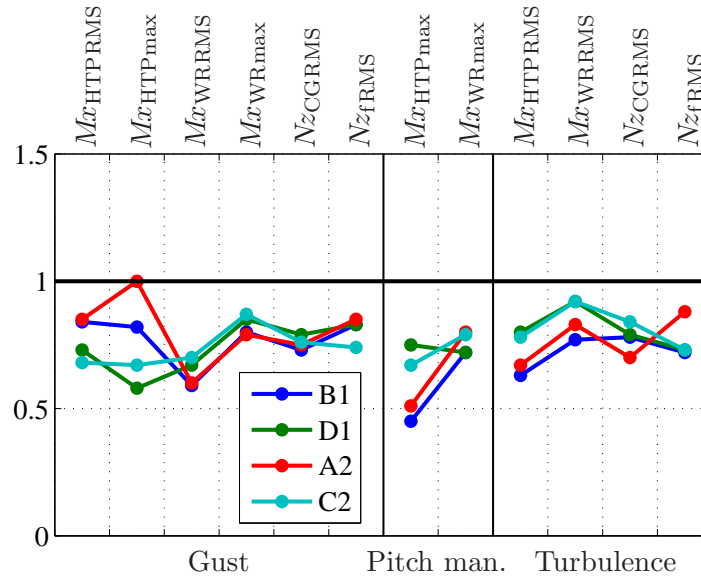


Figure 6.12: Control performance of optimized controller

control system for a large BWB passenger aircraft configuration. First results show an improvement in the posed goals, although not all goals are yet fulfilled. Moreover, the conflicting nature of various goals became evident, namely rise time and structural loads.

Benefits of the proposed approach include the efficient testing of design ideas, full validation and arbitrary goals, as well as exploiting powerful optimization tools in a systematic way for a highly complex process.

Possible extensions to this methodology include robust controller order reduction (while preserving robust performance, see [19]) and the use of comprehensive design parameter optimization using metaheuristics (see [119] and the MATLAB[®] Evolutionary Design Toolbox by these authors). An interesting future functionality is to freely construct weighting functions and interconnection structures through optimization.

Chapter 7

Extensions to Robust Controller Order Reduction

The problem of order reduction of linear dynamic controllers is addressed in this chapter. For two cases — related to nominal performance and to robust performance, respectively — suitable frequency-domain weightings are derived so that FWBR (see Sec.2.2) can be applied to the controller in a feedback loop. Two corresponding examples in which the obtained reduction procedure is compared against classic unweighted balanced reduction show the performance of the proposed formulations.

7.1 Introduction

The fundamental challenge in system modeling is to provide a model that describes the relevant behavior of a considered system sufficiently well while at the same time reducing model complexity to a minimum. This requirement is also present in control design: on one hand, the analysis of systems and the design of controllers is easier with a simplified problem setup. On the other hand, simple low-order control laws are much easier and cheaper to implement in real-time systems and in safety-critical environments. This generates a need for suitable order reduction methods, because many modern control design algorithms, such as those discussed in Chap. 3, produce complex control laws whose dynamical order depends on the dynamical order of the plant system model and is often undesirably high.

Thus, methods and tools to reduce the dynamical order of systems to the relevant behavior are vital and highly relevant to obtain design plants in control design. Likewise, suitable methods are needed to reduce the complexity of dynamic control laws without or with only minimal adverse effects on closed-loop stability and performance.

One of the most commonly applied reduction methods for LTI dynamic systems is the so-called balanced reduction in various forms (see Sec. 2.2). The standard setting of the unweighted balanced reduction, has been extended in [26] to the one- and two-sided weighted case (FWBR) which broadens the method's applicability. FWBR can be utilized to perform controller order reduction and allows to preserve closed-loop stability or performance quantities. An overview on the treatment of controller order reduction in literature is given in the following.

7.1.1 Literature Review of Controller Order Reduction Methods

In [49], system order reduction methods are surveyed and also the controller order reduction case is discussed. One of the investigated groups of methods is balanced reduction (called Lyapunov balancing there due to the arising Lyapunov equations in the balancing step).

In [4], the authors give formulations of one- and two-sided FWBR weightings for stability and performance preservation in standard feedback architectures. However, an exogenous performance path is not considered explicitly as in the $\mathbf{P} - \mathbf{K} - \text{Form}$. Comparative results at various problems and with various reduction methods are given.

In [43], Goddard & Doyle propose a formulation of frequency weightings to preserve \mathcal{H}_∞ nominal and robust performance based on the parametrization of all internally stabilizing \mathcal{H}_∞ -suboptimal controllers (see Sec. 3.2) and consider the reduction of the central controller. The proposed solution requires finding a rational approximation of complex matrix factorizations which may lead to a large order of the resulting frequency weights. Moreover it is noted that this formulation essentially bounds the admissible controller perturbations in a weighted \mathcal{L}_∞ -norm sense which may result in a conservative result. However, this limitation is present also in other methods which rely on the FWBR approach. Also in [100] and [101], the authors derive nominal performance weightings for reduction of the central controller using the parametrization of all internally stabilizing \mathcal{H}_∞ -suboptimal controllers, however without the need of building rational approximations.

A direct closed-loop balanced reduction approach for both, plant and controller, is given in [167]. The approach also allows to reduce unstable plants or controllers as long as the closed loop is stable. However, no stability guarantees are given in this work. A similar closed-loop controller reduction approach is taken in [175] which gives guarantees on closed-loop stability and bounds performance degradation.

The controller reduction problem in the robust control domain using the structured singular value μ (see Sec. 2.3.1) has been tackled recently in [19], where an admissible error bound for the unweighted \mathcal{H}_∞ controller reduction error has been derived to guarantee Robust Performance (RP), provided that RP is achieved by the original full-order controller. However, only unweighted balanced reduction has been utilized, thus the controller reduction itself does not exploit the extracted frequency-domain information.

This chapter provides two main contributions:

- Firstly, two-sided weightings are derived to use FWBR techniques for controller reduction for nominal performance preservation in the $\mathbf{P} - \mathbf{K} - \text{form}$. A simple variational onset is used to obtain the weightings and it is not necessary to obtain an \mathcal{H}_∞ -suboptimal parametrization of controllers for which the given controller is central (as in [43, 100, 101]).
- Secondly, extending the findings in [19], an algorithm is proposed to exploit the extracted robustness information by fitting suitable weightings for FWBR to perform controller reduction for RP preservation. Although the a priori RP preservation guarantee of the unweighted reduction in [19] is lost, a significantly improved reduction performance is seen in numeric validation.

The performance of the developed methods is demonstrated at two examples — an aircraft controller reduction case study and a 2×2 distillation column process from literature [143].

7.2 Nominal Performance Frequency Weighting Derivation in $\mathbf{P} - \mathbf{K} - \text{Form}$

Consider a plant \mathbf{P} which is internally stabilized by a controller \mathbf{K} and interconnected in the $\mathbf{P} - \mathbf{K} - \text{form}$ as in Sec. 2.3.1. The goal is to derive two-sided weightings suitable for FWBR of the controller so that the frequency-weighted error measure

$$\varepsilon := \|\mathbf{W}_{\text{out}} (\mathbf{K} - \mathbf{K}_{\text{red}}) \mathbf{W}_{\text{in}}\|_\infty \quad (7.1)$$

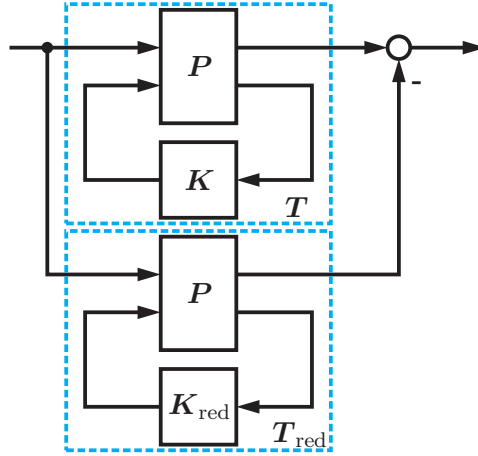


Figure 7.1: The Nominal Performance-preserving controller order reduction as model matching problem: Find \mathbf{K}_{red} of fixed order n_{red} to minimize $\|\mathbf{T} - \mathbf{T}_{\text{red}}\|_{\infty}$

reflects the \mathcal{H}_{∞} error in the closed-loop performance transfer path $\mathbf{w} \rightarrow \mathbf{z}$ which is given by

$$\varepsilon_{\text{P}} = \|\mathbf{T} - \mathbf{T}_{\text{red}}\|_{\infty} = \|\mathcal{F}_l(\mathbf{P}, \mathbf{K}) - \mathcal{F}_l(\mathbf{P}, \mathbf{K}_{\text{red}})\|_{\infty}, \quad (7.2)$$

which can be interpreted as a model matching problem as seen in Fig. 7.1.

Note that, different to a control design problem, the goal is not to reduce the gain of the performance path, but instead to approximate the performance transfer with low error.

Having found these weighting functions \mathbf{W}_{out} and \mathbf{W}_{in} , a FWBR is conducted to obtain the reduced controller \mathbf{K}_{red} of specified order, and the resulting controller is validated in closed loop.

To derive the frequency weightings, (7.2) is first expressed (approximately) as function of the controller reduction error

$$\Delta \mathbf{K} := \mathbf{K} - \mathbf{K}_{\text{red}}. \quad (7.3)$$

Therefore, the LFTs in (7.2) are expanded,

$$\mathbf{T} = \mathbf{P}_{11} + \mathbf{P}_{12} \mathbf{K} \mathbf{S} \mathbf{P}_{21} \quad (7.4)$$

$$\mathbf{T}_{\text{red}} = \mathbf{P}_{11} + \mathbf{P}_{12} \mathbf{K}_{\text{red}} \mathbf{S}_{\text{red}} \mathbf{P}_{21}, \quad (7.5)$$

where

$$\mathbf{S} = (\mathbf{I} - \mathbf{P}_{22} \mathbf{K})^{-1}, \quad (7.6)$$

$$\mathbf{S}_{\text{red}} = (\mathbf{I} - \mathbf{P}_{22} \mathbf{K}_{\text{red}})^{-1} \quad (7.7)$$

hold.

By help of the Woodbury identity [166] for a square nonsingular matrix \mathbf{A} and compatible matrices \mathbf{B} and \mathbf{C} ,

$$(\mathbf{A} - \mathbf{B}\mathbf{C})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1} \mathbf{B} (\mathbf{I} - \mathbf{C}\mathbf{A}^{-1} \mathbf{B})^{-1} \mathbf{C}\mathbf{A}^{-1}, \quad (7.8)$$

equation (7.7) can be expanded into the following form:

$$\mathbf{S}_{\text{red}} = \mathbf{S} - \mathbf{S} \mathbf{P}_{22} \underbrace{(\mathbf{I} + (\Delta \mathbf{K}) \mathbf{S} \mathbf{P}_{22})^{-1}}_{\approx \mathbf{I}} (\Delta \mathbf{K}) \mathbf{S}, \quad (7.9)$$

$$\mathbf{S}_{\text{red}} \approx \mathbf{S} - \mathbf{S} \mathbf{P}_{22} (\Delta \mathbf{K}) \mathbf{S} \quad (7.10)$$

where the approximation is justified if $\|(\Delta\mathbf{K})\mathbf{S}\mathbf{P}_{22}\| \ll \|\mathbf{I}\|$ holds which is fulfilled for sufficiently small $\|\Delta\mathbf{K}\|$.

The performance path perturbation can thus be expressed as

$$\Delta\mathbf{T} = \mathbf{T} - \mathbf{T}_{\text{red}} = \mathbf{P}_{12} [\mathbf{K}\mathbf{S} - \mathbf{K}_{\text{red}}\mathbf{S}_{\text{red}}] \mathbf{P}_{21} \quad (7.11)$$

$$\approx \mathbf{P}_{12} [\mathbf{K}\mathbf{S} - (\mathbf{K} - \Delta\mathbf{K})(\mathbf{S} - \mathbf{S}\mathbf{P}_{22}(\Delta\mathbf{K})\mathbf{S})] \mathbf{P}_{21} \quad (7.12)$$

$$\approx \mathbf{P}_{12} \left[\mathbf{I} + \mathbf{K}\mathbf{S}\mathbf{P}_{22} - \underbrace{(\Delta\mathbf{K})\mathbf{S}\mathbf{P}_{22}}_{\approx 0} \right] (\Delta\mathbf{K})\mathbf{S}\mathbf{P}_{21} \quad (7.13)$$

$$\approx \underbrace{\mathbf{P}_{12} [\mathbf{I} + \mathbf{K}\mathbf{S}\mathbf{P}_{22}]}_{\widetilde{\mathbf{W}}_{\text{out}}} (\Delta\mathbf{K}) \underbrace{\mathbf{S}\mathbf{P}_{21}}_{\widetilde{\mathbf{W}}_{\text{in}}}, \quad (7.14)$$

where the second approximation is done (justified if $\|(\Delta\mathbf{K})\mathbf{S}\mathbf{P}_{22}\| \ll \|\mathbf{I} + \mathbf{K}\mathbf{S}\mathbf{P}_{22}\|$ holds, which again is valid for sufficiently small $\|\Delta\mathbf{K}\|$).

From the assumed internal stability of the original closed loop it follows that \mathbf{P}_{11} , \mathbf{P}_{12} , \mathbf{P}_{21} , \mathbf{S} as well as $\mathbf{K}\mathbf{S}\mathbf{P}_{22}$ (and thus also $\widetilde{\mathbf{W}}_{\text{out}}$ and $\widetilde{\mathbf{W}}_{\text{in}}$) are all stable.

Note moreover that $\widetilde{\mathbf{W}}_{\text{out}}$ describes the closed-loop transfer from a disturbance injected at \mathbf{u} to \mathbf{z} , whereas $\widetilde{\mathbf{W}}_{\text{in}}$ is the closed-loop transfer from \mathbf{w} to \mathbf{y} , which can be directly verified by formulating these from the block diagram in Fig. 7.1.

For FWBR, the weightings however must also be stably invertible, so they need to be minimum-phase, square, and bi-proper (with full-rank feedthrough matrix). Generally, none of these requirements are fulfilled for the derived weightings from the outset. Heuristic methods to obtain admissible FWBR weighting functions are proposed in the following, considering that only their magnitude is relevant for FWBR.

Proposed steps to obtain admissible FWBR weightings: Starting out with the derived functions $\widetilde{\mathbf{W}}_{\text{out}}$ and $\widetilde{\mathbf{W}}_{\text{in}}$, the goal is to generate two FWBR weighting functions \mathbf{W}_{out} and \mathbf{W}_{in} which fulfill all listed requirements.

Square weights The derived weights $\widetilde{\mathbf{W}}_{\text{out}}$ and $\widetilde{\mathbf{W}}_{\text{in}}$ are in general non-square (of I/O-dimensions $(n_z \times n_u)$ respectively $(n_y \times n_w)$), but \mathbf{W}_{out} and \mathbf{W}_{in} need to be square ($(n_u \times n_u)$ respectively $(n_y \times n_y)$, or scalar). The simplest possible choice is to utilize scalar FWBR weights which are fitted to the maximum singular value of the respective non-square weight. However, more structure can be exploited in the MIMO case when considering the SVD (see Sec. A.3). Utilizing suitable subsets of the singular vectors and corresponding singular values to also retain the directional information could be beneficial for FWBR. This issue is, however, not investigated here in more detail and a potential subject of future research.

Bi-properness with full-rank feedthrough matrix If \mathbf{P}_{12} or \mathbf{P}_{21} are strictly proper, so is $\widetilde{\mathbf{W}}_{\text{out}}$ respectively $\widetilde{\mathbf{W}}_{\text{in}}$. However, the inverse of a strictly proper transfer function is improper and thus cannot be realized in state-space representation. To obtain realizable inverses, the feedthrough matrix moreover must be square and of full rank. To obtain this property for a system \mathbf{W} , a small perturbation in form of a diagonal matrix $\alpha\|\mathbf{W}\|_{\infty}\mathbf{I}$ with $0 < \alpha \ll 1$ can be added to the \mathbf{D} -matrix. Note that α should be chosen sufficiently large (compared to computation accuracy) to avoid bad numeric conditioning but small enough not to distort the relevant weighting function shape; a value of 10^{-3} has been found suitable for the studied problems.

Minimum-phase fitting As last step, if the weighting functions exhibit RHP transmission zeros, the weighting function magnitudes need to be fitted by minimum-phase transfer

functions (or state-space realizations) to ensure that all zeros lie in the LHP so that stable inversion is possible. Note that this step is required after the other steps (obtaining square weights and feedthrough full-rank-perturbation) because these can affect the zero locations. Various algorithms for minimum-phase fitting exist, notable Cepstrum algorithms or the log-Chebyshev method implemented in the MATLAB[®] Robust Control Toolbox (for example, via the command `fitmagfrd`) [6].

If the controller \mathbf{K} is not stable, one proposed solution is to separate \mathbf{K} into a stable part \mathbf{K}_s and an unstable part \mathbf{K}_{ns} where

$$\mathbf{K} = \mathbf{K}_s + \mathbf{K}_{ns} \quad (7.15)$$

holds. This is a factorization problem and readily implemented in MATLAB[®] via the command `stabsep`. Then, the interconnection can be rewritten into

$$\tilde{\mathbf{P}} = \mathcal{F}_1(\mathbf{P}, \mathbf{K}_{ns}) \quad (7.16)$$

and

$$\mathbf{T} = \mathcal{F}_1(\mathbf{P}, \mathbf{K}) = \mathcal{F}_1(\tilde{\mathbf{P}}, \mathbf{K}_s) \quad (7.17)$$

holds true. Now, the FWBR procedure can be applied as before with $\tilde{\mathbf{P}}$ and \mathbf{K}_s instead of \mathbf{P} and \mathbf{K} , respectively.

7.3 Case Study: NP-Preserving Controller Order Reduction

The reduction performance of NP-preserving weightings is demonstrated in the following at a controller order reduction problem for a high-order \mathcal{H}_∞ controller for rigid-body dynamics shaping of a flexible aircraft model. The utilized model is that of the BWB aircraft model referred to in Sec. 4.2, however focusing on longitudinal control. The utilized high-order plant is of order 54 in minimal realization, the signals are $w = u_{EL}$ (disturbance on elevator), $\mathbf{u} = [u_{EL}, u_{AIL3}, u_{MINIFL}]^T$ (symmetric deflections of elevator and of two outer flaps), $z = Nz_{CG}$ (vertical acceleration), and $\mathbf{y} = [Nz_{CG}, q_{CG}, Nz_{law}, Nz_{pax}]^T$ are the vertical acceleration at CG, the pitch rate, as well as vertical acceleration signals derived in terms of a symmetric modal wing bending sensor law (Nz_{law}) and at a comfort-relevant position in the cabin (Nz_{pax}).

This particular controller is of low authority and of low-pass behavior with total dynamic order 61. This complexity seems, especially when considering the low control authority, unnecessarily high and undesirable for implementation.

Figure 7.2 shows the results of a truncation-type reduction to order 2 by the unweighted balanced reduction (BR), yielding \mathbf{K}_{BR} , as well as by FWBR with NP-preserving weightings as derived in the previous section, yielding \mathbf{K}_{FWBR} , in terms of the maximum singular value. In the upper left plot, the controller singular values are plotted. The difference between the full-order controller and the reduced controllers are plotted in the lower left plot. Note that, as expected, the unweighted balanced reduction yields smaller maximum singular values of this unweighted reduction error. However, observing the closed-loop performance paths in the plots on the right reveals that the NP-FWBR approach yields far smaller weighted reduction errors than the unweighted BR.

In Fig. 7.3, the results of unweighted balanced reduction as well as the proposed NP-FWBR onset are shown for reduced controller orders from 21 down to a static gain. Thereby, both reduction variants, truncation and residualization, are computed for each reduction order and the lowest relative \mathcal{H}_∞ error

$$\frac{\|\mathbf{T}_{zw} - \mathbf{T}_{zw,red}\|_\infty}{\|\mathbf{T}_{zw}\|_\infty} \quad (7.18)$$

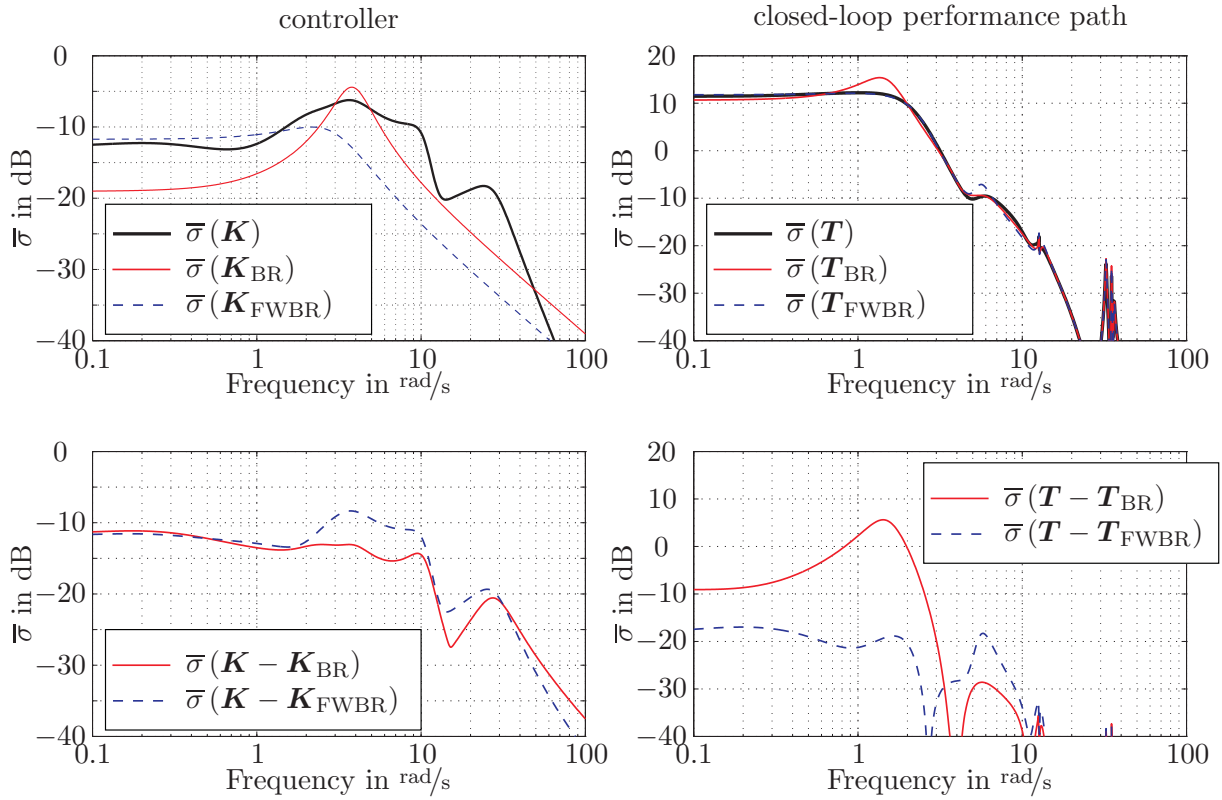


Figure 7.2: Maximum singular values $\bar{\sigma}$ of reduced controllers and their closed-loop performance: unweighted BR vs. NP-FWBR

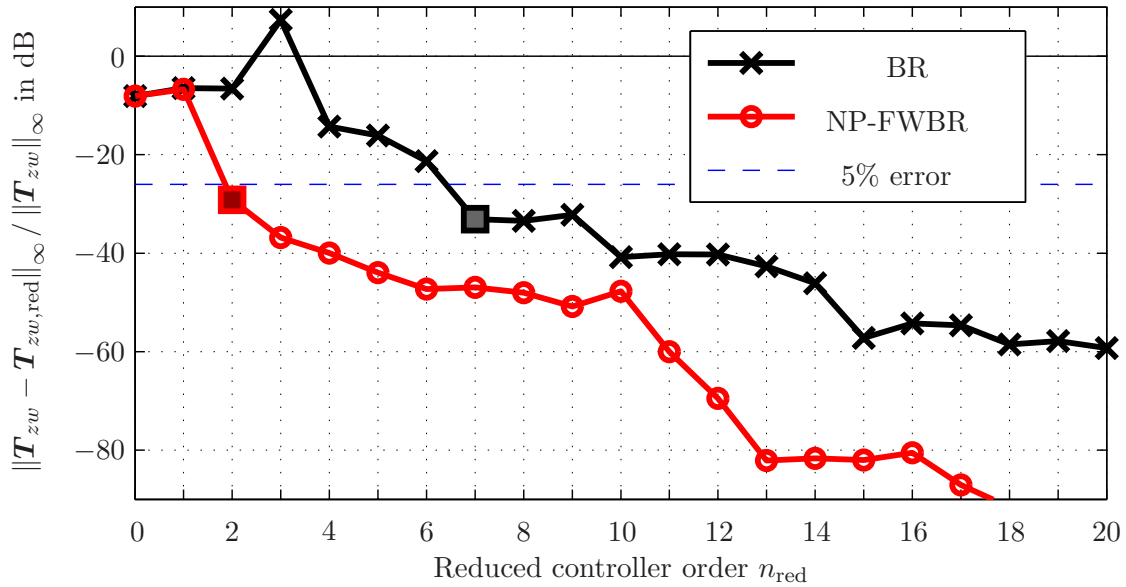


Figure 7.3: Comparison of best (smallest) achieved relative \mathcal{H}_∞ closed-loop performance error with unweighted balanced reduction (BR) and NP-preserving FWBR. The boxes indicate the lowest-order controller achieving less than 5% error.

is plotted. For all evaluated reduced controller orders it is clearly evident that the NP-preserving FWBR is far superior in efficiency. Unweighted balanced reduction requires controller order 7 to retain this error quantity below 5%, but the FWBR method produces even a controller of order 2 for this performance level (which is the one shown in Fig. 7.2).

As a concluding remark, it should be noted that this controller is, compared to its actually effective dynamics, of very high order and in this sense more efficient designs (with reduced-order design models, for example) would be possible. However, this case demonstrates that the specific FWBR formulation derived here is able to extract the relevant dynamics, tailored for a given nominal performance formulation. Note that simple tuning of these FWBR weights is possible to further shape the reduction process.

7.4 Robust Performance Frequency Weighting Derivation: An Extension

The previous section addresses nominal performance only, but robustness issues are not accounted for in the controller reduction method. In this section, an FWBR procedure aiming to retain robust performance (RP) in terms of the RP μ value in the reduction process. This work extends a recently published formulation of controller perturbation bounds [19] in the robust control framework by utilizing these obtained bounds to generate FWBR weightings. Remarkable results in controller order reduction can be achieved at a challenging RP problem from literature.

7.4.1 Previous Results

Consider a $\mathbf{P} - \mathbf{K} - \mathbf{\Delta}$ - interconnection structure (see Sec. 2.3.1) composed of an augmented plant \mathbf{P} , a feedback controller \mathbf{K} of order n_K , and an uncertainty block $\mathbf{\Delta}$ so that robust performance (RP) is fulfilled, that is,

$$\mu_{\widehat{\mathbf{\Delta}}}(\mathbf{M}(j\omega)) < 1, \quad \forall \omega_i \in \Omega \quad (7.19)$$

holds where $\widehat{\mathbf{\Delta}}$ is the uncertainty block used for RP evaluation on the frequency grid $\Omega = \{\omega_1, \dots, \omega_{n_\omega}\}$, see Sec. 2.3.1.

The idea followed in [19] is to capture the reduction error of \mathbf{K} in terms of an associated uncertainty $\mathbf{\Delta}_K$ and an appropriate weighting \mathbf{W}_K . Given that the full-order controller obtains robust performance, a conservative scalar bound (template) $A(\omega_i) \bar{\sigma}(\mathbf{K}(j\omega_i))$ for the admissible reduction error is derived. This involves the solution of a μ fitting problem (a skew- μ problem, see Sec. 2.3.1) at each ω_i (refer to [19] for details). Finally, the authors of [19] apply the unweighted balanced reduction to the controller to find the reduced controller \mathbf{K}_{red} of lowest order n_{red} for which

$$\|\mathbf{K} - \mathbf{K}_{\text{red}}\|_\infty \leq 2 \sum_{j=n_{\text{red}}+1}^{n_K} \sigma_j < \min_{\omega_i \in \Omega} A(\omega_i) \bar{\sigma}(\mathbf{K}(j\omega_i)) \quad (7.20)$$

holds, thus guaranteeing RP for the reduced controller in the considered $\mathbf{P} - \mathbf{K} - \mathbf{\Delta}$ - interconnection structure.

7.4.2 Extension into an RP-preserving FWBR method

The procedure outlined above is an unweighted balanced reduction of the controller itself, which, as shown in Sec. 7.3, yields low reduction performance for the controller order reduction problem. It could be argued that, in a practical control design setting, the a priori guaranteed error bounds

(which requires the knowledge of the HSVs) are of minor importance because they are typically not tight and often very conservative, and detailed analysis and validation of reduced controllers has to be carried out in any case. These reasons justify the investigation of reduction methods such as FWBR formulations also if they cannot provide guaranteed error bounds.

In the following, the controller perturbation bound $A(\omega_i) \bar{\sigma}(\mathbf{K}(j\omega_i)) > 0 \forall \omega_i \in \Omega$ from [19] will be utilized to obtain one- and two-sided scalar weighting functions for RP-preserving FWBR as follows:

- Written frequency-wise, the perturbation bound guaranteeing RP (evaluated with respect to the sufficiently fine frequency gridding Ω) reads

$$\bar{\sigma}(\mathbf{K}(j\omega_i) - \mathbf{K}_{\text{red}}(j\omega_i)) < A(\omega_i) \bar{\sigma}(\mathbf{K}(j\omega_i)) \quad \forall \omega_i \in \Omega. \quad (7.21)$$

- By dividing at each ω_i by the right-hand side of the inequality one obtains

$$\underbrace{\frac{1}{A(\omega_i) \bar{\sigma}(\mathbf{K}(j\omega_i))}}_{\hat{w}(\omega_i)} \bar{\sigma}(\mathbf{K}(j\omega_i) - \mathbf{K}_{\text{red}}(j\omega_i)) < 1 \quad \forall \omega_i \in \Omega, \quad (7.22)$$

where the constant right-hand side is equal to 1 for each ω_i .

- A scalar stable, minimum-phase, and bi-proper frequency-domain weighting function $W(s)$ is fitted to $\hat{w}(\omega_i)$. Under the condition that $\bar{\sigma}(W(j\omega)) \bar{\sigma}(\mathbf{K}(j\omega) - \mathbf{K}_{\text{red}}(j\omega)) < 1$ for all $\omega \in \mathbb{R}$, the frequency-weighted \mathcal{H}_∞ norm condition

$$\|W(\mathbf{K} - \mathbf{K}_{\text{red}})\|_\infty < 1 \quad (7.23)$$

follows which allows to identify the one-sided FWBR weight as $W(s)$ by inspection. Since the weight is scalar, no difference arises whether it is used as input or output weighting.

Remark: It is also possible to obtain two-sided weightings by fitting $W_{\text{out}}(s) = W_{\text{in}}(s)$ to $\sqrt{\hat{w}(\omega_i)}$, but numeric studies (see Sec. 7.5) show that the FWBR problem does not produce better results and numeric conditioning may be worse than in the one-sided case.

7.5 Case Study: RP-Preserving Controller Order Reduction

To test the robust controller reduction performance, the publicly available control design example from the textbook [143, pp. 330–335], a DK-iteration design for a (2×2) distillation plant model, is utilized. Although automatic DK-iterations do not achieve RP, a manually optimized controller with \mathbf{D} -scalings was found in [91] (see [143] for its design parameters) which just attains RP with an RP μ value of $\mu_{\text{RP}}^* = 0.974$. The values of μ are flat and close to the maximum value across a large region of the considered frequency range. The augmented plant order is 6, the controller is of order $n_K = 16$. Although this controller complexity is reasonably low for actual implementation, this well-known example provides a challenging RP problem and is thus believed to be well-suited to demonstrate the effectiveness of the RP-preserving controller order reduction procedure.

Frequency weightings for both, the NP-preserving (disregarding the modeled uncertainty) and the RP-preserving FWBR methods are derived. For the μ analysis steps in the RP-preserving FWBR weight computation, the same frequency gridding as in the textbook control design is utilized (61 logarithmically-spaced points from 10^{-3} to 10^3 rad/s). The minimum-phase weighting functions were found by a fitting procedure known as log-Chebyshev magnitude design

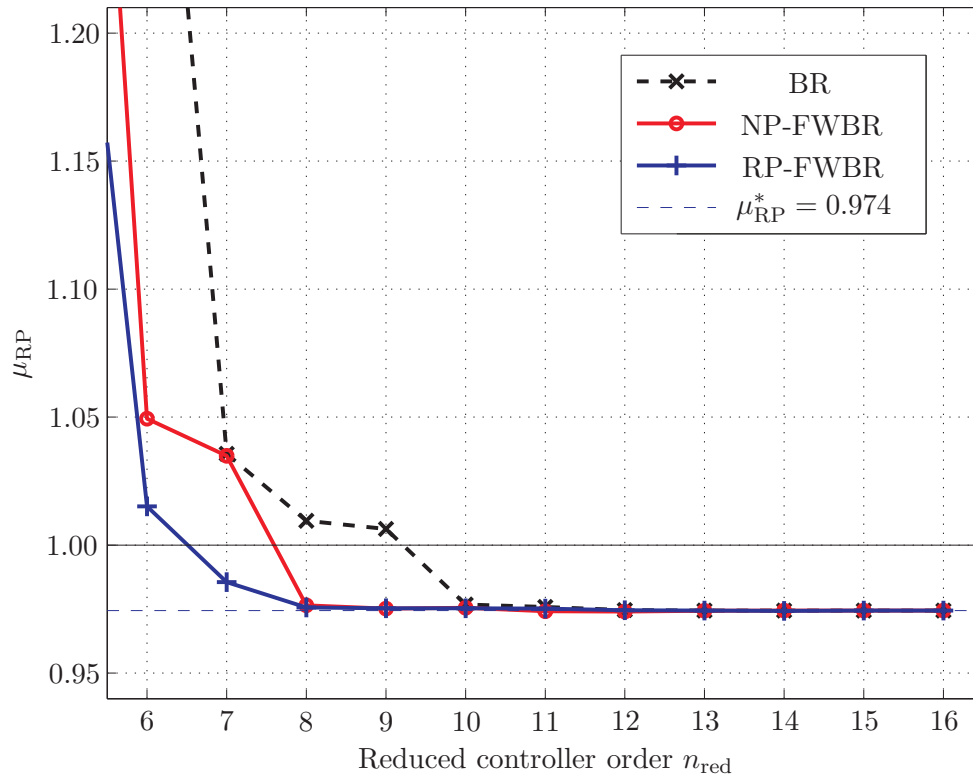


Figure 7.4: Comparison of best achieved RP μ with unweighted balanced reduction (BR), NP-, and RP-preserving FWBR

(MATLAB[®] command `fitmagfrd`, see [12]), yielding a one-sided weighting function of order 17 and, for comparison, two-sided weighting functions of order 19 each.

Robust performance preservation is evaluated for all dynamic orders $n_{\text{red}} = 1, \dots, n_K$ of the reduced controller by a RP μ analysis. For each reduction method, both, the truncation and the residualization variants are tested. Additionally, the RP-preserving FWBR method is tested both with one-sided and two-sided weightings. At each n_{red} and for each reduction method, the lowest value of μ_{RP} is plotted in Fig. 7.4. While no noticeable differences occur for controller orders down to 10 (particularly no deterioration of μ_{RP}), further reduction shows differences: The lowest controller orders which fulfill RP are 10 for the unweighted balanced reduction, 8 for the NP-preserving FWBR, and 7 for the RP-preserving FWBR. With a controller order as low as 6, the RP- and NP-FWBR methods yield controllers which slightly violate RP ($\mu_{\text{RP}} = 1.02$ respectively 1.05), whereas the unweighted balanced reduction yields a significantly larger value of $\mu_{\text{RP}} = 1.39$.

Further analysis shows no clear advantages of the two- or the one-sided weighted cases of the RP-preserving FWBR; the same applies to choice between truncation and residualization. Testing all combinations to obtain optimal results is thus advisable. Note that, having obtained the error bound it is computationally usually less expensive to carry out weight fitting and the FWBR procedure itself than to perform the subsequent high validations which involve a μ analysis for each produced controller.

Chapter 8

Gain Scheduling & Linear System Interpolation Studies

Gain scheduling concepts are defined in various ways [122]. The main task, however, is to find and realize a nonlinear control law which exploits system parameter information for a nonlinear system if a single (robust) linear controller cannot provide satisfactory stability or performance at the nonlinear plant. Gain scheduling allows to use linear control design tools to obtain finally a parametrized nonlinear control law with the aim to provide superior stability and performance over large system operation ranges.

One central task in many gain-scheduling design methods is the interpolation of linear parameter-dependent system dynamics to obtain the final controller realization. This can be eased by exploiting structure wherever possible. One such case is LFR-based scheduling of compensators in an observer-based representation (see Appendix Chap.B). However, typically this is only possible for a part of the system dynamics. The remaining dynamics must be interpolated by generic methods such that closed-loop stability (and performance) is maintained. Various onsets for the interpolation of specific linear system representations have been studied in the literature, but these most often cannot guarantee stability or performance of the closed loop. It is thus of interest to study interpolation methods and their properties for unstructured dynamics which is addressed herein.

This chapter is structured as follows: In Sec. 8.1 an overview of gain-scheduling onsets and interpolation methods for linear systems is given. Section 8.2 presents the LFR-based scheduling of controllers in observer-based representations and demonstrates the need of stability-preserving system interpolation methods. In Sec. 8.3, two well-known interpolation methods which are based on specific canonical state-space representations are analyzed in their stability properties, and stability results are established. Section 8.4 presents first results on a novel interpolation method, based on Geometric Algebra concepts and modal interpolation. This method preserves stability and exploits the system structure in a clear geometric fashion. An academic example is given in Sec. 8.5 in which the novel approach produces surprisingly good results. Finally, Sec. 8.6 discusses and compares the studied interpolation methods and gives an outlook.

8.1 Overview

8.1.1 Gain Scheduling

Gain scheduling of control laws has a long history in control engineering. Comprehensive surveys are given in [122] and [81]. Today, two important groups of methods are the so-called linearization-based scheduling and the LPV (linear parameter-varying) system approaches. The first treats the system

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{F}(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \\ \mathbf{y} &= \mathbf{G}(\mathbf{x}, \mathbf{u})\end{aligned}\tag{8.1}$$

via families of equilibrium or off-equilibrium linearizations. For these model families, controllers are designed and/or defined via interpolation, and the controller corresponding to the value of the scheduling parameter signal $\boldsymbol{\theta}(t)$ is utilized.

In contrast, LPV approaches use a more implicit representation of the system's parameter dependency. LPV-based control design directly yields a parametrized controller (which is also an LPV system). While these approaches have strong analytical results with respect to performance and stability, the existing onsets (involving large LMI problems) do not scale well with the design plant order. This can render direct LPV design computationally infeasible as the number of relevant states grows, for example in structure control problems with many modes. Applications of LPV structure / flight control can be found in [168] and most recently in [157].

One common representation of an LPV system parametrization, if the dependency of the state-space matrices on the parameters is polynomial or rational, is obtained as a Linear Fractional Representation (LFR). This structure is equivalent to a parametric uncertainty description (see Sec. 2.3.2) in which the Δ -block is real-valued and diagonal and contains the (typically repeated) normalized system parameters.

If a plant parametrization is available as LFR during operation, the so-called LFR-based scheduling onset can be utilized for gain-scheduled control of controllers in observer-based realization (see Sec. B.3). In [151], this scheduling onset is realized for the scheduled control of a flexible aircraft. The design of such controllers can coarsely be categorized into four steps:

- First, a parametrized design model must be obtained as LFR.
- Then, controllers are designed at numerous design points, and these controllers are transformed to a mutually compatible OBR.
- The scheduled controller is implemented by utilizing the parametrized plant matrices to construct the observer part and interpolation of the observer gain, the state feedback gain, and the Youla parameter, see Sec. B.3.
- Stability of the scheduled controller is tested in high-fidelity validation simulations.

The key issue lies in the last two steps: If stability cannot be achieved, it is not immediately clear how to repair the design. To prevent this, it is of interest to perform the scheduling such that closed-loop stability is preserved. Thereby, stability is assumed in the design grid points as direct result of the control designs.

8.1.2 Linear System Interpolation

Local linear models are a versatile structure to model non-linear / parameter-varying systems. However, a meaningful interpolation between the individual models is not a trivial task. Given a set of LTI state space models ("grid models") at known parameter grid points, linear system interpolation aims to generate systems with meaningful dynamics at parameter values between these parameter values.

The need for interpolation of linear system dynamics is also relevant in data-based modeling of non-linear parametric systems. Local linear models are a natural onset for black-box models of non-linear processes. Suitable interpolation laws attempt to obtain a parametrized global model with a large validity region. Local linear model networks (LLMNs), Takagi-Sugeno fuzzy models, or radial basis function networks are well-known implementations of this idea, see [102]

for details. In control, interpolation (system parametrization) arises in two major control disciplines: robust control and gain-scheduled control. Typically, the plant parametrization is put under a Linear Fractional Representation (LFR) [143]. Efficient LFR modeling from LTI grid models or from complex analytic relations is thus relevant for control design and validation, see [111] and [29]. Scheduled control laws aim to utilize online parameter information to improve control performance. The works [81] and [122] are two well-known surveys on gain scheduling technology. Recently, various approaches to linear systems interpolation, often in the context of linear parameter-varying (LPV)/LFR modeling and identification, have been published: [15] surveys analytic and data-based modeling approaches of LPV/LFR system architectures and proposes to exploit their synergies. Application-oriented works utilize specific system interpolation approaches: the modal form is utilized in [121], a re-scaled companion form is utilized in [29].

Other interpolation approaches are given in [104] (interpolating the zero-pole-gain form which can be seen as a modal interpolation onset); [72] (interpolation in balanced representation) and [87] (a balanced subspace approach); a framework for parametric order reduction of large-scale systems is provided in [108]. Hence, many contributions on system interpolation report on methods which are either based on ad-hoc principles or specifically tailored to certain applications.

8.2 LFR-based Scheduling of Compensators in Observer-based Representation

The observer-based state-space representation (OBR) of compensators is detailed in Sec. B.3 and useful both in convex control design (see Sec. 3.6.3) and in obtaining a gain-scheduled controller. Given such compensator in OBR as well as a plant model parametrized in LFR form, a widely-applied scheduling variant can be constructed as shown in the following. For results of a flight control design and this form of controller scheduling, see [151].

8.2.1 Interpolation of Full- and Augmented-Order Compensators

For a number of k design points (plant operating points or parameter cases) indexed by $j \in \{1, \dots, k\}$, let a full- or augmented-order compensator \mathbf{K}_j for each j (each of the same order $n_K \geq n$) be given in OBR. Each \mathbf{K}_j is realized by the observer part which itself consists of the observer gain $\mathbf{K}_{f,j}$ and the plant state space matrices $\mathbf{A}_j, \mathbf{B}_j, \mathbf{C}_j, \mathbf{D}_j$, as well as the state feedback gain $\mathbf{K}_{c,j}$ and a Youla parameter $\mathbf{Q}_j = \left[\begin{array}{c|c} \mathbf{A}_{Q,j} & \mathbf{B}_{Q,j} \\ \mathbf{C}_{Q,j} & \mathbf{D}_{Q,j} \end{array} \right]$ of order $n_K - n$.

Now assume that a sufficiently accurate LFR parametrization $\mathbf{P}_{\text{LFR}} = \mathcal{F}_u(\mathbf{P}, \mathbf{\Delta})$ exists with \mathbf{P} as the nominal plant and $\mathbf{\Delta}$ as a real diagonal matrix whose entries are the (possibly repeated) physical parameters of the plant, such that for each parameter matrix $\mathbf{\Delta}_j$ corresponding to design point j , the design plants are (approximately) obtained:

$$\mathbf{P}_{\text{LFR}}|_{\mathbf{\Delta}_j} = \mathcal{F}_u(\mathbf{P}, \mathbf{\Delta}_j) \cong \mathbf{P}_j. \quad (8.2)$$

Then, a scheduled compensator can be realized in OBR by utilizing the parametrized plant model matrices $\mathbf{A}_{\text{LFR}}(\mathbf{\Delta}), \mathbf{B}_{\text{LFR}}(\mathbf{\Delta}), \mathbf{C}_{\text{LFR}}(\mathbf{\Delta}), \mathbf{D}_{\text{LFR}}(\mathbf{\Delta})$ in the observer part. The observer gain $\mathbf{K}_f := \mathbf{K}_f(\mathbf{\Delta})$ and the state feedback gain $\mathbf{K}_c := \mathbf{K}_c(\mathbf{\Delta})$ are typically obtained by linear or polynomial matrix interpolation in the parameter space. The same applies to obtaining a static interpolated Youla parameter $\mathbf{Q} = \mathbf{Q}(\mathbf{\Delta})$ in the full-order case ($n = n_K$). However, in the augmented-order case, the Youla parameter is dynamic and its system dynamics must be suitably interpolated.

Remarks on Stability

The interpolation procedure above does not guarantee closed-loop stability between the design points. As seen from Theorem 3.6.2 in Sec. 3.6.3 and from the separation principle in Sec. B.2.1, for the interpolation of full- and augmented-order compensators, the set of closed-loop poles (the spectrum of the closed-loop system matrix) is the union of the spectrum of the matrices $\mathbf{A} - \mathbf{B}_2\mathbf{K}_c$, $\mathbf{A} - \mathbf{K}_f\mathbf{C}_2$, and \mathbf{A}_Q . They all need to be stable to ensure closed-loop stability. In the interpolated case, for fixed parameter values Δ , it must thus be ensured that $\mathbf{A}(\Delta) - \mathbf{B}_2(\Delta)\mathbf{K}_c(\Delta)$, $\mathbf{A}(\Delta) - \mathbf{K}_f(\Delta)\mathbf{C}_2(\Delta)$, and $\mathbf{A}_Q(\Delta)$ are all stable, given that this is fulfilled by design at $\Delta = \Delta_j, j \in \{1, \dots, k\}$. In this work the stability conditions for the first two matrix expressions are not studied in more depth, but it is evident that also the interpolation of the Youla parameter must be stable. This justifies the study of the stability properties of generic interpolation methods in the following.

The case of reduced-order compensators in OBR cannot be represented in the form above. Instead, the observer dynamics must itself be considered a dynamic system which must be interpolated under the constraint of providing sufficient closed-loop stability and performance properties.

8.3 Stability Properties of Classic System Interpolation Methods

While low-order control laws can often be satisfactorily scheduled in a heuristical ad-hoc manner, this does not extend to complex controllers of high dynamic order. Then, structured approaches such as the observer-based representation of full- and augmented-plant order controllers can be utilized. However, while the latter approach allows to exploit a known or modeled plant parametrization, the remaining dynamics, in particular if dynamic Youla parameters are present, still need to be interpolated without further structure.

8.3.1 Problem statement

Let a general, non-linear, parameter-dependent state space system be given as:

$$\dot{\mathbf{x}}_{\text{NL}} = \mathbf{f}(\mathbf{x}_{\text{NL}}, \boldsymbol{\rho}_{\text{NL}}, \mathbf{u}_{\text{NL}}), \quad \mathbf{y}_{\text{NL}} = \mathbf{g}(\mathbf{x}_{\text{NL}}, \boldsymbol{\rho}_{\text{NL}}, \mathbf{u}_{\text{NL}}). \quad (8.3)$$

The system has n states ($\mathbf{x}_{\text{NL}} \in \mathbb{R}^n$), m parameters ($\boldsymbol{\rho}_{\text{NL}} \in \mathbb{R}^m$), r inputs ($\mathbf{u}_{\text{NL}} \in \mathbb{R}^r$), and p outputs ($\mathbf{y}_{\text{NL}} \in \mathbb{R}^p$). The vector functions $\mathbf{f} : \mathbb{R}^{n \times m \times r} \rightarrow \mathbb{R}^n$ and $\mathbf{g} : \mathbb{R}^{n \times m \times r} \rightarrow \mathbb{R}^p$ are both assumed continuously differentiable.

Furthermore, let a set of k equilibria tuples $(\mathbf{x}_{\text{NL},j}, \boldsymbol{\rho}_{\text{NL},j}, \mathbf{u}_{\text{NL},j})$ be stationary points of (8.3). Linearization of (8.3) at these points for $\boldsymbol{\rho}_{\text{NL}} = \boldsymbol{\rho}_{\text{NL},j} = \text{const.}$ yields a set of k LTI systems

$$\dot{\mathbf{x}}_j = \mathbf{A}_j\mathbf{x}_j + \mathbf{B}_j\mathbf{u}_j, \quad \mathbf{y}_j = \mathbf{C}_j\mathbf{x}_j + \mathbf{D}_j\mathbf{u}_j, \quad (8.4)$$

where \mathbf{x} , \mathbf{u} , and \mathbf{y} are the deviation variables associated to their non-linear counterparts. The matrices \mathbf{A}_j , \mathbf{B}_j , \mathbf{C}_j , and \mathbf{D}_j are of compatible dimensions (system-, input-, output-, feed-through matrices, respectively).

The interpolation problem can be stated as follows: Given a known set of grid models (8.4) and their parameter values $\boldsymbol{\rho}_{\text{NL},j}$, an interpolated system is sought for a given parameter value $\boldsymbol{\rho}_{\text{NL}} \in \text{Conv}\{\boldsymbol{\rho}_{\text{NL},j}\}$ (Conv $\{\cdot\}$ indicates the convex hull of the argument vectors) which is, ideally speaking, as close as possible to the local linearization of (8.3) at $\boldsymbol{\rho}_{\text{NL}}$. However, due to the lack of structural knowledge on (8.3), this formulation is too general and it is necessary to resort to generic properties based on eigendynamics or signal- and system-norms on error

signals. This work focuses on the local stability of the interpolated systems. The grid models (8.4) actually differ not only in their matrices, but also in their stationary values. This affine term can affect global stability; however for local stability analysis it can be neglected. Details in the context of fuzzy systems are given in [28] and references therein. A general matrix interpolation onset for $\mathbf{x} = \mathbf{x}_1 = \dots = \mathbf{x}_k$, $\mathbf{u} = \mathbf{u}_1 = \dots = \mathbf{u}_k$, and $\mathbf{y} = \mathbf{y}_1 = \dots = \mathbf{y}_k$ is given as:

$$\dot{\mathbf{x}} = \sum_{j=1}^k \varphi_j(\boldsymbol{\rho}) \mathbf{A}_j \mathbf{x} + \sum_{j=1}^k \varphi_j(\boldsymbol{\rho}) \mathbf{B}_j \mathbf{u}, \quad \mathbf{y} = \sum_{j=1}^k \varphi_j(\boldsymbol{\rho}) \mathbf{C}_j \mathbf{x} + \sum_{j=1}^k \varphi_j(\boldsymbol{\rho}) \mathbf{D}_j \mathbf{u}, \quad (8.5)$$

where $\varphi_j(\boldsymbol{\rho})$ are interpolation weighting functions (activation functions) that satisfy $\sum_{j=1}^k \varphi_j(\boldsymbol{\rho}) = 1 \forall \boldsymbol{\rho}$, $0 \leq \varphi_j(\boldsymbol{\rho}) \leq 1 \forall j, \boldsymbol{\rho}$.

For simplicity and clarity, only the case $k = 2$ with a scalar parameter $\rho \in [0; 1]$ is considered in the following. Linear interpolation of the system dynamics for a given parameter value ρ results in:

$$\dot{\mathbf{x}} = ((1 - \rho) \mathbf{A}_1 + \rho \mathbf{A}_2) \mathbf{x} + ((1 - \rho) \mathbf{B}_1 + \rho \mathbf{B}_2) \mathbf{u}, \quad (8.6)$$

$$\mathbf{y} = ((1 - \rho) \mathbf{C}_1 + \rho \mathbf{C}_2) \mathbf{x} + ((1 - \rho) \mathbf{D}_1 + \rho \mathbf{D}_2) \mathbf{u}. \quad (8.7)$$

8.3.2 Companion form interpolation

In a recent paper [29], the authors generate a parametrized LFR of a flexible aircraft model by interpolating a given model grid. They choose to transform all grid models, obtained after order reduction of FE-based models, to a rescaled companion form and obtain satisfactory and reasonable results for the given application. The author of [29] points out that no need for mode assignment (as in modal form interpolation, see below) arises. The (unscaled) companion form of a fully controllable LTI system is:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ -a_0 & -a_1 & \dots & -a_{n-2} & -a_{n-1} \end{bmatrix} \mathbf{x} + \mathbf{B} \mathbf{u}. \quad (8.8)$$

The last row in the system matrix contains the coefficients of the monic characteristic polynomial. Linearly interpolating these state space representations element-wise thus corresponds to linearly interpolating the coefficients of their characteristic polynomials.

However, even if all grid models are stable, the interpolated plant is not necessarily stable. A suitable stability test is Bialas' Theorem [9] as formulated in [2]:

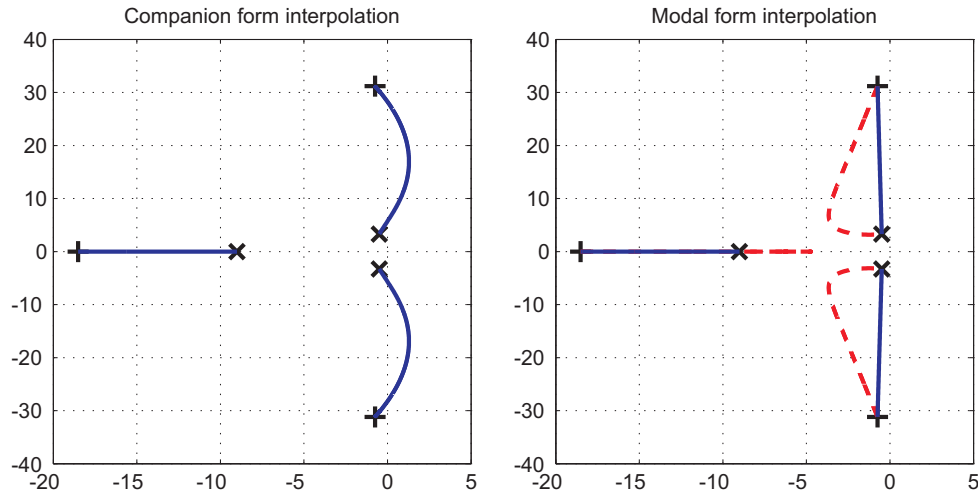


Figure 8.1: Root locus plots of the interpolated 3rd-order systems (8.12)–(8.13) with companion form interpolation (*left*) and modal form interpolation (*right*; *blue*: matching pole structure - stability guaranteed, *red dashed*: pole structure mismatch).

Theorem 8.3.1: Bialas [9]

Let $\mathbf{H}_1, \mathbf{H}_2$ be the Hurwitz matrices of

$$p_1(s) = a_0 + a_1s + a_2s^2 + \dots + a_ns^n, \quad a_n > 0, \quad (8.9)$$

$$p_2(s) = b_0 + b_1s + b_2s^2 + \dots + b_ns^n, \quad b_n > 0, \quad (8.10)$$

respectively. The polynomial family

$$P(s, \rho) = \{(1 - \rho)p_1(s) + \rho p_2(s) | \rho \in [0; 1]\} \quad (8.11)$$

is stable if and only if:

1. $p_1(s)$ is stable, and
2. the matrix $(\mathbf{H}_1)^{-1}\mathbf{H}_2$ has no non-positive *real* eigenvalues.

Proofs of this theorem are given in [9] and [2].

□

Already the interpolation of stable systems of order 3 can lead to unstable results. When choosing, for example, the two stable polynomials

$$p_1(s) = 18000 + 1000s + 20s^2 + s^3, \quad (8.12)$$

$$p_2(s) = 100 + 20s + 10s^2 + s^3, \quad (8.13)$$

condition 2 in Theorem 8.3.1 is violated ($(\mathbf{H}_1)^{-1}\mathbf{H}_2$ has 2 non-positive real eigenvalues) and the interpolation leads to unstable systems for $\rho \in [0.207; 0.987]$, see the root locus plot in Fig. 8.1 (*left*).

8.3.3 Modal form interpolation

Let a grid model (index j) be given in the real block-diagonal modal form [39]

$$\dot{\mathbf{x}} = \left[\begin{array}{c|cccc} \text{diag}(\lambda_{j,r,1\dots n_r}) & & & & \\ \hline & \mathbf{A}_{j,m,1} & \mathbf{0} & \dots & \mathbf{0} \\ & \mathbf{0} & \mathbf{A}_{j,m,2} & \dots & \mathbf{0} \\ & \vdots & & \ddots & \vdots \\ & \mathbf{0} & \dots & \mathbf{0} & \mathbf{A}_{j,m,n_c} \end{array} \right] \mathbf{x} + \mathbf{B}_j \mathbf{u} \quad (8.14)$$

with n_r real eigenvalues and n_c complex-conjugate eigenvalue pairs. The (2×2) -blocks $\mathbf{A}_{j,m,i}$ for each complex-conjugate mode $i = 1, \dots, n_c$ are:

$$\mathbf{A}_{j,m,i} = \begin{bmatrix} \Re\{\lambda_{j,i}\} & \Im\{\lambda_{j,i}\} \\ -\Im\{\lambda_{j,i}\} & \Re\{\lambda_{j,i}\} \end{bmatrix}. \quad (8.15)$$

This form can be established if a full-rank basis of eigenvectors can be obtained, which is fulfilled for systems with only distinct eigenvalues. The system matrix is first diagonalized via a similarity transformation with this eigenvector matrix and subsequently transformed to the form (8.14).

Under the prerequisite that all grid models are compatible in the structure of their eigenmodes, i.e., that all have the same number of real and complex-conjugate poles, the element-wise linear interpolation (8.6) within a set of grid models in the modal form (8.14) thus leads to linear interpolation of their eigenvalue locii in the complex plane, see Fig. 8.1 (*right*).

Theorem 8.3.2: Pole locations in structure-matched modal interpolation

Given a set of systems in real block-diagonal form with system matrices \mathbf{A}_j , $j = 1, \dots, k$ partitioned as in (8.14). Let $\mathcal{P} = \text{Conv}\{\lambda_{j,r,i_r}, \lambda_{j,c,i_c}, \overline{\lambda_{j,c,i_c}}\}$ ($j = 1, \dots, k$, $i_r = 1, \dots, n_r$, $i_c = 1, \dots, n_c$) be the convex hull over all real and complex eigenvalues of the model grid in the complex plane. Then, all interpolated system matrices resulting from (8.5) have eigenvalues inside \mathcal{P} .

Proof: The preserved structure across all plants allows to view the interpolation task mode-wise. Due to the structure of $\mathbf{A}_{j,m,i}$ as in (8.15) and the constraints on φ_j in (8.5), the eigenvalues of the convex combination $\sum_{j=1}^k \varphi_j(\boldsymbol{\rho}) \mathbf{A}_{j,m,i}$ always lie within the convex hull \mathcal{P}_i of the eigenvalues of $\mathbf{A}_{j,m,i}$, $j = 1, \dots, k$. The convex hull \mathcal{P} by definition spans all regions \mathcal{P}_i , $i = 1, \dots, n_c$. The same argument holds for real modes by replacing $\mathbf{A}_{j,m,i}$ by the real scalar $\lambda_{j,r,i}$, $i = 1, \dots, n_r$. \square

Corollary 1 (Stability preservation). *Let (the convex hull of) all poles \mathcal{P} of a system set in Theorem 8.3.2 lie in the left half plane. Then the structure-matched mode-wise modal interpolation yields a stable LTI system.*

Remark 1. *Analogous results to 1 hold for convex restricted stability regions and for discrete-time plants.*

Mode assignment

The central prerequisite of the modal interpolation onset is mode structure compatibility: The number of real and complex-conjugate poles must be the same for all parameter cases to apply Theorem 8.3.2. Multiple poles and the transition from real poles to complex-conjugate pairs or

vice versa requires special treatment and is subject of further studies. Moreover, pole-matching must be carried out explicitly through sorting. This is in fact an increased design freedom, and one particular choice is given by the companion form interpolation. Mode correspondence can be obtained by pole tracking algorithms [109] or by similarity measures of eigenvectors, input-, or output-directions [93].

8.4 Geometric Algebra Interpolation

In order to introduce structural properties into the system description and to exploit them for interpolation, one can resort to formal mathematical methods. Choosing to apply generalized ideas of geometry, using *Geometric Algebra*, introduces strong structural properties into the problem and may be advantageous for interpolation. Geometric Algebra (GA) is closely related to Clifford algebra [65] and is widely applied in (astro-)physics, quantum mechanics, computer graphics and robotics. It enables to handle geometry relations in a coordinate-free way.

8.4.1 Basic Idea

The basic idea is to view a linear dynamic state-space system (more exact: its eigendynamics) as a linear mapping $f : \mathbb{R}^n \mapsto \mathbb{R}^n$. The associated state space $\mathcal{Z} \subset \mathbb{R}^n$ is a linear space over which a geometric algebra can be defined. For example, for each complex-conjugate pair of eigenvalues, their eigenvectors can be seen as basis that spans a two-dimensional subspace $\mathcal{Z}^2 \subset \mathcal{Z}$, and one interpretation is that they thereby span an oscillating plane for the associated eigendynamics.

Given two systems S_1, S_2 (between which to interpolate), each with a complex-conjugate pair of eigenvalues $\lambda_{1+} = a + bj, \lambda_{1-} = a - bj, \lambda_{2+} = c + dj, \lambda_{2-} = c - dj$ and their associated eigenvectors $\mathbf{v}_{1+}, \mathbf{v}_{1-}, \mathbf{v}_{2+}, \mathbf{v}_{2-}$, a modal interpolation strategy is to smoothly parameterize the posture and orientation of the oscillatory plane. Usual subspace computations do not include information about orientation (since any eigenvector is defined up to a scalar real nonzero factor). However, using formal geometric algebra representations, this information – being intrinsic to the algebra itself – is available and can be exploited.

8.4.2 Interpolation of the System Matrix

Given grid models in their original state space representations (8.4), an interpolated system for a given parameter ρ is computed as follows [150]:

1. Establish the modal form of the grid models, perform pole matching. Let $\mathbf{A}_j = \mathbf{T}_j \mathbf{\Lambda}_j \mathbf{T}_j^{-1}$ be a pole-matched eigendecomposition where $\mathbf{T}_j \in \mathbb{C}^{n \times n}$ is the complex, full-rank, column-wise eigenvector matrix of \mathbf{A}_j and $\mathbf{\Lambda}_j$ the associated diagonal matrix of eigenvalues.
2. Formulate each mode's eigenvectors as GA objects (vectors, bivectors [150]): Specifically, the eigenvector pair $\mathbf{t}_{j,c,i} = \mathbf{u}_j \pm j\mathbf{v}_j$ (columns of \mathbf{T}_j) of the i th complex-conjugate mode $\lambda_{j,c,i}, \overline{\lambda_{j,c,i}}$ has the real-valued representation $\mathbf{u}_j, \mathbf{v}_j$. These vectors span the mode's oscillation plane which is represented in GA as a bivector $\mathbf{u}_j \wedge \mathbf{v}_j$ (constructed by the outer product \wedge of both vectors).
3. Perform geometric interpolation by scaling and rotation operations: the attitude, orientation, and magnitudes of the GA objects are interpolated. For two systems $j \in \{1, 2\}$ and the interpolation parameter $\rho \in [0, 1]$, the interpolated eigenvector of each real mode is obtained by linear interpolation: $\mathbf{t}_{\text{int},r,i} = (1 - \rho)\mathbf{t}_{1,r,i} + \rho\mathbf{t}_{2,r,i}$. For each complex-conjugate mode, the interpolated oscillation plane (bivector) is obtained by

$$B_{\text{int}}(\rho) = (1 - \rho)\mathbf{u}_1 \wedge \mathbf{v}_1 + \rho\mathbf{u}_2 \wedge \mathbf{v}_2. \quad (8.16)$$

The attitude of the concrete interpolated vector $\mathbf{u}_{\text{int}}(\rho)$ is obtained by asserting that it lies both on the interpolated oscillatory plane $B_{\text{int}}(\rho)$ and on the plane spanned by $\mathbf{u}_1, \mathbf{u}_2$. This can be expressed in various ways:

$$\mathbf{u}_{\text{int}}(\rho) = \alpha \text{meet}(\mathbf{u}_1 \wedge \mathbf{u}_2, B_{\text{int}}(\rho)) \quad (8.17)$$

via the `meet` operator which intersects the argument subspaces and a real scalar α . If \mathbf{u}_1 and \mathbf{u}_2 are not co-linear, the following outer product expressions both hold:

$$\mathbf{u}_{\text{int}}(\rho) \wedge (\mathbf{u}_1 \wedge \mathbf{u}_2) = 0 \quad \text{and} \quad \mathbf{u}_{\text{int}}(\rho) \wedge B_{\text{int}}(\rho) = 0. \quad (8.18)$$

Alternatively, $\mathbf{u}_{\text{int}}(\rho)$ can be expressed via a scaled rotation which provides the advantage of preserved orientation:

$$\mathbf{u}_{\text{int}}(\rho) = r_{u_{\text{int}}} R \frac{\mathbf{u}_1}{|\mathbf{u}_1|} R^{-1}, \quad R = \exp\left(-\frac{\mathbf{u}_1 \wedge \mathbf{u}_2}{|\mathbf{u}_1 \wedge \mathbf{u}_2|} \frac{\theta_u \rho}{2}\right), \quad \cos \theta_u = \frac{\mathbf{u}_1 \cdot \mathbf{u}_2}{|\mathbf{u}_1| |\mathbf{u}_2|}. \quad (8.19)$$

The interpolated vector's length $r_{u_{\text{int}}}$ is chosen to be interpolated between the lengths of \mathbf{u}_1 and \mathbf{u}_2 :

$$r_{u_{\text{int}}} = |\mathbf{u}_{\text{int}}| = (1 - \rho)|\mathbf{u}_1| + \rho|\mathbf{u}_2|. \quad (8.20)$$

The vector \mathbf{v}_{int} is computed analogously.

4. Construct the interpolated eigenvalue matrix $\mathbf{T}_{\text{int}}(\rho)$ mode-wise. The complex-conjugate pairs of eigenvectors are

$$[\mathbf{t}_{\text{int},c,i,1}, \mathbf{t}_{\text{int},c,i,2}] (\rho) = [\mathbf{u}_{\text{int}} + \mathbf{j}\mathbf{v}_{\text{int}}, \mathbf{u}_{\text{int}} - \mathbf{j}\mathbf{v}_{\text{int}}] (\rho). \quad (8.21)$$

5. Define the interpolated eigenvalue matrix $\mathbf{\Lambda}_{\text{int}}(\rho)$ via linear interpolation:

$$\mathbf{\Lambda}_{\text{int}}(\rho) = (1 - \rho)\mathbf{\Lambda}_1 + \rho\mathbf{\Lambda}_2 \quad (8.22)$$

6. Generate the interpolated system matrix in original coordinates:

$$\mathbf{A}_{\text{int}}(\rho) = \mathbf{T}_{\text{int}}(\rho)\mathbf{\Lambda}_{\text{int}}(\rho)(\mathbf{T}_{\text{int}}(\rho))^{-1}. \quad (8.23)$$

This algorithm can directly be implemented for systems up to order 3 via the MATLAB[®] Gable toolbox [22]. Fig. 8.2 illustrates the interpolation of an oscillatory eigenplane (eigenbivector). Note that the orientation of the rotation is preserved as an inherent feature of the GA representation. Numeric tools to tackle arbitrary-order systems and multi-parameter interpolation are subject of future research. Also, the task of obtaining optimal, possibly constrained realizations of an interpolation of system input and output matrices \mathbf{B} and \mathbf{C} require further studies.

8.4.3 An onset for the interpolation of the \mathbf{B} and \mathbf{C} matrices

Several sensible onsets to interpolate the \mathbf{B} and \mathbf{C} -matrices seem possible:

- One could argue that element-wise matrix interpolation is sensible because the proposed system matrix interpolation onset yields the system in coordinates that may be interpreted as original, physical coordinates.
- Alternatively, also along the lines of the original coordinates - argument, one can attempt to obtain interpolated \mathbf{B} and \mathbf{C} matrices in the modal coordinates and utilize the transformation obtained by the algorithm at the interpolation parameter value to transform the resulting matrices back to the coordinates of the final interpolated system.

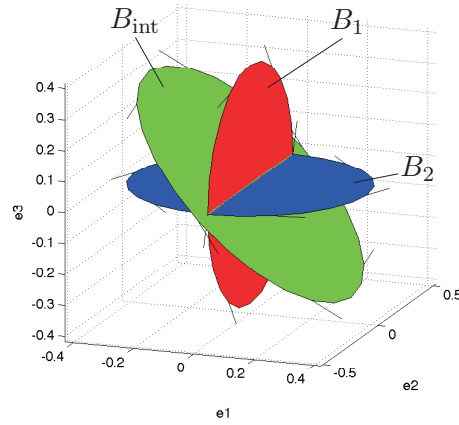


Figure 8.2: Geometric Algebra concepts help to interpolate oscillatory modes of systems meaningfully; the orientation information is inherently considered (taken from [150]).

- Other onsets could view the system as a stable coprime factorization and treat its numerator and denominator dynamics both by the interpolation onset presented above for the system matrix.

The second onset will be detailed in the following. Let the state-space matrices \mathbf{A}_j , \mathbf{B}_j , \mathbf{C}_j , \mathbf{D}_j of system $j = 1$ corresponding to $\rho = 0$ and system $j = 2$ ($\rho = 1$) and the transformation matrices to modal form \mathbf{T}_j be given, such that

$$\mathbf{A}_{j,m} = \mathbf{T}_j^{-1} \mathbf{A}_j \mathbf{T}_j \quad (8.24)$$

$$\mathbf{B}_{j,m} = \mathbf{T}_j^{-1} \mathbf{B}_j \quad (8.25)$$

$$\mathbf{C}_{j,m} = \mathbf{C}_j \mathbf{T}_j \quad (8.26)$$

$$\mathbf{D}_{j,m} = \mathbf{D}_j \quad (8.27)$$

yields the modal (or complex-diagonal) forms of the systems. Then, with the matrix $\mathbf{T}_{\text{int}}(\rho)$ obtained by the Geometric-Algebra-based system interpolation onset above, the interpolated input and output matrices according to this interpolation onset are given by

$$\mathbf{B}_{\text{int}}(\rho) = \mathbf{T}_{\text{int}}(\rho) [(1 - \rho) \mathbf{B}_{1,m} + \rho \mathbf{B}_{2,m}] \quad (8.28)$$

$$= \mathbf{T}_{\text{int}}(\rho) [(1 - \rho) \mathbf{T}_1^{-1} \mathbf{B}_1 + \rho \mathbf{T}_2^{-1} \mathbf{B}_2] \quad (8.29)$$

$$\mathbf{C}_{\text{int}}(\rho) = [(1 - \rho) \mathbf{C}_{1,m} + \rho \mathbf{C}_{2,m}] \mathbf{T}_{\text{int}}^{-1}(\rho) \quad (8.30)$$

$$= [(1 - \rho) \mathbf{C}_1 \mathbf{T}_1 + \rho \mathbf{C}_2 \mathbf{T}_2] \mathbf{T}_{\text{int}}^{-1}(\rho). \quad (8.31)$$

8.5 Example

This section gives an academic example to illustrate the effectiveness of modal interpolation. Consider the two 4th order LTI SISO state-space systems \mathbf{G}_a and \mathbf{G}_b , associated to the parameter values $\rho = 0$ and $\rho = 1$, respectively:

$$\mathbf{G}_a = \left[\begin{array}{cccc|c} 26.7526 & 6.4234 & 18.3248 & -0.4283 & 0.6478 \\ -38.5609 & -15.1009 & -17.0941 & 1.3578 & -0.3176 \\ -32.9193 & -4.7152 & -26.2832 & -0.5044 & 1.7690 \\ 10.2865 & 1.5569 & 10.5580 & -3.3685 & 1.5106 \\ \hline 0.6737 & -0.6691 & -0.4003 & -0.6718 & 0.0 \end{array} \right], \quad (8.32)$$

$$\mathbf{G}_b = \left[\begin{array}{cccc|c} 78.4360 & 49.3806 & 36.1644 & -25.4122 & 0.1640 \\ -66.8822 & -45.0610 & -22.4980 & 18.9232 & -0.2828 \\ -57.5637 & -26.9409 & -37.0469 & 12.7819 & 1.1522 \\ 36.5198 & 24.4588 & 20.4721 & -19.1281 & -1.1465 \\ \hline 0.5756 & -0.7781 & -1.0636 & 0.5530 & 0.0 \end{array} \right]. \quad (8.33)$$

These plants have poles at the following locations:

$$\boldsymbol{\lambda}_a^T = \{-10, -2, -3 + 4j, -3 - 4j\}, \quad (8.34)$$

$$\boldsymbol{\lambda}_b^T = \{-15, -2.4, -2.7 + 12j, -2.7 - 12j\}. \quad (8.35)$$

The following four interpolation onsets will be compared:

1. direct, element-wise interpolation of the original state-space matrices,
2. controllability-companion-form interpolation,
3. modal interpolation (mode order as in (8.34)–(8.35)) with the Geometric Algebra-based algorithm and element-wise interpolation of the \mathbf{B} and \mathbf{C} -matrices, and
4. modal interpolation with the Geometric Algebra-based algorithm and the second onset in Sec. 8.4.3.

The controllability companion forms of the systems are

$$\mathbf{G}_{a,\text{ccf}} = \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -500 & -420 & -117 & -18 & 1 \\ \hline 3517.5 & 1090.1 & 58.317 & -1.0741 & 0 \end{array} \right], \quad (8.36)$$

$$\mathbf{G}_{b,\text{ccf}} = \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -5446.4 & -2826.8 & -281.25 & -22.80 & 1 \\ \hline 16429 & 3241.2 & 128.15 & -1.545 & 0 \end{array} \right]. \quad (8.37)$$

Figure 8.3 shows the pole paths during the interpolation. The onsets based on element-wise interpolation and especially the one based on companion-form interpolation show significant deviations from the linear pole interpolation obtained by the modal onset. Figure 8.4 shows the resulting DC gains over the interpolation parameter. It can be seen that the modal interpolation approaches, particularly onset 4, yield a DC gain which depends almost linearly on the interpolation parameter. In contrast, the other approaches deviate significantly. Thereby it is notable that the companion-form interpolation remains within the DC gain limits of the vertex plants, whereas the element-wise interpolation leaves this regime quickly and shows strongly altered behavior.

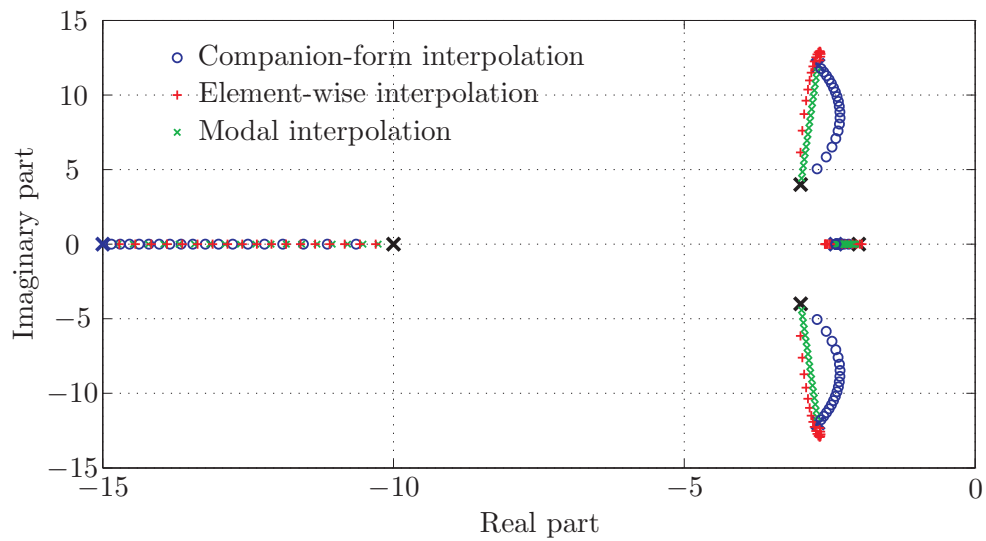


Figure 8.3: Interpolation pole paths of 4th order systems in Sec. 8.5

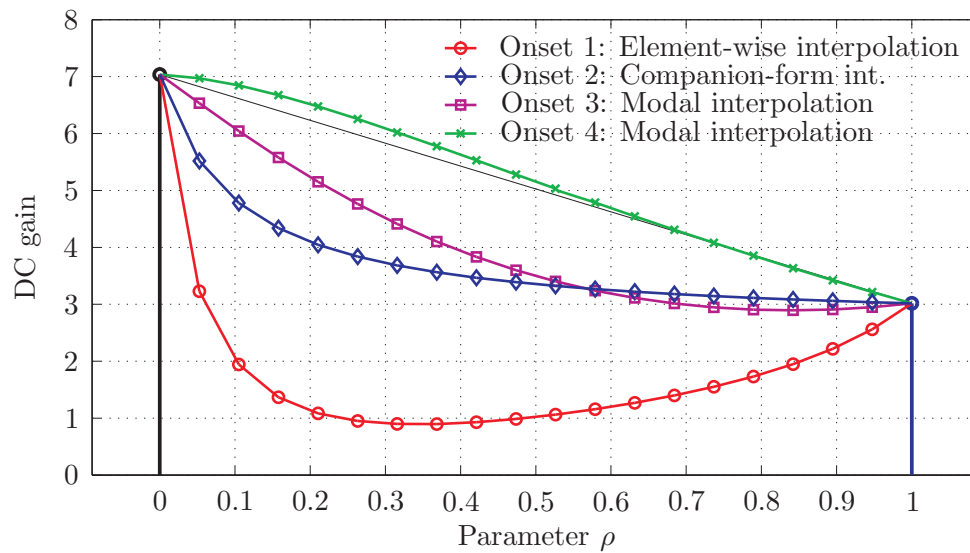


Figure 8.4: Interpolation DC gains of 4th order systems in Sec. 8.5

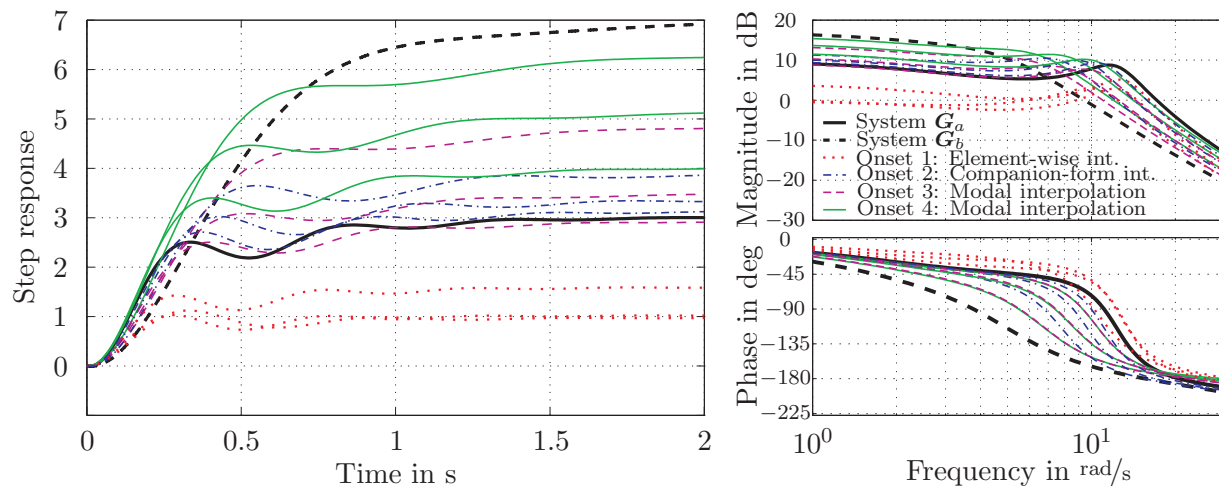


Figure 8.5: Interpolation step and Bode plots of 4th order systems in Sec. 8.5

Finally, the step and Bode response plots in Fig. 8.5, evaluated for 3 intermediate models only, illustrate these observations further. It is evident that the step responses of the vertex plants differ significantly in dominant behavior and DC gain, and a significant phase shift is seen in the frequency response. The element-wise interpolation (onset 1) yields highly distorted characteristics both in the time-domain as well as in the frequency domain, especially in terms of DC gain and phase which is problematic from a control design perspective. Onsets 2 and 3 may be considered acceptable, depending on the application. Their responses almost always lie within the bounds given by the vertex systems in terms of DC gain, magnitude, and phase. However, only onset 4 lies within these bounds and interpolates these properties almost linearly which can be considered beneficial. Finally note that these observations are based on plausibility in this example. An evaluation (and the related optimization) of an interpolation onset always depends on the definition of the desired structural properties. Methodologically, this gives rise to an optimization problem to minimize the deviations of the interpolation results from these properties, for example from a perfectly linear variation of the DC gain. These considerations potentially are subject of future research in the area.

8.6 Discussion

Summarizing, many gain scheduling methods need system interpolation to realize the parametrized control laws. In the case of an LFR-based interpolation of controllers in OBR, the parametrized plant matrices can be utilized in the observer realization, however, remaining interpolation tasks are those of the gain matrices and of the Youla parameter. They all need to provide stable interpolated systems or system matrices, so it is of interest to study the stability properties of system interpolation onsets.

The *companion form* interpolation procedure (element-wise interpolation of system matrices in controllability companion form) merely requires the systems to be fully state controllable and of the same dynamic order. High-order systems suffer from bad numeric conditioning of the standard companion form, which can be alleviated by a rescaled variant [29]. In the unscaled case, the companion form interpolation is equivalent to interpolating the characteristic polynomial coefficients. Stability cannot be guaranteed in general and a suitable stability criterion is given in Sec. 8.3.2. Its main relative advantage over the modal interpolation approach is implicit handling of multiple poles. Finally, note that the user cannot influence the parametrized

paths of the eigenmodes that are realized by the interpolation. Instead, one particular mode correspondence results from the formulation which may be unsuitable if interpolation stability fails.

The *modal form* interpolation operates in a mode-wise fashion. This requires the systems to be diagonalizable which is fulfilled if only distinct poles are present. The case of multiple poles becomes more involved and is not covered here. An additional requirement, besides the same dynamic order of all grid models, is mode compatibility: the same number of real- and of complex-conjugate poles are required for all grid models here. The transition across break-in or break-away points requires additional assumptions and case distinctions, which is a limitation of the modal approach. On the other hand, due to direct control on pole locations, the modal interpolation procedure provides stability guarantees that other approaches cannot, see Theorem 8.3.2. The correspondence of poles between grid models must be established explicitly through eigenvalue ordering. Contrary to the view in [29] it can be argued that this provides increased flexibility in the design of the interpolation.

Finally, the presented *Geometric Algebra (GA)*-based interpolation approach can be viewed as an extension to the modal interpolation approach. Thus it faces the same requirements and limitations, but also the advantages, in particular the stability property, of the modal approach. Additionally, the GA-based interpolation retains the original system coordinates and operates on its eigenstructure through geometric formalisms. This approach is subject of current and future research; first results on low-order systems are encouraging [150] and show the potential to optimally exploit geometric information in the grid models.

In conclusion, the interpolation of linear-dynamic grid models is generally not unique and various approaches exist. Here, three such approaches have been reviewed, discussed, and assessed, specifically in terms of stability of the interpolated system: The companion form interpolation corresponds to characteristic polynomial coefficient interpolation. The modal form interpolation interpolates the grid models mode-wise and provides interesting stability guarantees. Geometric Algebra interpolation is a novel extension to modal interpolation. The latter provides meaningful interpolation of the systems in its original coordinates using geometric relations. First results are promising, however, numeric tools and theoretical aspects are subject of future research.

Chapter 9

Summary and Outlook

9.1 Summary of the Results and Contributions

This work presents some central challenges of modern complex multivariable control designs in the field of flight control design for large flexible passenger aircraft. Novel contributions are developed in four key areas: convex control design, design parameter optimization, controller order reduction, and system interpolation. These are preceded by a comprehensive introduction of fundamental modeling, optimization, and robust control concepts as well as the state of the art in optimal and robust control design methods (Chapters 1–3).

The novel formulations are applied in numerous case studies which mainly address design tasks in the lateral control design of integrated flight control laws for a large, flexible BWB passenger aircraft. The corresponding aircraft models are introduced in Chap. 4, followed by the core chapters which detail the novel findings:

9.1.1 Extensions to Convex Control Design

Chapter 5 reports on the development of a novel, high-level optimization framework to formulate and carry out convex control design tasks in MATLAB[®]. Time- and frequency-domain objectives and constraints as well as tools for both feedback and feed-forward control design are implemented. Effective heuristic algorithms for an efficient, adaptive formulation of large-scale problems are developed. The framework's object-oriented structure is discussed and detailed in Appendix Chap. D.

The problem of strong stabilization, that is to obtain a stable and stabilizing feedback controller, is reviewed in the present context of convex control design. Suitable convex constraints which approximate the originally non-convex strong stabilization constraint are discussed and formulated. Finally, a convex feedback control design of the integrated lateral flight control for the considered BWB aircraft is carried out successfully.

Next, an onset to perform robust scheduled feed-forward control design with multiple models is investigated. A second case study shows the maneuver load alleviation potential of such design for the BWB aircraft. The successful completion of these demanding, high-dimensional case studies clearly demonstrates the efficiency and performance of the developed methods and the optimization framework as a design tool.

9.1.2 Design Parameter Optimization

In a more general context, a design parameter optimization framework is designed and implemented in Chap. 6. The idea is to automate the tedious parameter tuning of complex control design tasks based on the fulfillment of control goals in validation results. A genetic algorithm

is utilized to perform the actual optimization, whereas the evaluation of its fitness function includes the automated control design, validation, and goal attainment evaluation steps.

In two case studies, this onset is applied to optimize control designs for the BWB aircraft as well as for a conventional flexible aircraft (which is taken from [162]). The proposed methodology enables the engineer to efficiently test numerous design variants. Also, it becomes possible to address specifications which otherwise cannot directly be incorporated in the utilized design methods, for example time-domain specifications in \mathcal{H}_∞ -based control design.

9.1.3 Controller Order Reduction

The important task of reducing control law complexity is addressed in Chap. 7. Frequency-weighted balanced state reduction forms the basis for two novel formulations for controller order reduction. Firstly, a formulation is found which preserves nominal performance in the standard robust control feedback interconnection architecture by a variational derivation. Secondly, a recent result for robust controller order reduction is extended to the frequency-weighted case. Both novel methods show significantly improved reduction performance over the unweighted balanced reduction method at a flight control case study and at a publicly available robust control example taken from literature.

9.1.4 A Novel System Interpolation Onset

Chapter 8 focuses on control law scheduling, specifically on the interpolation of linear-dynamic systems. Motivated by LFR-based scheduling onsets, several methods for the interpolation of linear dynamic systems are reviewed with respect to their stability-preserving properties. Stability results for interpolation of systems in controllability companion form as well as in modal form are established. Moreover, a novel modal state-space interpolation method is sketched which exploits Geometric Algebra concepts to consistently treat geometric relations in the interpolation onset. A nontrivial academic example shows promising results for the latter onset, however further investigations and developments are necessary to explore its potential.

9.1.5 Results: Lateral Control for Large Flexible BWB Aircraft

In this thesis, various control design onsets for lateral control have been investigated at the considered model of a large flexible BWB passenger aircraft. An integrated BWB aircraft model including its rigid-body flight mechanics and flexible dynamics as well as their aeroelastic coupling formed the starting point for these investigations.

Based on the aircraft model preconditioned by a low-order initial controller which performs partial eigenstructure assignment, Sec. 5.3 shows the design of an optimized feedback controller by convex control design. Time-domain and frequency-domain constraints and objectives are considered and a stable controller could be obtained via the proposed technique to approximate this non-convex constraint by a convex one. The design results in a stable controller which preserves favorable properties of the initial control law (decoupling, rigid-body dynamics response shaping) while improving disturbance rejection and decoupling properties.

Building on these results, a robust, scheduled feed-forward reference command shaper is designed with the same convex control methodology in Sec. 5.5. Again, time-domain and frequency-domain constraints and objectives have directly been considered, thus ensuring that the aircraft response satisfies tight time-domain specifications in all considered parameter cases. The optimization's main objective, however, was the minimization of structural loads in the aircraft due to reference roll maneuvers. Two feed-forward architectures, both in a non-scheduled and in a scheduled setting (on the fuel filling parameter) have been compared. Compared to static decoupling with feedback control only, the loads reduction is significant for all considered design

variants. Moreover, the benefit of scheduling for the fuel filling parameter turned out to be significant.

In Sec. 6.4, the developed design parameter optimization framework is utilized to optimize a different lateral feedback control law design onset for the BWB aircraft. The control law is constructed following a multi-loop control architecture from a robust inner loop and PI SISO control loops. The design parameters to be optimized include both parameters of the weighting functions in the robust design task as well as the PI controller parameters. Although not all of the stringent control specifications can be fulfilled eventually in the followed one-degree-of-freedom setting, the case study shows significant improvement in difficult time-domain goals compared to a reasonably-tuned manual design.

9.2 An Outlook on Potential Future Research

The development of the listed contributions unveiled several issues and ideas which, from the author's point of view, deserve attention in future research:

Multi-model Youla parametrization The Youla parametrization (see Sec. 3.6.3) is formulated with respect to a nominal plant which prevents the use of direct multi-model onsets in convex feedback design. However, if perturbations of the plant or of the controller from an initial point are small, first-order approximations are valid and can robustify and optimize the convex design while retaining all its advantages. Tests of a robustified optimization of time-domain objectives has shown to work well for moderate plant perturbations.

Strong Stabilization Problem as BMI problem An alternative to approximate the strong stabilization constraint is to solve the resulting BMI problem directly using recent dedicated BMI solvers (see the related discussion in Sec. 5.2.2). However, an efficient formulation and the numeric properties and complexity of this onset needs to be investigated.

$Q - \mu$ -synthesis with Adaptive Constraint Refinement The $Q - \mu$ -synthesis algorithm in Sec. 3.6.6 (and variants of it) can directly be implemented in the developed control design framework. Moreover, the adaptive constraint refinement procedure bears high potential to efficiently solve this robust control design problem.

FWBR with SVD-based direction exploitation in weighting preparation The utilization of multivariable frequency weightings in FWBR is expected to further improve reduction performance, see Sec. 7.2. Moreover, the performance of the FWBR approach should be compared to that of other frequency-weighted reduction approaches, such as frequency-weighted Hankel approximation [49].

Optimal Geometric Algebra-interpolation of B and C matrices As denoted in Sec. 8.4.3, optimality criteria need to be defined to investigate and optimize interpolation laws with respect to these criteria.

Mode Matching / Tracking in Modal Interpolation Methods The tasks of mode matching between different plant linearizations and, ultimately, that of relating physically corresponding modes across parameter variation, require physical insight. However, supporting tools for semi-automated mode tracking are needed to perform this matching efficiently. For one onset which tracks the pole motion from open to closed loop in an element-wise matrix interpolation by a continuation-based method, see [109].

Closed-loop Interpolation Stability Criteria As outlined in Sec. 8.2.1 in the case of an LFR-based interpolation of compensators in OBR, for a stable interpolated closed loop

the interpolated Youla parameter and two additional matrix interpolation expressions need to be stable. If their stability can be guaranteed a priori, scheduling laws of this type can guarantee closed-loop stability.

Part III
Appendices

Appendix A

Selected Linear Algebra Concepts and Tools

A.1 Vector and Function Spaces

Vector and matrix calculus are fundamental requirements in systems theory and, in particular, in control theory. In the following several formal definitions of useful terms are given. The reader is referred to graduate textbooks such as [120] for a thorough treatment of various linear algebra topics. For a well-readable undergraduate text with a novel perspective, covering both standard linear algebra and geometric algebra fundamentals, refer to [92].

Definition A.1.1: Field

A field \mathbb{F} is a set \mathbf{F} , together with an *addition* $+$ and a *multiplication* \cdot operation, such that $\mathbb{F} := \langle \mathbf{F}, +, \cdot \rangle$ is a commutative group and the following axioms hold for all $a, b, c \in \mathbb{F}$:

F1.: $a + b \in \mathbb{F}, \quad a \cdot b \in \mathbb{F}$	Closure
F2.: $a + (b + c) = (a + b) + c, \quad a \cdot (b \cdot c) = (a \cdot b) \cdot c$	Associativity
F3.: $a + b = b + a, \quad a \cdot b = b \cdot a$	Commutativity
F4.1: $\exists 0 \in \mathbb{F} : 0 + a = a$	Additive identity (0 is called the null element)
F4.2: $\exists 1 \in \mathbb{F} : 1 \cdot a = a$	Multiplicative identity (1 is called the one element)
F5.1: $\exists (-a) : a + (-a) = 0$	Additive inverse
F5.2: $\forall a \in \mathbb{F} \setminus \{0\} : \exists (a^{-1}) : a \cdot (a^{-1}) = 1$	Multiplicative inverse
F6.: $a \cdot (b + c) = a \cdot b + a \cdot c$	Distributivity of multiplication over addition

Example A.1.1: Common fields

Commonly used fields are the field of real numbers, denoted \mathbb{R} , and the field of complex numbers \mathbb{C} with their usual addition and multiplication definitions. All axiomatic properties given above are satisfied as one can directly verify. Also, the set of rational numbers \mathbb{Q} with usual addition and multiplication is a field.

Definition A.1.2: Vector space

A vector space \mathbf{V} is a set or collection of objects called vectors (based on Def. 2.1 in [92]). Two fundamental operations are defined on \mathbf{V} : the scalar multiplication $a\mathbf{v}$ and the vector addition $\mathbf{v} + \mathbf{w}$. There exist a zero vector denoted by $\mathbf{0} \in \mathbf{V}$ (additive identity) and a scalar multiplicative identity denoted by 1. The following defining axioms must hold for all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{V}$ and all scalars $a, b \in \mathbb{F}$ (here, mainly real and complex vector spaces are considered, $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$, respectively):

V1.: $a\mathbf{v} \in \mathbf{V}, \mathbf{v} + \mathbf{w} \in \mathbf{V}$	Closedness under scalar multiplication and vector addition
V2.: $\mathbf{v} + \mathbf{w} = \mathbf{w} + \mathbf{v}$	Vector addition is commutative
V3.: $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$	Vector addition is associative
V4.: $\mathbf{v} + \mathbf{0} = \mathbf{v}$	$\mathbf{0}$ is the additive identity
V5.: $0\mathbf{v} = \mathbf{0}$	
V6.: $1\mathbf{v} = \mathbf{v}$	1 is the multiplicative identity
V7.: $a(b\mathbf{v}) = (ab)\mathbf{v}$	
V8.: $a(\mathbf{v} + \mathbf{w}) = a\mathbf{v} + a\mathbf{w}$	Scalar multiplication distributes over vector addition
V9.: $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$	

Example A.1.2: Common vector spaces

Commonly used vector spaces are the n -vector space over the real numbers, $\mathbb{R}^n = \{\mathbf{x} : x_i \in \mathbb{R}, i = 1, \dots, n\}$ or over the complex numbers \mathbb{C}^n . Also the set of polynomial functions $f(x) = r_0 + r_1x + \dots + r_nx^n$ with $r_0, \dots, r_n \in \mathbb{F}$, also called the polynomial ring $F[x]$ can be interpreted as a vector space over \mathbb{F}^n whose elements are the parameter vectors $\mathbf{r} \in \mathbb{F}^n$. There also exist infinite-dimensional vector spaces which arise particularly in the form of topological vector spaces such as Banach or Hilbert spaces (see [174] for some of their properties and their linkage to signal and system norms).

Definition A.1.3: Subspace

Let \mathbf{U} be a set of vectors in a vector space \mathbf{V} . If \mathbf{U} , together with the same scalar multiplication and vector addition operations as \mathbf{V} , is a vector space itself (that is, if all axioms V1-V9 in Def. A.1.2 hold for \mathbf{U}), then \mathbf{U} is a subspace of \mathbf{V} . [92]

A.2 Some Matrix Properties

Definition A.2.1: Range, Null Space & Rank

The Range $\text{range}(\mathbf{A})$ of an $(m \times n)$ -matrix \mathbf{A} is defined by

$$\text{range}(\mathbf{A}) = \{\mathbf{y} \in \mathbb{R}^m : \mathbf{y} = \mathbf{A}\mathbf{x} \text{ for some } \mathbf{x} \in \mathbb{R}^n\}, \quad (\text{A.1})$$

the Null space $\text{null}(\mathbf{A})$ of a matrix \mathbf{A} is defined by

$$\text{null}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{0}\}. \quad (\text{A.2})$$

For the column partitioning $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n]$,

$$\text{range}(\mathbf{A}) = \text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_n\} \quad (\text{A.3})$$

holds. The rank of \mathbf{A} is defined by

$$\text{rank}(\mathbf{A}) = \dim(\text{range}(\mathbf{A})). \quad (\text{A.4})$$

Important properties are $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^T)$ and $\dim(\text{null}(\mathbf{A})) + \text{rank}(\mathbf{A}) = n$. [44]

Definition A.2.2: Orthogonal matrices

A real matrix $\mathbf{Q} \in \mathbb{R}^{(n \times n)}$ is called an orthogonal matrix if it fulfills $\mathbf{Q}^T \mathbf{Q} = \mathbf{Q} \mathbf{Q}^T = \mathbf{I}_n$, where $(\cdot)^T$ denotes the matrix transpose.

Definition A.2.3: Unitary matrices

A complex matrix $\mathbf{Q} \in \mathbb{C}^{(n \times n)}$ is called a unitary matrix if it fulfills $\mathbf{Q}^H \mathbf{Q} = \mathbf{Q} \mathbf{Q}^H = \mathbf{I}_n$, where $(\cdot)^H$ denotes the Hermitian (complex-conjugate) matrix transpose.

Unitary matrices over the complex field correspond to orthogonal matrices over the real field. Moreover, a transformation by an orthogonal or unitary matrix preserves the 2-norm (see Def. A.4.2 and (A.12)).

A.3 Singular Value Decomposition (SVD)

The singular value decomposition (SVD) of a real or complex matrix is one of the most important matrix factorizations [44]:

Definition A.3.1: Singular Value Decomposition (SVD)

The SVD of an $(m \times n)$ real or complex matrix \mathbf{A} is

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\text{H}} \quad (\text{A.5})$$

$$\mathbf{\Sigma} = \begin{cases} \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \end{bmatrix} & \text{if } m > n \\ \mathbf{\Sigma}_1 & \text{if } m = n \\ \begin{bmatrix} \mathbf{\Sigma}_1 \\ \mathbf{0} \end{bmatrix} & \text{if } m < n \end{cases}, \quad \mathbf{\Sigma}_1 = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_p \end{bmatrix}, \quad p = \min(m, n) \quad (\text{A.6})$$

Thereby, \mathbf{U} and \mathbf{V} are $(m \times m)$ and $(n \times n)$ unitary matrices, respectively, and $\mathbf{\Sigma}_1$ is a $(p \times p)$ diagonal matrix whose diagonal entries $\sigma_1 \geq \dots \geq \sigma_p \geq 0$ are called the singular values. The $(m \times n)$ matrix $\mathbf{\Sigma}$ is of the same dimensions as \mathbf{A} and is obtained by appropriate stacking of $\mathbf{\Sigma}_1$ with a zero matrix. The corresponding column vectors \mathbf{u}_i of \mathbf{U} and row vectors \mathbf{v}_i^{H} of \mathbf{V}^{H} are called (left respectively right) singular vectors.

The SVD is an extremely useful tool in numerical linear algebra because the singular values σ_i and the associated singular vectors $\mathbf{u}_i, \mathbf{v}_i^{\text{H}}$ condense vital structure information on the matrix. Some examples and interpretations are given in the following.

Example A.3.1: Lower-rank approximations via SVD

A 2-norm optimal approximation of \mathbf{A} of reduced rank $k < p$ can be constructed by truncating the last $p - k$ singular values and the associated columns in \mathbf{U} and rows in \mathbf{V}^{H} , respectively [44]. The reduced-rank matrix \mathbf{A}_k ($\text{rank}(\mathbf{A}_k) = k$) is obtained by¹

$$\mathbf{A}_k = \mathbf{U}(:, 1:k)\mathbf{\Sigma}(1:k, 1:k)(\mathbf{V}(:, 1:k))^{\text{H}}, \quad (\text{A.7})$$

and this matrix fulfills

$$\min_{\text{rank}(\mathbf{B})=k} \|\mathbf{A} - \mathbf{B}\|_2 = \|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1} \quad (\text{A.8})$$

where \mathbf{B} is a rank- k matrix.

In the context of linear dynamic systems, a related methodology is the well-known balanced system reduction to perform order reduction (see Sec. 2.2). The singular values can also be utilized to define a system's \mathcal{H}_∞ norm (see Sec. A.4), and singular value plots (see [143]) represent a generalization of SISO Bode magnitude plots to the multivariate case.

¹The MATLAB[®] notation for row/column selection of a matrix is utilized: $\mathbf{X}(1:2, 3:5)$ selects rows 1 to 2 and columns 3 to 5 of a matrix \mathbf{X} , a single colon $:$ selects the entire corresponding dimension.

A.4 Norms

Norms are utilized in order to measure the size of an algebraic object. These objects can be scalars, vectors, matrices, signals in the time- or frequency domains, or linear operators such as transfer functions. Their norms all have common properties – the defining properties of a norm. However, a norm’s interpretation varies with the context of the considered object, and the notation used in literature strongly varies. In this work, the common conventions of [143] are utilized. For a detailed discussion on norms in system theory, the reader is referred to [143], [174], and [44]. The following definitions are collected and summarized from these sources.

Definition A.4.1: Norm

A *norm* of \mathbf{x} (where \mathbf{x} is an element of a vector space \mathbf{V} over a field \mathbb{F}) is a real number, denoted $\|\mathbf{x}\|$, that satisfies the following properties:

- | | |
|--|---|
| N1.: $\ \mathbf{x}\ \geq 0$ | Non-negativity |
| N2.: $\ \mathbf{x}\ = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$ | Positivity (For <i>semi-norms</i> , only $\mathbf{x} = \mathbf{0} \Rightarrow \ \mathbf{x}\ = 0$ is true.) |
| N3.: $\ \alpha\mathbf{x}\ = \alpha \ \mathbf{x}\ $ for all scalars $\alpha \in \mathbb{F}$ | Homogeneity |
| N4.: $\ \mathbf{x}_1 + \mathbf{x}_2\ \leq \ \mathbf{x}_1\ + \ \mathbf{x}_2\ $ | Triangle inequality |

A.4.1 Vector- and Matrix Norms

If \mathbf{x} in Def. A.4.1 is a constant vector or matrix, the associated norms are called *spatial* norms. Finite-dimensional vectors and matrices have finite-dimensional norms. Details on these norms are given in [44] and [68]; the following formulations are taken from [143].

Definition A.4.2: Vector p -norm

For a vector $\mathbf{x} = [x_1, \dots, x_m]^T \in \mathbb{F}^m$ with m elements, the vector p -norm is defined as:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^m |x_i|^p \right)^{\frac{1}{p}}, \quad p \geq 1. \quad (\text{A.9})$$

A shortlist of common vector norms is given in the following:

- | | | |
|--|--------------|--|
| Vector 1-norm (sum norm): | $p = 1$ | $\ \mathbf{x}\ _1 = \sum_{i=1}^m x_i $ |
| Vector 2-norm (Euclidian norm): | $p = 2$ | $\ \mathbf{x}\ _2 = \sqrt{\sum_{i=1}^m x_i ^2} = \sqrt{\bar{\mathbf{x}}^T \mathbf{x}}$, where $\bar{\mathbf{x}}$ denotes the conjugate if $\mathbb{F} = \mathbb{C}$ (thus, $\bar{\mathbf{x}}^T = \mathbf{x}^H$ is the Hermitian transpose of \mathbf{x}). |
| Vector ∞-norm (max norm): | $p = \infty$ | $\ \mathbf{x}\ _\infty = \max_{i=1}^m x_i $ |

Definition A.4.3: Matrix norm, Generalized matrix norm

A norm of a constant $(m \times n)$ -matrix $\mathbf{A} \in \mathbb{F}^{m \times n}$ is called a *matrix norm* if it fulfills the four axioms N1–N4 in Def. A.4.1 and additionally the *multiplicative property*:

$$\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\| \quad (\text{A.10})$$

for all \mathbf{A}, \mathbf{B} for which the product \mathbf{AB} exists (i.e., which are of compatible inner dimensions). If only the four axioms N1–N4 in Def. A.4.1 are fulfilled, the norm is called *generalized matrix norm* and has weaker properties.

A shortlist of common matrix norms is given in the following for a matrix $\mathbf{A} = [a_{ij}] \in \mathbb{C}^{l \times k}$:

Sum matrix norm:

$$\|\mathbf{A}\|_{\text{sum}} = \sum_{i,j} |a_{ij}|$$

Frobenius matrix norm (Euclidean norm):

$$\|\mathbf{A}\|_{\text{F}} = \sqrt{\sum_{i,j} |a_{ij}|^2} = \sqrt{\text{trace}(\mathbf{A}^{\text{H}}\mathbf{A})}$$

where $(\cdot)^{\text{H}}$ denotes the Hermitian transpose.

Induced norms of linear operators, particularly of matrices, are of special interest:

Definition A.4.4: Induced norm

Given a constant matrix \mathbf{A} and the linear mapping $\mathbf{z} = \mathbf{A}\mathbf{w}$ for compatible vectors \mathbf{z}, \mathbf{w} , the induced p -norm $\|\cdot\|_p$ is defined as:

$$\|\mathbf{A}\|_{ip} = \max_{\mathbf{w} \neq \mathbf{0}} \frac{\|\mathbf{z}\|_p}{\|\mathbf{w}\|_p} = \max_{\mathbf{w} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{w}\|_p}{\|\mathbf{w}\|_p}. \quad (\text{A.11})$$

The induced norm is a matrix norm and thus satisfies the multiplicative property (A.10).

For the case $p = 2$, the **singular value norm** or **spectral norm** is obtained:

$$\|\mathbf{A}\|_{i2} = \bar{\sigma}(\mathbf{A}) = \sqrt{\rho(\mathbf{A}^{\text{H}}\mathbf{A})}, \quad (\text{A.12})$$

where $\bar{\sigma}(\cdot)$ is the largest singular value (see Sec. A.3) and $\rho(\cdot)$ is the spectral radius:

Definition A.4.5: Spectral radius

The spectral radius $\rho(\mathbf{A})$ of a matrix \mathbf{A} is the maximum of the eigenvalue magnitudes:

$$\rho(\mathbf{A}) = \max_i |\lambda_i(\mathbf{A})|. \quad (\text{A.13})$$

The spectral radius is not a norm, but represents an important lower bound on any matrix norm [143]:

$$\rho(\mathbf{A}) \leq \|\mathbf{A}\| \quad (\text{A.14})$$

A.4.2 Signal norms

The following description is based on the exposition given in [143]. Signal norms, also called temporal norms, are formulated for time-varying (or frequency-varying) signals $e(t)$. They are considered infinite-dimensional as they operate on infinite-dimensional function spaces. While different spatial norms on the same vector or matrix always yield similar results (they can only differ by at most a constant finite factor), choosing different temporal norms, on the contrary, can fundamentally change the norm value – for example, the same signal can have a well-defined, finite 2-norm, while its ∞ -norm is infinite, or vice-versa.

Signal norms are defined in two steps: firstly, evaluating a spatial norm at a given fixed time (or frequency), and secondly, summing up these spatial norms over all times (or frequencies). Typically, the same p -norms in both steps are utilized.

Definition A.4.6: Temporal p -norm, l_p -norm

Given a time-varying vector signal $\mathbf{z}(t) = [z_1(t), \dots, z_n(t)]$, its temporal p -norm or l_p -norm $\|\mathbf{z}(t)\|_p$ is defined by

$$\|\mathbf{z}(t)\|_p = \left(\int_{-\infty}^{\infty} \sum_{i=1}^n |z_i(\tau)|^p d\tau \right)^{\frac{1}{p}} \quad (\text{A.15})$$

A shortlist of common temporal norms of signals is given in the following:

1-norm in time (integral absolute error, IAE): $p = 1 \quad \|\mathbf{z}(t)\|_1 = \int_{-\infty}^{\infty} \sum_{i=1}^n |z_i(\tau)| d\tau$

2-norm in time (quadratic norm, integral square error, ISE, energy of the signal): $p = 2 \quad \|\mathbf{z}(t)\|_2 = \sqrt{\int_{-\infty}^{\infty} \sum_{i=1}^n |z_i(\tau)|^2 d\tau}$

∞ -norm in time (peak value in time): $p = \infty \quad \|\mathbf{z}(t)\|_{\infty} = \max_{\tau} (\max_{i=1}^n |z_i(\tau)|)$

Note that the commonly used signal RMS (root mean square) norm, also called power norm, is only a semi-norm (compare Def. A.4.1, N2 is not fulfilled – a nonzero signal may very well have an RMS value of zero):

$$\|\mathbf{z}(t)\|_{\text{pow}} = \|\mathbf{z}(t)\|_{\text{RMS}} = \lim_{T \rightarrow \infty} \sqrt{\frac{1}{2T} \int_{-T}^T \sum_{i=1}^n |z_i(\tau)|^2 d\tau} \quad (\text{A.16})$$

A.4.3 System norms

The concept of system norms is distinct from that of temporal norms. Given a linear dynamic system \mathbf{G} , system norms are typically defined as the worst-case temporal p -norm of the output signal of \mathbf{G} which is excited by an input signal of a specific class $\mathbf{u} \in \mathcal{U}$.

Consequently, many different system norms result from the potential combinations of input signal class and output signal temporal norm. In the following, the two most widely used system norms in control, the \mathcal{H}_2 and the \mathcal{H}_{∞} system norms, are defined and their specific features are stated from a control perspective. A thorough discussion on system norms is given in [143] and [174].

Definition A.4.7: \mathcal{H}_2 system norm

Given a causal, stable, strictly proper linear system $\mathbf{G}(s)$ as in (2.8) with $\mathbf{D} = \mathbf{0}$, its \mathcal{H}_2 norm for $s = j\omega, \omega \in \mathbb{R}$ is defined by the following equivalent definitions [143]:

$$\|\mathbf{G}\|_2 = \max_{\mathbf{w}(t) \text{ unit impulses}} \|\mathbf{z}(t)\|_2 = \|g(t)\|_2 = \sqrt{\sum_{i,j} \int_0^\infty |g_{ij}(\tau)|^2 d\tau} \quad (\text{A.17})$$

where the input $\mathbf{w}(t)$ is composed of unit impulses at each scalar channel, one at a time, and with an (ideally infinite) settling time to the next impulse so that the system output can settle to zero after each impulse. The alternative formulation utilizes the impulse response matrix $g(t) = [g_{ij}(t)]$ which can be directly interpreted as temporal norm over the matrix signal $g(t)$ (note that $g(t) = \mathbf{0}$ for $t < 0$ in causal systems).

If \mathbf{G} is unstable or not strictly proper, its \mathcal{H}_2 norm is infinite.

By Parseval's theorem,

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega, \quad (\text{A.18})$$

which relates the integrals of the time-domain signal $x(t)$ and its spectral representation $X(\omega)$, the \mathcal{H}_2 system norm can also be formulated as:

$$\|\mathbf{G}\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace}(\mathbf{G}(j\omega)^H \mathbf{G}(j\omega)) d\omega}, \quad (\text{A.19})$$

where the integrand is the squared Frobenius norm of $\mathbf{G}(j\omega)$.

For norm computation, given a state space representation $\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ (see Sec. 2.1.1), the following forms are beneficial:

$$\|\mathbf{G}\|_2 = \sqrt{\text{trace}(\mathbf{B}^T \mathbf{W}_o \mathbf{B})} = \sqrt{\text{trace}(\mathbf{C} \mathbf{W}_c \mathbf{C}^T)}, \quad (\text{A.20})$$

where \mathbf{W}_o and \mathbf{W}_c are the observability and the controllability Gramians, respectively [143]; and

$$\|\mathbf{G}\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \sum_i \sigma_i^2(\mathbf{G}(j\omega)) d\omega}, \quad (\text{A.21})$$

where $\sigma_i(\cdot)$ is the i th singular value of the $(k \times l)$ matrix argument with $i \in \{1, \dots, \min(k, l)\}$, see Sec. A.3.

Remarks & Interpretation:

- The \mathcal{H}_2 norm is **not** an induced norm and thus it does not fulfill the multiplicative property:

$$\|\mathbf{A}(s)\mathbf{B}(s)\|_2 \not\leq \|\mathbf{A}(s)\|_2 \|\mathbf{B}(s)\|_2 \quad (\text{A.22})$$

- Common interpretations of the \mathcal{H}_2 norm are:

average gain over all frequencies: the energy gain that the system produces for white noise input signals,

deterministic – impulse response norm: the output signal 2-norm resulting from sequentially applying unit impulses to each input, letting the output settle to zero after each impulse (see (A.17)),

stochastic: the expected RMS (root mean square) value of the output in response to white noise excitation, and

singular value interpretation: sum of squared singular values over all frequencies (“average direction, average frequency” [143], see also the conceptual illustration in Fig. A.1 (left))

- The \mathcal{H}_2 norm of a system which is not stable or not strictly proper is infinite. This is in accordance with the interpretations above, because the response of an unstable system to an arbitrary nonzero input signal is unbounded. If a system is stable but bi-proper, its impulse response is infinite at $t = 0$ and the 2-norm of the impulse response is infinite as well.²

Definition A.4.8: \mathcal{H}_∞ system norm

Given a proper, stable linear system $\mathbf{G}(s)$ as in (2.8), its \mathcal{H}_∞ system norm is defined as

$$\|\mathbf{G}\|_\infty = \max_{\mathbf{w}(t) \neq \mathbf{0}} \frac{\|\mathbf{z}(t)\|_2}{\|\mathbf{w}(t)\|_2} = \max_{\|\mathbf{w}(t)\|_2=1} \|\mathbf{z}(t)\|_2. \quad (\text{A.23})$$

It is evident from Def. A.4.4 that the \mathcal{H}_∞ system norm is an induced norm which is crucial for many applications. The multiplicative property (A.10) is central to \mathcal{H}_∞ -based system analysis and control design techniques.

For stable, proper $\mathbf{G}(s)$, another definition of $\|\mathbf{G}\|_\infty$ based on the singular values (see Sec. A.3) of $\mathbf{G}(s)$ is:

$$\|\mathbf{G}\|_\infty = \sup_{\omega \in \mathbb{R}} \bar{\sigma}(\mathbf{G}(j\omega)). \quad (\text{A.24})$$

Thus the \mathcal{H}_∞ norm is the peak of the maximum singular value magnitude over all frequencies.

For the numeric computation of the \mathcal{H}_∞ norm of a given system, an iterative bi-section procedure is typically used to approximate the \mathcal{H}_∞ norm up to a demanded precision. The following norm definition involves a Hamiltonian matrix \mathbf{H} formed from the system state space matrices. Assuming $\mathbf{G}(s)$ stable, then

$$\|\mathbf{G}\|_\infty = \inf_{\gamma > 0} \gamma : \mathbf{H}(\gamma) \text{ has no imaginary eigenvalues} \quad (\text{A.25})$$

holds, where $\mathbf{H}(\gamma) = \begin{bmatrix} \mathbf{A}_H(\gamma) & \mathbf{B}\mathbf{R}^{-1}(\gamma)\mathbf{B}^T \\ -\mathbf{C}^T(\mathbf{I} + \mathbf{D}\mathbf{R}^{-1}(\gamma)\mathbf{D}^T)^{-1}\mathbf{C} & -\mathbf{A}_H(\gamma) \end{bmatrix}$ with $\mathbf{A}_H(\gamma) = \mathbf{A} + \mathbf{B}\mathbf{R}^{-1}(\gamma)\mathbf{D}^T\mathbf{C}$, and $\mathbf{R}(\gamma) = \gamma^2\mathbf{I} - \mathbf{D}^T\mathbf{D}$.

If \mathbf{G} is not stable, its \mathcal{H}_∞ norm is infinite.

²Note that the MATLAB[®] command `impz` computes the impulse response of a given strictly proper LTI system. If the given system is bi-proper, however, the value at $t = 0$ is not infinite and thus differs from the true impulse response.

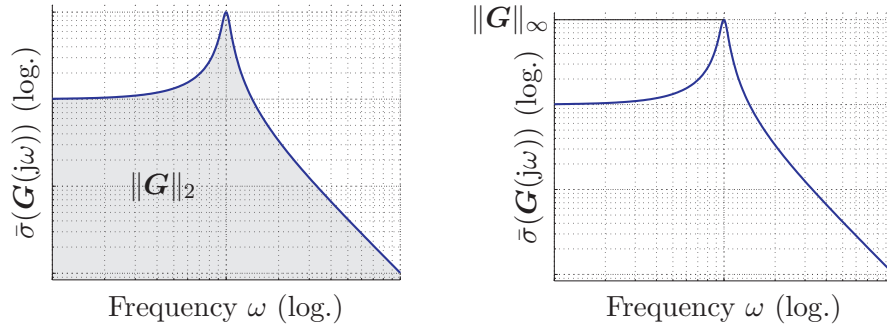


Figure A.1: Illustrations of \mathcal{H}_2 (left, conceptual only) and \mathcal{H}_∞ (right) system norms. Note, however, that the \mathcal{H}_2 norm is the integrated sum of all squared singular values σ_i over all frequencies (see (A.21)).

Remarks & Interpretations:

- One key property of the \mathcal{H}_∞ system norm is that it is an induced norm and thus fulfills the multiplicative property (A.10). This enables effective system manipulation by weighting functions which can be designed by exploiting this property, see Sec. 2.3.1.
- Common interpretations of the \mathcal{H}_∞ system norm are:
 - worst-case gain over all frequencies:** the maximum steady-state energy gain that the system produces for the sinusoidal inputs at any frequency over all possible input directions,
 - stochastic:** it is equal to the induced power (RMS) norm and enables statements on the expected values of stochastic signals, and
 - singular value interpretation:** peak of the largest singular value over all frequencies (“worst direction, worst frequency” [143], see Fig. A.1 (right))
- Note that the \mathcal{H}_∞ norm of a system which is not stable is infinite. In contrast, the singular values of such system may very well be finite for all frequencies. This is a point of caution for interpreting singular value plots. Moreover, this fact poses limitations to numeric control design approaches (see the according discussion on Strong Stabilization, Sec. 5.2).

A.5 Eigenvalues and Eigenvectors

The standard eigenvalue problem associated to a real or complex square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ or $\mathbf{A} \in \mathbb{C}^{n \times n}$ can be stated as:

$$\mathbf{A}\mathbf{u}_i = \mathbf{u}_i\lambda_i, \quad i = 1, \dots, n \quad (\text{A.26})$$

where $\mathbf{u}_i \in \mathbb{C}^n$, $\lambda_i \in \mathbb{C}$ are the i th right eigenvector and the associated eigenvalue, respectively.

Eigenvalue/eigenvector problems arise in a wide range of mathematical problems, and elaborate algorithms are available to solve them (QR factorization, for example, see [44]).

Each eigenvector defines a one-dimensional *invariant subspace* for \mathbf{A} , which means that the result of its pre-multiplication by \mathbf{A} is still within the same subspace. The result is scaled by the corresponding eigenvalue λ_i .

More generally, a subspace $\mathbf{S} \subseteq \mathbb{C}$ is invariant for \mathbf{A} iff

$$\mathbf{x} \in \mathbf{S} \Rightarrow \mathbf{A}\mathbf{x} \in \mathbf{S}. \quad (\text{A.27})$$

A.5.1 Characteristic Polynomial

The characteristic polynomial of a square matrix \mathbf{A} reads

$$p(t) := \det(t\mathbf{I} - \mathbf{A}) = \sum_{i=0}^n a_i t^i. \quad (\text{A.28})$$

Its n roots $\{\lambda_i : p(\lambda_i) = 0\}$ are just the eigenvalues of \mathbf{A} .

A.5.2 Theorem of Cayley-Hamilton

A square matrix $\mathbf{A} \in \mathbb{R}^{(n \times n)}$ or $\mathbf{A} \in \mathbb{C}^{(n \times n)}$ satisfies its own characteristic equation. That is to say, utilizing the characteristic polynomial's coefficients a_i , $i = 1, \dots, n$ from (A.28), the theorem states that

$$p(\mathbf{A}) := \sum_{i=1}^n a_i \mathbf{A}^i = 0. \quad (\text{A.29})$$

Various proofs of this theorem can be found in literature, notably a compact and elegant proof can be done via Geometric Algebra [66].

A.5.3 Multiple Eigenvalues/-vectors and their Computation

Rearranging (A.26) to $(\lambda\mathbf{I} - \mathbf{A})\mathbf{u}_i = \mathbf{0}$ for a known λ_i casts the problem of finding the associated eigenvectors of \mathbf{A} into the problem of finding the eigenvectors corresponding to the matrix $(\lambda\mathbf{I} - \mathbf{A})$ and its zero eigenvalue(s). This is equivalent with finding a basis for its null space (see Def. A.2.1).

Given an eigenvalue λ with algebraic multiplicity $r = 2$, a solution onset can be chosen as:

$$\mathbf{x} = c_1 \mathbf{u} e^{\lambda t} + c_2 \mathbf{v} t e^{\lambda t}. \quad (\text{A.30})$$

Differentiating this equation yields

$$\dot{\mathbf{x}} = \lambda c_1 \mathbf{u} e^{\lambda t} + c_2 (\mathbf{v} e^{\lambda t} + \lambda \mathbf{v} t e^{\lambda t}). \quad (\text{A.31})$$

This is still a linear system, so this system can be written as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} = \mathbf{A} (c_1 \mathbf{u} e^{\lambda t} + c_2 \mathbf{v} t e^{\lambda t}). \quad (\text{A.32})$$

Comparing the coefficients in $e^{\lambda t}$ and in $t e^{\lambda t}$, one can identify the associated eigenvalue equations:

$$e^{\lambda t} : \quad \lambda c_1 \mathbf{u} + c_2 \mathbf{v} = c_1 \mathbf{A}\mathbf{u} \quad (\text{A.33})$$

$$t e^{\lambda t} : \quad c_2 \lambda \mathbf{v} = c_2 \mathbf{A}\mathbf{v}. \quad (\text{A.34})$$

From (A.34) the relation

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v} \Rightarrow (\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0} \quad (\text{A.35})$$

is directly obtained, while from (A.33) it can be stated that

$$c_1 (\mathbf{A} - \lambda\mathbf{I})\mathbf{u} = c_2 \mathbf{v} \quad (\text{A.36})$$

holds. Pre-multiplying this with $(\mathbf{A} - \lambda\mathbf{I})$ yields

$$c_1 (\mathbf{A} - \lambda\mathbf{I})^2 \mathbf{u} = c_2 (\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}. \quad (\text{A.37})$$

As a result, it is evident that for any eigenvalue λ with algebraic multiplicity m a complete (m -dimensional) basis of generalized eigenvectors can be found as null $\left((\lambda\mathbf{I} - \mathbf{A})^{m-1} \right)$, see also [5]. The lower powers of $(\lambda\mathbf{I} - \mathbf{A})$ can have lower-dimensional nullspaces, which reflects information on the geometric multiplicities of λ that add up to its algebraic multiplicity m .

Appendix B

Observers & Observer-based Structures

This chapter presents the Luenberger state observer, utilized to reconstruct the states of a dynamic plant first. The Luenberger observer structure, equations, and properties are detailed, and its important use in the design of control systems, consisting of a state vector feedback gain and a state observer, is outlined. The separation principle allows to separate both design tasks and carry them out independently. Furthermore, the concept of the OBR of feedback compensators is presented. In OBR, a compensator consists of a state observer of the plant states, a static state feedback gain, and a remainder term which can be interpreted as Youla parameter. Given an arbitrary feedback compensator, transformation procedures to obtain an equivalent OBR are summarized and main properties and applications of this representation are sketched.

The chapter is structured as follows: Sec. B.1 summarizes the Luenberger observer, Sec. B.2 presents the state vector feedback gain / observer combination and the separation principle, and Sec. B.3 discusses the observer-based representation (OBR) of feedback compensators.

B.1 Luenberger observer

A state observer¹ is a dynamic system which reconstructs the states of a system, based on information on its inputs and outputs as well as on a (sufficiently good) model of its system dynamics. Back in 1964, Luenberger firstly formulated the concept of a state observer [90]. He later extended the formulation to multivariable systems in [88]. An introductory overview on observers and their properties is given in [89].

Figure B.1 shows a block diagram of a linear state observer \mathbf{K}_{obs} connected to a strictly-proper² linear dynamic system in state space representation $\mathbf{G} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{0} \end{array} \right]$. The estimated states are denoted by $\hat{\mathbf{x}}$ and the static observer gain is \mathbf{K}_f .

The basic observer equations are derived as follows, based on [143] and [132]. Let the state estimation error $\boldsymbol{\varepsilon}$ be defined by

$$\boldsymbol{\varepsilon} := \mathbf{x} - \hat{\mathbf{x}}. \quad (\text{B.1})$$

Choosing the linear relation

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{G}\mathbf{u} + \mathbf{K}_f\mathbf{y}, \quad (\text{B.2})$$

¹The term observer is used in a deterministic context, whereas the term estimator is used for the related concept in a stochastic problem setting. However, this distinction is not sharply done in literature and in a specific problem setting it is up to the control engineer which aspects are focused on.

²The observer can also be extended to the bi-proper case ($\mathbf{D} \neq \mathbf{0}$), see [174, pp. 63–65].

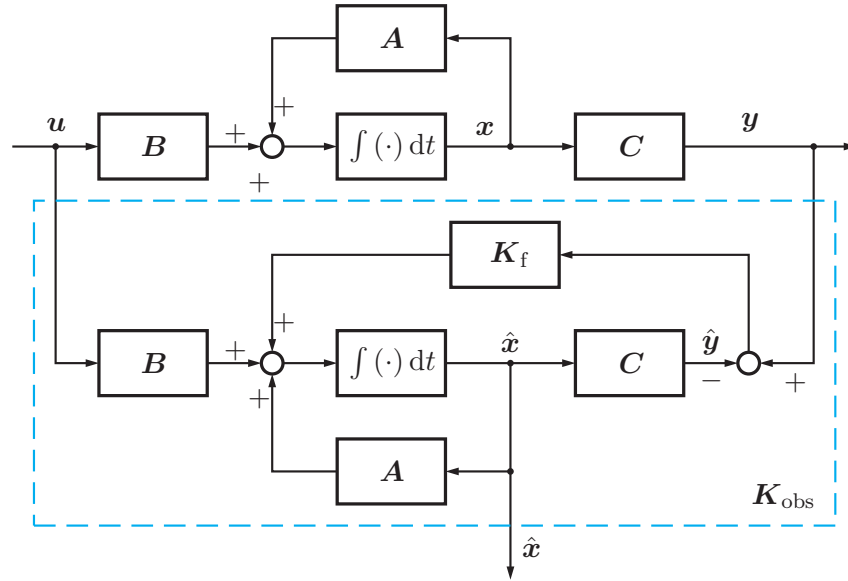


Figure B.1: Luenberger observer

for state estimation, the following error dynamics is obtained:

$$\dot{\varepsilon} = \mathbf{F}\varepsilon + (\mathbf{A} - \mathbf{K}_f\mathbf{C} - \mathbf{F})\mathbf{x} + (\mathbf{B} - \mathbf{G})\mathbf{u}. \quad (\text{B.3})$$

If $\mathbf{F} = \mathbf{A} - \mathbf{K}_f\mathbf{C}$ and $\mathbf{G} = \mathbf{B}$ hold, and if the real parts of the eigenvalues of \mathbf{F} are negative, the error dynamics is stable, $\hat{\mathbf{x}}$ converges to the plant state vector \mathbf{x} , and the observer equation

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}_f(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}) \quad (\text{B.4})$$

is obtained. This equation is realized in Fig. B.1. The design of an observer with this structure is methodologically equivalent (often also called dual) to state vector feedback control design for the plant.

One basic requirement for the observer as a dynamic system is stability. Moreover, certain dynamic requirements on the dynamics of the estimation error ε should be fulfilled. These are the same typical requirements as addressed in state vector feedback control design, and the problem structure (designing a static feedback gain) is the same. Thus, the same methods as for state vector feedback control design can be applied to design an observer.

The observer presented above is sometimes called full, full-state, or full-order observer because it reconstructs the entire state vector. It is itself a dynamic system \mathbf{K}_{obs} of order n – the same order as the plant (where the system matrix \mathbf{A} is $(n \times n)$). Many other variants of observers are available in the literature which specialize on specific system properties, some well-known variants are:

Reduced-order observer, minimum-order observer: When a subset of the states is available as measurements, the dynamic complexity of the observer can be reduced, see [89].

Unknown-input observer If one or more inputs of the system are unknown to the observer, they, too, can be reconstructed if certain existence conditions are fulfilled, see for example [48] and references therein.

B.2 State vector feedback control using a state observer

In many important control problem formulations, state vector feedback control laws of the form

$$u = -K_c x \tag{B.5}$$

are chosen. Well-known design problems following this onset include for example pole placement designs (eigenstructure assignment, see Sec. 3.6.2) or optimal control designs such as LQR design (see Sec. 3.4).

In actual physical systems, a commonly encountered property is that not all system states are available as measured signals. In such cases, one possibility to apply state vector feedback methods is to substitute the unknown state vector x in (B.5) by its estimate which is obtained by a state observer.

The corresponding block diagram is shown in Fig. B.2.

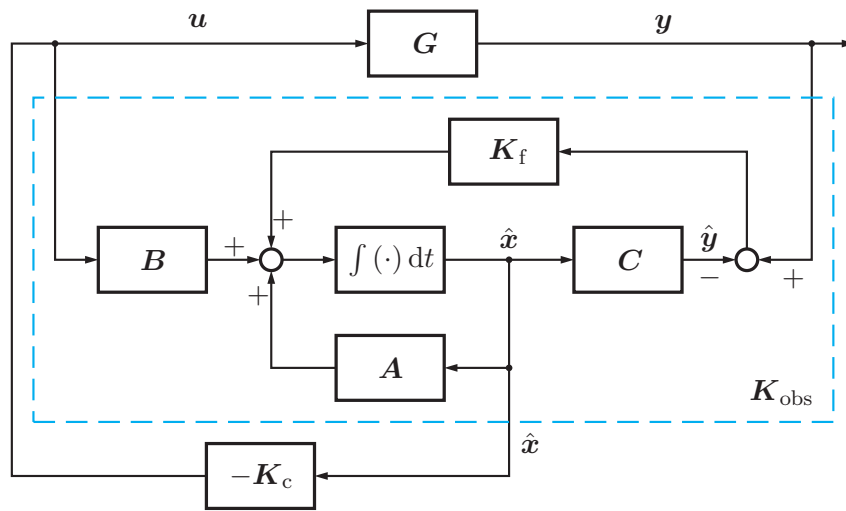


Figure B.2: Interconnection architecture: observer and static state vector feedback

B.2.1 Separation principle

In a state vector feedback loop with full state observer as in Fig. B.2, the so-called separation principle is an important property [143].

Definition B.2.1: Separation principle

Given a closed-loop system interconnection as in Fig. B.2 of a system with a state observer and a state vector feedback gain, the set of poles of the closed-loop system is the union of the poles of the state-feedback regulator dynamics $A - BK_c$ and the poles of the state observer dynamics $A - K_f C$.

This can be seen by formulating the closed-loop dynamics in terms of the plant states and the observation error. First note that the plant and observer system equations, the plant output

equation, and the control law are

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (\text{B.6})$$

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}_f(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}), \quad (\text{B.7})$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}, \quad (\text{B.8})$$

$$\mathbf{u} = -\mathbf{K}_c\hat{\mathbf{x}}. \quad (\text{B.9})$$

Eliminating \mathbf{u}, \mathbf{y} , and substituting $\mathbf{x} - \varepsilon$ for $\hat{\mathbf{x}}$ yields

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\varepsilon} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} - \mathbf{B}\mathbf{K}_c & \mathbf{B}\mathbf{K}_c \\ \mathbf{0} & \mathbf{A} - \mathbf{K}_f\mathbf{C} \end{bmatrix}}_{\mathbf{A}_{cl}} \begin{bmatrix} \mathbf{x} \\ \varepsilon \end{bmatrix}. \quad (\text{B.10})$$

Now consider the fact that the determinant of a partitioned matrix \mathbf{X} can be expressed as

$$\det \left(\left[\begin{array}{c|c} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \end{array} \right] \right) = \det(\mathbf{X}_{11}) \cdot \det(\mathbf{X}_{22} - \mathbf{X}_{21}\mathbf{X}_{11}^{-1}\mathbf{X}_{12}) \quad (\text{B.11})$$

whenever \mathbf{X}_{11} is invertible, see [110, Sec. 9.1.2]. Note in particular that if $\mathbf{X}_{21} = \mathbf{0}$ or $\mathbf{X}_{12} = \mathbf{0}$ holds, this simplifies to $\det(\mathbf{X}) = \det(\mathbf{X}_{11}) \cdot \det(\mathbf{X}_{22})$.

Applying this fact to \mathbf{A}_{cl} in (B.10) it is evident that

$$\det(\mathbf{A}_{cl}) = \det(\mathbf{A} - \mathbf{B}\mathbf{K}_c) \cdot \det(\mathbf{A} - \mathbf{K}_f\mathbf{C}) \quad (\text{B.12})$$

holds. The product of the eigenvalues of a matrix are just their determinant, so that the separation principle is proven if $\mathbf{A} - \mathbf{B}\mathbf{K}_c$ and $\mathbf{A} - \mathbf{K}_f\mathbf{C}$ have distinct eigenvalues and at least one of them is invertible.

One important consequence of the separation principle is that the regulator design task (that is, the computation of the feedback gain matrix \mathbf{K}_c) and the observer design task (the computation of \mathbf{K}_f) are independent of each other and can be carried out separately. Note that this holds true only if the model utilized in the observer is exact.

B.3 Observer-based Representation (OBR) of a feedback compensator

Consider an LTI state space plant model of order n and a dynamic LTI compensator, that is, a dynamic LTI feedback control law, of order n_K in closed-loop interconnection with this plant. The compensator is called

- full-order if $n_K = n$ holds,
- augmented-order if $n_K > n$ holds, and
- reduced-order $n_K < n$ holds.

A central result shown in [3] is that any full- or augmented-order feedback compensator \mathbf{K} can be transformed into an OBR, consisting of a full-order state observer, a state vector feedback gain matrix, and a so-called Youla parameter \mathbf{Q} (which is an LTI dynamic system of dynamic order $n_K - n$ or a static gain if $n_K = n$), see Fig. B.3.

One major advantage of this representation is that if a (trusted) plant model $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is available during operation, the observer part of the controller can directly exploit this information by utilizing these \mathbf{A} , \mathbf{B} , and \mathbf{C} matrices in its realization. This often provides good results in scheduled control laws for slowly parameter-varying systems (see Sec. 8.2).

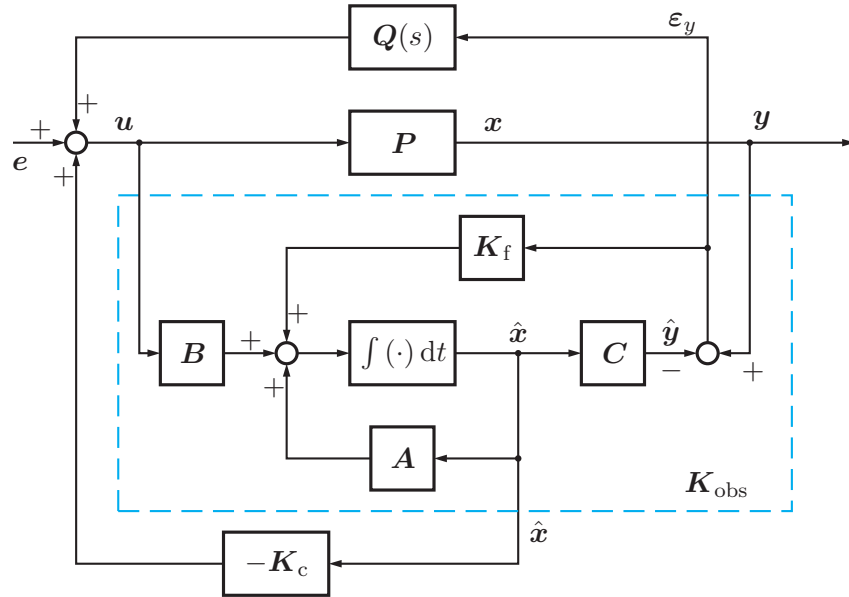


Figure B.3: Compensator in observer-based representation with observer \mathbf{K}_{obs} , feedback gain \mathbf{K}_c and Youla parameter $Q(s)$

For reduced-order compensators, a form based on a reduced- or minimum order observer can be obtained if some existence conditions are met. However, in the latter case the reduced-order OBR cannot be realized with the plant matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , so the plant structure cannot be exploited as in the full- or augmented-order cases. As one possible solution, these structural properties can be recovered with reduced-order plant model matrices.

B.3.1 Transformation of Arbitrary Compensators to an OBR

The main results of [3] state the transformation steps of arbitrary compensators into observer-based representations and are summarized in the following. It is assumed throughout that the compensator yields a stable closed loop, that is, it is assumed internally stabilizing.

Transformation of Full-Order Compensators

Consider the plant and compensator state space representations,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (\text{B.13})$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}, \quad (\text{B.14})$$

and

$$\dot{\boldsymbol{\eta}} = \mathbf{A}_K\boldsymbol{\eta} + \mathbf{B}_K\mathbf{y} \quad (\text{B.15})$$

$$\mathbf{u} = \mathbf{C}_K\boldsymbol{\eta} + \mathbf{D}_K\mathbf{y}, \quad (\text{B.16})$$

respectively. First, a transformation \mathbf{T} is sought such that the compensator is a Luenberger observer of the variables $\mathbf{z} = \mathbf{T}\mathbf{x}$, or equivalently, that the compensator states $\boldsymbol{\eta} = \hat{\mathbf{z}}$ are estimates of $\mathbf{z} = \mathbf{T}\mathbf{x}$.

As in Sec. B.1, this problem is equivalent to finding \mathbf{T} , \mathbf{F} , and \mathbf{G} such that for the system

$$\dot{\hat{\mathbf{z}}} = \mathbf{F}\hat{\mathbf{z}} + \mathbf{G}\mathbf{y} + \mathbf{T}\mathbf{B}\mathbf{u} \quad (\text{B.17})$$

the observation error $\mathbf{z} - \hat{\mathbf{z}}$ vanishes as t goes to infinity. This is fulfilled when the constraint

$$\mathbf{T}\mathbf{A} - \mathbf{F}\mathbf{T} = \mathbf{G}\mathbf{C} \quad (\text{B.18})$$

is fulfilled [89] and if \mathbf{F} is stable.

The compensator equations become

$$\dot{\hat{\mathbf{z}}} = (\mathbf{F} + \mathbf{T}\mathbf{B}\mathbf{C}_K) \hat{\mathbf{z}} + (\mathbf{G} + \mathbf{T}\mathbf{B}\mathbf{D}_K) \mathbf{y} \quad (\text{B.19})$$

$$\mathbf{u} = \mathbf{C}_K \hat{\mathbf{z}} + \mathbf{D}_K \mathbf{y}, \quad (\text{B.20})$$

and the \mathbf{F} and \mathbf{G} can be identified as

$$\mathbf{F} = \mathbf{A}_K - \mathbf{T}\mathbf{B}\mathbf{C}_K, \quad (\text{B.21})$$

$$\mathbf{G} = \mathbf{B}_K - \mathbf{T}\mathbf{B}\mathbf{D}_K. \quad (\text{B.22})$$

These matrices have to fulfill the constraints posed by Luenberger, so inserting (B.21)–(B.22) into (B.18) yields a generalized non-symmetric and rectangular Riccati equation in the unknown matrix variable \mathbf{T} :

$$\mathbf{A}_K \mathbf{T} - \mathbf{T}(\mathbf{A} + \mathbf{B}\mathbf{D}_K \mathbf{C}) - \mathbf{T}\mathbf{B}\mathbf{C}_K \mathbf{T} + \mathbf{B}_K \mathbf{C} = \mathbf{0}, \quad (\text{B.23})$$

which can be rewritten as

$$\begin{bmatrix} -\mathbf{T} & \mathbf{I} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{A} + \mathbf{B}\mathbf{D}_K \mathbf{C} & \mathbf{B}\mathbf{C}_K \\ \mathbf{B}_K \mathbf{C} & \mathbf{A}_K \end{bmatrix}}_{\mathbf{A}_{\text{cl}}} \begin{bmatrix} \mathbf{T} \\ \mathbf{I} \end{bmatrix} = \mathbf{0}. \quad (\text{B.24})$$

A solution \mathbf{T} (whose existence conditions are discussed in [3]) is obtained by the following steps:

- A basis \mathbf{U} of an n -dimensional invariant subspace of the associated eigenvalue problem

$$\mathbf{A}_{\text{cl}} \mathbf{U} = \mathbf{U} \mathbf{\Lambda} \quad (\text{B.25})$$

is computed via eigenvalue decomposition or Schur factorizations, see [3], [44]. This subspace is defined by the choice of eigenvalues of \mathbf{A}_{cl} that are included in $\mathbf{\Lambda}$. It turns out that these correspond to the poles of $\mathbf{A} - \mathbf{B}\mathbf{K}_c$ in the resulting OBR.

- The transformation \mathbf{T} is obtained by $\mathbf{T} = \mathbf{U}_2 \mathbf{U}_1^{-1}$ where $\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix}$ is partitioned such that \mathbf{U}_1 is $(n \times n)$ which is compatible to \mathbf{A}_{cl} .

Then the OBR of the compensator is obtained as in Fig. B.3 with the matrices

$$\mathbf{K}_f = \mathbf{T}^{-1} \mathbf{B}_K - \mathbf{B}\mathbf{D}_K \quad (\text{B.26})$$

$$\mathbf{K}_c = -\mathbf{C}_K \mathbf{T} - \mathbf{D}_K \mathbf{C} \quad (\text{B.27})$$

$$\mathbf{Q}(s) = \mathbf{D}_Q = \mathbf{D}_K. \quad (\text{B.28})$$

Note that the choice of poles in $\mathbf{\Lambda}$ when obtaining \mathbf{T} is not unique, but instead is taken from a finite number of combinatorial choices, and determines the realization of the OBR. Some constraints can be derived from existence conditions on \mathbf{T} , for example, the uncontrollable poles of (\mathbf{A}, \mathbf{B}) must be in $\mathbf{\Lambda}$, whereas the unobservable poles of (\mathbf{A}, \mathbf{C}) must not be contained in $\mathbf{\Lambda}$ for \mathbf{T} to exist. Moreover, some heuristic constraints can be of help (“assign those poles of \mathbf{A}_{cl} which are (in some sense) close to the open-loop plant poles of \mathbf{A} to $\mathbf{A} - \mathbf{B}\mathbf{K}_c$, i.e., include them in $\mathbf{\Lambda}$ ”). These are discussed in [3], but no rigorous analysis of the choices, for example in terms of implications on robustness of the chosen OBR against various types of uncertainty is available.

Remarks: Transformations of Augmented-Order and Reduced-Order Compensators

A detailed derivation of the OBR of an augmented-order compensator as well as of a reduced-order compensator is given in [3] and will not be reproduced here. However, some basic facts are pointed out in the following.

For the augmented-order case, the transformation matrix \mathbf{T} (now a $(n_K \times n)$ matrix) and the matrices \mathbf{F} and \mathbf{G} are obtained as in the full-order case. Then, via a Schur decomposition of \mathbf{F} , the dynamic Youla parameter (of order $n_K - n$) and the observer dynamics of order n are separated which involves an additional eigenvalue selection task. A heuristic approach mentioned in [3] is to include the fastest poles in $\mathbf{Q}(s)$ so that it acts approximately as feedthrough term as in the case of a static Youla parameter.

The reduced-order case does not lead to a structure as in Fig. B.3 because no full-order observer utilizing the plant matrices \mathbf{A} , \mathbf{B} , and \mathbf{C} can be constructed. However, the scheme of a reduced-order observer can be followed if $n_K + n_y \geq n$ holds (where the number of available measurements \mathbf{y} is n_y), see [89]. Then, the state estimate is constructed via

$$\hat{\mathbf{x}} = \mathbf{H}_1 \hat{\mathbf{z}} + \mathbf{H}_2 \mathbf{y} \quad (\text{B.29})$$

where \mathbf{H}_1 and \mathbf{H}_2 have to fulfill $\mathbf{H}_1 \mathbf{T} + \mathbf{H}_2 \mathbf{C} = \mathbf{I}$, see [3] for more details. If $n_K + n_y < n$ holds, these results (particularly the separation principle) do not hold for the original plant model. One might, however, construct a valid reduced-/full-/augmented-order OBR for an order-reduced plant model.

Appendix C

LQ-based Lateral Control Designs for BWB Aircraft

As a first step in lateral control law design, this section presents two LQ-based MIMO controllers for the BWB aircraft's lateral motion. First, an LQG controller is designed to fulfill damping and maneuver response requirements. Then, an LQI architecture is used to provide asymptotic tracking and disturbance rejection. The control laws are successfully validated at varying fuel and payload mass cases for the cruise flight condition of the considered BWB configuration.

C.1 Lateral LQG Control Design

C.1.1 LQG Architecture and Design Parameters

Traditionally, to increase the damping of the DR mode and to improve disturbance rejection in β or Ny_{CG} , the yaw rate and Ny_{CG} are fed back to the rudder in two SISO loops, respectively. Additionally, the required roll response would naturally be shaped via a roll angle to (antisymmetric) aileron feedback.

However, since SISO control of the DR mode and β or Ny_{CG} faces said limitations, a MIMO architecture with control input commands for rudder RU and combined aileron deflections AIL and the following subset of measured outputs (compare Sec. 4.2) are used:

- Yaw rate $y_1 = r$
- Roll angle $y_2 = \phi$
- Lateral acceleration $y_3 = Ny_{CG}$

It turned out that the roll rate is not needed as feedback signal to achieve the requirements, although it is common in lateral / roll attitude control to do so. In order to incorporate reference tracking specifications into LQG design, a roll reference input r_ϕ is added and the roll angle output is replaced by the roll tracking error output $\phi_r - \phi$. This way, an output-weighted LQG design can directly shape the reference step response.

The control design plant was chosen at parameter values in the middle of the considered fuel and passenger mass parameter intervals. This choice leads to acceptable performance at the extremal parameter values and best performance at the design plant configuration.

The noisy plant configuration as in (3.37)–(3.38) (see Sec. 3.4.1) is considered and an output-weighted LQG design (see Sec. 3.4.2) is carried out.

For the LQR design, the design parameters

$$\tilde{\mathbf{Q}} = 10^3 \cdot \text{diag}(1 \ 10 \ 1), \quad (\text{C.1})$$

$$\tilde{\mathbf{R}} = 20 \cdot \mathbf{I}_{[2 \times 2]}, \quad (\text{C.2})$$

corresponding to the output-weighted objective (3.50) was carried out. This emphasizes the roll tracking error in the design and leads to a well-performing controller for all mass cases of the aircraft model.

The Kalman estimator was designed with process noise modeled as if it was acting via the control input channels. This corresponds to the case $\mathbf{E} = \mathbf{B}$ in (3.37)–(3.38) (see Sec. 3.4.1), and $\mathbf{D} = \mathbf{0}$ holds. The process noise covariance was chosen high to improve convergence (in the sense of so-called Q-stabilization, compare [40]).

The chosen process noise parameters for Kalman-Bucy filter design (see Sec. 3.4.1) are:

$$\mathbf{W} = 25^2 \cdot \mathbf{I}_{[2 \times 2]}, \quad (\text{C.3})$$

$$\mathbf{V} = 10^{-3} \cdot \mathbf{I}_{[3 \times 3]}. \quad (\text{C.4})$$

C.1.2 LQG Validation Results

The resulting (nominal) closed loop performance can be seen in Fig. C.1 (roll reference step response) and Fig. C.2 (response due to a step in the vertical wind disturbance). It is evident that the DR mode damping is increased and that the reference roll response largely fulfills the given requirement. Also disturbance rejection is effective (in open loop, the system drifts away), but this controller yields a non-zero position error.

However, the limited effect of the roll demand and the disturbance on β and Ny_{CG} are an important step towards a coordinated turn control.

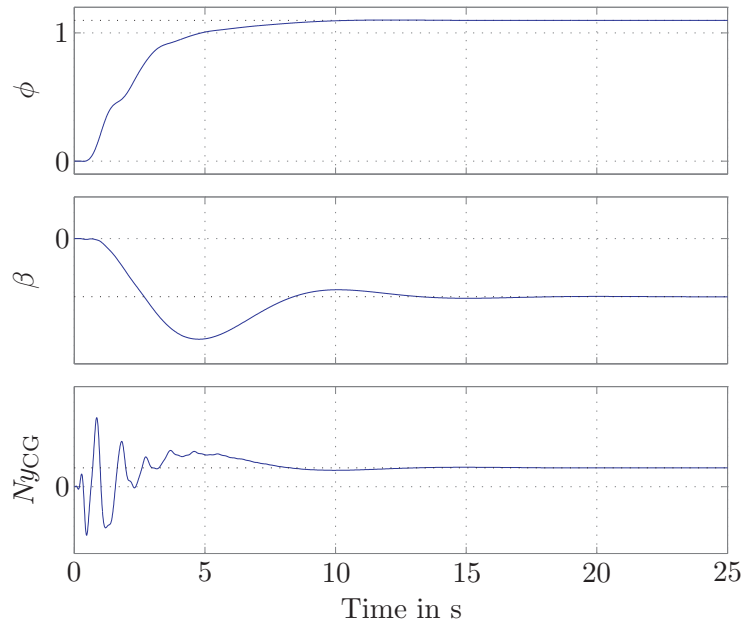


Figure C.1: LQG closed-loop results: Responses to a roll reference step (roll angle ϕ , sideslip angle β , lateral acceleration Ny_{CG})

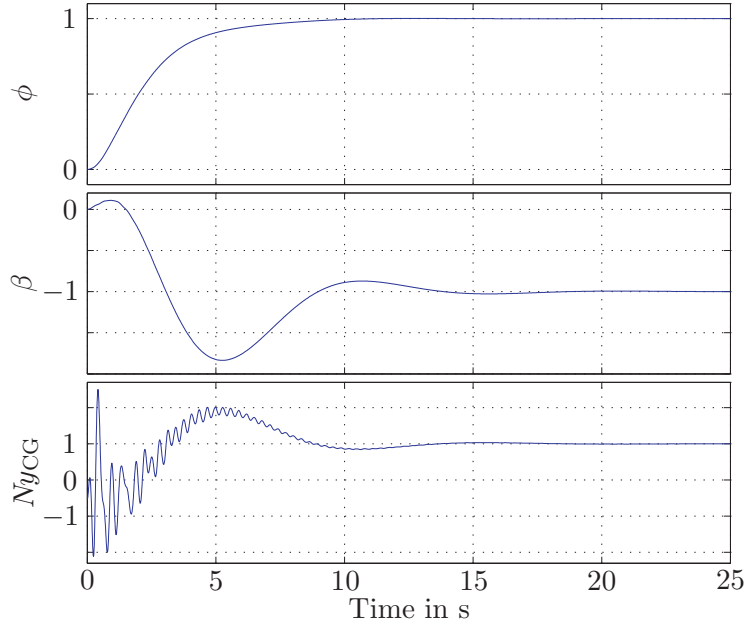


Figure C.2: LQG closed-loop results: Responses to a wind disturbance step (roll angle ϕ , sideslip angle β , lateral acceleration Ny_{CG} , all normalized with respect to their final values)

C.2 Lateral LQI Control Design

C.2.1 Motivation

While the LQG controller in Sec. C.1 performs well in terms of roll response shaping and DR mode damping, it cannot reject low-frequency exogenous disturbances. To achieve this, an integrator is added in the inner loop control loop, arriving at a two-degree-of-freedom LQI architecture as shown in Fig. 3.5, see Sec. 3.4.3.

This architecture is expected to yield zero position error with respect to step references in roll and step disturbances. While it is usually not the objective of inner loop control to provide asymptotic setpoint tracking, asymptotic disturbance rejection is a feature which can reason an integral control law at this stage. However, adding integrators can conflict with robustness, bandwidth-, or phase/gain margin requirements.

C.2.2 Design Parameters

As presented in Sec. C.1 the control design plant was again chosen at central fuel and payload parameter values. The following design weights were chosen for the LQI control design:

$$\mathbf{Q} = \begin{bmatrix} 10^3 \mathbf{I}_{[3 \times 3]} & & & & \\ & \mathbf{0} & & & \\ & & 0.1 & & \\ & & & 1 & \\ & & & & 0.1 \end{bmatrix}, \quad (\text{C.5})$$

$$\mathbf{R} = 10^{-2} \mathbf{I}_{[2 \times 2]}, \quad (\text{C.6})$$

where the state weighting matrix \mathbf{Q} focuses the control action on the rigid body modes of the aircraft, leaves the flexible modes largely untouched, and also emphasizes asymptotic roll reference tracking in the design. The input weighting \mathbf{R} was chosen under the trade off of control

law sensitivity with respect to the mass variation (requires low control input magnitudes, i.e. high \mathbf{R} values) versus (nominal) performance requirements.

The Kalman estimator was designed using the non-augmented design plant identically as in Sec. C.1.

C.2.3 LQI Results: Performance

Roll Reference Tracking Results are shown in Fig. C.3 via the step response of a large reference step of the roll angle. It can be seen that the overshoot is minimal and that the rise time is around 7 s. Moreover, the control surface deflections for a -30° to 30° roll reference step stay well within saturation and rate limits.

Another feature of this control law is evident, namely the realization of a coordinated turn: after the roll maneuver is completed, the side slip β and the lateral acceleration Ny_{CG} both approach zero.

Disturbance Rejection Results are demonstrated in the following. Figure C.4 shows the Bode magnitude plot of the rudder disturbance \rightarrow yaw rate transfer function of the closed loop. It shows (quasi-) globally derivative behavior, while the open loop shows quasi-global integral behavior (compare Fig. 4.3). Also, the strongly increased damping of the DR mode is evident. The disturbance step response illustrates this particularly, see Fig. C.5.

The disturbance rejection properties of the LQI controller are highly favorable and pose the major reason to consider such architecture in the inner flight control loop.

Finally, the rejection of the wind disturbance is also evident in the roll angle ϕ , β , and Ny_{CG} as depicted in Fig. C.6.

C.2.4 LQI Results: Validation at various Mass Cases

In order to validate the LQI control law, it is applied to a validation set that includes 6 plant models at several mass cases (including the minimum and maximum fuel and payload cases). The design process was tuned (by selecting the best-suited design plant mass case and adjusting the control input weighting matrices \mathbf{R}) to obtain robustly performing controllers. Note that this robustness is validated only with respect to the used discrete validation set.

All closed-loop systems were stable, and Fig. C.7 depicts the roll reference step response of each validation plant. The small differences prove that, albeit significant mass variation, the dominant low-frequency dynamics is robustly shaped. Figure C.8 shows the Bode magnitude plot from wind disturbance to yaw rate, both for open loop (OL) and closed loop (CL). From this transfer function it is evident that the DR mode is strongly dependent on the mass variations. However, the LQI controller attenuates it robustly.

The LQG validation details are omitted for brevity but show analogous results.

C.3 Discussion

The lateral dynamics of the investigated BWB aircraft shows slow RHP zeros in important SISO transfer functions which represent fundamental limitations for traditional SISO lateral control design.

The proposed Linear-Quadratic (LQ)-based MIMO controllers (an LQG architecture without and an LQI architecture with asymptotic reference tracking and disturbance rejection) can circumvent these limitations and meet the principal control goals. However, the additional control goals are yet to be tackled: loads alleviation and comfort improvement.

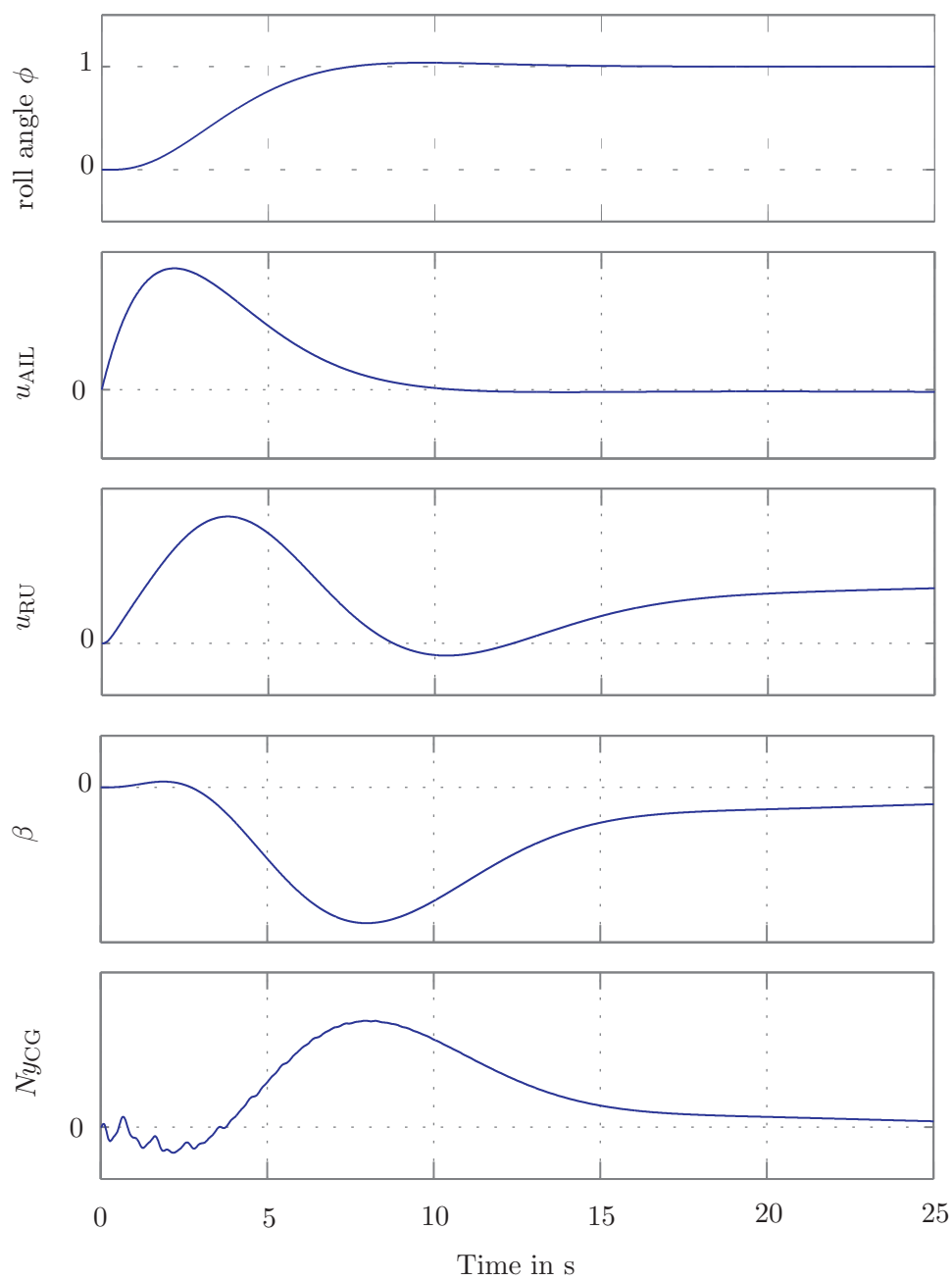


Figure C.3: LQI closed-loop results: Responses to a roll reference step

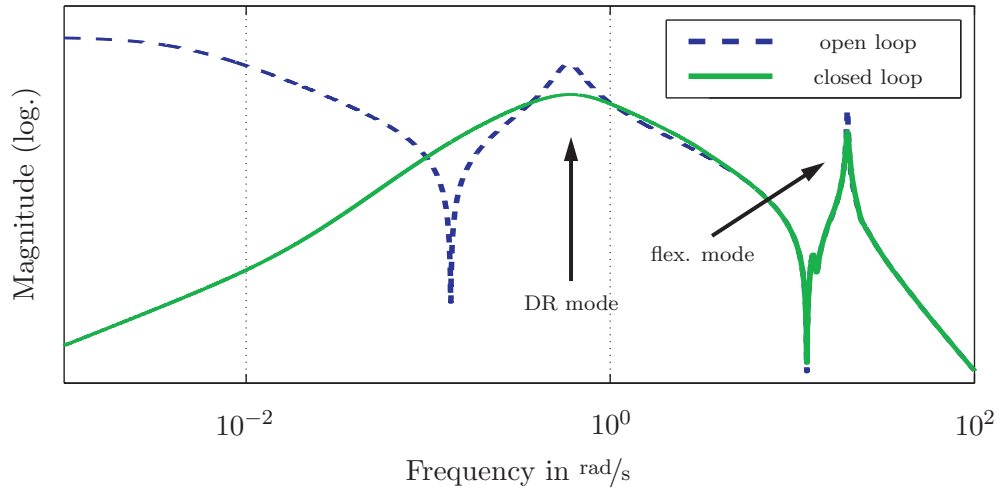


Figure C.4: LQI closed-loop results: Bode magnitude of rudder \rightarrow yaw rate transfer

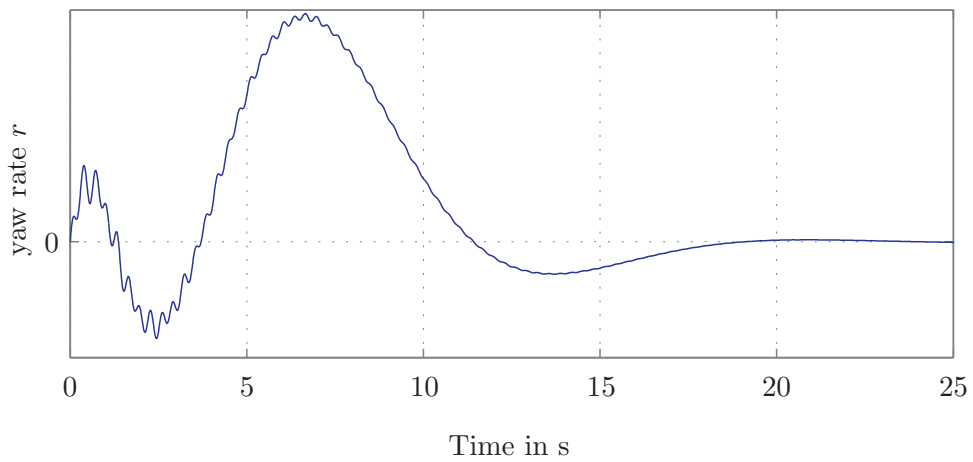


Figure C.5: LQI closed-loop results: Response of yaw rate r to a wind disturbance step

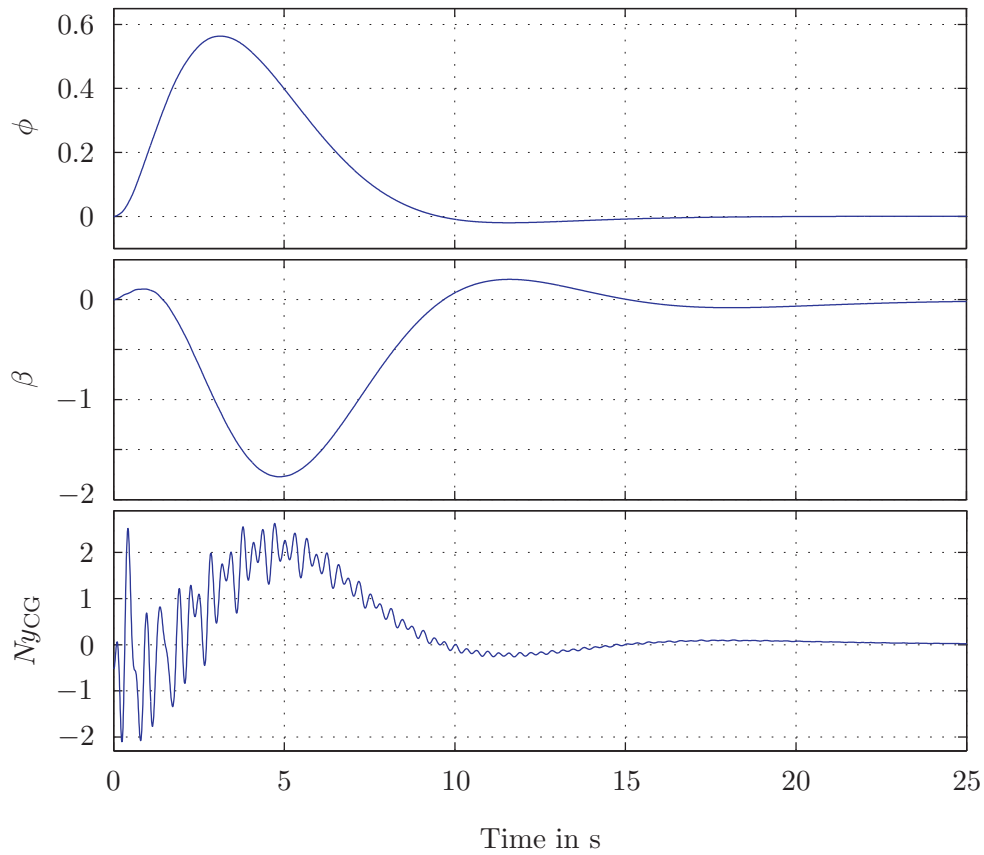


Figure C.6: LQI closed-loop results: Responses to a wind disturbance step (roll angle ϕ , sideslip angle β , lateral accel. Ny_{CG} , all normalized using the final values of the LQG disturbance case (compare Fig. C.2))

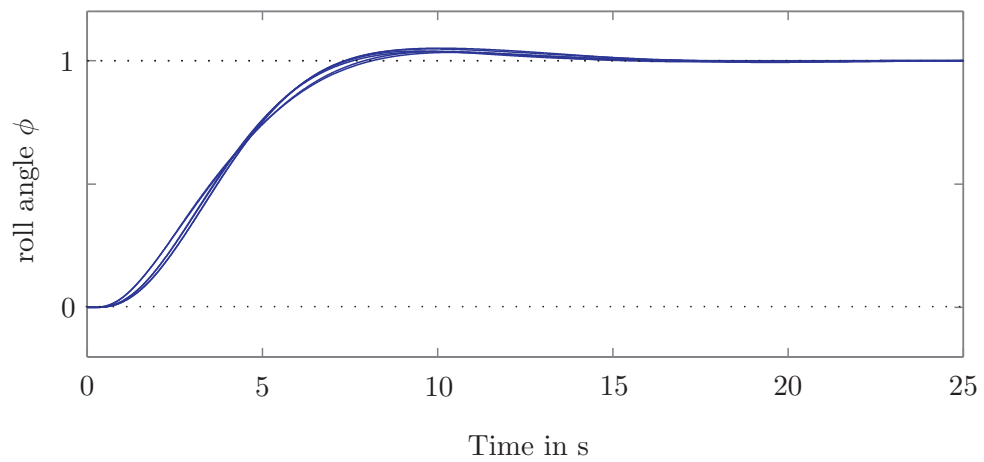


Figure C.7: LQI closed-loop results: Roll response to step command for all mass cases

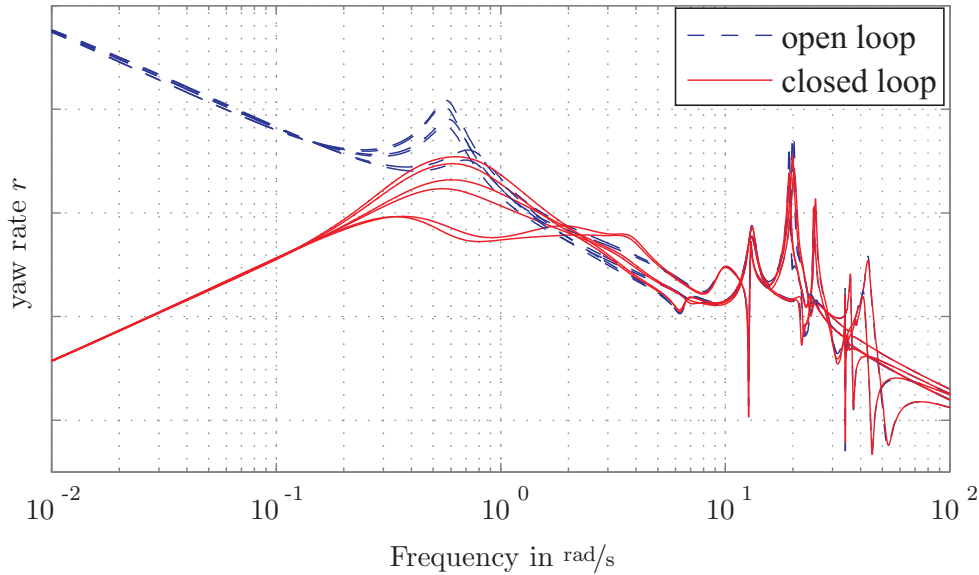


Figure C.8: LQI closed-loop results: Bode magnitude of wind disturbance \rightarrow yaw rate transfer for all mass cases

Applying robust control design methods on this aircraft's lateral dynamics is the next logical step and is subject of ongoing research by the authors and their project partners. From the first studies and the results obtained here, the low-frequency dynamics are likely to be unproblematic with respect to robustness. However, the structural modes' variations over mass might significantly limit achievable robust performance in that frequency range.

Further studies will include different design methodologies to fulfill all control goals robustly with maximized robust performance, for example using DK iteration, Q - μ iterations (based on the Youla parametrization) and fixed-structure \mathcal{H}_∞ optimization.

The investigated LQG controller is directly synthesized in an observer-based form and thus amendable to be used as starting point for convex synthesis methods based on the Youla parametrization [18].

At the same time, the longitudinal dynamics is under study by the authors [162], showing significant coupling between rigid-body and flexible modes. As evident from the results above, this is less severe for the lateral motion in the present BWB configuration with the given requirements. The longitudinal - lateral coupling has yet to be investigated and its consequences on control design will be of high research interest.

C.4 Conclusions

Two LQ-based MIMO controllers (LQG and LQI) are presented for the lateral aircraft dynamics. The main control goals (Dutch roll damping, coordinated turn control, and roll reference tracking) are successfully reached. While the LQG controller only partially fulfills the posed requirements, the designed LQI controller yields good performance with respect to all requirements: it asymptotically rejects disturbances and performs well in roll tracking. The control performance has been validated at a set of validation plants with varying fuel and payload mass. Ongoing and future studies include robust control design for the lateral and longitudinal dynamics of the BWB aircraft.

Appendix D

Classes of the Object-Oriented Framework for Convex Control Design

A listing of each class of the optimization framework for convex control design follows. The framework structure is described in Sec. 5.1, and the following information is intended as quick reference for future development work on and with the framework.

Each class is given a name and a short description of its role in the framework. The properties of the class (local variables for the created objects of this class) and its methods (functions acting on objects of the class) are listed without further comment. However, care has been taken to give sensible names to the classes, properties, and methods to be largely self-explanatory. Abstract properties or methods are those which have to be defined (in a non-abstract, i.e., concrete way) by subclasses so that actual objects can be created.

D.1 Classes in the ConvexOptimizer Hierarchy

Class name:	<code>ConvexOptimizer</code>
Description:	implements top-level functionality to formulate and solve convex control design problems via an LMI formulation
Superclass:	none
Properties:	<code>Sys_OL</code> , <code>Sys_CL_Q</code> , <code>Alphas</code> , <code>Thetas</code> , <code>History</code> , <code>Q</code> , ...
Abstract properties:	<code>K</code>
Methods:	<code>generateQBase</code> , <code>addObjective</code> , <code>addConstraint</code> , <code>solveLMI</code> , <code>iterateLMI</code> , ...
Abstract methods:	<code>refineLMIs</code> , <code>refineGrids</code> , <code>plot</code>

Class name:	<code>ConvexOpt_SingleModel</code>
Description:	specializes <code>ConvexOptimizer</code> to problems with a single plant model
Superclass:	<code>ConvexOptimizer</code>
Properties:	<code>P11</code> , <code>P12</code> , <code>P21</code> , <code>P22</code> , <code>A</code> , <code>B1</code> , <code>B2</code> , <code>C1</code> , <code>C2</code> , <code>D11</code> , ..., <code>D22</code>
Methods:	<code>refineLMIs</code> , <code>refineGrids</code> , <code>plot</code>
Abstract methods:	<code>getCL</code>

Class name:	<code>ConvexOpt_MultiModel</code>
Description:	specializes <code>ConvexOptimizer</code> to problems with multiple plant models
Superclass:	<code>ConvexOptimizer</code>
Properties:	none
Methods:	<code>refineLMIs</code> , <code>refineGrids</code> , <code>plot</code>
Abstract methods:	<code>getCLs</code>

Class name:	<code>ConvexOpt_FB_OL</code>
Description:	realizes convex feedback control design problem with zero initial controller
Superclass:	<code>ConvexOpt_SingleModel</code>
Properties:	<code>K</code>
Methods:	<code>generateAffineCL</code> , <code>getCL</code>

Class name:	<code>ConvexOpt_FB_OL</code>
Description:	realizes convex feedback control design problem with OBR initial controller
Superclass:	<code>ConvexOpt_SingleModel</code>
Properties:	<code>Sys_CL_initK</code> , <code>F</code> , <code>L</code> , <code>Q0</code> , <code>K</code>
Methods:	<code>generateOBRCL</code> , <code>getCL</code>

D.2 Classes in the Constraint/Objective Hierarchy

Class name:	<code>ConstrDef</code>
Description:	general class for constraint and objective definitions which implements basic design/validation grid handling, common data structures and method interfaces
Superclass:	none
Properties:	<code>Name</code> , <code>Sys</code> , <code>SysID</code> , <code>GroupID</code> , <code>PlotID</code> , <code>SubplotID</code> , <code>InChannel</code> , <code>OutChannel</code> , <code>Template</code> , ...
Constant properties:	<code>REFINE_GRID_MAX_ADD</code>
Methods:	<code>addToDesignGrid</code> , <code>preparePlot</code> , <code>importGrids</code> , <code>exportGrids</code>
Abstract methods:	<code>defLMI</code> , <code>plot</code> , <code>refineDesignGrid</code>
Static methods:	<code>getLocalMaxima</code>

Class name:	<code>ObjDef</code>
Description:	specializes <code>ConstrDef</code> to objectives by adding free bounding variables “alpha”
Superclass:	<code>ConstrDef</code>
Properties:	<code>AlphaID</code> , <code>AlphaScaling</code>
Abstract methods:	<code>assignAlphaScalingByRefSys</code> , <code>refineLMIDef</code>
<hr/>	
Class name:	<code>ConstrH2</code>
Description:	implements \mathcal{H}_2 constraint $\ \mathbf{T}(\boldsymbol{\theta})\ _2 \leq \text{Target}$, <code>Target</code> const.
Superclass:	<code>ConstrDef</code>
Properties:	<code>Target</code> , <code>ObjH2H</code>
Methods:	<code>defLMI</code> , <code>plot</code> , <code>refineDesignGrid</code> , <code>cleanupDesignGrid</code> , <code>refineLMIDef</code> , <code>resetCache</code> , <code>saveCache</code> , <code>loadCache</code>
<hr/>	
Class name:	<code>ConstrHInf</code>
Description:	implements \mathcal{H}_∞ constraint $\bar{\sigma}(\mathbf{T}(j\omega_k, \boldsymbol{\theta})) \leq \gamma_k$ where γ_k is defined via a <code>Template</code> object
Superclass:	<code>ConstrDef</code>
Properties:	none
Methods:	<code>defLMI</code> , <code>plot</code> , <code>refineDesignGrid</code> , <code>cleanupDesignGrid</code> , <code>refineLMIDef</code>
<hr/>	
Class name:	<code>ConstrLInflB</code>
Description:	implements time-domain lower bound l_∞ constraint $z_L(t_k) \leq z(t_k) - z_{\text{ref}}(t_k)$ of the response to a given <code>InSignal</code> where $z_{\text{ref}}(t_k)$ is defined via a <code>Template</code> object
Superclass:	<code>ConstrDef</code>
Properties:	<code>InSignal</code>
Methods:	<code>defLMI</code> , <code>plot</code> , <code>refineDesignGrid</code> , <code>cleanupDesignGrid</code> , <code>refineLMIDef</code>
<hr/>	
Class name:	<code>ConstrLInfUB</code>
Description:	implements time-domain upper bound l_∞ constraint $z_{\text{ref}}(t_k) \geq z_U(t_k)$ of the response to a given <code>InSignal</code> where $z_{\text{ref}}(t_k)$ is defined via a <code>Template</code> object
Superclass:	<code>ConstrDef</code>
Properties:	<code>InSignal</code>
Methods:	<code>defLMI</code> , <code>plot</code> , <code>refineDesignGrid</code> , <code>refineLMIDef</code> , <code>cleanupDesignGrid</code>

Class name:	<code>ConstrLin</code>
Description:	implements linear (LP) constraint $\mathbf{a}^T \boldsymbol{\theta} \leq c$
Superclass:	<code>ConstrDef</code>
Properties:	<code>a, c</code>
Methods:	<code>defLMI, plot, refineDesignGrid, cleanupDesignGrid</code>
<hr/>	
Class name:	<code>ConstrLyap</code>
Description:	implements iterative Lyapunov strong stabilization constraint $\mathbf{P}_{\text{lyap}} \mathbf{A}_K(\boldsymbol{\theta}) + \mathbf{A}_K(\boldsymbol{\theta})^T \mathbf{P}_{\text{lyap}} \prec \mathbf{0}$ with fixed $\mathbf{P}_{\text{lyap}} = \mathbf{P}_{\text{lyap}}^T \succ \mathbf{0}$ or solves for \mathbf{P}_{lyap} with fixed \mathbf{A}_K
Superclass:	<code>ConstrDef</code>
Properties:	<code>P_lyap, P22, c</code>
Methods:	<code>calcLyap, defLMI, plot, refineDesignGrid, cleanupDesignGrid</code>
Static methods:	<code>constructAk</code>
<hr/>	
Class name:	<code>ConstrRSmu</code>
Description:	implements a singular value constraint of mixed- μ bounds analogous to $Q - \mu$ -synthesis
Superclass:	<code>ConstrDef</code>
Properties:	<code>Dleft, Dright, Gleft, Gmiddle, Gright</code>
Methods:	<code>updateRSMuScalings, defLMI, plot, refineDesignGrid, cleanupDesignGrid</code>
<hr/>	
Class name:	<code>ConstrHypCube</code>
Description:	implements parameter hypercube constraint $\boldsymbol{\theta}^- \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^+$ for strong stabilization heuristics
Superclass:	<code>ConstrDef</code>
Properties:	<code>p_min, p_max, ObjCvOpt</code>
Methods:	<code>defLMI, plot, refineDesignGrid, cleanupDesignGrid</code>
<hr/>	
Class name:	<code>ObjH2</code>
Description:	implements \mathcal{H}_2 objective $\ \mathbf{T}(\boldsymbol{\theta})\ _2 \leq \alpha$ with LMI variable α
Superclass:	<code>ObjDef</code>
Properties:	<code>Target, ObjH2H</code>
Methods:	<code>defLMI, plot, refineDesignGrid, cleanupDesignGrid, refineLMIDef, resetCache, saveCache, loadCache, assignAlphaScalingByRefSys, refineLMIDef</code>

Class name:	ObjHInf
Description:	implements \mathcal{H}_∞ objective $\bar{\sigma}(\mathbf{T}(j\omega_k, \boldsymbol{\theta})) \leq \alpha\gamma_k$ where γ_k is defined via a <code>Template</code> object and α is an LMI variable
Superclass:	<code>ObjDef</code>
Properties:	none
Methods:	<code>defLMI</code> , <code>plot</code> , <code>refineDesignGrid</code> , <code>cleanupDesignGrid</code> , <code>refineLMIDef</code> , <code>assignAlphaScalingByRefSys</code>
Class name:	ObjLInflB
Description:	implements a time-weighted, one-sided time-domain lower bound l_∞ objective $(-\alpha)z_L(t_k) \leq z(t_k) - z_{\text{ref}}(t_k)$ of the response to a given <code>InSignal</code> where α is an LMI variable and $z_{\text{ref}}(t_k)$ is defined via a <code>Template</code> object
Superclass:	<code>ObjDef</code>
Properties:	<code>InSignal</code> , <code>TimeWeightingExp</code> , <code>TimeWeightConst</code>
Methods:	<code>defLMI</code> , <code>plot</code> , <code>refineDesignGrid</code> , <code>cleanupDesignGrid</code> , <code>refineLMIDef</code> , <code>calcAlphaScalingByRefSys</code> , <code>assignAlphaScalingByRefSys</code>
Class name:	ObjLInfUB
Description:	implements time-domain upper bound l_∞ objective $z(t_k) - z_{\text{ref}}(t_k) \geq \alpha z_U(t_k)$ of the response to a given <code>InSignal</code> where α is an LMI variable and $z_{\text{ref}}(t_k)$ is defined via a <code>Template</code> object
Superclass:	<code>ObjDef</code>
Properties:	<code>InSignal</code> , <code>TimeWeightingExp</code> , <code>TimeWeightConst</code>
Methods:	<code>defLMI</code> , <code>plot</code> , <code>refineDesignGrid</code> , <code>cleanupDesignGrid</code> , <code>refineLMIDef</code> , <code>calcAlphaScalingByRefSys</code> , <code>assignAlphaScalingByRefSys</code>
Class name:	ObjLInf
Description:	implements a time-weighted, two-sided time-domain lower and upper bound l_∞ objective $(-\alpha)z_L(t_k) \leq z(t_k) - z_{\text{ref}}(t_k) \geq \alpha z_U(t_k)$ of the response to a given <code>InSignal</code> where α is an LMI variable and $z_{\text{ref}}(t_k)$ is defined via a <code>Template</code> object
Superclass:	<code>ObjDef</code>
Properties:	<code>InSignal</code> , <code>TimeWeightingExp</code> , <code>TimeWeightConst</code> , <code>ObjLB</code> , <code>ObjUB</code>
Methods:	<code>defLMI</code> , <code>plot</code> , <code>refineDesignGrid</code> , <code>refineLMIDef</code> , <code>importGrids</code> , <code>exportGrids</code> , <code>calcAlphaScalingByRefSys</code> , <code>assignAlphaScalingByRefSys</code>

D.3 Auxiliary Classes

Class name:	<code>CTemplate</code>
Description:	defines a common interface to parametric and non-parametric templates
Superclass:	none
Properties:	none
Methods:	<code>plot</code>
Abstract methods:	<code>eval</code>
Class name:	<code>CTemplParametric</code>
Description:	implements a parametric template defined by the time-domain step response, bode magnitude, or complex transfer values at given time respectively frequency values of an LTI system <code>Sys</code>
Superclass:	<code>CTemplate</code>
Properties:	<code>Sys, Grid, Response</code>
Methods:	<code>eval</code>
Class name:	<code>CTemplNonParametric</code>
Description:	implements a non-parametric template defined by a set of (x_i, y_i) points where the y_i s constitute <code>Data</code> and the x_i s are stored in <code>Grid</code> . Various types of interpolation can be used when evaluating the template's value at x .
Superclass:	<code>CTemplate</code>
Properties:	<code>InterpolationMethod, ExtrapolationMethod, Data, Grid</code>
Methods:	<code>eval</code>
Class name:	<code>CYoulaModel</code>
Description:	implements the affine parametrization of transfer functions in the form $T_1 + T_2 Q T_3$ where Q is affine in the parameters θ including basis construction, response computation, and transparent response data caching
Superclass:	none
Properties:	<code>SysID, T1, T2, T3, QBase, Sched_Factor, ncont, nmeas, nThetaTotal, nThetaLocal, ...</code>
Methods:	<code>generateQBase, queryStepCache, addStepCache, getStepResp, getFreqResp</code>
Static methods:	<code>computeCacheKey</code>

Class name:	H2Handler
Description:	implements the computation and caching of the precomputed scalar, vector, and matrix data that define the discretized \mathcal{H}_2 norm
Superclass:	none
Properties:	ObjConstr, beta, gamma, Q (corresponds to $\mathbf{\Gamma}$), L, filename
Methods:	resetCache, buildCache, calcH2norm, saveCache, loadCache, validateCache

Class name:	CTools
Description:	collects miscellaneous helper functions
Superclass:	none
Properties:	none
Static methods:	processPVpairs, getQBaseOrtho, getStepUneven, solveAdaptiveLMIs

List of Figures

2.1	The $P - K - \Delta$ - form and LFTs	16
2.2	Additive uncertainty model	20
2.3	Multiplicative uncertainty model	21
2.4	Parametric state space uncertainty	23
3.1	Standard form of the feedback control design problem	38
3.2	Mixed-sensitivity problems in standard form	40
3.3	Exemplary 2DOF augmented plant interconnection structure	42
3.4	LQG architecture	46
3.5	LQI architecture	49
3.6	Standard feedback interconnection, Youla parametrization	55
4.1	Aircraft system model with actuator and sensor dynamics	62
4.2	BWB aircraft schematics	63
4.3	Open-loop Bode magnitude: Rudder \rightarrow yaw rate transfer	64
4.4	Open-loop: yaw rate response to a rudder step	64
4.5	Open-loop: roll rate response to a wind disturbance step	64
4.6	Bode magnitude plot of lateral wind v_{lat} - lateral acceleration Ny_{CG}	64
4.7	Benefit of system scaling for balanced system reduction	67
5.1	Class hierarchy of ConvexOptimizer and its subclasses	71
5.2	Constraint/objective class hierarchy	72
5.3	Auxiliary classes	72
5.4	Stability region slice (white) for fixed $\theta_{12} = \theta_{21} = 0$	78
5.5	Stability region slice (white) for fixed $\theta_{11} = \theta_{22} = 0$	78
5.6	Stability region slice (white) for fixed $\theta_{11} = -1, \theta_{22} = -0.5$	78
5.7	Hypercube constraint $\theta \in \alpha \mathbf{E}_{sstab}$ with downscaling factor α	81
5.8	Polytope embedding, variant 1: Orthogonal hyperplanes	83
5.9	Polytope embedding, variant 2: Tangential hyperplanes	84
5.10	Step responses: open loop, initial controller, and optimized controller	87
5.11	Optimization results	89
5.12	Two feed-forward interconnection architectures	90
5.13	Step responses without, with LTI and with scheduled feed-forward controller	94
5.14	Shaped response $\phi_{ref} \rightarrow Mx_{fin}$	95
5.15	Shaped response $\phi_{ref} \rightarrow Ny_{CG}$	95
6.1	Automated control design, validation, and optimization: process and data flow	99
6.2	Class structure of the implementation	102
6.3	Two-degree-of-freedom (2DOF) control architecture	103
6.4	Goal attainment achieved by the initial controller	106

6.5	Goal attainment achieved by the optimized controller	107
6.6	Roll reference step response for all mass cases	107
6.7	Side-slip reference step response for all mass cases	108
6.8	Bode magnitude plot for closed-loop roll reference to wing root load output	108
6.9	Bode magnitude plot for closed-loop roll reference to fin root load output	109
6.10	Obtained vibration reduction of the targeted mode when weight parameter ω_{peak} is varied	110
6.11	Control performance (controller taken from [162])	110
6.12	Control performance of optimized controller	111
7.1	The Nominal Performance-preserving controller order reduction as model matching problem	114
7.2	Maximum singular values of reduced controllers and closed loop	117
7.3	Comparison of relative \mathcal{H}_∞ closed-loop performance error: BR vs. FWBR	117
7.4	Comparison of best achieved RP μ : BR vs. NP-FWBR vs. RP-FWBR	120
8.1	Root locus plots of the interpolated 3rd-order systems	126
8.2	Geometric Algebra interpolation illustration	130
8.3	Interpolation pole paths of 4th order systems in Sec. 8.5	132
8.4	Interpolation DC gains of 4th order systems in Sec. 8.5	132
8.5	Interpolation step and Bode plots of 4th order systems in Sec. 8.5	133
A.1	\mathcal{H}_2 and \mathcal{H}_∞ norm illustrations	149
B.1	Luenberger observer	152
B.2	Interconnection architecture: observer and static state vector feedback	153
B.3	Compensator in observer-based representation	155
C.1	LQG results: Responses to a roll reference step	159
C.2	LQG results: Responses to a wind disturbance step	160
C.3	LQI results: Responses to a roll reference step	162
C.4	LQI results: Bode magnitude of rudder \rightarrow yaw rate transfer	163
C.5	LQI results: Response of yaw rate r to a wind disturbance step	163
C.6	LQI results: Responses to a wind disturbance step	164
C.7	LQI results: Roll response to step command, all mass cases	164
C.8	LQI results: Bode magnitude of wind disturbance \rightarrow yaw rate transfer, all mass cases	165

List of Tables

2.1	Genetic Algorithm pseudo code	29
4.1	Lateral control goals for large flexible BWB aircraft	65
5.1	Strong stabilization test results: l_∞ objective, 41 LMI variables.	85
5.2	Objectives (O) and constraints (C) for feedback control optimization	88
5.3	Objectives (O) and constraints (C) for feed-forward control optimization	93
5.4	Cumulative cost values on the design plant set, relaxed constraints	94
6.1	BWB lateral control goals	105

Bibliography

- [1] ACFA 2020 Consortium. Active control of flexible 2020 aircraft (ACFA 2020), EU FP7 project no. 213321. URL: <http://www.acfa2020.eu>, last retrieved: Sep. 8th 2011.
- [2] J. Ackermann. *Robust Control - The Parameter Space Approach*. Springer, London, UK, 2002.
- [3] D. Alazard and P. Apkarian. Exact observer-based structures for arbitrary compensators. *International Journal of Robust and Nonlinear Control*, 9:101–118, 1999.
- [4] B.D.O. Anderson and Y. Liu. Controller reduction: concepts and approaches. *Automatic Control, IEEE Transactions on*, 34(8):802–812, 1989.
- [5] S.J. Axler. *Linear algebra done right*. Undergraduate texts in mathematics. Springer, New York, NY, 2nd edition, 1997.
- [6] G. Balas, R. Chiang, A. Packard, and M. Safonov. *MATLAB Robust Control Toolbox 3, User's Guide*. The MathWorks, 2009.
- [7] D. Bates and I. Postlethwaite. *Robust Multivariable Control of Aerospace Systems*. DUP Science, Ios Pr Inc, 2002.
- [8] C. Benatzky. *Theoretical and experimental investigation of an active vibration damping concept for metro vehicles*. Dissertation, Vienna University of Technology, Vienna, 2006.
- [9] S. Bialas. A necessary and sufficient condition for the stability of convex combinations of stable polynomials or matrices. *Bulletin of the Polish Academy of Sciences*, 33:437–480, 1985.
- [10] S. Borguet and O. Léonard. The Fisher information matrix as a relevant tool for sensor selection in engine health monitoring. *International Journal of Rotating Machinery*, 2008, 2008.
- [11] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), 1994.
- [12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [13] R. Brockhaus. *Flugregelung*. Springer Verlag, 2001.
- [14] S.L. Campbell. Linearization of DAEs along trajectories. *Zeitschrift für Angewandte Mathematik und Physik (ZAMP)*, 46(1):70–84, 1995.
- [15] F. Casella and M. Lovera. LPV/LFT modelling and identification: overview, synergies and a case study. In *2008 IEEE Int. Symp. Comp.-Aided Cont. Sys. Design*, pages 852–857, 2008.

- [16] G. Chesi. LMI techniques for optimization over polynomials in control: A survey. *IEEE Transactions on Automatic Control*, 55(11):2500–2510, 2010.
- [17] G.B. Dantzig. *Linear Programming and Extensions*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 1998.
- [18] I. Dardenne. *Développement de méthodologies pour la synthèse de lois de commande d'un avion de transport souple*. PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (SUPAERO), France, 1999. (Title (engl.): Development of methodologies for control law synthesis for a flexible transport aircraft).
- [19] V. R. Dehkordi and B. Boulet. Robust controller order reduction. In *Proc. American Control Conf.*, pages 3083–3088, USA, 2009.
- [20] F. Demourant. *Interactions identification-commande robuste: méthodes et applications à l'avion souple*. PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (SUPAERO), France, 2002. (Title (engl.): Interactions of identification and robust control: methods and applications to flexible aircraft).
- [21] C.A. Desoer and M. Vidyasagar. *Feedback systems: input-output properties*, volume 55. Society for Industrial Mathematics, 2009.
- [22] L. Dorst, S. Mann, and T. Bouma. GABLE: a Matlab tutorial for Geometric Algebra, version 1.3. Technical report, University of Waterloo, Ontario, Canada, 2002. URL: <http://www.cgl.uwaterloo.ca/~smann/GABLE/tutorial.pdf>, last retrieved Sep. 8th, 2011.
- [23] A. Douik, H. Liouane, and H. Messaoud. Optimized eigenstructure assignment by ant system and LQR approaches. *Int. Journal of Computer Science and Applications*, 5(4):45–56, 2008.
- [24] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis. State-space solutions to standard \mathcal{H}_2 and \mathcal{H}_∞ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.
- [25] S.C. Eisenstat and I.C.F. Ipsen. Relative perturbation results for eigenvalues and eigenvectors of diagonalisable matrices. *BIT Numerical Mathematics*, 38(3):502–509, 1998.
- [26] D.F. Enns. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In *Decision and Control, 1984. The 23rd IEEE Conference on*, volume 23, pages 127–132, 1984.
- [27] F. Fahroo and Y. Wang. Optimal location of piezoceramic actuators for vibration suppression of a flexible structure. In *Proceedings of the 36th Conference on Decision and Control*, pages 1966–1971, San Diego, California USA, 1997.
- [28] G. Feng. A survey on analysis and design of model-based fuzzy control systems. *IEEE Trans. Fuzzy Sys.*, 14(5):676–697, 2006.
- [29] G. Ferreres. Computation of a flexible aircraft LPV/LFT model using interpolation. *IEEE Trans. Cont. Sys. Tech.*, 19(1):132–139, 2011.
- [30] G. Ferreres, J.M. Biannic, and J.F. Magni. A skew mu toolbox (SMT) for robustness analysis. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 309–314, sep 2004.

- [31] G. Ferreres and G. Puyou. Feasibility of \mathcal{H}_∞ design specifications: an interpolation method. *Int. Journal of Control*, 78(12):927–936, 2005.
- [32] G. Ferreres and C. Roos. Efficient convex design of robust feedforward controllers. In *44th IEEE Conference on Decision and Control and European Control Conference. CDC-ECC'05.*, pages 6460–6465. IEEE, 2005.
- [33] R.A. Fisher. Theory of statistical estimation. *Proc. Cambridge Philos. Soc.*, 22:700–725, 1925.
- [34] P. J. Fleming and R. C. Purshouse. Genetic algorithms in control systems engineering. In *Proc. of the 12th IFAC World Congress*, pages 383–390, 2001.
- [35] P.J. Fleming and R.C. Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10:1223–1241, 2002.
- [36] C.M. Fonseca and P.J. Fleming. Multiobjective optimal controller design with genetic algorithms. In *Int. Conf. on Control'94*, volume 1, pages 745–749, 1994.
- [37] K. Fukuda. Polyhedral computation FAQ, 2004. URL: <http://www.ifor.math.ethz.ch/~fukuda/polyfaq/node22.html>, last retrieved: Sep. 2nd, 2011.
- [38] P. Gahinet and A. Nemirovskii. LMI Control Toolbox: the LMI Lab., 1995.
- [39] W. Gawronski. *Advanced structural dynamics and active control of structures*. Springer, New York, 2004.
- [40] A. Gelb, editor. *Applied Optimal Estimation*. The Analytic Sciences Corporation, MIT Press, 1974.
- [41] T. Glad and L. Ljung. *Control Theory: Multivariable and Nonlinear Methods*. Taylor & Francis Ltd., 2000.
- [42] K. Glover and J.C. Doyle. State-space formulae for all stabilizing controllers that satisfy an \mathcal{H}_∞ -norm bound and relations to risk sensitivity. *Systems and Control letters*, 11:167–172, 1988.
- [43] P.J. Goddard and K. Glover. Controller reduction: weights for stability and performance preservation. In *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, pages 2903–2908. IEEE, 1993.
- [44] G.H. Golub and C.F. van Loan. *Matrix computations*. The John Hopkins University Press, 3rd edition, 1996.
- [45] M.S. Grewal and A.P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. John Wiley & Sons, Inc., 2nd edition, 2001.
- [46] B. Grünbaum. *Convex Polytopes*. Graduate Texts in Mathematics. Springer, 2nd edition, 2003.
- [47] G. Grübel and H.-D. Joos. Multi-objective parameter synthesis (MOPS). In J.-F. Magni, S. Bennani, and J. Terlouw, editors, *Robust Flight Control: A Design Challenge*, pages 13–21. Garteur, 1997.
- [48] Y. Guan and M. Saif. A novel approach to the design of unknown input observers. *Automatic Control, IEEE Transactions on*, 36(5):632–635, may 1991.

- [49] S. Gugercin and A.C. Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8):748–766, 2004.
- [50] S. Gumussoy, D. Henrion, M. Millstone, and M. Overton. Multiobjective Robust Control with HIFOO 2.0. In *IFAC Symposium on Robust Control Design*, Haifa, Israel, 2009. LAAS-CNRS Research Report.
- [51] A. Hać. Distribution of actuators in vibration control of adaptive structures. In *Proceedings of the American Control Conference*, pages 4295–4299, Seattle, Washington USA, 1995.
- [52] A. Hać and L. Liu. Sensor and actuator location in motion control of flexible structures. *Journal of Sound and Vibration*, 167(2):239–261, 1993.
- [53] M. Hanel. *Robust Integrated Flight and Aeroelastic Control System Design for a Large Transport Aircraft*, volume 8. VDI-Verlag, 2001.
- [54] T. Hanis and M. Hromcik. Optimal sensors placement and elimination of undesirable mode shapes. In *Proc. of the European Control Conference 2009 (ECC 2009)*, MTA SZTAKI - Hungarian Academy of Sciences, Budapest, Hungary, 2009.
- [55] T. Hanis and M. Hromcik. Information-based sensor placement optimization for BWB aircraft. In *Preprints of the 18th IFAC World Congress*, pages 2236–2241, Milano, Italy, 2011.
- [56] S. Hecker. *Generation of low order LFT Representations for Robust Control Applications*. PhD thesis, Technische Universität München, Lehrstuhl für Steuerungs- und Regelungstechnik, 2006.
- [57] S. Hecker and A. Varga. Symbolic manipulation techniques for low order LFT-based parametric uncertainty modelling. *International Journal of Control*, 79(11):1485–1494, 2006.
- [58] S. Hecker, A. Varga, and J.F. Magni. Enhanced LFR-toolbox for MATLAB. *Aerospace science and technology*, 9(2):173–180, 2005.
- [59] M. Hemedi. *Input-Output optimization for Blended Wing Body civil transport aircraft (working title)*. Dissertation, Vienna University of Technology, 2011. in preparation.
- [60] M. Hemedi, A. Schirrer, C. Westermayer, and M. Kozek. Performance evaluation of an input/output selection criterion via normalized LQG closed-loop comparison. In *Proceedings of the 17th IEEE Mediterranean Conference on Control & Automation*, 2009.
- [61] M. Hemedi, A. Schirrer, C. Westermayer, and M. Kozek. Integrated input-output selection strategy for robust control of complex parameter varying systems. In *Proceedings of the 10th International Conference on Motion and Vibration Control (MOVIC 2010)*, number No. 10-203, Tokyo, Japan, 2010. Japan Society of Mechanical Engineers JSME.
- [62] M. Hemedi, A. Schirrer, C. Westermayer, and M. Kozek. Integrated input-output selection strategy for robust control of complex parameter depending systems. *Journal of System Design and Dynamics (Special Issue of Motion and Vibration Control 2010)*, 5(5):1106–1118, 2011. Invited.
- [63] D. Henrion, D. Arzelier, D. Peaucelle, and M. Sebek. An LMI condition for robust stability of polynomial matrix polytopes. *Automatica*, 37(3):461–468, 2001.

- [64] D. Henrion, J.B. Lasserre, and J. Löfberg. GloptiPoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.
- [65] D. Hestenes. *New Foundations for Classical Mechanics (Fundamental Theories of Physics)*. Springer / Kluwer, 2nd edition, 2008. (reprint).
- [66] D. Hestenes and G. Sobczyk. *Clifford algebra to geometric calculus: a unified language for mathematics and physics*. Springer Netherlands, 1984.
- [67] F.M. Hoblit. *Gust loads on aircraft: concepts and applications*. American Institute of Aeronautics and Astronautics, New York, 1988.
- [68] R.A. Horn and C.R. Johnson. *Matrix analysis*. Cambridge Univ Press, 1990.
- [69] H. Ito, H. Ohmori, and A. Sano. Design of stable controllers attaining low hinfinitly weighted sensitivity. *Automatic Control, IEEE Transactions on*, 38(3):485–488, 1993.
- [70] M. Jeanneau, J. Lamolie, G. Puyou, and N. Aversa. Awiators design of multi-objectives control laws. In *Proc. 16th IFAC World Congress*, Prague, Czech Rep., 2005.
- [71] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, STOC '84, pages 302–311, New York, NY, USA, 1984. ACM.
- [72] M.G. Kellet. Continuous scheduling of \mathcal{H}_∞ controllers for a ms760 paris aircraft. In *Robust control system design using \mathcal{H}_∞ and related methods*, pages 197–219. P.H. Hammond, 1991.
- [73] S.W. Kim, B.D.O. Anderson, and A.G. Madievski. Error bound for transfer function order reduction using frequency weighted balanced truncation. *Systems & Control Letters*, 24(3):183–192, 1995.
- [74] M. Kocvara and M. Stingl. PENBMI user's guide (version 2.1). URL: http://www.penopt.com/doc/penbmi2_1.pdf, last retrieved 8th Sep. 2011, 2006.
- [75] M. Kozek, C. Benatzky, A. Schirrer, and A. Stribersky. Vibration damping of a flexible car body structure using piezo stack actuators. *Control Engineering Practice*, 19(3):298–310, 2009. invited.
- [76] N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *Evolutionary Computation, IEEE Transactions on*, 9(5):474–488, 2005.
- [77] A.J. Krener. On the equivalence of control systems and the linearization of nonlinear systems. *Siam J. Control*, 11(4):670–676, 1973.
- [78] A. Kron, J. de Lafontaine, and D. Alazard. Robust 2-DOF H-infinity controller for highly flexible aircraft: Design methodology and numerical results. *Canadian Aeronautics and Space Journal*, 49:19–29, 2003.
- [79] F. Kubica and T. Livet. Design of flight control system for an aeroelastic aircraft. In *Control Applications, 1994., Proc. of the Third IEEE Conf. on*, volume 1, pages 335–340, 1994.
- [80] F. Kubica, T. Livet, X. Le Tron, and A. Bucharles. Parameter-robust flight control system for a flexible aircraft. *Control Engineering Practice*, 3(9):1209–1215, 1995.

- [81] D.J. Leith and W.E. Leithead. Survey of gain-scheduling analysis and design. *Int. J. Contr.*, 73(11):1001–1025, 2000.
- [82] S. Leleu, H. Abou-Kandil, and Y. Bonnasieux. Piezoelectric actuators and sensors location for active control of flexible structures. *IEEE Transactions on Instrumentation and Measurement*, 50(6):1577–1582, 2001.
- [83] K. Lenz, P. Khargonekar, and J. Doyle. When is a controller \mathcal{H}_∞ -optimal? *Mathematics of Control, Signals, and Systems (MCSS)*, 1:107–122, 1988. 10.1007/BF02551404.
- [84] F. Lewis. *Optimal Estimation*. John Wiley & Sons, 1986.
- [85] D.S. Li, H.N. Li, and C.P. Fritzen. The connection between effective independence and modal kinetic energy methods for sensor placement. *Journal of sound and vibration*, 305(4-5):945–955, 2007.
- [86] J. Löfberg. YALMIP: a toolbox for modeling and optimization in MATLAB. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.
- [87] M. Lovera and G. Mercère. Identification for gain-scheduling: a balanced subspace approach. In *Proc. 2007 American Cont. Conf.*, pages 858–863, 2007.
- [88] D. Luenberger. Observers for multivariable systems. *Automatic Control, IEEE Transactions on*, 11(2):190–197, 1966.
- [89] D. Luenberger. An introduction to observers. *Automatic Control, IEEE Transactions on*, 16(6):596–602, 1971.
- [90] D. G. Luenberger. Observing the state of a linear system. *IEEE Transactions on Military Electronics*, 8(2):74–80, 1964.
- [91] P. Lundström. *Studies on Robust Multivariable Distillation Control*. PhD thesis, University of Science and Technology, Trondheim, Norway, 1994.
- [92] A. Macdonald. *Linear and Geometric Algebra*. Createspace, 2010.
- [93] J.F. Magni. *Robust Modal Control with a Toolbox for Use with MATLAB*. Kluwer Academic, 2002.
- [94] G. Meinsma. Unstable and non-proper weights in \mathcal{H}_∞ control. *Automatica*, 31(1):1655–1658, 1995.
- [95] M. Meo and G. Zumpano. On the optimal sensor placement techniques for a bridge structure. *Engineering Structures*, 27(10):1488–1497, 2005.
- [96] B. Mialon and M. Hepperle. Flying wing aerodynamics studies at ONERA and DLR. In *CEAS Katnet Conference on Key Aerodynamic Technologies*, Germany, 2005.
- [97] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *Automatic Control, IEEE Transactions on*, 26(1):17–32, feb 1981.
- [98] B.G. Morton. New applications of mu to real-parameter variation problems. In *Decision and Control, 1985 24th IEEE Conference on*, volume 24, pages 233–238, dec 1985.
- [99] B.G. Morton and R.M. McAfoos. A mu-test for robustness analysis of a real-parameter variation problem. In *American Control Conference, 1985*, pages 135–138, jun 1985.

- [100] T. Nagado and S. Usui. A frequency-weighted reduction method for \mathcal{H}_∞ controllers. In *SICE 2002. Proceedings of the 41st SICE Annual Conference*, volume 1, pages 213–214. IEEE, 2002.
- [101] T. Nagado and S. Usui. Controller reduction via frequency-weightings considering \mathcal{H}_∞ property preservation. *IEEJ Transactions on Electronics, Information and Systems*, 127:247–252, 2007.
- [102] O. Nelles. *Nonlinear system identification*. Springer, Berlin, 2001.
- [103] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*, volume 13. SIAM, 1994.
- [104] R.A. Nichols, R.T. Reichert, and W.J. Rugh. Gain scheduling for H-infinity controllers: A flight control example. *Control Systems Technology, IEEE Transactions on*, 1(2):69–79, 1993.
- [105] B. Ninness and F. Gustafsson. A unifying construction of orthonormal bases for system identification. *Automatic Control, IEEE Transactions on*, 42(4):515–521, apr 1997.
- [106] R.C.L.F. Oliveira and P.L.D. Peres. LMI conditions for robust stability analysis based on polynomially parameter-dependent Lyapunov functions. *Systems & Control Letters*, 55(1):52–61, 2006.
- [107] A. Packard and J. C. Doyle. The complex structured singular value. *Automatica*, 29(1):71–109, 1993.
- [108] H. Panzer, J. Mohring, R. Eid, and B. Lohmann. Parametric model order reduction by matrix interpolation. *at - Automatisierungstechnik*, 58(8):475–484, 2010.
- [109] P.C. Pellanda, P. Apkarian, and D. Alazard. Gain-scheduling through continuation of observer-based realizations – applications to \mathcal{H}_∞ and μ controllers. In *Proc. 39th IEEE Conf. Dec. & Cont.*, pages 2787–2792, 2000.
- [110] K.B. Petersen and M.S. Pedersen. The matrix cookbook, 2008. URL: <http://orion.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf>, version Nov. 14, 2008.
- [111] H. Pfifer and S. Hecker. Generation of optimal linear parametric models for LFT-based robust stability analysis and control design. In *Proc. 47th IEEE Conf. on Decision & Control*, pages 3866–3871, Cancun, Mexico, 2008.
- [112] H. Pfifer and S. Hecker. Generation of optimal linear parametric models for LFT-based robust stability analysis and control design. *Control Systems Technology, IEEE Transactions on*, 19(1):118–131, jan 2011.
- [113] Y. Pigué, U. Holmberg, and R. Longchamp. A minimax approach for multi-objective controller design using multiple models. *International Journal of Control*, 72(7/8):716–726, 1999.
- [114] S. Popprath, C. Benatzky, C. Bilik, M. Kozek, A. Stribersky, and J. Wassermann. Experimental modal analysis of a scaled car body for metro vehicles. In *Proceedings of the 13th International Congress on Sound and Vibration (ICSV13)*, July 2–6, 2006. ISBN: 3-9501554-5-7, Paper ID 683.

- [115] S. Popprath, A. Schirrer, C. Benatzky, M. Kozek, and J. Wassermann. Experimental modal analysis of an actively controlled scaled metro vehicle car body. In *Proceedings of the 14th International Congress on Sound and Vibration (ICSV14)*, Cairns, Australia, July 9–12, 2007.
- [116] W.L. Poston, R.H. Tolson, and D.C. Kammer. Maximizing the determinant of the information matrix with the effective independence method. reply. *Journal of Guidance, Control, and Dynamics*, 15(6):1513–1514, 1992.
- [117] A. Preumont. *Vibration control of active structures: an introduction*. Kluwer, 1997.
- [118] G. Puyou and C. Chiappa. A multiobjective method for flight control law design. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–11, 2004.
- [119] F. O. Ramos. Observer-based form for a computational-intelligence designed satellite attitude controller. In *Proc. of the 3rd European Conf. for Aerospace Sciences*, 2009.
- [120] S. Roman. *Advanced Linear Algebra (Graduate Texts in Mathematics)*. Springer, 2009.
- [121] C. Roos. Generation of flexible aircraft LFT models for robustness analysis. In *Proceedings of the 6th IFAC Symposium on Robust Control Design*, pages 349–354, Oxford, UK, 2009. Elsevier Ltd.
- [122] W.J. Rugh and J.S. Shamma. Survey paper: Research on gain scheduling. *Automatica*, 36(10):1401–1425, 2000.
- [123] M.G. Safonov, D.J.N. Limebeer, and R.Y. Chiang. Simplifying the H_∞ theory via loop-shifting, matrix-pencil and descriptor concepts. *International Journal of Control*, 50(6):2467–2488, 1989.
- [124] M. Sanayei and C.N. Javdekar. Sensor placement for parameter estimation of structures using fisher information matrix. In *Applications of Advanced Technologies in Transportation (Proc. 7th Int. Conf. on Applications of Advanced Technology in Transportation)*, pages 385–386, Cambridge, MA, USA, 2002. American Society of Civil Engineers (ASCE).
- [125] G. Schandl. *Methodenuntersuchung zur aktiven Schwingungsreduktion eines Schienenfahrzeugwagenkastens*. Dissertation, Vienna University of Technology, Vienna, 2005.
- [126] A. Schirrer. Co-Simulation of Rail Car Body Vibration Control with SIMPACK[®]. Diploma thesis, Vienna University of Technology, Vienna, 2007.
- [127] A. Schirrer. *Co-Simulation of Rail Car Body Vibration Control with SimPACK[®]*. VDM Verlag, Saarbrücken, 2010.
- [128] A. Schirrer and M. Kozek. Co-simulation as effective method for flexible structure vibration control design validation and optimization. In *Proceedings of the 16th Mediterranean Conference on Control and Automation, MED'08*, Ajaccio, Corsica, France, 2008.
- [129] A. Schirrer and M. Kozek. Assessment of interpolation methods for Linear-Dynamic systems. In *Brno 2nd International Conference on Computational Intelligence, Modelling and Simulation 2011 (CSSim2011)*, 2011. Brno, Czech Republic.
- [130] A. Schirrer, M. Kozek, and C. Benatzky. Piezo stack actuators in flexible structures: Experimental verification of a nonlinear modeling and identification approach. In *Proceedings of the 6th EUROMECH Conference ENOC 2008*, St. Petersburg, Russia, 2008.

- [131] A. Schirrer, M. Kozek, A. Plank, M. Neumann, S. Badshah, and J. Wassermann. Vibration analysis of an actively controlled flexible structure using speckle interferometry. In *Proceedings of 15th International Congress on Sound and Vibration (ICSV15)*, Daejeon, Korea, July 2008.
- [132] A. Schirrer, M. Kozek, and J. Schöftner. MIMO Vibration Control for Flexible Rail Car Body: LQG and H2 Designs and Experimental Validation. In F. Beltran-Carbajal, editor, *Vibration Analysis and Control – New Trends and Developments*. Intech Publishing House, 2011.
- [133] A. Schirrer, C. Westermayer, M. Hemedi, and M. Kozek. A comprehensive robust control design and optimization methodology for complex flexible-structure systems. In *Control and Automation, 2010. MED '10. 18th Mediterranean Conference on*, Marrakech, Morocco, 2010.
- [134] A. Schirrer, C. Westermayer, M. Hemedi, and M. Kozek. LQ-based design of the inner loop lateral control for a large flexible BWB-type aircraft. In *2010 IEEE International Conference on Control Applications; Part of 2010 IEEE Multi-Conference on Systems and Control*, Yokohama, Japan, 2010. IEEE CCA.
- [135] A. Schirrer, C. Westermayer, M. Hemedi, and M. Kozek. Robust \mathcal{H}_∞ control design parameter optimization via genetic algorithm for lateral control of a bwb type aircraft. In *Proceedings of the Workshop on Intelligent Control Systems WICS2010*, available at <http://www.ifac-papersonline.net/>, Sinaia, Romania, 2010.
- [136] A. Schirrer, C. Westermayer, M. Hemedi, and M. Kozek. Actuator and sensor positioning optimization in control design for a large BWB passenger aircraft. *ISRN Mechanical Engineering*, vol. 2011, 2011.
- [137] A. Schirrer, C. Westermayer, M. Hemedi, and M. Kozek. Multi-model convex design of a scheduled lateral feedforward control law for a large flexible BWB aircraft. In *Preprints of the 18th IFAC World Congress*, pages 2126–2131, Milano, Italy, 2011.
- [138] A. Schirrer, C. Westermayer, M. Hemedi, and M. Kozek. Robust convex lateral feedback control synthesis for a BWB aircraft. In *Preprints of the 18th IFAC World Congress*, pages 7262–7267, Milano, Italy, 2011.
- [139] A. Schirrer, C. Westermayer, M. Hemedi, and M. Kozek. Robust feedback lateral control using a parameterized LFR model and DGK-iteration. In *Proc. of the 4th EUCASS (European Conf. f. Aerospace Sciences)*, St. Petersburg, Russia, 2011.
- [140] J. Schuler. *Flugregelung und aktive Schwingungsdämpfung für flexible Großraumflugzeuge (engl. Flight control and active vibration damping for large flexible aircraft)*. PhD thesis, Universität Stuttgart, Germany, 1997.
- [141] A. M. Simões, D. Alazard, H. D. Tuan, and P. Apkarian. Lateral flight control design for a highly flexible aircraft using a nonsmooth method. In *Proc. of Joint 48th IEEE Conf. on Decision and Control and 28th Chinese Control Conf.*, pages 696–701, 2009.
- [142] D. Sima. LMI optimization in connection with quadratic and nonlinear functions. In *Proceedings of the 3rd Niconet Workshop on Numerical Software in Control and Engineering*, pages 91–96, Belgium, 2001.
- [143] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. Wiley & Sons, 2nd edition, 2005.

- [144] V. Sreeram and B.D.O. Anderson. Frequency weighted balanced reduction technique: a generalization and an error bound. In *Proc. 34th IEEE Conf. on Decision and Control*, volume 4, pages 3576–3581, 1995.
- [145] B.L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, 2003.
- [146] M. Stingl. *On the Solution of Nonlinear Semidefinite Programs by Augmented Lagrangian Methods*. PhD thesis, Friedrich-Alexander Universität Erlangen-Nürnberg, Aachen, Germany, 2006.
- [147] F. Stroscher, Ö. Petersson, and M. Leitner. Structural optimization framework for aircraft subject to transient maneuver and gust loads. In *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, Texas, 2010.
- [148] F. Stroscher, Ö. Petersson, and M. Leitner. Aircraft structural optimization subject to flight loads - application to a wide body commercial aircraft configuration. In *EASN Intl. Workshop on Aerostructures*, 2010.
- [149] J.F. Sturm. Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods & Software*, 11(1):625–653, 1999.
- [150] E. Thonhofer. Interpolation of system dynamics. Master’s thesis, Vienna University of Technology, Vienna, 2011.
- [151] J. Torralba, G. Puyou, and F. Demourant. Self-scheduling multiobjective control law design for a flexible aircraft. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Chicago, USA, 2009.
- [152] M. van de Wal and B. de Jager. A review of methods for input/output selection. *Automatica*, 37:487–510, 2001. Survey Paper.
- [153] A. Varga and B.D.O. Anderson. Accuracy-enhancing methods for balancing-related frequency-weighted model and controller reduction. *Automatica*, 39(5):919–927, 2003.
- [154] M. Vidyasagar. *Control System Synthesis: A Factorization Approach, Part II*, volume 2. Morgan & Claypool Publishers, 2011. Reprint of MIT press edition (1985).
- [155] G. Wang, V. Sreeram, and W.Q. Liu. A new frequency-weighted balanced truncation method and an error bound. *IEEE Transactions on Automatic Control*, 44(9):1734–1737, 1999.
- [156] L. Wang, Z. Wang, L. Zhang, and W. Yu. Edge theorem for multivariable systems. *IMA Preprint Series*, 2002. 69925307, URL: <http://arxiv.org/abs/math/0211014>.
- [157] C. Westermayer. *2DOF parameter-dependent longitudinal control of a blended wing body flexible aircraft*. Dissertation, Vienna University of Technology, 2011.
- [158] C. Westermayer, A. Schirrer, M. Hemedi, and M. Kozek. An advanced criterion for optimal actuator and sensor placement on complex flexible structures. In *Proc. of the IFAC Workshop on Control Applications of Optimisation 09*, Jyväskylä, Finland, 2009. URL: <http://www.ifac-papersonline.net/>.
- [159] C. Westermayer, A. Schirrer, M. Hemedi, and M. Kozek. Linear parameter-varying control of a large blended wing body flexible aircraft. In *Proceedings of the 18th IFAC Symposium on Automatic Control in Aerospace*, Nara, Japan, 2010.

- [160] C. Westermayer, A. Schirrer, M. Hemedi, and M. Kozek. An h-infinity full information approach for the feedforward controller design of a large BWB flexible aircraft. In *Proc. of the 4th EUCASS (European Conf. f. Aerospace Sciences)*, St. Petersburg, Russia, 2011.
- [161] C. Westermayer, A. Schirrer, M. Hemedi, M. Kozek, T. Hanis, and M. Hromcik. Optimal sensor placement in flexible structures: energy vs. information based criteria. In *Proceedings of the 6th Mathmod Vienna 09*, number 35, Vienna, Austria, 2009. Argesim Report.
- [162] C. Westermayer, A. Schirrer, M. Hemedi, M. Kozek, and A. Wildschek. Robust H-inf flight and load control of a flexible aircraft using a 2DOF multi-objective design. In *2009 CACS International Automatic Control Conference*, Taipei, Taiwan, 2009.
- [163] Wikipedia. Object-oriented programming — Wikipedia, The Free Encyclopedia, 2011. [Online; accessed August 25th, 2011].
- [164] A. Wildschek. *An Adaptive Feed-Forward Controller for Active Wing Bending Vibration Alleviation on Large Transport Aircraft*. PhD thesis, Technische Universität München, 2009.
- [165] A. Wildschek, R. Maier, M. Hromcik, T. Hanis, A. Schirrer, M. Kozek, C. Westermayer, and M. Hemedi. Hybrid controller for gust load alleviation and ride comfort improvement using direct lift control flaps. In *Proc. of the 3rd EUCASS (European Conf. f. Aerospace Sciences)*, Paris, France, 2009.
- [166] M.A. Woodbury. Inverting modified matrices. Technical Report Memorandum Report 42, Statistical Research Group, Princeton University, Princeton, N.J., USA, 1950.
- [167] P.M.R. Wortelboer, M. Steinbuch, and O.H. Bosgra. Iterative model and controller reduction using closed-loop balancing, with application to a compact disc mechanism. *International Journal of Robust and Nonlinear Control*, 9(3):123–142, 1999.
- [168] F. Wu. *Control of Linear Parameter Varying Systems*. PhD thesis, University of California at Berkeley, 1995.
- [169] YALMIP wiki main/home page, 2011. URL: <http://users.isy.liu.se/johanl/yalmip/>, last retrieved Sep. 8th 2011.
- [170] L. Yao, W.A. Sethares, and D.C. Kammer. Sensor placement for on-orbit modal identification via a genetic algorithm. *AIAA journal*, 31(10):1922–1928, 1993.
- [171] D.C. Youla, J.J. Bongiorno, and C.N. Lu. Single-loop feedback-stabilization of linear multivariable dynamical plants. *Automatica*, 10(2):159–173, 1974.
- [172] P.M. Young. Controller design with mixed uncertainties. In *American Control Conference, 1994*, volume 2, pages 2333–2337, 1994.
- [173] K. Zhou. Frequency-weighted model reduction with \mathcal{L}_∞ error bounds. *Systems & Control Letters*, 21(2):115–125, 1992.
- [174] K. Zhou, J. C. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, 1996.
- [175] K. Zhou, C. D’Souza, and J.R. Cloutier. Structurally balanced controller order reduction with guaranteed closed loop performance. *Systems & Control Letters*, 24:235–242, 1995.

Alexander Schirrer

Curriculum Vitae

September 2011

Address: Institute of Mechanics and Mechatronics, Vienna
University of Technology, Austria.
Phone: +43 1 58801 325521
Email: alexander.schirrer@tuwien.ac.at

Personal Information

born June 18th, 1981; Vienna
citizenship Austrian
marital status married, 1 child

Education and Qualifications

2001-2005	Bachelor degree: Business Informatics	Vienna University of Technology & University of Vienna, Austria
2005-2006	Erasmus stay (2 semesters)	KTH Stockholm, Sweden
2001-2007	Diploma degree: Mechanical Engineering	Vienna University of Technology, Austria
2007-2011	Doctoral studies: Mechanical Engineering (Robust Control)	Vienna University of Technology

Research Interests, Publications & Awards

In the field of mechanical engineering I am focusing on robust vibration control of flexible structures, including automation and control, Finite Element Methods and non-linear dynamics. In the field of computer science I am concentrating on numerical computation problems, optimization algorithms for convex and non-convex problem classes, and heuristic search algorithms. I have authored 27 papers, chapters, or books. I presented my first paper during my Business Informatics studies at the Metaheuristics Intl. Conference 2005; it was 2007 accepted as a book chapter. Focusing on modeling, simulation, and control in the context of mechanical engineering, the research results related to my diploma and doctoral research projects have been published from 2007 on. A list of these appears on pages 2–4.

Awards

2002 Scholarship for exceptional performance in studies, Vienna University of Technology
2005 Tabu Search Award within the course “approximation algorithms”, Vienna University of Technology
2008 Co-Author of a paper awarded the Best Application Paper Award at the IFAC World Congress 2008
2009 Participant in the FFG/ISB research project “Active vibration control for Metro vehicles” which was nominated for the 2009 Austrian state prize, section traffic

Language Skills

C2+	(negotiation level)	German, English
B2	(fluent communication)	Swedish, Romanian
A2..B1	(basic communication)	French, Spanish
A1	(beginner)	Italian, Russian, Greek, Czech

Work Experience

- 2007-2011 University assistant at the Vienna University of Technology, Institute of Mechanics and Mechatronics: Scientific research in active vibration control, robust control, identification and laboratory experiments, as well as teaching tasks.
- 2006 Embedded C development, implementation, and testing of optimal control algorithms, KTH Stockholm, Sweden as part of a student project in automatic control
- 2005 Development of heuristic algorithms to optimize an assignment problem for the Metaheuristics Intl. Conference (MIC) 2005 (C++ and Java programming) at the Institute of Production & Logistics of the University of Vienna
- 2004 Development of printer firmware (software engineering and architecture-specific C programming) at Sagem, Vienna
- 2003 J2EE development within an application framework during an international traineeship in Prague, Czech Republic (10 week IAESTE traineeship). Advancement to the vice teamleader of our development workgroup.
- earlier Project participation in the development of steam sterilization plants, tunnel radio communication plants, in-company business process reengineering, web design, software engineering, IT support

Publications

Books

1. Schirrer, A. (2010). *Co-Simulation of Rail Car Body Vibration Control with SimPACK®*. Saarbrücken: VDM Verlag, 123 pages. ISBN: 978-3-639-30692-7.

Book chapters

1. Schirrer, A., M. Kozek, and J. Schoeftner (2011). MIMO Vibration Control for Flexible Rail Car Body: LQG and H2 Designs and Experimental Validation. In: *Vibration Control*. Ed. by F. Beltran-Carbajal. In press. Intech Publishing House.
2. Schirrer, A., K. Doerner, and R. Hartl (2007). Reviewer Assignment for Scientific Articles Using Memetic Algorithms. In: *Metaheuristics: progress in complex systems optimization*. Ed. by K. Doerner. Springer, pp.113–134.

Publications in scientific journals (refereed)

1. Hemedi, M., A. Schirrer, C. Westermayer, and M. Kozek (2011). 10085 Integrated Input-output Selection Strategy for Robust Control of Complex Parameter Depending Systems. *Special Issue of Motion and Vibration Control 2010 (Journal of System Design and Dynamics)*. Invited; accepted for publication.
2. Schirrer, A., C. Westermayer, M. Hemedi, and M. Kozek (2011). Actuator and sensor positioning optimization in control design for a large BWB passenger aircraft. *ISRN Mechanical Engineering* **2011**, 11. doi: 10.5402/2011/635815.
3. Kozek, M., C. Benatzky, A. Schirrer, and A. Stribersky (2009). Vibration Damping of a Flexible Car Body Structure Using Piezo Stack Actuators. *Control Engineering Practice* **19**(3). invited, 298–310. doi: 10.1016/j.conengprac.2009.08.001.

Talks and poster presentations (with proceedings entry, refereed)

1. Paces, N., A. Voigt, S. Jakubek, A. Schirrer, and M. Kozek (2011). Combined Control of Combustion Load and Combustion Position in a Moving Grate Biomass Furnace. In: *Proc. of the 2011 Mediterranean Control Conference*. Corfu, Greece.
2. Schirrer, A. and M. Kozek (2011). Assessment of Interpolation Methods for Linear-Dynamic Systems. In: *Brno 2nd International Conference on Computational Intelligence, Modelling and Simulation 2011 (CSSim2011)*. Brno, Czech Republic.
3. Schirrer, A., C. Westermayer, M. Hemedi, and M. Kozek (2011). Multi-model convex design of a scheduled lateral feedforward control law for a large flexible BWB aircraft. In: *Proc. of the IFAC World Congress 2011*. Accepted for publication. Milan, Italy.

4. Schirrer, A., C. Westermayer, M. Hemedi, and M. Kozek (2011). Robust convex lateral feedback control synthesis for a BWB aircraft. In: *Proc. of the IFAC World Congress 2011*. Accepted for publication. Milan, Italy.
5. Hemedi, M., A. Schirrer, C. Westermayer, and M. Kozek (2010). Integrated input-output selection strategy for robust control of complex parameter varying systems. In: *Proceedings of the 10th International Conference on Motion and Vibration Control*. Talk: 10th International Conference on Motion and Vibration Control (MOVIC 2010), Tokyo, Japan; 2010-08-17 – 2010-08-20. No. 10-203, Tokyo, Japan: Japan Society of Mechanical Engineers JSME.
6. Schirrer, A., C. Westermayer, M. Hemedi, and M. Kozek (2010). A Comprehensive Robust Control Design and Optimization Methodology for Complex Flexible-Structure Systems. In: *Control and Automation, 2010. MED '10. 18th Mediterranean Conference on*. talk: 18th Mediterranean Conference on Control & Automation, Marrakesch, Marokko; 2010-06-23 – 2010-06-25. ISBN: 978-1-4244-8092-0.
7. Schirrer, A., C. Westermayer, M. Hemedi, and M. Kozek (2010). LQ-based design of the inner loop lateral control for a large flexible BWB-type aircraft. In: *2010 IEEE International Conference on Control Applications; Part of 2010 IEEE Multi-Conference on Systems and Control*. talk: 2010 IEEE Multi-Conference on Systems and Control, A port city on Tokyo Bay, Yokohama, Kanagawa, JAPAN; 2010-09-08 – 2010-09-10. CFP10CCA-CDR: Ieee Cca. ISBN: 978-1-4244-5363-4.
8. Westermayer, C., A. Schirrer, M. Hemedi, and M. Kozek (2010). Linear parameter-varying control of a large blended wing body flexible aircraft. In: *Proceedings of the 18th IFAC Symposium on Automatic Control in Aerospace*. talk: 18th IFAC Symposium on Automatic Control in Aerospace, Nara, Japan; 2010-09-06 – 2010-09-10.
9. Hemedi, M., A. Schirrer, C. Westermayer, and M. Kozek (2009). Performance evaluation of an input/output selection criterion via normalized LQG closed-loop comparison. In: *Proceedings of the 17th IEEE Mediterranean Conference on Control & Automation*. talk: 17th IEEE Mediterranean Conference on Control & Automation, Thessaloniki, Greece; 2009-06-24 – 2009-06-26. ISBN: 978-1-4244-4685-8.
10. Westermayer, C., A. Schirrer, M. Hemedi, and M. Kozek (2009). An advanced criterion for optimal actuator and sensor placement on complex flexible structures. In: *IFAC-PapersOnLine*. talk: IFAC Workshop on Control Applications of Optimisation 09, Jyväskylä, Finland; 2009-05-06 – 2009-05-08.
11. Westermayer, C., A. Schirrer, M. Hemedi, M. Kozek, T. Hanis, and M. Hromcik (2009). Optimal sensor placement in flexible structures: energy vs. information based criteria. In: *Proceedings of the 6th Mathmod Vienna 09*. talk: 6th Vienna Mathmod 09, Vienna; 2009-02-11 – 2009-02-13. No. 35: Argesim Report. ISBN: 978-3-901608-35-3.
12. Westermayer, C., A. Schirrer, M. Hemedi, M. Kozek, and A. Wildschek (2009). Robust H-inf flight and load control of a flexible aircraft using a 2DOF multi-objective design. In: *2009 CACS International Automatic Control Conference*. talk: 2009 CACS International Automatic Control Conference, Taipei, Taiwan; 2009-11-27 – 2009-11-29. ISBN: 978-986-84845-1-1.
13. Kozek, M., C. Benatzky, A. Schirrer, and A. Stribersky (2008). Vibration Damping of a Flexible Car Body Structure Using Piezo-Stack Actuators. In: *Proceedings of the 17th IFAC World Congress*. invited; talk: 17th IFAC World Congress, Seoul, Korea; 2008-07-06 – 2008-07-11.
14. Schirrer, A. and M. Kozek (2008). Co-Simulation as Effective Method for Flexible Structure Vibration Control Design Validation and Optimization. In: *Proceedings of the 16th Mediterranean Conference on Control and Automation, MED'08*. talk: 16th Mediterranean Conference on Control and Automation, MED'08, Ajaccio, Corsica, France; 2008-06-25 – 2008-06-27. ISBN: 978-1-4244-2505-1.
15. Schirrer, A., M. Kozek, and C. Benatzky (2008). Piezo stack actuators in flexible structures: Experimental verification of a nonlinear modeling and identification approach. In: *Proceedings of the 6th EUROMECH Conference ENOC 2008*. talk: 6th EUROMECH Conference ENOC 2008, Russia; 2008-06-30 – 2008-07-04.

Talks and poster presentations (with proceedings entry, not refereed)

1. Schirrer, A., C. Westermayer, M. Hemedi, and M. Kozek (2011). Robust feedback lateral control using a parameterized LFR model and DGK-iteration. In: *Proc. of the 4th Eucass (European Conf. f. Aerospace Sciences)*. St. Petersburg, Russia.
2. Westermayer, C., A. Schirrer, M. Hemedi, and M. Kozek (2011). An H-infinity full information approach for the feedforward controller design of a large BWB flexible aircraft. In: *Proc. of the 4th Eucass (European Conf. f. Aerospace Sciences)*. St. Petersburg, Russia.
3. Wildschek, A., R. Maier, M. Hromcik, T. Hanis, A. Schirrer, M. Kozek, C. Westermayer, and M. Hemedi (2009). Hybrid controller for gust load alleviation and ride comfort improvement using direct lift control flaps. In: *Proc. of the 3rd Eucass (European Conf. f. Aerospace Sciences)*. Paris, France.

4. Kozek, M., C. Benatzky, A. Schirrer, and A. Stribersky (2008). Aktive Schwingungsdämpfung an einer elastischen Wagenkasten Struktur. In: *Proceedings GMA Workshops FA 1.30 and 1.40 in Salzburg/Anif, Austria 2008*. Talk: GAMM-GMA-Workshop, Salzburg (invited).
5. Schirrer, A., M. Kozek, A. Plank, M. Neumann, S. Badshah, and J. Wassermann (2008). Vibration analysis of an actively controlled flexible structure using speckle interferometry. In: *Proceedings of 15th International Congress on Sound and Vibration (ICSV15), Daejeon, Korea*.
6. Popprath, S., A. Schirrer, C. Benatzky, M. Kozek, and J. Wassermann (2007). Experimental modal analysis of an actively controlled scaled metro vehicle car body. In: *Proceedings of the 14th International Congress on Sound and Vibration (ICSV14), Cairns, Australia*.