TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Steuerungs- und Regelungstechnik

Stability & Transparency Analysis of Teleoperation Systems

Thomas Peter Schauß

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Eckehard Steinbach

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss
2. Prof. Dr.-Ing. Angelika Peer,
   University of the West of England, Bristol

Die Dissertation wurde am 14.06.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 23.01.2017 angenommen.

Foreword

This thesis represents the research I have conducted during the past years at the Institute of Automatic Control Engineering (LSR) at TU München. Looking back now, the content of this thesis is quite different from my initial goals. I started out doing research on multi-user teleoperation within the German research project SFB-453 *High-Fidelity Telepresence and Teleaction* and the European research project Robot@CWE. The deeper I dug into this field, the more it became evident that there was a lack of tools which would be necessary to start thoroughly examining such systems. I believe that the tools and methods developed in this thesis are a big step in this direction.

A much greater number of people actually contributes to a thesis in some form than fits on the title page and I would like to thank you all. Prof. Martin Buss for providing an outstanding working environment, for critical comments, and for the freedom at the institute. Prof. Angelika Peer for supporting me when I changed the focus of my research, for fruitful discussions, and for fast and concise feedback. I am grateful for the opportunity to visit the McMaster University in Hamilton, Canada for several weeks: thanks to the SFB-453, for providing this opportunity, thank you Carolina and Benjamin Passenberg, and Pawel Malysz for the nice time we spent in Canada, and thank you to Prof. Sirouspour for invigorating discussions and a warm welcome. Thank you to my colleagues Ulrich Unterhinninghofen, Daniela Feth, and Nikolay Stefanov with whom I worked in the SFB-453: we had many interesting discussions and it was always fun working with you. During my time at the institute I was a system administrator for the institute servers and workstations for three years together with Wolfgang Jaschik, Jens Hölldampf, Sebastian Erhard, Stefan Sosnowski, and Mathias Rungger - working with you was a pleasure. My thanks also go to Larissa Schmid and Waltraud Werner for their administrative support and Tobias Stöber, Thomas Lowitz, Josef Gradl, Horst Kubick, and Domenik Weilbach for their technical support, especially during the SFB-453.

I would like to thank Stefan Sosnowski with whom I shared an office and had many enjoyable discussions and Tobias Göpel who was my office mate much too short. I would also like to thank my colleagues and friends which made life at the institute (and especially the coffee breaks) so enjoyable: Georg Bätz, Daniel Althoff, Markus Rank, Raphaela Groten, Michael Scheint, Klaas Klasing, Mathias Rambow, Martin Lawitzky, Ken Friedl, Andreas Lawitzky, José Medina, Alexander Pekarovskiy - thank you very much!

I want to thank my parents, my sister, and my parents-in-law for their constant encouragement and support. Finally, my deepest gratitude goes to my wife Chrissy and my son Alexander, who provided me with the time I needed to finish the dissertation and were the biggest possible incentive. Chrissy, you encouraged and supported me at all times and gave me a push or two when necessary - this wouldn't have been possible without you!

Munich, February 2016 Thomas Schauß

Abstract

The topic of this thesis is the stability and transparency analysis of haptic teleoperation systems. Such systems enable an operator to manipulate objects in a remote environment which might be far away, dangerous, or on a scale other than the human workspace. As is common in literature, the overall teleoperation system including operator and remote object, is modeled as a linear time-invariant system, and parametric uncertainties are used to represent the unknown dynamic properties of operator and environment. Two major objectives are generally pursued when designing teleoperation systems or parametrizing controllers. On the one hand, the system must be robustly stable when an operator with unknown impedance interacts with different remote environments. On the other hand, the system should be as transparent as possible, i.e., the operator should feel as if he is directly interacting with the remote environment.

To analyze the robust stability of teleoperation systems a new stability analysis method for time-delay systems with parametric uncertainties is developed in this thesis. The general idea is to determine stable regions in a low-dimensional parameter space. As additional parameters may be unknown but constrained to an interval, the method is well suited for the controller design of uncertain time-delay systems. The method is based on evaluating the value set of the characteristic function of the time-delay system, using a combination of Taylor Models and polynomials in Bernstein form. This allows for the formulation of an efficient branch and bound algorithm which makes use of the zero-exclusion principle to first map boundaries to the parameter space and then examine the stability of disjoint regions. The method is not limited to teleoperation systems but can be applied to a large class of time-delay systems including systems with multiple incommensurate delays and non-affine parameter dependencies. In contrast to existing robust stability analysis methods for time-delay systems with incommensurate delays which are generally quite conservative, the conservatism of our method only depends on the specifiable resolution with which boundaries are determined in the parameter space. Moreover, the results are non-conservative if stability is evaluated for a set of interval parameters instead of mapping the results to the parameter space.

The stability analysis method is complemented by a newly developed transparency analysis method: the distortion caused by the teleoperation system is determined for the whole range of relevant environment parameters which results in an intuitive parameter-space transparency analysis of teleoperation systems. The newly developed stability and transparency analysis methods are applied to a number of different teleoperation architectures and the suitability of different architectures is evaluated for different time delays. Stability and transparency generally deteriorate for increasing time delay, but some architectures, e.g., the transparency-optimized four-channel architecture and the position-based admittance control architecture can be parametrized so as to offer good transparency for very small time delays while being robustly stable for larger time delays.

Finally, a natural extension of the stability analysis method towards optimal parametrization of linear time-invariant systems with incommensurate time delays and parametric uncertainties is outlined.

Zusammenfassung

Das Thema dieser Arbeit ist die Stabilitäts- und Transparenzanalyse von haptischen Telepräsenzsystemen. Solche Systeme ermöglichen es einem Menschen Arbeiten in einer entfernten Umgebung durchzuführen. Wie in der Literatur üblich wird das Telepräsenzsystem als linear zeitinvariantes System modelliert, wobei die unbekannte Dynamik von Mensch und entfernter Umgebung durch parametrische Unsicherheiten modelliert wird. Zwei Ziele, die bei der Entwickelung und Reglerparametrierung von Telepräsenzsystemen in der Regel verfolgt werden sind die robuste Stabilität des Gesamtsystems, unabhängig von der Umgebung mit der man interagiert, sowie eine möglichst hohe Transparenz, also eine möglichst natürliche Darstellung der entfernten Umgebung.

Zur robusten Stabilitätsanalyse von Telepräsenzsystemen wird in dieser Arbeit eine neue Methode vorgestellt, welche für ein totzeitbehaftetes System stabile und instabile Regionen in einem niedrig-dimensionalen Parameterraum bestimmt. Die Methode eignet sich auch zum Reglerentwurf weil weitere Parameter auf ein Interval beschränkt sein können. Die Grundidee besteht darin, die Wertemenge der charakteristischen Funktion der Systemmatrix zu bestimmen und diese auf Nullausschluss zu überprüfen. Wegen der Totzeiten handelt es sich bei der charakteristischen Funktion um ein Quasipolynom welches durch ein Taylorpolynom mit Restintervall überapproximiert wird. Das Polynom wird in Bernsteinform transformiert und ein Branch and Bound Algorithmus bestimmt Nulleinschluss und -ausschluss. Die Methode ist nicht auf Telepräsenzsysteme beschränkt sondern lässt sich auf eine große Klasse totzeitbehafteter Systeme mit parametrischen Unsicherheiten anwenden, auch auf Systeme mit mehreren inkommensurablen Totzeiten. Wie konservativ die Ergebnisse der vorgestellten Methode sind, lässt sich direkt spezifizieren, wobei ein weniger konservatives Ergebnis lediglich zu einer längeren Berechnungszeit führt. Dies steht im Gegensatz zu existierende Stabilitätsanalyseverfahren für totzeitbehaftete System welche im Allgemeinen sehr konservativ sind, speziell im Fall mit mehreren Totzeiten, aber auch im Fall mit einer Totzeit und zusätzlichen parametrischen Unsicherheiten.

Die Stabilitätsanalysemethode wird duch eine neu entwickelte Methode zur Transparenzanalyse von Telepräsenzsystemen ergänzt. Kern dieser Methode ist die Untersuchung der Abweichung zwischen Umgebungsimpedanz und dem Bediener dargestellter Impedanz im Parameterraum. Die neu entwickelten Methoden zur Stabilitäts- und Transparenzanalyse werden verwendet um verschiedene Telepräsenzarchitekturen zu untersuchen. Der Hauptfokus dieser Untersuchung ist der Effekt kleiner aber nicht vernachlässigbare Totzeiten auf die Eigenschaften der verschiedenen Architekturen. Im Allgemeinen nimmt sowohl die Transparenz als auch die Größe des stabilen Parameterraums für steigende Totzeiten ab. Einige Architektueren, beispielsweise die transparenzoptimierte Vierkanalarchitektur oder die positionsbasierte Admittanzregelung mit Kraftaustausch können aber so parametriert werden, dass sie im Fall sehr kleiner Zeitverzögerungen sehr gute Transparenzeigenschaften bieten und auch für etwas größere Zeitverzögerungen Stabilität sichergestellt wird.

Abschließend wird ein Ausblick auf eine mögliche Erweiterung des Stabilitätsanalyseverfahrens hin zu einer automatischen Parametrierung linear zeitinvarianter Systeme mit mehreren Zeitverzögerungen und parametrischen Unsicherheiten gegeben.

Contents

1 Introduction 1
1.1 Problem Statement . . . 2
1.2 Contributions and Outline of the Thesis . . . 3

2 Mathematical Foundations 5
2.1 Definition of Value Set . . . 6
2.2 Intervals . . . 6
2.2.1 Arithmetic . . . 7
2.2.2 Elementary Functions . . . 7
2.2.3 Complex Intervals . . . 7
2.2.4 Approximation of Value Set . . . 8
2.3 Polynomials . . . 8
2.3.1 Notation . . . 8
2.3.2 Definition . . . 9
2.3.3 Normalization . . . 10
2.3.4 Bernstein Transformation . . . 10
2.3.5 Approximation of Value Set . . . 11
2.3.6 Bernstein Algorithms . . . 11
2.3.7 Value Set of Complex Polynomials . . . 14
2.4 Taylor Models . . . 16
2.4.1 Definition . . . 17
2.4.2 Arithmetic . . . 18
2.4.3 Elementary Functions . . . 19
2.4.4 Approximation of Value Set . . . 20
2.4.5 Value Set of Complex Taylor Models . . . 21
2.5 Summary . . . 23

3 Robust Stability Analysis of Time-Delay Systems 25
3.1 State of the Art . . . 26
3.1.1 Robust Stability Analysis of Delay-Free Systems . . . 27
3.1.2 Stability Analysis of Time-Delay Systems . . . 28
3.1.3 Robust Stability Analysis of Time-Delay Systems . . . 30
3.2 Problem Formulation and General Idea . . . 32
3.3 Overall Algorithm . . . 34
3.4 Regions in the Complex Plane . . . 35
3.4.1 Zero-Exclusion Principle . . . 35

3.4.2 Asymptotic Stability 38
3.4.3 Minimum Damping 39
3.4.4 Minimum Decay Rate 39
3.5 Transformation of Unbounded Regions 39
3.5.1 Asymptotic Stability 41
3.5.2 Minimum Damping 42
3.5.3 Time Delay 43
3.6 Branch and Bound Algorithm 44
3.6.1 Evaluation of Zero Exclusion and Inclusion 46
3.6.2 Desired Resolution 49
3.6.3 Subdivision 50
3.7 Example for Robust Stability Analysis 52
3.7.1 System Model 52
3.7.2 Hurwitz Stability 53
3.7.3 Minimum Damping 55
3.7.4 Time Delay 57
3.8 Examples for Stability Analysis with Incommensurate Delays 61
3.8.1 One time delay 61
3.8.2 Two incommensurate time delays 64
3.8.3 Three incommensurate time delays 65
3.9 Discussion 66
3.9.1 Computational Complexity 66
3.9.2 Conservatism 67
3.9.3 Marginal Stability 67
3.9.4 Marginal Stability for Time-Delay Systems 68
3.9.5 Convergence 68
3.9.6 BIBO-Stability 69
3.9.7 Mapping of unbounded regions 69
3.9.8 Comparability of Results 69
3.10 Summary and Future Work 70

4 Stability and Transparency Analysis of Teleoperation Systems 73
4.1 State of the Art 75
4.1.1 Stability Analysis 75
4.1.2 Transparency Analysis 76
4.2 System Model 77
4.2.1 Modeling Assumptions 77
4.2.2 Teleoperation Architecture 79
4.2.3 Human 82
4.2.4 Environment 82
4.2.5 Haptic Interface (Master) 83
4.2.6 Teleoperator (Slave) 83
4.3 Parameter-Space Stability Analysis 84
4.4 Parameter-Space Transparency Analysis 85
4.4.1 Transparency Transfer Function 85

4.4.2 System Identification 86
4.4.3 Graphical Analysis 87
4.5 Examples 88
4.5.1 Model Parameters 88
4.5.2 Force-Position Architecture (FP) 90
4.5.3 Force-Position Architecture with Wave Variables (FP-W) 100
4.5.4 Four-Channel Architecture (4C) 108
4.5.5 Position-Based Admittance Control with Force Exchange (FaFa) . . 116
4.6 Summary and Future Work 123

5 Towards Optimal Parametrization of LTI Systems 127
5.1 State of the Art 128
5.2 Problem Formulation and General Idea 129
5.3 Performance Specification 131
5.4 Two-Step Solution of the Constrained Optimization Problem 133
5.4.1 Step One: Stability and Performance Constraints 134
5.4.2 Step Two: Performance Optimization 135
5.5 Discussion 136
5.5.1 Necessary Extensions to Taylor-Bernstein Form 136
5.5.2 One-Step Solution of the Constrained Optimization Problem 137
5.6 Summary and Future Work 138

6 Conclusions and Outlook 139
6.1 Outlook 141

A Stability of LTI-Systems 143
A.1 Stability of LTI-Systems without Time Delay 143
A.1.1 Asymptotic Stability 143
A.1.2 Marginal Stability 143
A.1.3 Instability 144
A.2 Stability of LTI-Systems with Time Delay 144

Bibliography 147

Notations

Abbreviations

DoF	Degrees of Freedom
BIBO	Bounded Input Bounded Output
LTI	Linear Time-Invariant
RFDE	Retarded Functional Differential Equation

Conventions

Scalars, Vectors, and Matrices

Scalars are denoted by lower or upper case letters in italic type. *Vectors* are denoted by lower case letters in boldface, and the vector $\boldsymbol{x}$ is composed of elements x_i. *Matrices* are denoted by upper case letters in boldface, and the matrix $\boldsymbol{X}$ is composed of elements x_{ij} (i^{th} row, j^{th} column).

x or X	scalar
$\boldsymbol{x}$	vector
$\boldsymbol{X}$	matrix
$f(\cdot)$	scalar function
$\boldsymbol{f}(\cdot)$	vector function
$\dot{x}, \ddot{x}, \ldots$	first, second, ... time derivative of x
$X(\boldsymbol{s})$	frequency transform of $x(t)$
X	closed interval
$\underline{X}$	lower bound of interval X
$\overline{X}$	upper bound of interval X
$\boldsymbol{X}$	interval box

Subscripts and Superscripts

x_i	i^{th} component of vector $\boldsymbol{x}$
x_{ij}	element of matrix $\boldsymbol{X}$ at row i and column j
X_i	i-th element of interval box $\boldsymbol{X}$

Symbols

General

$f(\boldsymbol{X})$	value set of all values $f(\boldsymbol{x})$ for $\boldsymbol{x} \in \boldsymbol{X}$
u	system input
y	system output
$\boldsymbol{x}$	system state
t	time
$\boldsymbol{A}$	state matrix
$\boldsymbol{b}$	input vector
$\boldsymbol{c}$	output vector
d	feedthrough scalar
$\boldsymbol{s}$	Laplace operator
$\boldsymbol{Q}$	interval box of uncertain system parameters
$\boldsymbol{q}$	uncertain system parameters, $\boldsymbol{q} \in \boldsymbol{Q}$
$\boldsymbol{Q}$	interval box of uncertain time delays
$\boldsymbol{t}_d$	uncertain time delays, $\boldsymbol{t}_d in \boldsymbol{T}_d$
$\boldsymbol{\Gamma}$	region in the complex plane
$\partial\boldsymbol{\Gamma}$	boundary of the region $\boldsymbol{\Gamma}$
$G(\boldsymbol{s})$	transfer function
$Z(\boldsymbol{s})$	mechanical impedance
$Y(\boldsymbol{s})$	mechanical admittance
m	inertia
b	viscous damping
c	stiffness

Mathematical Foundations

X, A, B	intervals
$\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$	interval boxes
U	unit interval
$\boldsymbol{U}$	unit interval box
I, J	l-tuples representing a multi-indices
$p(\boldsymbol{x})$	multivariate polynomial
$\tilde{p}(\boldsymbol{u}, \boldsymbol{X})$	normalized multivariate polynomial
a_I	coefficient with multi-index I of multivariate polynomial
$\tilde{a}_I(\boldsymbol{X})$	coefficient with multi-index I of normalizedmultivariate polynomial
$b_I(\boldsymbol{X})$	Bernstein coefficient with multi-index I
$B_{N,I}(\boldsymbol{u})$	Ith Bernstein polynomial of degree N
S_0	multi-indices of the vertices of the interval box
$\boldsymbol{H}(\boldsymbol{X})$	convex hull of complex bernstein coefficients
E_1, E_2, E_3, E_4	multi-indices of edges of interval box
$\boldsymbol{H}_i(\boldsymbol{X})$	convec hull of Bernstein coefficients with multi-indices E_i
$T(\boldsymbol{X})$	Taylor Model consisting of normalized polynomial $\tilde{p}(\boldsymbol{u}, \boldsymbol{X})$ and interval remainder $R(\boldsymbol{X})$

c_T	constant part of the Taylor Model $T(\boldsymbol{X})$
$\bar{T}(\boldsymbol{X})$	constant-free Taylor Model $T(\boldsymbol{X})$

Robust Stability Analysis of Time-Delay Systems

$\boldsymbol{\Gamma}_+$, $\partial\boldsymbol{\Gamma}_+$	right half-plane, imaginary axis
$\boldsymbol{\Gamma}_D$, $\partial\boldsymbol{\Gamma}_D$	region for minimum damping, corresponding boundary
$\tilde{\boldsymbol{Q}}$	extended parameter set $\boldsymbol{Q} \times \boldsymbol{T}_d \times \boldsymbol{\Gamma}$ or $\tilde{\boldsymbol{Q}} = \boldsymbol{Q} \times \boldsymbol{T}_d \times \partial\boldsymbol{\Gamma}$
$f(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)$	characteristic function
$f(\boldsymbol{\Gamma}, \boldsymbol{Q}, \boldsymbol{T}_d) = f(\tilde{\boldsymbol{Q}})$	value set of the characteristic function
$\bar{f}(\boldsymbol{s}, \boldsymbol{Q}, \boldsymbol{T}_d)$	transformed characteristic function
$\mathcal{Q}$	set of unchecked regions in branch and bound algorithm
$\tilde{\boldsymbol{q}}_{\min}$	desired resolution of the boundary mapping
$\tilde{\boldsymbol{Q}}_0$	initial extended parameter set
$T(\tilde{\boldsymbol{Q}})$	Taylor Model consisting of normalized polynomial $\tilde{p}(\boldsymbol{U}, \tilde{\boldsymbol{Q}})$ and interval remainder $R(\tilde{\boldsymbol{Q}})$

Stability and Transparency Analysis of Teleoperation Systems

$f_h/F_h(\boldsymbol{s})$	interaction force between master and human
$\tilde{f}_h/\tilde{F}_h(\boldsymbol{s})$	measured force between master and human
$f_h^\star/\ F_h^\star(\boldsymbol{s})$	exogenous force of human
x_m, $\dot{x}_m/V_h(\boldsymbol{s})$, $\ddot{x}_m$	position, verlocity, acceleration of master
$Z_h(\boldsymbol{s})$	human arm impedance
m_h, b_h, c_h	inertia, damping, stiffness of human
m_{em}	inertia of the master end-effector
$f_m/F_m(\boldsymbol{s})$	desired master actuator force
$\tilde{f}_m/\tilde{F}_m(\boldsymbol{s})$	actual master actuator force
$Z_m(\boldsymbol{s})$	master impedance
m_m, b_m	inertia, damping of master
$f_e/F_e(\boldsymbol{s})$	interaction force between slave and remote environment
$\tilde{f}_h/\tilde{F}_h(\boldsymbol{s})$	measured force between slave and remote environment
x_s, $\dot{x}_s/V_e(\boldsymbol{s})$, $\ddot{x}_s$	position, verlocity, acceleration of slave
$Z_e(\boldsymbol{s})$	remote environment impedance
m_h, b_h, c_h	inertia, damping, stiffness of remote environment
m_{em}	inertia of the slave end-effector
$f_s/F_s(\boldsymbol{s})$	desired slave actuator force
$\tilde{f}_s/\tilde{F}_s(\boldsymbol{s})$	actual slave actuator force
m_s, b_s	inertia, damping of slave
$Z_s(\boldsymbol{s})$	slave impedance
τ_m, τ_s	actuator time constants for master and slave
τ_h, τ_e	sensor time constants for master and slave
$\mathcal{H}$	two-port of general four-channel architecture
$C_{1..6}(\boldsymbol{s})$, $C_{m,s}(\boldsymbol{s})$	controllers in the four-channel architecture
T_d	time delay in the communication channel

$\boldsymbol{q}_h$	parameter vector of human impedance
$\boldsymbol{q}_e$	parameter vector of environment impedance
$\boldsymbol{q}_m$	parameter vector of master
$\boldsymbol{q}_s$	parameter vector of slave
$Z_{to}(\boldsymbol{s})$	impedance transmitted to the operator
$\hat{Z}_{to}(\boldsymbol{s})$	approximated impedance transmitted to the operator
Z_{error}	error of impedance approaximation
K_f	force controller gain
K_p, K_d	proportional and derivative gain of position controller
b	wave damping
m_d, b_d	virtual inertia and damping of virtual admittance $Y_d(\boldsymbol{s}$
x_{dm}, $\dot{x}_{dm}$, $\ddot{x}_{dm}$	position, velocity, acceleration of virtual admittance (master side)
x_{ds}, $\dot{x}_{ds}$, $\ddot{x}_{ds}$	position, velocity, acceleration of virtual admittance (slave side)

Towards Optimal Parametrization of LTI Systems

g	cost function
f	characteristic function
$\boldsymbol{c}$	performance constraints
$\boldsymbol{p}$	tunable parameters
$\bar{\boldsymbol{q}}$	nominal parameters
$\bar{\boldsymbol{t}}_d$	nominal time delays
$\omega_{\max}$	upper frequency bound for performance optimisation and constraints
$G(\boldsymbol{s})$	plant model
$K(\boldsymbol{s})$	controller model
$F(\boldsymbol{s})$	prefilter
$G_d(\boldsymbol{s})$	desired closed-loop transfer function
r	reference input
w	filtered reference input
e	tracking error
e_m	meansured tracking error
e_M	model matching error
u'	controller output
u	actuator signal
y'	plant output
y	controlled signal
d_1	plant input disturbance
d_2	plant output disturbance
n	sensor noise
$S(\boldsymbol{s})$	sensitivity function
$T(\boldsymbol{s})$	complementary sensitivity function
$S_K(\boldsymbol{s})$	input sensitivity function
$S_G(\boldsymbol{s})$	output sensitivity function
$E(\boldsymbol{s})$	model matching error function
$\boldsymbol{P}_a$	set of admissible parametters

$\boldsymbol{P}_i$	set of parameters satisfying performance constraint i
$\boldsymbol{P}_c$	set of parameters satisfying all performance constraints
$\boldsymbol{P}_s$	set of parameters satisfying stability constraints

List of Figures

1.1 Principle of a multi-modal teleoperation system 1

2.1 Inner approximation of the value set of a complex polynomial 16
2.2 Outer approximation of the value set of a complex Taylor Model 22
2.3 Inner approximation of the value set of a complex Taylor Model 23

3.1 Flow-chart of overall algorithm used for robust stability analysis 34
3.2 $\mathbf{\Gamma}$-regions used for system analysis 38
3.3 Unbounded regions in the complex plane 41
3.4 Flow-chart of branch & bound algorithm 45
3.5 Crane 53
3.6 Hurwitz stable regions for crane example (2d) 54
3.7 Hurwitz stable regions for crane example (4d) 55
3.8 Regions with minimum damping for crane example (2d) 56
3.9 Regions with minimum damping for crane example (4d) 57
3.10 Hurwitz stable regions for crane example with time delay (4d) 58
3.11 Hurwitz stable regions for crane example with time delay (2d) 59
3.12 Simulation results for crane example with time delay 60
3.13 Stable regions for examples with one time delay 63
3.14 Stable regions for examples with two time delays 64

4.1 General Four-Channel Architecture of a Teleoperation System. 80
4.2 One-DoF linear axis teleoperation system. 88
4.3 FP architecture 90
4.4 Stability analysis of simplified FP architecture (initial parametrization) . . 92
4.5 Controller design for simplified FP architecture 93
4.6 Stability analysis of simplified FP architecture (final parametrization) . . . 94
4.7 Stability analysis of FP architecture (initial parametrization) 95
4.8 Controller design for FP architecture 96
4.9 Stability analysis of FP architecture (final parametrization) 97
4.10 Transparency analysis of FP architectures for $K_f = 1$ 98
4.11 Transparency analysis of FP architecture for $K_f = 0.1$ 99
4.12 FP-W architecture 101
4.13 Stability analysis of FP-W architecture (initial parametrization) 104
4.14 Controller design for FP-W architecture 105
4.15 Stability analysis of FP-W architecture (final parametrization) 106
4.16 Transparency analysis of FP-W architecture 107

4.17 Stability analysis of 4C architecture (initial parametrization) 110
4.18 Controller design for 4C architecture 111
4.19 Stability analysis of 4C architecture (final parametrization) 112
4.20 Transparency analysis of 4C architecture 113
4.21 Transparency analysis of 4C architecture (effect of position controller) . . . 114
4.22 Transparency analysis of 4C architecture (effect of force controller) 115
4.23 FaFa architecture 118
4.24 Stability analysis of FaFa architecture 119
4.25 Controller design for FaFa architecture 120
4.26 Transparency analysis of FaFa architecture 121
4.27 Transparency analysis of FaFa architecture (large virtual damping) 122

5.1 Flow-chart of overall algorithm used for performance optimization 130
5.2 Single-loop feedback system (Fig. 5.32 from [1]) 131

List of Tables

3.1 System matrices for different time-delay examples 62

4.1 Parameters of haptic interface and teleoperator. 89
4.2 Parameters of human and environment. 89

1 Introduction

Autonomous robotics have made great strides in the past years and many challenging problems have been tackled successfully. Nevertheless, the current state of robotic cognition is still very far from the cognitive skills of humans, especially when considering adaptation to unknown problems in unstructured environments. In contrast to autonomous robots, teleoperation systems keep the human in the loop. Using a teleoperation system, an operator can control the actions of a robotic system and thereby perform tasks in a remote environment which can be separated from the operator by different barriers. The remote environment can be far away from the operator (e.g., outer space), can be very small (e.g., microsurgery) or very large (e.g., construction work), can be dangerous (e.g., nuclear power plants), etc. For a thorough introduction to teleoperation and all aspects involved, see the book by *Sheridan* [2].

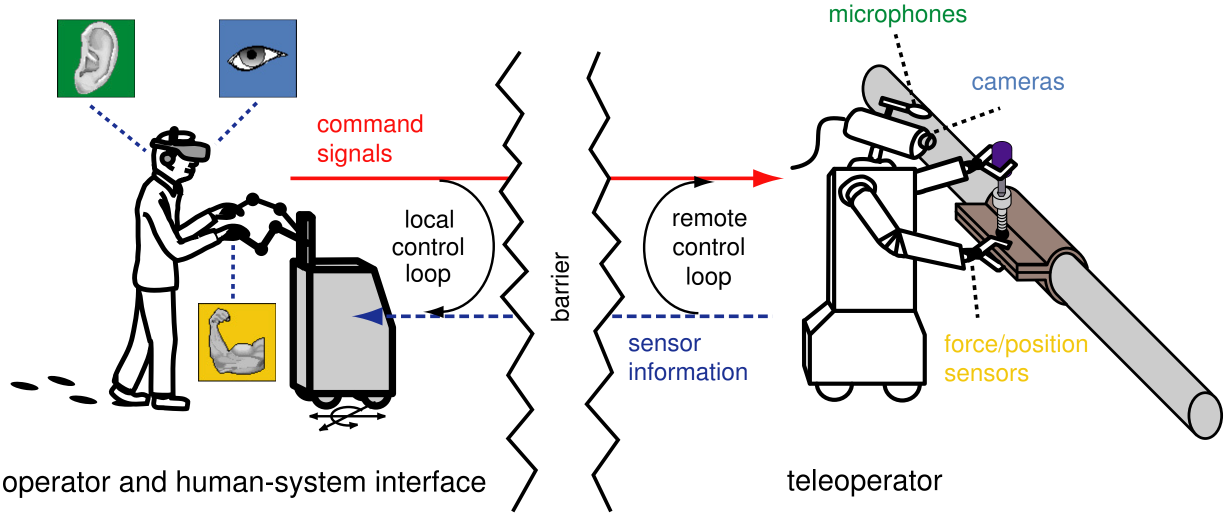


Figure 1.1: Principle of a multi-modal teleoperation system taken from [3].

In general, teleoperation systems provide feedback to the operator in different modalities, e.g., visual feedback, auditory feedback, and haptic feedback. The principle of a multi-modal teleoperation system is illustrated in Fig. 1.1. An example of a highly integrated multi-modal teleoperation system can be found in [3]. That system incorporates complex solutions for haptic, visual, and auditory feedback. A less complex realization of visual and auditory feedback is however very straightforward, e.g., by using a webcam with microphone at the remote site and displaying the video and audio using a computer monitor and loudspeakers. In contrast, even *simple* realizations of haptic feedback, are challenging due to the bilateral energy exchange involved, which can cause instabilities and potentially harm the operator or cause damage in the remote environment.

In this thesis, we focus on haptic teleoperation which enables the *manipulation* of objects in the remote environment and *reflects* interaction forces back to the operator, and is therefore often referred to as telemanipulation or bilateral force-reflecting teleoperation. There are two main goals when developing a telemanipulation system: a *robustly stable* interaction with a range of *unknown* remote environments is necessary while a *transparent* interaction with the remote environments is desirable. A perfectly *transparent* teleoperation system would not distort the impedance displayed to the operator, i.e., when interacting with a remote environment using this teleoperation system the operator would feel the same impedance as when directly interacting with the remote environment. In practice, a perfectly transparent teleoperation system can not be realized [4]. Instead, a compromise between stability and transparency must be accepted, i.e., a system must be developed which is robustly stable for different remote objects and at the same time sufficiently transparent. Haptic teleoperation has been a field of active research for several decades. An overview of the developments in this field is, e.g., given in [5]. This thesis introduces new methods for stability and transparency analysis of teleoperation systems. The general idea is to derive methods which are less conservative than existing methods and easily interpretable.

1.1 Problem Statement

Stability and transparency analysis of a complex teleoperation system like the one developed in [3] is extremely challenging. This is due to the fact that the overall system is very complex, e.g., due to unknown dynamic properties of the remote environment, unknown dynamic properties of the human operator, time delay between the operator site and remote site, non-linear dynamics of all subsystems involved (operator, haptic interface, communication channel, teleoperator, environment), as well as time-varying operator dynamics, environment dynamics, and time delay. As addressing all of these issues is not possible in practice, some simplifying assumptions are generally made which make a solution to the stability and transparency analysis feasible[1]: the teleoperation system is modeled as *linear* and *time-invariant*, and the remote environment and human operator are modeled as linear mass-spring-damper systems with unknown parameters constrained to intervals. Using these assumptions we can represent the closed-loop system including operator and environment by a Linear Time-Invariant (LTI) time-delay system with parametric uncertainties.

This represents a rather common system class to which a number of existing stability analysis methods are applicable. These methods are, however, generally very conservative. Therefore, a new method for the stability analysis of LTI time-delay systems with parametric uncertainties is developed in this thesis. This method is not only suitable for the stability analysis of teleoperation systems as is illustrated by several examples. When considering the transparency analysis of teleoperation systems, it becomes clear that existing methods require an in-depth understanding of teleoperation systems by the system designer, and the results are limited to special cases and not easily interpretable. In contrast, our new approach to transparency analysis focuses on offering an intuitive understanding of the effect of a teleoperation system and controller on transparency. The

[1] The assumptions are formulated and discussed in greater detail in Section 4.2.1.

results are easily interpretable by anyone with a basic understanding of simple impedance models.

1.2 Contributions and Outline of the Thesis

The first main result of this thesis is a stability analysis method for LTI systems which can either determine stable and unstable regions in the parameter space with minimal conservatism or non-conservatively check stability of a system with interval parameters. This method is not restricted to teleoperation systems but can be applied to arbitrary LTI systems of retarded type with parametric uncertainties and concentrated (possibly incommensurate) time delay. The method can handle non-affine dependencies of the characteristic equation on the uncertain parameters, i.e., the coefficients of the characteristic equation may depend polynomially on the uncertain parameters. Moreover, it is not limited in the number of uncertain parameters, scales to systems of high order, and can be used to determine regions in the parameter space which constrain eigenvalues of the system to a given region in the complex plane, thereby imposing further constraints on system dynamics, e.g., a desired minimum damping. The resulting stability regions are rigorously determined with a specifiable amount of conservatism. The method is based on interval-like computations using Taylor Models which result in polynomials and interval remainders. A transformation of the polynomial parts into Bernstein form allows for efficient branch and bound algorithms which allow to rigorously determine bounds on the polynomial parts. This, in essence, transforms sufficient, conservative conditions into necessary conditions (with a specifiable amount of remaining conservatism). More concretely, this approach is used to map regions from the complex plane to the parameter space by computing the so-called value set and making use of the zero exclusion principle. This results in an easily interpretable graphical stability check. The stability analysis method is presented in Chapter 3. It makes use of mathematical foundations introduced in Chapter 2. In these two chapters a general formulation is chosen which is not restricted to teleoperation systems. The generality of the developed method is further emphasized in Chapter 3 by some examples taken from literature on robust stability analysis which are not related to teleoperation.

The second main contribution is a novel transparency analysis method which is introduced in Section 4.4. Common transparency measures for haptic teleoperation consider one or two fixed environments only, usually the extreme cases of free space and stiff contact. In contrast, our approach offers insights into the range of environments for which a system achieves a desired transparency by approximating the impedance displayed to the operator by means of a simple, physically interpretable mechanical impedance. A numerical optimization is used to determine the parameters characterizing this impedance model which results in an intuitive graphical representation of the transparency of a teleoperation system. As the method is based on a general four-channel architecture it can be used to analyze a large number of different teleoperation setups.

The third main result is a detailed stability and transparency analysis for a number of different teleoperation architectures in Chapter 4 which mainly focuses on evaluating the effect of time delay on stability and transparency. Therefore, we combine our newly developed stability and transparency analysis methods. For each architecture we first design a controller which stabilizes the teleoperation system for a range of uncertain parameters (due to unknown environment and operator dynamics) and time delays between the operator side and teleoperator side. Then, our transparency analysis method is used to evaluate the transparency for different time delays.

Finally, a possible future extension of the stability analysis algorithm towards an automatic optimal parametrization of LTI systems with parametric uncertainties and (possibly incommensurate) time delays is outlined in Chapter 5. The proposed method consists of combining the stability analysis method with performance constraints and a cost function defined as weighted sum of different frequency-dependent performance metrics and then solving a constrained global optimization problem. A general formulation is again chosen which is not restricted to teleoperation. However, one possible application of this method would be an automatic parametrization of a teleoperation system which optimizes transparency while assuring robust stability.

2 Mathematical Foundations

Summary. *In this chapter the mathematical foundations used in this thesis are presented. This includes a short summary of interval arithmetic, Bernstein polynomials, and Taylor Models. Using these mathematical tools a method to evaluate the value set of a Taylor Model is developed, which results in an inner and outer approximation of the exact value set. This value set evaluation is the main result of this chapter, and forms the basis of the robust stability analysis method developed in this thesis.*

The stability analysis method developed in this thesis is based on evaluating the value set of the characteristic function of a dynamic system with uncertain system parameters. A value set represents the set of values a function takes for given parameter ranges. As a calculation of the exact value set of a characteristic function is not possible in general [6, Section 4.4.2], we develop a method to determine an inner and outer approximation of the value set.

In the case considered, a linear time-invariant system with parametric uncertainties and time delay, the characteristic function takes the form of an uncertain quasi-polynomial. More concretely, the characteristic function consists of a complex-valued polynomial depending on parameters constrained to given intervals and exponential terms with complex-valued exponent, again depending on interval parameters. Therefore, we must determine the value set of a complex-valued function with interval parameters (to deal with the parametric uncertainty) and exponential terms (to deal with time delay).

In this chapter, a method which is based on [7] is developed which evaluates the value set of such a function using Taylor Models and Bernstein polynomials. While the method in [7] is limited to value sets of polynomials, our method can determine the value set of arbitrary holomorphic functions. A function is holomorphic on the domain $\boldsymbol{U}$ if it is complex differentiable at every point $\boldsymbol{u} \in \boldsymbol{U}$ [8]. A holomorphic function maps domains, i.e. open connected sets, to domains [9]. Polynomials, trigonometric functions, exponential functions, and quasi-polynomials are holomorphic functions [8]. In contrast to the method presented in [7] our method is therefore applicable to linear time-invariant systems with time delay as these can be described by a quasi-polynomial.

This chapter is structured as follows. In Section 2.1, the definition of a value set is given and the general approach used to determine the value set is described. Interval arithmetic, which offers the most straightforward approach to computing bounds of arbitrary functions, is introduced in Section 2.2. Then, multivariate polynomials are defined, the Bernstein transformation is introduced, and it is shown how the value set of a polynomial can efficiently be determined using algorithms for subdivision and derivative estimation that operate directly on Bernstein coefficients in Section 2.3. Finally, Taylor Models are introduced in Section 2.4. Taylor Models represent an arbitrary function as a polynomial and additional interval remainder, and allow an efficient evaluation of the value set of a holomorphic function using the novel method presented in Section 2.4.5.

2.1 Definition of Value Set

In this thesis we consider a complex-valued function $f(x): \mathbb{R}^l \to \mathbb{C}$. We would like to determine the set of values this function takes for a given set of l-dimensional parameter values $x \in \boldsymbol{X}$. In the following we call this the value set $f(\boldsymbol{X})$ which is defined as

$$f(\boldsymbol{X}) := \{f(\boldsymbol{x}) \mid \boldsymbol{x} \in \boldsymbol{X}\}. \tag{2.1}$$

Throughout the rest of the thesis we consider the set $\boldsymbol{X}$ to be a closed l-dimensional interval box.

An exact calculation of the value set $f(\boldsymbol{X})$ is not possible in general. When considering the characteristic function of a time delay system, e.g., an analytic solution can only be found for a few special cases [6, Section 4.4.2]. Instead, we calculate an outer approximation and an inner approximation of the value set. The complex function, evaluated for any parameter within the given parameter-set is always contained within the outer approximation, i.e., the exact value set is a subset of the outer approximation of the value set. Likewise, the exact value set contains the inner approximation of the value set, i.e., the inner approximation is a subset of the exact value set.

In this thesis, the following novel method is used to calculate an inner and outer approximation of the value set of a complex-valued function. First, the function is evaluated using Taylor Models. This results in a complex Taylor Model, i.e., a complex normalized multivariate polynomial and complex interval remainder. The polynomial part is then transformed into Bernstein form resulting in complex Bernstein coefficients. Inner and outer approximations of the polynomial part can efficiently be determined directly from the complex Bernstein coefficients [7], see Section 2.3.7. Finally, inner and outer approximations of the complete Taylor Model are determined by taking the interval remainder into account as presented in Section 2.4.5. As a prerequisite, a definition of intervals as well as interval arithmetic is required which is introduced in the following section.

2.2 Intervals

Interval arithmetic in its current form was introduced by *R. E. Moore* around 1960. Notable publications include his dissertation [10] and shortly afterwards the well known book *Interval Analysis* [11]. Interval arithmetic offers the most straightforward approach to rigorously determine the value set of arbitrary functions.

A closed interval $X \in [\mathbb{R}]$ with lower bound $\underline{X}$ and upper bound $\overline{X}$ is defined as

$$X := [\underline{X}, \overline{X}] = \{x \in \mathbb{R} \mid \underline{X} \leq x \leq \overline{X}\}.$$

An l-dimensional interval box $\boldsymbol{X} \in [\mathbb{R}]^l$ is defined as

$$\boldsymbol{X} := [X_1 \times X_2 \times \ldots \times X_l] = \{\boldsymbol{x} \in \mathbb{R}^l \mid \underline{X}_k \leq x_k \leq \overline{X}_k \;\forall\; 1 \leq k \leq l\}.$$

For future use, we also define the *unit interval* U and *unit interval box* $\boldsymbol{U}$ of dimension l as

$$U := \{u \in \mathbb{R} \mid 0 \leq u \leq 1\} \tag{2.2}$$

$$\boldsymbol{U} := [U_1 \times U_2 \times \ldots \times U_l] = \{\boldsymbol{u} \in \mathbb{R}^l \mid 0 \leq u_k \leq 1 \;\forall\; 1 \leq k \leq l\}. \tag{2.3}$$

2.2.1 Arithmetic

Arithmetic operations on intervals are defined in such a way that the inclusion property

$$[\min_{a\in A,\, b\in B}(a \circ b),\ \max_{a\in A,\, b\in B}(a \circ b)] \subseteq A \circ B$$

holds for all operations $\circ$ on the two intervals A and B where the result is again an interval. Exact definitions may be given for all arithmetic operations.

Addition and multiplication are defined as

$$\begin{aligned} A + B :=& [\underline{A} + \underline{B},\ \overline{A} + \overline{B}] \\ A\,B :=& [\min(\underline{A}\,\underline{B},\ \underline{A}\,\overline{B},\ \overline{A}\,\underline{B},\ \overline{A}\,\overline{B}),\ \max(\underline{A}\,\underline{B},\ \underline{A}\,\overline{B},\ \overline{A}\,\underline{B},\ \overline{A}\,\overline{B})] \end{aligned}$$

where min() and max() return the minimum and maximum of the elements respectively. Subtraction and division are achieved with the help of unary operations for negation and multiplicative inverse, i.e.

$$\begin{aligned} A - B :=& A + (-B) \\ -B :=& [-\overline{B},\ -\underline{B}] \\ A/B :=& A(1/B) \\ 1/B :=& \begin{cases} [1/\overline{B},\ 1/\underline{B}] & \text{for } 0 \notin B \\ \mathbf{NaI} & \text{for } 0 \in B \end{cases} \end{aligned}$$

where **NaI** stands for *Not an Interval*. This is the interval-equivalent to *Not a Number* and implies that any operation involving **NaI** again returns **NaI**.

2.2.2 Elementary Functions

In addition to the arithmetic operations, elementary functions for the exponential e^x or alternatively the trigonometric functions $\sin(x)$ and $\cos(x)$ are required to represent time delays. Such elementary functions $f(A)$ may be determined using the inclusion property as

$$[\min_{a\in A} f(a),\ \max_{a\in A} f(a)] \subseteq f(A).$$

For monotonic functions the interval is given by the value at the end-points. For non-monotonic functions (e.g. the trigonometric functions) minima and maxima between the two end-points must be taken into account!

2.2.3 Complex Intervals

In the following complex functions are evaluated, i.e., complex intervals are required in addition to real intervals. A closed complex interval $X \in [\mathbb{C}]$ with lower bound $\underline{X}$ and upper bound $\overline{X}$ is defined as

$$X := [\underline{X}, \overline{X}] = \{x \in \mathbb{C} \mid \mathrm{Re}(\underline{X}) \leq \mathrm{Re}(x) \leq \mathrm{Re}(\overline{X}), \mathrm{Im}(\underline{X}) \leq \mathrm{Im}(x) \leq \mathrm{Im}(\overline{X})\}.$$

Unit interval U and *unit interval box* $\boldsymbol{U}$ of dimension l may be defined for the complex case as

$$U := \{u \in \mathbb{C} \mid 0 \leq \mathrm{Re}(u) \leq 1, 0 \leq \mathrm{Im}(u) \leq 1, \} \tag{2.4}$$

$$\boldsymbol{U} := \{\boldsymbol{u} \in \mathbb{C}^l \mid 0 \leq \mathrm{Re}(u_k) \leq 1, 0 \leq \mathrm{Im}(u_k) \leq 1 \; \forall \, 1 \leq k \leq l\}. \tag{2.5}$$

Arithmetic operations and elementary functions on complex intervals may be determined directly from the definition of arithmetic operations and elementary functions on complex numbers. E.g., addition and multiplication of the complex intervals $A, B \in [\mathbb{C}]$ are defined as

$$A + B := \mathrm{Re}(A) + \mathrm{Re}(B) + j(\mathrm{Im}(A) + \mathrm{Im}(B))$$

and

$$A\,B := \mathrm{Re}(A)\,\mathrm{Re}(B) - \mathrm{Im}(A)\,\mathrm{Im}(B) + j\,(\mathrm{Re}(A)\,\mathrm{Im}(B) + \mathrm{Im}(A)\,\mathrm{Re}(B))$$

where the operations on real intervals $(\mathrm{Re}(A), \mathrm{Re}(B), \mathrm{Im}(A), \mathrm{Im}(B))$ are given in Section 2.2.1.

2.2.4 Approximation of Value Set

Interval arithmetic can be used to directly calculate outer bounds of a function with interval parameters and thereby approximate the value set. However, due to the dependency problem interval arithmetic generally suffers from a large overapproximation when determining outer bounds of functions with dependent terms, see, e.g., the recent *Introduction to Interval Analysis* by *Moore et al.* [12]. Thus, pure interval arithmetic is not employed in this thesis to calculate bounds of functions.

2.3 Polynomials

In this section a definition of multivariate polynomials is given. Then, the Bernstein Transformation is introduced which results in Bernstein polynomials and Bernstein coefficients. The Bernstein coefficients can be used to efficiently determine bounds of a polynomial and thereby approximate the value set of a polynomial. In the following, first a *multi-index* is introduced which is then used for a compact formulation of multivariate polynomials.

2.3.1 Notation

A multi-index I of length l is an l-tuple defined as

$$I := (i_1, i_2, \ldots, i_l) \quad \text{with } i_k \in \mathbb{N}_0, 1 \leq k \leq l. \tag{2.6}$$

We define several operations on multi-indices to allow for a compact formulation in the following sections. Let I and J be multi-indices of length l as defined in (2.6) and $\boldsymbol{x} \in \mathbb{R}^l$, then

$$\boldsymbol{x}^I := \prod_{k=1}^{l} x_k^{i_k} \tag{2.7}$$

which is used to compactly formulate polynomials in Section 2.3.2. A compact formulation of binomial coefficients for multi-indices is defined as

$$\binom{I}{J} := \prod_{k=1}^{l} \binom{i_k}{j_k} \tag{2.8}$$

where

$$\binom{i_k}{j_k} = \frac{i_k!}{(i_k - j_k)!\, j_k!}. \tag{2.9}$$

The operator $\leq$ is used to define the set of all multi-indices for which all elements of each multi-index are smaller than or equal to the corresponding index of a given multi-index N, i.e.

$$I \leq N \Leftrightarrow I \in \{(i_1, i_2, \ldots, i_l) \mid i_k \leq n_k \; \forall\, 1 \leq k \leq l\}\,. \tag{2.10}$$

Subtraction of two multi-indices is defined as

$$J - I := (j_1 - i_1, j_2 - i_2, \ldots, j_l - i_l) \tag{2.11}$$

and we define an operation to modify one element of a multi-index of length l as

$$I_{r,\mu} := (i_1, \ldots, i_r + \mu, \ldots, i_l). \tag{2.12}$$

This notation is mostly adopted from [7] which is the publication this section is largely based on.

2.3.2 Definition

A multivariate polynomial $p(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathbb{R}^l$ may be written as

$$p(\boldsymbol{x}) = \sum_{I \leq N} a_I \boldsymbol{x}^I$$

where all definitions are given in Section 2.3.1, the multi-index N is referred to as the degree of the polynomial, and $a_I \in \mathbb{R}$ is a scalar coefficient of the multivariate polynomial. Any multivariate polynomial in l variables of order N may be described by a set of multi-indices $I \leq N$ and corresponding coefficients a_I. The number of multi-indices is uniquely defined by the order of the multivariate polynomial N. An arbitrary number of coefficients a_I with $I \leq N$ may however be zero.

The stability analysis in the following chapters involves multivariate polynomials with complex coefficients and complex variables. These may be defined analogously, with variables $\boldsymbol{x} \in \mathbb{C}^l$ and scalar coefficients $a_I \in \mathbb{C}$.

2.3.3 Normalization

The methods introduced in the following sections require a normalized multivariate polynomial where each variable is in the range $U = [0,1]$ (real case) or $U = [0,1] + j[0,1]$ (complex case). Any multivariate polynomial $p(\boldsymbol{x})$ on the l-dimensional interval box $\boldsymbol{X}$ can be transformed to the unit interval box $\boldsymbol{U}$ using a coordinate transformation, i.e.

$$p(\boldsymbol{x}) = \sum_{I \leq N} a_I \boldsymbol{x}^I = \sum_{I \leq N} \tilde{a}_I(\boldsymbol{X}) \boldsymbol{u}^I = \tilde{p}(\boldsymbol{u}, \boldsymbol{X}), \tag{2.13}$$

where the multi-index N is referred to as the degree of the polynomial, operations on multi-indices are defined in Section 2.3.1, and $\boldsymbol{u} \in \boldsymbol{U}$ with $\boldsymbol{U}$ the l-dimensional unit interval box (real or complex).

2.3.4 Bernstein Transformation

In this section the multivariate polynomial is transformed into Bernstein form. As will become evident in the following, this allows for an efficient evaluation of the value set of a multivariate polynomial. See Section 2.3.1 for the definitions of the operations on multi-indices which are used here for a compact formulation.

A normalized multivariate polynomial $\tilde{p}(\boldsymbol{u}, \boldsymbol{X})$ (see Section 2.3.3) may be transformed into Bernstein form by

$$\tilde{p}(\boldsymbol{u}, \boldsymbol{X}) = \sum_{I \leq N} \tilde{a}_I(\boldsymbol{X}) \boldsymbol{u}^I = \sum_{I \leq N} b_I(\boldsymbol{X}) B_{N,I}(\boldsymbol{u}), \tag{2.14}$$

where I is a multi-index which is element-wise smaller or equal to the multi-index N which is the degree of the polynomial (see Section 2.3.1), and $b_I(\boldsymbol{X})$ are the Bernstein coefficients while $B_{N,I}(\boldsymbol{u})$ are Bernstein polynomials.

The Bernstein coefficients $b_I(\boldsymbol{X})$ are computed from the coefficients $\tilde{a}_J(\boldsymbol{X})$ (J is another multi-index) by

$$b_I(\boldsymbol{X}) = \sum_{J \leq I} \frac{\binom{I}{J}}{\binom{N}{J}} \tilde{a}_J(\boldsymbol{X}) \tag{2.15}$$

and $B_{N,I}(\boldsymbol{u})$, the Ith Bernstein polynomial of degree N is[1]

$$B_{N,I}(\boldsymbol{u}) = \binom{N}{I} \boldsymbol{u}^I (\boldsymbol{1} - \boldsymbol{u})^{(N-I)}.$$

The Bernstein transformation and calculation of Bernstein coefficients is valid for real as well as complex polynomials.

[1] An approximation of the value set as well as the other necessary algorithms (subdivision and derivative estimation) only require the Bernstein coefficients $b_I(\boldsymbol{X})$. Therefore, it is not actually necessary to calculate the Bernstein polynomials $B_{N,I}(\boldsymbol{u})$.

2.3.5 Approximation of Value Set

For real polynomials the Bernstein transformation has two interesting properties with respect to evaluating the value set[2]. On the one hand, an outer bound for the range of a real polynomial is given by the smallest and largest coefficient of the corresponding Bernstein polynomial, i.e.

$$\min_{I \leq N}(b_I(\boldsymbol{X})) \leq \min_{\boldsymbol{x} \in \boldsymbol{X}}(p(\boldsymbol{x})) \leq p(\boldsymbol{x})|_{\boldsymbol{x} \in \boldsymbol{X}} \leq \max_{\boldsymbol{x} \in \boldsymbol{X}}(p(\boldsymbol{x})) \leq \max_{I \leq N}(b_I(\boldsymbol{X})). \tag{2.16}$$

On the other hand, an inner bound for the range of the polynomial can easily be determined from the multi-indices S_0 of the Bernstein polynomial corresponding to the vertices of the interval box $\boldsymbol{X}$, which are given by

$$S_0 := \{(i_1, i_2, \ldots, i_l) \mid i_k = 0 \vee i_k = n_k \; \forall \; 1 \leq k \leq l\}, \tag{2.17}$$

where n_k is element k of multi-index N. The Bernstein coefficients of the multi-indices S_0 on the vertices of the interval box are exact, i.e., they have exactly the same value as the polynomial when evaluated for the values corresponding to these vertices. Therefore, these Bernstein coefficients represent inner bounds of the polynomial, i.e.

$$\min_{I \in S_0}(b_I(\boldsymbol{X})) \geq \min_{\boldsymbol{x} \in \boldsymbol{X}}(p(\boldsymbol{x})) \wedge \max_{I \in S_0}(b_I(\boldsymbol{X})) \leq \max_{\boldsymbol{x} \in \boldsymbol{X}}(p(\boldsymbol{x})). \tag{2.18}$$

Combining the properties for inner and outer bounds we can derive an approximation of the minimum value $\min_{\boldsymbol{x} \in \boldsymbol{X}}(p(\boldsymbol{x}))$ and maximum value $\max_{\boldsymbol{x} \in \boldsymbol{X}}(p(\boldsymbol{x}))$ of the polynomial for all values $\boldsymbol{x} \in \boldsymbol{X}$ as

$$\min_{I \leq N}(b_I(\boldsymbol{X})) \leq \min_{\boldsymbol{x} \in \boldsymbol{X}}(p(\boldsymbol{x})) \leq \min_{I \in S_0}(b_I(\boldsymbol{X})) \tag{2.19}$$

$$\max_{I \leq N}(b_I(\boldsymbol{X})) \geq \max_{\boldsymbol{x} \in \boldsymbol{X}}(p(\boldsymbol{x})) \geq \max_{I \in S_0}(b_I(\boldsymbol{X})). \tag{2.20}$$

This represents an approximation of the value set of the polynomial.

If the upper and lower bound for the minimum or maximum value are identical we know the minimum or maximum value exactly, i.e., the bound of the value set is *sharp*. This can be checked using the following conditions:

$$\min_{I \in S_0}(b_I(\boldsymbol{X})) = \min_{I \leq N}(b_I(\boldsymbol{X})) \rightarrow \min_{\boldsymbol{x} \in \boldsymbol{X}}(p(\boldsymbol{x})) = \min_{I \leq N}(b_I(\boldsymbol{X})) \tag{2.21}$$

$$\max_{I \in S_0}(b_I(\boldsymbol{X})) = \max_{I \leq N}(b_I(\boldsymbol{X})) \rightarrow \max_{\boldsymbol{x} \in \boldsymbol{X}}(p(\boldsymbol{x})) = \max_{I \leq N}(b_I(\boldsymbol{X})) \tag{2.22}$$

2.3.6 Bernstein Algorithms

As described in the previous section, Bernstein polynomials offer a fast approximation of the value set of a polynomial. The resulting value set suffers from two difficulties.

[2] These properties also hold for the real and imaginary part of complex polynomials. More elaborate results for the value set of complex polynomials may however be determined by the method described in Section 2.3.7.

On the one hand, the value set is not *sharp* in general, i.e., we can only determine that the minimum value and maximum value of the polynomial lie within certain ranges which may be very large in practice. Therefore, two Bernstein algorithms are presented in this section which allow to determine the value set of the polynomial up to a desired accuracy[3]

On the other hand, the stability analysis algorithm in Chapter 3 requires a subdivision of the original interval box $\boldsymbol{X}$ so as to find regions in $\boldsymbol{X}$ for which the value set of the polynomial is within a given range. The same two Bernstein algorithms can be used to efficiently solve this problem within a branch and bound algorithm which is presented in Chapter 3.

The algorithms presented in the following are a subdivision algorithm and a derivative estimation algorithm. The subdivision algorithm efficiently determines Bernstein coefficients of sub-boxes from the Bernstein coefficients of the original interval-box, i.e., the original interval box is split into two new interval-boxes and the Bernstein coefficients of these interval-boxes are determined from the Bernstein coefficients of the original interval box. This subdivision algorithm was first introduced in [13]. We could simply split the original box along each dimension in turn, until the desired accuracy is reached for the value set, or an interval-box is found, for which the value set is within a given range. This is however very inefficient. Instead, the subdivision direction is selected using a derivative estimation algorithm which efficiently estimates partial derivatives of the polynomial. This algorithm was first introduced in [7].

Subdivision Algorithm

Consider a Bernstein polynomial with Bernstein coefficients $b_I(\boldsymbol{X})$, $I \leq N$ as defined in (2.14) and (2.15) on the l-dimensional interval box $\boldsymbol{X}$. The algorithm presented in [13], reformulated in [7], and summarized in this section allows to split the interval box $\boldsymbol{X}$ along dimension r at point λ into two interval boxes $\boldsymbol{Y}$, $\boldsymbol{Z}$ so that

$$Y_i = Z_i = X_i \quad \forall\, 1 \leq i \leq l, i \neq r \tag{2.23}$$

$$Y_r = [\underline{X}_r, \underline{X}_r + \lambda(\overline{X}_r - \underline{X}_r)] \tag{2.24}$$

$$Z_r = [\underline{X}_r + \lambda(\overline{X}_r - \underline{X}_r), \overline{X}_r] \tag{2.25}$$

and then determine the Bernstein coefficients $b_I(\boldsymbol{Y})$ and $b_I(\boldsymbol{Z})$ directly from the Bernstein coefficients $b_I(\boldsymbol{X})$, i.e., without performing a Bernstein transformation. The value of λ is restricted to $0 < \lambda < 1$, for subdivision into two equally sized parts $\lambda = 0.5$ is used.

First, set

$$b_I^{(0)}(\boldsymbol{Y}) = b_I(\boldsymbol{X}) \tag{2.26}$$

for all $I \leq N$. Then, calculate $b_I^{(k)}$ recursively for $k = 1, 2, \ldots, n_r$ using

$$b_I^{(k)}(\boldsymbol{Y}) = \begin{cases} b_I^{(k-1)}(\boldsymbol{Y}) & \text{for } 0 \leq i_r < k \\ (1-\lambda) b_{I_{r,-1}}^{(k-1)}(\boldsymbol{Y}) + \lambda b_I^{(k-1)}(\boldsymbol{Y}) & \text{for } k \leq i_r \leq n_r \end{cases} \tag{2.27}$$

where for each value of k, the computation is carried out for all $I \leq N$ in order of increasing i_r, i.e., starting with $i_r = 0$ and continuing with $i_r = 1, 2, \ldots, n_r$. Note that apart from the order of i_r the order within $I \leq N$ is not relevant.

[3] Only limited by machine precision.

The Bernstein coefficients $b_I(\boldsymbol{Y})$ and $b_I(\boldsymbol{Z})$ of the two new interval boxes $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are then given by

$$b_I(\boldsymbol{Y}) = b_I^{(n_r)}(\boldsymbol{Y}) \tag{2.28}$$

$$b_{(i_1,\ldots,n_r-k,\ldots,i_l)}(\boldsymbol{Z}) = b^{(k)}_{(i_1,\ldots,n_r,\ldots,i_l)}(\boldsymbol{Y}) \tag{2.29}$$

for all multi-indices $I \le N$.

Derivative Estimation

Two different cases must be considered when deciding in which direction the interval box should be split.

On the one hand, we would like to determine an accurate value set. Therefore, we would like to split the interval box in the direction where the bounds of the polynomial become as accurate as possible. We can make use of the fact that the bounds of a polynomial which is linear in all variables are exact. This is known as the *linear precision* property of the Bernstein approximation, see, e.g., [14]. The second derivative of a polynomial which is linear in all variables is zero. Therefore, we try to minimize the second derivative by splitting the interval box in the direction where the absolute value of the second partial derivative is largest.

On the other hand, we would like to determine the value set of the polynomial as exactly as possible for different sub-boxes. Therefore, we would like to split the interval box in the direction where the sizes of the value sets of the resulting sub-boxes are minimal. The size of the value set of a polynomial with first derivative zero is zero. Therefore, we try to minimize the first derivative by splitting the polynomial in the direction where the absolute value of the first partial derivative is largest.

An efficient method exists to estimate the μth partial derivative of a Bernstein polynomial and thereby select a direction for subdivision of the Bernstein polynomial which was first presented in [7].

Using the difference operator $\Delta_r^{(k)}$ defined as

$$\Delta_r^{(k)} b_I(\boldsymbol{X}) := \begin{cases} b_I(\boldsymbol{X}) & \text{for } k = 0 \\ \Delta_r^{(k-1)} b_{I_{r,1}}(\boldsymbol{X}) - \Delta_r^{(k-1)} b_I(\boldsymbol{X}) & \text{for } k > 0 \end{cases} \tag{2.30}$$

the μth partial derivative of the polynomial $\tilde{p}(\boldsymbol{u}, \boldsymbol{X})$ in direction r may be determined by

$$\frac{\partial^\mu \tilde{p}(\boldsymbol{u}, \boldsymbol{X})}{\partial u_r^\mu} = \frac{n_r!}{(n_r - \mu)!} \sum_{I \le N_{r,-\mu}} \Delta_r^{(\mu)} b_I(\boldsymbol{X}) B_{N_{r,-\mu},I}(\boldsymbol{u}). \tag{2.31}$$

See, e.g., [15] for a detailed derivation of this result. Equation (2.31) may be used to determine the partial derivative of a polynomial given in Bernstein form for any point in the normalized interval box $\boldsymbol{u} \in \boldsymbol{U}$.

From (2.31) an efficient algorithm for estimating the maximum partial derivatives

$$\max_{u \in U} \left| \frac{\partial^\mu \tilde{p}(\boldsymbol{u}, \boldsymbol{X})}{\partial u_r^\mu} \right| \le \tilde{I}_r^{(\mu)} \tag{2.32}$$

is derived in [7] as

$$\tilde{I}_r^{(\mu)} = \frac{n_r!}{(n_r-\mu)!} \max_{I \le N_{r,-\mu}} |\Delta_r^{(\mu)} b_I(\boldsymbol{X})| \tag{2.33}$$

by employing the triangle inequality and properties

$$\sum_{I \le N} B_{N,I}(\boldsymbol{u}) = 1 \qquad \forall\, \boldsymbol{u} \in \boldsymbol{U} \tag{2.34}$$

$$B_{N,I}(\boldsymbol{u}) \ge 0 \qquad \forall\, \boldsymbol{u} \in \boldsymbol{U}, I \le N \tag{2.35}$$

of the Bernstein polynomial[4].

2.3.7 Value Set of Complex Polynomials

A complex multivariate polynomial may be described by

$$\begin{aligned} p(\boldsymbol{x}) &= \sum_{I \le N} a_I \boldsymbol{x}^I && (2.36)\\ &= \sum_{I \le N} \mathrm{Re}(a_I \boldsymbol{x}^I) + j \sum_{I \le N} \mathrm{Im}(a_I \boldsymbol{x}^I). && (2.37) \end{aligned}$$

with $\boldsymbol{x} \in \mathbb{C}^l$ and $a_I \in \mathbb{C}$ for all $I \le N$ where N is a multi-index of the degree of the polynomial (see definitions in Section 2.3.1).

When determining the value set of such a complex polynomial using Bernstein coefficients two approaches are possible. On the one hand, we can separately determine Bernstein coefficients for the real and imaginary part of the complex polynomial. The resulting approximations of the value set are conservative in general, as the result is constrained to a complex interval-box with horizontal and vertical edges. On the other hand, complex Bernstein coefficients $b_I(\boldsymbol{X})$ can be determined which results in a more exact outer approximation than when separately considering the real and imaginary part. In addition, this allows to determine inner approximations of the value set.

The calculation of complex Bernstein coefficients $b_I(\boldsymbol{X})$ from the polynomial is identical to the calculation of real coefficients. First, the multivariate polynomial is normalized (see Section 2.3.3) resulting in

$$p(\boldsymbol{x}) = \sum_{I \le N} a_I \boldsymbol{x}^I = \sum_{I \le N} \tilde{a}_I(\boldsymbol{X}) \boldsymbol{u}^I = \tilde{p}(\boldsymbol{u}, \boldsymbol{X}). \tag{2.38}$$

Then, the complex Bernstein coefficients $b_I(\boldsymbol{X})$ are calculated as described in Section 2.3.4. These Bernstein coefficients are used in the following to determine inner and outer approximations of the value set.

[4] In [7] equation (2.32) is given as approximation instead of inequality. The inequality given here is a more precise formulation.

Outer Approximation

According to [7] the region in the complex plane in which the polynomial $\tilde{p}(\boldsymbol{u}, \boldsymbol{X})$ lies for $\boldsymbol{u} \in \boldsymbol{U}$ is bounded by the convex hull $\boldsymbol{H}(\boldsymbol{X})$ of the complex Bernstein coefficients $b_I(\boldsymbol{X})$ which is given by

$$\boldsymbol{H}(\boldsymbol{X}) = \text{hull}_{I \leq N}(b_I(\boldsymbol{X})) \tag{2.39}$$

where the operation $I \leq N$ is described in Section 2.3.1 and *hull* represents an operation to determine the convex hull of a set of complex values. Efficient algorithms exist to determine the convex hull of a set of 2D points, e.g., Graham's Scan [16]. These algorithms are directly applicable to a set of complex values. This results in an outer approximation of the value set of $\tilde{p}(\boldsymbol{u}, \boldsymbol{X})$ for $\boldsymbol{u} \in \boldsymbol{U}$ which can be used to determine whether a point (or interval) is excluded from the exact value set.

Inner Approximation

A more complicated approach is necessary to determine an inner approximation of the value set. Here, the method developed in [7] is used which is summarized below. The general idea is as follows: A simple method exists to determine the outer approximation of a value set. Instead of determining such an outer approximation for the complete parameter set an outer approximation of the value set is determined for different edges of the parameter set. An inner approximation of a face of the value set is then given by the region surrounded by the outer approximation of the edges surrounding this face. This inner approximation is then completely included in the exact value set.

This method is illustrated in Fig. 2.1 for a case with three parameters. On the left the parameter cube is depicted where one face has been selected (red) and the edges surrounding this face are marked in different colors. On the right, the convex hulls of the Bernstein coefficients associated with the different edges are depicted in the same colors. The red region surrounded by the four convex hulls represents an inner approximation of the value set for parameters on the red face of the parameter cube.

For the general case we consider a complex normalized polynomial $\tilde{p}(\boldsymbol{u}, \boldsymbol{X})$ depending on l parameters $\boldsymbol{u} \in \boldsymbol{U} \in [\mathbb{R}]^l$. There are

$$n_f = 2^{l-2} \binom{l}{2} \tag{2.40}$$

faces on the l-dimensional parameter hyper-cube. Each face has 4 edges of which the corresponding multi-indices E_1, E_2, E_3, E_4 can easily be determined as

$$E_i := \{(e_1, e_2, \ldots, e_k, \ldots, e_l) \mid 0 \leq e_k \leq n_k \wedge e_j = f_j \; \forall \; j \neq k\} \tag{2.41}$$

with index f_j an element of the multi-index F which is fixed for the edge and defined as

$$F := (f_1, f_2, \ldots, f_l) \quad \text{with } f_j = 0 \vee f_j = n_j, \; 0 \leq j \leq l, \tag{2.42}$$

i.e., each entry f_j of F can have one of the two extreme values 0 and n_j. Therefore, for an edge E_i where variable k is varied, there are $n_k + 1$ multi-indices in E_i and the k-th entry of the multi-index takes each value from 0 to n_k. For all of the multi-indices in E_i all entries j with $j \neq k$ are fixed to the values in F which depend on *which edge* of the parameter set is being considered.

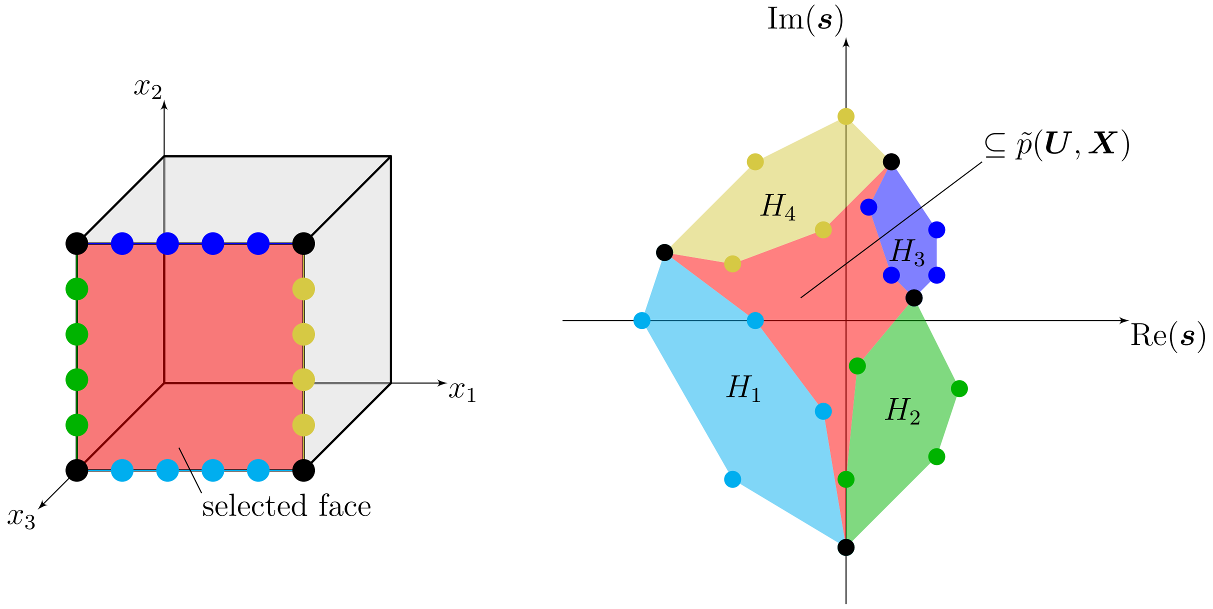

x_2
x_1
x_3
selected face
$\mathrm{Im}(\boldsymbol{s})$
$\subseteq \tilde{p}(\boldsymbol{U}, \boldsymbol{X})$
H_4
H_3
H_1
H_2
$\mathrm{Re}(\boldsymbol{s})$

Figure 2.1: Inner approximation of the value set of a complex polynomial. The parameter box (in this case 3 parameters) is depicted on the left. One face of the parameter box (red) is selected and the parameter values associated with Bernstein coefficients on these edges are marked in different colors. On the right, the corresponding Bernstein coefficients are depicted in the same colors together with the convex hull of the Bernstein coefficients associated with each edge. The red region on the right then represents an inner approximation of the Bernstein polynomial for parameters from the red face on the parameter box. By considering different faces of the parameter box different inner approximations may be derived.

The convex hulls $\boldsymbol{H}_i$ of the edges may then be determined from the Bernstein coefficients of the edges by

$$\boldsymbol{H}_i(\boldsymbol{X}) = \mathrm{hull}_{I \in E_i}(b_I(\boldsymbol{X})). \tag{2.43}$$

These hulls represent outer approximations of the value sets of the different edges of the parameter set. Thus, an inner approximation of the value set associated with the face of the parameter set surrounded by these edges is given by the region surrounded by the value sets of the four edges. Such an inner approximation can be determined for each face of the parameter set which results in different inner approximations of the value set of $\tilde{p}(\boldsymbol{u}, \boldsymbol{X})$ for $\boldsymbol{u} \in \boldsymbol{U}$ that can be used to determine whether a point (or interval) is included in the exact value set of $p(\boldsymbol{x})$.

2.4 Taylor Models

The Bernstein transformation introduced in the previous section allows to efficiently approximate the value set of multivariate polynomials. For the methods developed in the following chapters it is however necessary to determine the value set of *quasi-polynomials*. A *quasi-polynomial* is a polynomial where the coefficients are not fixed but depend on some variable non-linearly.

Value sets of a quasi-polynomial could be determined directly by evaluating the quasi-polynomial using interval arithmetic. This would however lead to a large overapproximation of the bounds due to the dependency problem. To avoid this problem, different centered forms and higher order inclusion functions have been developed over the years [17]. In this work we use Taylor Models, a higher order inclusion function which can be calculated efficiently and offers a tight enclosure of the exact function. For a thorough introduction to Taylor Models, see [18]. For a critical discussion on Taylor Models and related methods, see [17].

A Taylor Model may represent an arbitrary non-linear function with uncertainties using a multivariate polynomial and interval remainder. The approach to determine a Taylor Model of a function, which is introduced in [19] and used in this thesis, consists of first determining one Taylor Model for each uncertain parameter (specified by an interval). The Taylor Model of the complete function is then determined by evaluating the function using the arithmetic operations and intrinsic functions given in Section 2.4.2. In contrast to directly determining the Taylor Model of the complete function this has the advantage that no complex symbolic calculations are necessary[5].

Once the Taylor Model of the complete complex-valued function has been determined the value set of the polynomial part may be calculated by using the Bernstein transformation from the previous section. The value set of the overall function is then determined by taking the interval remainder into account, see Section 2.4.5.

2.4.1 Definition

A Taylor-Model $T(\boldsymbol{X})$ defined on the interval box $\boldsymbol{X}$ consists of a polynomial part $\tilde{p}(\boldsymbol{u}, \boldsymbol{X})$ of order m which is normalized to the unit interval box $\boldsymbol{U}$ (see Section 2.3.3) and an interval remainder $R(\boldsymbol{X})$, i.e.

$$T(\boldsymbol{X}) = (\tilde{p}(\boldsymbol{u}, \boldsymbol{X}), R(\boldsymbol{X})).$$

A Taylor Model can be used to approximate any arbitrary function. As $\tilde{p}(\boldsymbol{u}, \boldsymbol{X})$ may be a multivariate polynomial, any multivariate function $f(\boldsymbol{x})$ may be approximated by a Taylor Model. The exact value of the function must then always lie within an interval (the interval remainder) around the value of the polynomial, i.e.,

$$f(\boldsymbol{x}) \in \tilde{p}(\tilde{\boldsymbol{x}}, \boldsymbol{X}) + R(\boldsymbol{X}) \quad \forall\, \boldsymbol{x} \in \boldsymbol{X} \tag{2.44}$$

where

$$\tilde{\boldsymbol{x}} = \frac{\boldsymbol{x} - \underline{\boldsymbol{X}}}{\overline{\boldsymbol{X}} - \underline{\boldsymbol{X}}}. \tag{2.45}$$

To determine the Taylor Model of a mathematical expression, first each uncertain parameter is transformed to a Taylor Model. Given the l-dimensional uncertain parameter box $\boldsymbol{X}$ the corresponding Taylor models normalized to the unit interval box $\boldsymbol{U}$ are

$$\begin{aligned} T_i(\boldsymbol{X}) &= (\tilde{p}_i(\boldsymbol{u}, \boldsymbol{X}), R_i(\boldsymbol{X})) \\ \tilde{p}_i(\boldsymbol{u}, \boldsymbol{X}) &= \underline{X}_i + (\overline{X}_i - \underline{X}_i)\, u_i \\ R_i &= [0, 0] \end{aligned}$$

[5] It must however be pointed out that calculating a Taylor Series expansion of the complete function might lead to a smaller interval remainder.

with $1 \le i \le l$, i.e., each uncertain parameter X_i is represented as Taylor Model T_i with a polynomial of order one (offset $\underline{X}_i$, scale $(\overline{X}_i - \underline{X}_i)$) and empty interval remainder $R_i = [0, 0]$.

Then, the Taylor Model of the overall function $f(\boldsymbol{x})$ may be evaluated using the arithmetic operations and intrinsic functions given in Section 2.4.2.

For use in the following sections an operation to determine outer bounds of the value set of a Taylor model or polynomial is defined which is denoted by $B(\ldots)$. To bound the polynomial we can, e.g., use the Bernstein transformation as shown in Section 2.3.4. The bounds of the complete Taylor Model are then determined by the sum of the bounds of the polynomial part and the interval remainder:

$$B(T(\boldsymbol{X})) = B(\tilde{p}(\boldsymbol{u}, \boldsymbol{X})) + R(\boldsymbol{X}). \tag{2.46}$$

The definitions given in this section are valid for real and complex Taylor Models. In the following, we first consider the case of real Taylor Models. Special considerations for complex Taylor Models are discussed in Section 2.4.5.

2.4.2 Arithmetic

Arithmetic operations and many elementary functions have been defined for real Taylor Models [18]. For our case, the arithmetic operations addition, multiplication, negation, and multiplicative inverse are required. Note, that in the following round-off errors which arise in practical implementations are not taken into account[6].

Let $T_1(\boldsymbol{X})$ and $T_2(\boldsymbol{X})$ be two distinct Taylor Models with different polynomial part and interval remainder but same interval box $\boldsymbol{X}$:

$$T_1(\boldsymbol{X}) = (\tilde{p}_1(\boldsymbol{u}, \boldsymbol{X}), R_1(\boldsymbol{X})) = (\tilde{p}_1, R_1) \tag{2.47}$$
$$T_2(\boldsymbol{X}) = (\tilde{p}_2(\boldsymbol{u}, \boldsymbol{X}), R_2(\boldsymbol{X})) = (\tilde{p}_2, R_2) \tag{2.48}$$

To shorten notation, the dependence of the polynomial part and remainder on $\boldsymbol{X}$ and $\boldsymbol{u}$ is not given explicitly in the following definitions of arithmetic operations and required intrinsic functions.

Addition

An addition of two Taylor Models is performed by separately adding the polynomials and interval remainders, i.e.

$$T_1(\boldsymbol{X}) + T_2(\boldsymbol{X}) := (\tilde{p}_1 + \tilde{p}_2, R_1 + R_2). \tag{2.49}$$

Negation

Negation of a Taylor Model is achieved by simply negating the polynomial part and interval remainder, i.e.

$$-T_1(\boldsymbol{X}) := (-\tilde{p}_1, -R_1). \tag{2.50}$$

[6] In general, floating-point implementations have a finite precision and therefore only some values are exactly representable. For all other values, round-off errors occur. This must be taken into account in practical implementations to assure the inclusion property of Taylor Models.

Multiplication

The multiplication of two Taylor Models is slightly more complex as multiplication of two polynomials of order m would result in a polynomial of order $2m$. Thus, the Taylor Model multiplication is defined as

$$T_1(\boldsymbol{X})T_2(\boldsymbol{X}) := (\deg_m^-(\tilde{p}_1\tilde{p}_2), R_3) \tag{2.51}$$

with

$$R_3 := B(\deg_{m+1}^+(\tilde{p}_1\tilde{p}_2)) + B(\tilde{p}_1)R_2 + B(\tilde{p}_2)R_1 + R_1R_2) \tag{2.52}$$

where $\deg_m^-(\tilde{p}_1\tilde{p}_2)$ corresponds to all terms of $\tilde{p}_1\tilde{p}_2$ with degree smaller or equal m and $\deg_{m+1}^+(\tilde{p}_1\tilde{p}_2)$ corresponds to all terms of $\tilde{p}_1\tilde{p}_2$ with degree larger or equal $m+1$.

Multiplicative Inverse

The multiplicative inverse of a Taylor Model $T(\boldsymbol{X})$ is determined using the Taylor Series expansion as

$$\frac{1}{T(\boldsymbol{X})} := \sum_{k=0}^{m} \left((-1)^k \frac{\bar{T}(\boldsymbol{X})^k}{c_T^{k+1}} \right) + \left((-1)^{m+1} \frac{\bar{T}(\boldsymbol{X})^{m+1}}{c_T^{m+2} \left(1 + U\,\bar{T}(\boldsymbol{X})/c_T\right)^{m+2}} \right) \tag{2.53}$$

where U is the unit interval as defined in (2.2), c_T is the constant part of the Taylor Model $T(\boldsymbol{X})$, i.e.,

$$c_T := \tilde{p}(\mathbf{0}, \boldsymbol{X}) \tag{2.54}$$

and $\bar{T}(\boldsymbol{X})$ is the constant-free Taylor Model defined as

$$\bar{T}(\boldsymbol{X}) := T(\boldsymbol{X}) - c_T. \tag{2.55}$$

The last term of (2.53) which contains the *unit interval* U (see (2.2)) represents a pure interval while the sum contributes to the polynomial part and interval remainder.

2.4.3 Elementary Functions

In addition to the arithmetic operations, elementary functions for the natural exponential function e^x (also written as $\exp(x)$) or, alternatively, the trigonometric functions $\sin(x)$ and $\cos(x)$, are required to represent time delays. Using the definitions (2.54) and (2.55) the natural exponential function, Sine, and Cosine are derived using the Taylor Series expansion as

$$\exp(T(\boldsymbol{X})) := \exp(c_T) \sum_{k=0}^{m} \left(\frac{\bar{T}(\boldsymbol{X})^k}{k!} \right) + \exp(c_T) \frac{\bar{T}(\boldsymbol{X})^{m+1}}{(m+1)!} \exp(U\,\bar{T}(\boldsymbol{X})) \tag{2.56}$$

$$\begin{aligned} \sin(T(\boldsymbol{X})) := &\sum_{k=0}^{m} \left(\operatorname{trig}(c_T, k) \frac{\bar{T}(\boldsymbol{X})^k}{k!} \right) \\ &+ \operatorname{trig}(c_T, m+1) \frac{\bar{T}(\boldsymbol{X})^{m+1}}{(m+1)!} \operatorname{trig}(c_T + U\,\bar{T}(\boldsymbol{X}), m+1) \end{aligned} \tag{2.57}$$

$$\begin{aligned} \cos(T(\boldsymbol{X})) := &\sum_{k=0}^{m} \left(\operatorname{trig}(c_T, k+1) \frac{\bar{T}(\boldsymbol{X})^k}{k!} \right) \\ &+ \operatorname{trig}(c_T, m+2) \frac{\bar{T}(\boldsymbol{X})^{m+1}}{(m+1)!} \operatorname{trig}(c_T + U\,\bar{T}(\boldsymbol{X}), m+2) \end{aligned} \tag{2.58}$$

where the function trig(a, i) is defined as

$$\operatorname{trig}(a, i) := \begin{cases} \sin(a) & \text{for } \operatorname{mod}(i, 4) = 0 \\ \cos(a) & \text{for } \operatorname{mod}(i, 4) = 1 \\ -\sin(a) & \text{for } \operatorname{mod}(i, 4) = 2 \\ -\cos(a) & \text{for } \operatorname{mod}(i, 4) = 3 \end{cases} \tag{2.59}$$

and the last terms again represents a pure interval while the sum contributes to polynomial part and interval remainder.

2.4.4 Approximation of Value Set

The result of evaluating a mathematical expression using Taylor Models is a multivariate polynomial of maximum order m and an interval remainder. Therefore, bounds of an arbitrary function may be determined by bounding the multivariate polynomial and adding the interval remainder.

The main difficulty in determining tight bounds of a Taylor Model lies in determining tight bounds of the multivariate polynomial. The simplest approach to determine bounds of the polynomial is to use interval arithmetic. Herefore, each variable in the polynomial is replaced by the corresponding interval and interval arithmetic is used to determine bounds for the value of the polynomial. Much tighter bounds on the polynomial may be determined using more complex algorithms. In [18] a linear-dominated bounder is used to determine tighter bounds of the polynomial than would be achieved using pure interval arithmetic. Here, we instead transform the polynomial into Bernstein form [20]. Bernstein coefficients can be used to efficiently determine bounds of a polynomial. Moreover, Bernstein polynomials allow to determine bounds of complex polynomials which arise when evaluating the value set of a characteristic equation of a dynamic system.

The second aspect influencing how tight the bounds of a Taylor Model are, is the interval remainder. The size of the interval remainder is of order $m + 1$ with respect to the size of the interval-box $\boldsymbol{X}$ of the Taylor Model [18][7]. Thus, the size of the interval remainder may be reduced efficiently by subdividing the interval-box $\boldsymbol{X}$ of the Taylor Model. Note that a one sided normalized polynomial ($U \in [0,1]$) is used in our definition of Taylor Models compared to a centered normalization ($U \in [-1,1]$), e.g., in [18]. On the one hand, this leads to a larger interval remainder (for the same size of $\boldsymbol{X}$) which, however, retains the convergence order. On the other hand, the domain of the Taylor Model is the *unit interval box* and, therefore, no additional normalization is necessary before determining the Bernstein coefficients.

2.4.5 Value Set of Complex Taylor Models

The method presented in Section 2.3.7 can be used to determine the value set of complex-valued polynomials. In this section we extend the method and make it applicable to complex-valued Taylor Models. Thereby, arbitrary holomorphic functions can be examined which allows analyzing stability of time-delay systems.

A complex Taylor Model consists of a complex polynomial and complex interval remainder. Arithmetic operations and elementary functions on complex Taylor Models may be determined straightforwardly from the definition of arithmetic operations and elementary functions on complex numbers. These definitions describe how to determine the real and complex part of the result from the real and complex parts of the argument(s). Thus, the result of arithmetic operations and elementary functions on complex Taylor Models may be determined separately for the real and complex part using these definitions for complex numbers and the rules for real Taylor Models from Section 2.4.2 and Section 2.4.3.

We consider a complex Taylor Model $T(\boldsymbol{X}) = (\tilde{p}(\boldsymbol{u},\boldsymbol{X}), R(\boldsymbol{X}))$ with complex normalized polynomial $\tilde{p}(\boldsymbol{u},\boldsymbol{X})$ and complex interval remainder $R(\boldsymbol{X})$. This Taylor Model represents an approximation of the complex function $f(x):\ \mathbb{R}^l \to \mathbb{C}$, i.e.,

$$f(\boldsymbol{x}) \in \tilde{p}(\tilde{\boldsymbol{x}},\boldsymbol{X}) + R(\boldsymbol{X}) \quad \forall\, \boldsymbol{x} \in \boldsymbol{X} \tag{2.60}$$

where

$$\tilde{\boldsymbol{x}} = \frac{\boldsymbol{x} - \underline{\boldsymbol{X}}}{\overline{\boldsymbol{X}} - \underline{\boldsymbol{X}}}. \tag{2.61}$$

The goal is to determine an inner and outer approximation of the value set $f(\boldsymbol{X})$. Therefore, the method developed in [7] and summarized in Section 2.3.7 is modified to incorporate the interval remainder.

Outer Approximation

Calculating an outer approximation of the value set of $f(\boldsymbol{X})$ is relatively straightforward. First, an outer approximation of the polynomial part $\tilde{p}(\boldsymbol{U},\boldsymbol{X})$ is determined as in Section 2.3.7. Then, this is enlarged by adding the complex interval remainder to each point

[7] This means that if a Taylor model of order m is recomputed for an interval box with half the size, the size of the new interval remainder will be $(1/2)^m$ of the size of the original interval remainder.

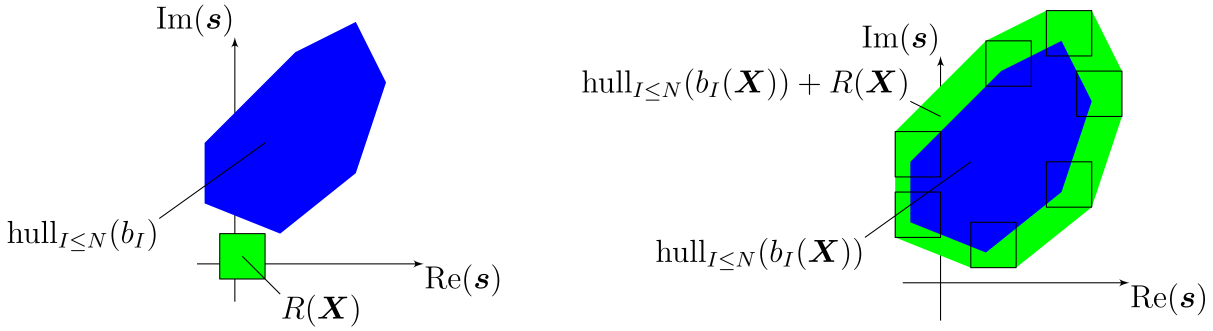

$\mathrm{Im}(\boldsymbol{s})$
$\mathrm{hull}_{I\leq N}(b_I)$
$R(\boldsymbol{X})$
$\mathrm{Re}(\boldsymbol{s})$
$\mathrm{Im}(\boldsymbol{s})$
$\mathrm{hull}_{I\leq N}(b_I(\boldsymbol{X})) + R(\boldsymbol{X})$
$\mathrm{hull}_{I\leq N}(b_I(\boldsymbol{X}))$
$\mathrm{Re}(\boldsymbol{s})$

Figure 2.2: Outer approximation of the value set of a complex Taylor Model. The convex hull of the Bernstein coefficients (dark blue) and the interval remainder (light green) are depicted on the left. The convex hull enlarged by adding the interval remainder, i.e. the outer approximation of the value set of the Taylor Model, is shown on the right (light green and dark blue region).

of the resulting value set. Graphically, the resulting value set can be constructed by moving the interval remainder rectangle along the edge of the value set of the polynomial part as illustrated in Fig. 2.2. The outer approximation of the value set of $f(\boldsymbol{X})$ is depicted on the right. It consists of the light green as well as the dark blue regions.

Inner Approximation

An inner approximation of the value set of $f(\boldsymbol{X})$ may be calculated analogously. As described in Section 2.3.7, an inner approximation of the polynomial part $\tilde{p}(\boldsymbol{U}, \boldsymbol{X})$ is determined using edges of the parameter set. Instead of directly using the convex hulls $\boldsymbol{H}_i$ of the Bernstein coefficients of the edges, these convex hulls are enlarged. This is done by adding the interval remainder to each point of each convex hull or, graphically, as in the previous section, by moving the interval remainder along the edges of the convex hulls. The area surrounded by the enlarged convex hulls represents an inner approximation of the value set of $f(\boldsymbol{X})$.

The method is illustrated in Fig. 2.3 for the same polynomial part as in Fig. 2.1. Note that the inner approximation of $f(\boldsymbol{X})$ (surrounded by the black line) is smaller than the inner approximation of the polynomial part $\tilde{p}(\boldsymbol{U}, \boldsymbol{X})$. As in the case of a pure polynomial, several different inner approximations may be determined by determining inner approximations for different faces of the parameter set.

A property of holomorphic functions, which is necessary in order for this inner approximation to be valid, is the so-called preservation of domains [9]. This states that a holomorphic function maps an open connected set to another open connected set. This implies that the exact value set cannot have any holes, which is essential for the inner approximation to be valid[8].

[8] Consider the case where the value set *can* have holes and edges of the parameter set are mapped to the complex plane and surround some region in the complex plane. As there can be a hole in the value set, the function does not necessarily take every value within this region for some parameter value on the corresponding face of the parameter set. Therefore, we cannot determine an inner approximation of the value set in this case, i.e., a region in the complex plane of which the function takes every value for some parameter value within the parameter set. Hence, the preservation of domains is a property of holomorphic functions which is essential for this inner approximation to be valid.

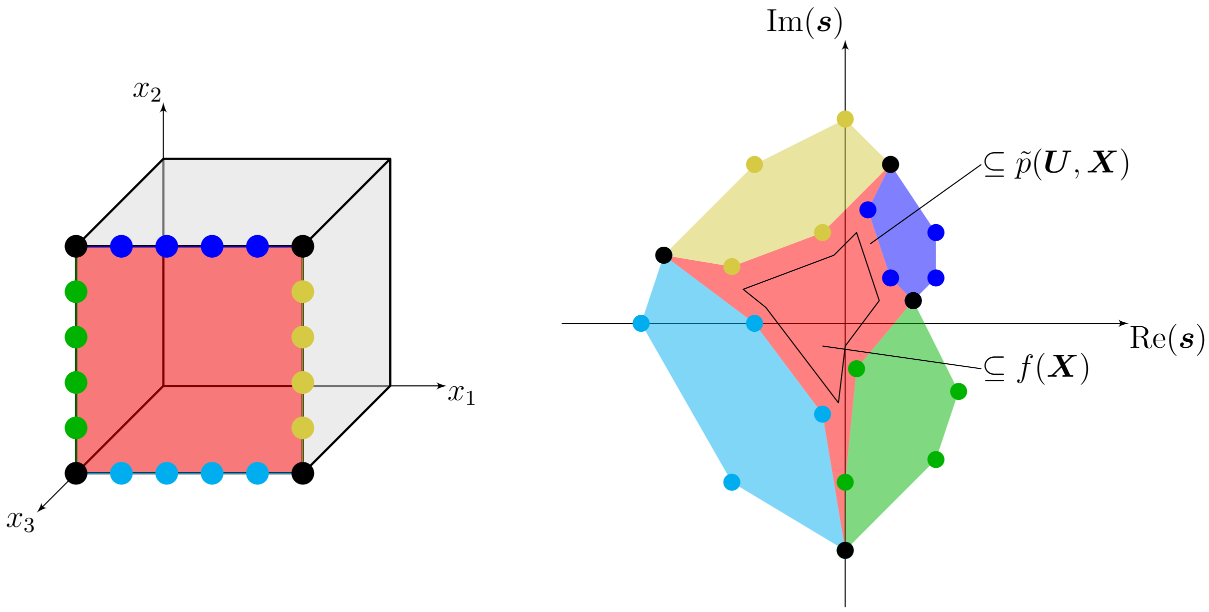

x_2
x_1
x_3
$\mathrm{Im}(\boldsymbol{s})$
$\mathrm{Re}(\boldsymbol{s})$
$\subseteq \tilde{p}(\boldsymbol{U}, \boldsymbol{X})$
$\subseteq f(\boldsymbol{X})$

Figure 2.3: Inner approximation of the value set of a complex Taylor Model. The parameter box (in this case 3 parameters) is depicted on the left. One face of the parameter box (red) is selected and the parameter values associated with Bernstein coefficients on these edges are marked in different colors. On the right, the corresponding Bernstein coefficients are depicted in the same colors together with the convex hull of the Bernstein coefficients associated with each edge. The red region on the right then represents an inner approximation of the Bernstein polynomial for parameters from the red face on the parameter box. This region is shrunk by enlarging each of the convex hulls of the edges by the interval remainder. This then results in the inner region surrounded by the black line, which represents an inner approximation of the value set of the complete complex Taylor Model. By considering different faces of the parameter box different inner approximations may be derived.

Implementation Details

Considering the implementation of complex Taylor Models it is important to distinguish between a *complex Taylor Model with imaginary value zero* and a purely *real Taylor Model*. This is important when considering multiplication with an unbounded Taylor Model. Multiplying two *real Taylor Models* should again result in a *real Taylor Model*, even if the real part of one of these Taylor Models is unbounded. If a *real Taylor Model* were simply a *complex Taylor Model with imaginary value zero* this would not be the case, as multiplying zero with an unbounded interval results in **NaI**, i.e. *Not an Interval*.

2.5 Summary

In this chapter, algorithms to evaluate the value set of holomorphic functions are introduced. These are based on algorithms which were first introduced by *Zettler, Garloff, et al.* in [7, 21]. They make use of the convex hull of Bernstein coefficients as well as the property that some Bernstein coefficients are sharp, to efficiently evaluate inner and outer approximations of a value set.

The main contribution of this chapter is an extension of these algorithms to examine the value set of Taylor Models instead of polynomials, thereby making it possible to consider holomorphic functions, e.g., quasi-polynomials. The result is an inner and outer approximation of the value set of the complex-valued function. The inner and outer approximation of the value set can be used to check whether a point is included in or excluded from the exact value set. If the point is excluded from the outer approximation it must also be excluded from the exact value set. Similarly, if a point is included in the inner approximation it must also be included in the exact value set. This property is used in the following chapter within a robust stability analysis method for time-delay systems.

3 Robust Stability Analysis of Time-Delay Systems

Summary. *In this chapter a method for stability analysis of Linear Time-Invariant (LTI) systems with incommensurate time delays and parametric uncertainties is developed. The method is based on interval-like computations using Taylor Models and polynomials in Bernstein form. In addition to examining Hurwitz stability, eigenvalues of a system may also be restrained to a region in the complex plane, e.g., to impose desired damping properties. The final result is a stability analysis method which can*

- *non-conservatively check stability of time-delay systems with interval parameters or*
- *map stable regions to a low-dimensional parameter space while taking additional interval parameters into account.*

This chapter introduces a stability analysis method for LTI systems with parametric uncertainties and time delay. More precisely, we consider time-delay systems of retarded type[1] with pointwise delays. This class of systems is found in many practical applications, e.g., teleoperation systems, chemical processes, transportation systems, and economics.

Existing methods for stability analysis of LTI systems with parametric uncertainties and time delay are limited. Only a few methods can simultaneously handle time delay and parametric uncertainty. In general, these methods are conservative or do not scale to real-world problems. For an overview of state of the art methods for stability analysis of systems with parametric uncertainties and/or time delay, see Section 3.1.

The general idea of the stability analysis method developed in this chapter is as follows. Using a branch and bound algorithm the value set of the characteristic function of a dynamic system is calculated for some boundary in the complex plane, e.g., the imaginary axis if asymptotic stability is examined. This allows mapping the boundary from the complex plane to the parameter space. Then, for each disjoint region in the parameter space (separated by a boundary), we check whether the region is stable or not using the same algorithm as for boundary mapping.

The resulting stability analysis method is applicable to time-delay systems with parametric uncertainties and incommensurate time delays. It can be used for parameter-space stability analysis or controller synthesis by determining stable and unstable regions in a

[1] For time-delay systems of retarded type, also known as Retarded Functional Differential Equation (RFDE), the derivative only depends on the current and past states and not on the past derivative, see [22].

low-dimensional (e.g., two-dimensional) parameter space. Additional parameters may be unknown (i.e., lie within a closed interval) and the desired resolution (and thus conservatism) of the boundary mapping may be specified directly in the parameter space (a higher resolution leads to longer calculation times). The algorithm can also be used to simply *check stability* for a *complete* set of interval parameters instead of *mapping* stability to the parameter space. For this simple stability check the results are non-conservative. The method can handle non-affine dependencies of the characteristic equation on the uncertain parameters, i.e., the coefficients of the characteristic equation may depend polynomially on the uncertain parameters.

The usefulness of the value set for robust stability analysis has been known for some time and is the foundation, e.g., of Kharitonov's Theorem [23] and the Edge Theorem [24]. The main difficulty in applying this test to systems, where there is a non-affine dependence of the characteristic equation on the system parameters, lies in the complexity of rigorously checking for zero inclusion/exclusion. In [7] it is shown that Bernstein Polynomials can be used to perform this check efficiently for the *delay-free* case. We build on the results from [7] and extend the method in several ways. We extend the branch and bound algorithm so as to not only check stability of a complete set of interval parameters, but also map stable regions to the parameter space. And, more importantly, by incorporating Taylor Models our method becomes applicable to time-delay systems, more specifically, to systems with multiple incommensurate time delays. A C++ implementation of the resulting algorithm is available for download at http://www.github.com/schauss/glob_stab.

This chapter is structured as follows. First, related work is discussed in Section 3.1. The problem formulation and general idea of the developed method is presented in Section 3.2. An overview of the overall algorithm developed for robust stability analysis is then introduced in Section 3.3. A more detailed description of the different steps of the algorithm is given in the following sections. Different regions in the complex plane are defined in Section 3.4 and implications of mapping these regions to the parameter space are discussed. As these regions are unbounded in general a numerically feasible method to evaluate the value set of unbounded regions in the complex plane is developed in Section 3.5. Then, a branch and bound algorithm is used to map bounded regions from the complex plane to the parameter space in Section 3.6. The stability analysis algorithm is applied to some simple examples with parametric uncertainties in Section 3.7 and to some examples with multiple incommensurate time delays in Section 3.8. The chapter ends with a discussion of our newly developed method in Section 3.9 and a conclusion as well as an outlook on future work in Section 3.10.

3.1 State of the Art

In this section an overview of existing methods for stability analysis of LTI systems with time delay and parametric uncertainties is given. As many methods are only applicable either to systems with parametric uncertainty or time delay, but not both, the literature review is split into three parts. First, analysis methods for systems with parametric uncertainty and no time delay are shortly summarized. The emphasis is on methods which are closely related to or form the basis of the method developed in this chapter. Then, a selection of methods for stability analysis of time-delay systems without parametric uncertainty is shortly summarized. Finally, the few methods that are applicable to time-delay systems with parametric uncertainties are introduced.

3.1.1 Robust Stability Analysis of Delay-Free Systems

The stability analysis of LTI systems with parametric uncertainty has been an active field of research for several decades. Within this field *Kharitonov's Theorem* [23] is probably the most well known theorem. It states that stability of a polynomial with independent interval coefficients can be checked by examining stability of four polynomials with fixed coefficients. On the one hand, this theorem is especially interesting as the number of polynomials that must be checked is independent of the system order. On the other hand, only few practical systems possess the required structure as there are generally dependencies between different coefficients of the polynomial. In this case, *Kharitonov's Theorem* is only a sufficient but not necessary condition and very conservative in general.

The *Edge Theorem* [24] builds on *Kharitonov's Theorem* and extends the (non conservative) applicability to a larger class of systems. Using the *Edge Theorem* systems with polytopic characteristic equation, i.e. systems with affine dependence of the coefficients of the characteristic equation on uncertain parameters, can be examined non-conservatively.

The parameter-space approach [1] is applicable to an even larger class of systems with polynomial dependence of the coefficients of the characteristic equation on uncertain parameters. It offers a straightforward intuitive approach to robust stability analysis. However, it is practically only applicable to systems with up to two (or at the most three) uncertain parameters as it is a graphical approach and cannot handle time delay. Moreover, the conditions are evaluated by a frequency sweep and depending on the step-width and upper bound of the frequency sweep instabilities may be missed.

A different approach to robust stability analysis is to use classic stability analysis methods and evaluate these methods using interval arithmetic. One such approach, based on evaluating the Routh-Array using interval arithmetic, is presented in [25]. Practical applicability is however limited to very simple systems due to the conservatism introduced by the dependency problem.

For practical problems with uncertainties the H_∞-norm [26] and structured singular value μ, introduced to robust control by *Safonov* [27] and *Doyle* [28], are often used. These methods can also capture other types of uncertainties, e.g. dynamic uncertainties. The H_∞-norm is conservative in general as it does not take the structure of the uncertainty into account. In contrast, the structured singular value μ, as the name suggests, takes the structure of the uncertainty into account and thus reduces conservatism. However, an exact calculation of the structured singular value is not possible in general. Therefore, bounds of the singular value must be approximated leading to significant conservatism [6]. For a thorough introduction to the structured singular value, see [29].

Finally, another group of methods is introduced which has achieved considerable results but is not very widely known. They are based on Bernstein Polynomials and have mainly been developed by *Garloff et al.*. In [30], a robustness analysis is carried out using Bernstein Polynomials. In [13] bounds of the Hurwitz Determinants are determined using Bernstein Polynomials and a branch and bound algorithm. To circumvent the computational complexity of calculating the Hurwitz Determinants, the method is modified in [7,21] to instead check for zero exclusion using the value set. As the value set is directly given by the characteristic equation this makes costly intermediate calculations unnecessary. The characteristic equation is transformed into Bernstein form. The resulting Bernstein coefficients represent points on the complex plane. Using the convex hull to determine exclusion of zero and an edge-check to determine inclusion of zero, stability is examined within a branch and bound algorithm. This results in a fast robust stability check for linear time-invariant systems which is applicable to systems of high order and with a large number of uncertain parameters[2]. This method is the basis of the stability analysis method which is developed in this chapter. As presented in [7, 21] the method is only applicable to delay-free systems. In this work, the method is extended so as to become applicable to systems with time delay.

All of the methods presented here operate in the frequency domain. Thereby, they often offer some interesting insights into the examined system. There are however also several time-domain stability analysis methods which allow a computationally efficient robust stability analysis. These are generally based on solving Lyapunov-functions using linear matrix inequalities (LMIs), see [31] for one example. Time-domain methods for robust stability analysis of delay-free systems are not further discussed here as this is outside the scope of this thesis.

3.1.2 Stability Analysis of Time-Delay Systems

Stability analysis of time-delay system has been studied extensively in the past. A good overview of many existing methods is given, e.g., in [6, 32, 33]. The presentation in [6] forms the basis of the literature review in this section. In addition, some newer methods are shortly summarized.

Two classes of time-delay systems are considered in the following. Systems with commensurate delays, i.e., all time delays of a system are integer multiples of one constant and systems with incommensurate delays where this condition does not hold. Naturally, systems with a single delay are a sub-class of systems with commensurate delays, while systems with commensurate delays are a sub-class of systems with incommensurate delays. Only few existing stability analysis methods are applicable to systems with incommensurate time delays. Moreover, a recent comparison of several methods shows that most existing methods are extremely conservative [34]. This is the case, although no parameter uncertainties were present in this study. Many of the methods in the following are only applicable to systems with commensurate delays. If a method is also applicable to systems with incommensurate delays it is mentioned explicitly. Even less stability analysis methods are applicable to systems with uncertainties and time delays, i.e., the robust stability analysis of time-delay systems. Of these methods, many have mainly been applied to systems without uncertainties but are also applicable to systems with uncertainties (in theory, often very conservative). These are introduced in this section and referred to in Section 3.1.3.

[2] *High order* refers to an order of around 10 to 20 while the allowed *number of uncertain parameters* is on the order of 10. Note that there are no exact limits to these numbers and they strongly depend on the exact problem and the hardware setup used for calculating the problem.

Additional methods, especially developed for robust stability analysis of time-delay systems are introduced in Section 3.1.3. This section mainly discusses frequency-domain methods but ends with a short introduction of some time-domain methods.

Frequency Domain

There are several stability tests in the frequency domain which are applicable to time-delay systems. These methods are based on the fact that stability of a time-delay system, as in the delay-free case is determined by the roots of the characteristic equation: if the real parts of all roots are negative then the system is stable. A detailed analysis can be found in the 1963 book by Bellman and Cooke [22] which is considered the basis of frequency-domain stability analysis of time-delay systems.

The main difficulty, in comparison with delay-free systems, is caused by the fact that the characteristic equation of a time-delay system is a quasi-polynomial with an infinite number of roots. Nevertheless, several methods have been developed which allow to examine whether all roots are in the open left half-plane.

For systems with commensurate delays several classical stability checks exist. Different *2d stability tests* exist which map the right half-plane to a unit disk or half-disk [6]. Another classical stability test for time-delay systems is the *pseudo-delay method*, also known as Rekasius substitution [35–38]. This, effectively, reduces the stability analysis of a time-delay system to the stability analysis of a delay-free system with a parameter-dependent polynomial thereby allowing the use of standard stability analysis tools, e.g., the Routh-Hurwitz criterion. In addition a *direct method* can be used to determine the zero-crossing frequencies and thereby stability which is based on the conjugate symmetry property of the quasi-polynomial [6]. These classical stability checks can be used to examine delay-independent as well as delay-dependent stability. The applicability of these checks is unfortunately limited to a low system order due to the need for complex symbolic calculations. Moreover, an extension to robust stability analysis is not straightforward.

An exact method for the stability analysis of systems with commensurate delays but without parametric uncertainties was introduced in [39] and is based on the Rekasius substitution. An extension to systems with parametric uncertainties is not straightforward, but further investigation may be of interest.

For systems with incommensurate delay, there are only few practically applicable stability checks in the frequency domain. These include some frequency sweeping tests based on the structured singular value μ as, e.g., developed by *Gu* [6]. These tests are conservative in general as it is not possible to calculate the structured singular value μ exactly, but nearly exact results are obtainable for some special cases, e.g., norm-bounded uncertainties [6].

Recently, two interesting methods which are applicable also to systems with incommensurate delays have been developed by the *Allgöwer* group [40]: a stability condition based on the Rekasius substitution is efficiently evaluated by sum of squares techniques [41] or, alternatively, linear programming [42]. These methods result in less conservative stability bounds than common time-domain approaches [34, 40] and are theoretically applicable to uncertain systems. To our best knowledge an application of these methods to uncertain systems has however not been published, and conservatism of this method when applied to uncertain systems has not been evaluated.

Another method, developed in [43] and compared to some other methods in [34], is known which results in non-conservative stability results for systems with incommensurate delays. This method computes stability-boundaries in the delay-space by solving a quadratic eigenvalue problem which is fast and non-conservative. However, as presented in [43] and used in [34] the problem is solved for a finite number of values from an interval which would actually have to be checked completely. Thus, an unstable system may falsely be characterized as stable. Moreover, it is not clear how and if this method can be extended to systems with parametric uncertainties. An analytic approach is pursued by *Gu et al.* who examine the stability crossing curves in the delay-space for general systems with two [44] and three [45] incommensurate time delays. The results are especially interesting as they are exact (i.e., only limited by machine precision) and offer some interesting insights into the stability boundaries of time-delay systems. An extension to systems with parametric uncertainties is, however, not easily possible. Finally, a stability analysis method which can examine systems with commensurate and incommensurate interval time delays excluding zero is introduced in [46]. For the case with one delay the authors claim their method is non-conservative, while for the case of incommensurate time delays a nearly exact stable box in the delay-space can be determined. However, the center point and aspect ratio of the box must be given. Moreover, an extension of the method to the stability analysis of systems with parametric uncertainties does not seem possible.

Time Domain

Stability of time-delay systems may also be examined in the time domain instead of the frequency domain. Thereby, generally, one of two stability criteria are utilized: *Lyapunov-Krasovskii functionals* or the *Razumikhin Theorem*. These criteria were first developed in the 1950s and 1960s. Especially in the field of *Lyapunov-Krasovskii functionals* much research has been performed in the past decades. This is due to the fact that although theoretically rather straightforward, practical application is difficult due to the *functional*. One of the most interesting methods which has been developed in this context is the *discretized Lyapunov functional* method which is applicable to systems with incommensurate delays and uncertainties [47]. Results have however only been presented for norm-bounded uncertainties and subpolytopic uncertainties. Several other methods employing *Lyapunov-Krasovskii functionals* or the *Razumikhin Theorem* have been developed. These are mostly based on linear matrix inequalities (LMIs) which result in a computationally efficient solution. The results are however quite conservative in practice, especially for systems with incommensurate delays [34] or uncertainties [6]. This is due to the fact that an exact solution of the *Lyapunov-Krasovskii functional* or the *Razumikhin Theorem* is not possible and therefore conservative sufficient conditions are checked which in turn guarantee that the exact conditions hold. The interested reader is referred to, e.g., [6] or [32] for an overview of several prominent methods.

3.1.3 Robust Stability Analysis of Time-Delay Systems

This section gives an overview of methods applicable to robust stability analysis of time-delay systems. Several of the presented methods have already been mentioned in the previous sections as they are also used either for (non-robust) stability analysis of time delay systems or for robust stability analysis of delay-free systems. In addition, some other methods are presented which were developed especially for robust stability analysis of time-delay systems. The section mainly concentrates on frequency-domain methods.

Several frequency-domain based methods for robust stability analysis of time-delay systems exist. As for the non-robust case, these are all based on the location of roots of the characteristic equation: if the real part of all roots is negative, the system is stable. This also holds for systems with time delay, see [22].

Based on this property several stability analysis methods which have been developed for robust stability analysis of delay-free systems have been extended to the case with delays. In *Gu et al.* [6], e.g., the *Edge Theorem*, and the *Multivariate Polynomial* approach are described for delay-dependent stability analysis. The *Edge Theorem*, first introduced in [24] and extended to the case with time delay in [48], evaluates the value set and zero exclusion principle, as does the method presented in this thesis, but is limited to systems with polytopic uncertainties and fixed time delay [48]. The *Multivariate Polynomial* approach transforms the stability analysis of time-delay systems where the characteristic function is a quasi-polynomial into the stability analysis of a multivariate polynomial. This remains challenging, and only for some special cases, e.g. an interval multivariate polynomial or, more general, a diamond family of multivariate polynomials a concrete solution is presented. A more in-depth presentation of this method can be found in [49–51]. In principle, the multivariate polynomial approach is applicable also to systems with incommensurate delays. Recently, the *Parameter-Space Approach* [1] has been extended to cases with time delay for some special cases, e.g., PID-controller design with one time delay in [52] and cascade control with time-delay in every cascade in [53]. In both cases, the limitations of the *Parameter-Space Approach* remain valid, i.e., at the most two to three uncertain parameters can be examined and a frequency-sweep is used which would theoretically have to be performed with infinitely small step size.

Two interesting methods based on an efficient evaluation of conditions derived using the Rekasius substitution [40–42] have already been discussed in Section 3.1.2. These methods are efficient for systems with one or two delays and no uncertainty [34]. However, although theoretically applicable to uncertain systems no evaluation has been performed, and it is expected that the results could be quite conservative in this case, depending on how the uncertainty is taken into account.

Finally, several of the time-domain methods based on *Laypunov-Krasovskii functionals* or the *Razumikhin Theorem* may be extended to the case of robust stability analysis, see, e.g., [6]. However, these conditions are already very conservative for the case without uncertainty [34] and generally become even more conservative when additional uncertainty is introduced [6]. Probably the most interesting time-domain method which has been developed in this context is the *discretized Lyapunov functional* method which is applicable to systems with incommensurate delays and norm-bounded uncertainties or subpolytopic uncertainties [47].

3.2 Problem Formulation and General Idea

We consider a Linear Time-Invariant (LTI) Retarded Functional Differential Equation (RFDE) with concentrated delays given by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)\boldsymbol{x} \tag{3.1}$$

with n states $\boldsymbol{x}$ and the system matrix $\boldsymbol{A}$ defined as

$$\boldsymbol{A}(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d) = \boldsymbol{A}_0(\boldsymbol{q}) + \sum_{i=1}^{m} \boldsymbol{A}_i(\boldsymbol{q}) e^{-t_{d,i}\boldsymbol{s}} \tag{3.2}$$

where $\boldsymbol{A}_i(\boldsymbol{q}) : \mathbb{R}^l \to \mathbb{R}^{n \times n}$. The system dynamics depend on l uncertain parameters $\boldsymbol{q} \in \boldsymbol{Q} \in [\mathbb{R}]^l$ and m uncertain time delays $\boldsymbol{t}_d \in \boldsymbol{T}_d \in [\mathbb{R}]^m$ (for interval notation, see Section 2.2) with

$$\boldsymbol{t}_d = [t_{d,1}\ t_{d,2}\ \dots\ t_{d,m}]^T \tag{3.3}$$

$$t_{d,i} > 0\ \ \forall\, 1 \le i \le m. \tag{3.4}$$

An equivalent representation of the system dynamics is

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_0(\boldsymbol{q})\boldsymbol{x}(t) + \sum_{i=1}^{m} \boldsymbol{A}_i(\boldsymbol{q})\boldsymbol{x}(t - t_{d,i}). \tag{3.5}$$

The overall goal is to determine stable and unstable regions in $\boldsymbol{Q} \times \boldsymbol{T}_d$. More specifically, the goal is to determine regions in $\boldsymbol{Q} \times \boldsymbol{T}_d$ for which there are no eigenvalues of $\boldsymbol{A}(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)$ in a specifiable region $\boldsymbol{\Gamma}$ in the complex plane. In the following this will be referred to as *mapping* a region from the complex plane to the parameter space. On the one hand, this allows to determine stable and unstable regions in the parameter space or simply check stability for a complete set of interval parameters. On the other hand, different constraints on the system dynamics, e.g., constraints on system damping can be introduced.

The characteristic function $f(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)$ of the system given by (3.1) is

$$f(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d) = \det(\boldsymbol{s}\boldsymbol{I} - \boldsymbol{A}(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)) = \sum_{i=0}^{m} \sum_{j=0}^{n} a_{ji}(\boldsymbol{q}) s^j e^{-t_{d,i}\boldsymbol{s}} \tag{3.6}$$

where $t_{d,0} = 0$ and thereby $e^{-t_{d,0}\boldsymbol{s}} = 1$. Alternatively, the characteristic function may be written as

$$f(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d) = p_0(\boldsymbol{s}, \boldsymbol{q}) + \sum_{i=1}^{m} p_i(\boldsymbol{s}, \boldsymbol{q}) e^{-t_{d,i}\boldsymbol{s}} = \sum_{j=0}^{n} a_j(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d) \boldsymbol{s}^j \tag{3.7}$$

with

$$p_i(\boldsymbol{s}, \boldsymbol{q}) = \sum_{j=0}^{n} a_{ji}(\boldsymbol{q}) s^j \tag{3.8}$$

$$a_j(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d) = \sum_{i=0}^{m} a_{ji}(\boldsymbol{q}) e^{-t_{d,i}\boldsymbol{s}}. \tag{3.9}$$

The roots of the characteristic function (3.6) correspond to the eigenvalues of the system matrix $\boldsymbol{A}(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)$ in (3.1). Thus, the roots of the characteristic function can be used to examine system stability and other system characteristics, e.g., system damping.

The system is asymptotically stable if and only if all roots of the characteristic function have a negative real part[3]. This is a well known fact for systems without time delay where the characteristic function is a polynomial and basis of many classical stability proofs. This property also holds for systems with time delay where the characteristic function is a quasi-polynomial as in (3.6), see [22] for the proof.

Finding the roots $\boldsymbol{s}_0$ of the characteristic function, i.e. solving the characteristic equation

$$f(\boldsymbol{s}_0, \boldsymbol{q}, \boldsymbol{t}_d) := 0 \tag{3.10}$$

for $\boldsymbol{s}_0$ is not easy in general due to the uncertain parameters $\boldsymbol{q} \in \boldsymbol{Q}$. Moreover, the characteristic equation (3.10) has an infinite number of solutions due to the time delays $\boldsymbol{t}_d \in \boldsymbol{T}_d$. However, instead of directly computing the roots we can compute the value set $f(\boldsymbol{\Gamma}, \boldsymbol{Q}, \boldsymbol{T}_d)$ of the characteristic function. Then, the following holds:

Lemma 3.1. There is *at least one root* of f in the region $\boldsymbol{\Gamma}$ for *some* parameter $\boldsymbol{Q}$ and *some* time delay $\boldsymbol{T}_d$ if and only if $f(\boldsymbol{\Gamma}, \boldsymbol{Q}, \boldsymbol{T}_d)$ *includes* the point *zero*.

Proof. If there is a root of f in the region $\boldsymbol{\Gamma}$, then $f(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d) = 0$ for some $\boldsymbol{q} \in \boldsymbol{Q}$, $\boldsymbol{t}_d \in \boldsymbol{T}_d$, $\boldsymbol{s} \in \boldsymbol{\Gamma}$, i.e., *zero* must be included in the value set $f(\boldsymbol{\Gamma}, \boldsymbol{Q}, \boldsymbol{T}_d)$ as the value set includes all values $f(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)$ takes for any values $\boldsymbol{q} \in \boldsymbol{Q}$, $\boldsymbol{t}_d \in \boldsymbol{T}_d$, $\boldsymbol{s} \in \boldsymbol{\Gamma}$. Likewise, if there is no root of f in the region $\boldsymbol{\Gamma}$, then $f(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d) \neq 0$ for all $\boldsymbol{q} \in \boldsymbol{Q}$, $\boldsymbol{t}_d \in \boldsymbol{T}_d$, $\boldsymbol{s} \in \Gamma$, i.e., *zero* cannot be included in the value set $f(\boldsymbol{\Gamma}, \boldsymbol{Q}, \boldsymbol{T}_d)$ as the value set does not include any values the function $f(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)$ does not take for any values $\boldsymbol{q} \in \boldsymbol{Q}$, $\boldsymbol{t}_d \in \boldsymbol{T}_d$, $\boldsymbol{s} \in \boldsymbol{\Gamma}$. □

Lemma 3.2. From Lemma 3.1 it directly follows that there is *no root* of f in the region $\boldsymbol{\Gamma}$ for *any* combination of parameter $\boldsymbol{Q}$ and time delay $\boldsymbol{T}_d$ if and only if $f(\Gamma, \boldsymbol{Q}, \boldsymbol{T}_d)$ *excludes* the point *zero*.

Lemma 3.1 and 3.2 are also the basis of the *zero exclusion principle* which is introduced in detail in Section 3.4.1 and allows to first examine the boundary of the $\boldsymbol{\Gamma}$-region and subsequently check stability of disjoint regions.

Up to now, applicability of the value set and zero exclusion principle has been limited as existing methods for calculating the value set are either conservative or cannot handle time delay or are limited to a few special forms of uncertainty. The novel contribution in this chapter is an algorithm which allows a calculation of the value set up to an arbitrary precision for a general LTI system with time delays and polynomial dependence of the coefficients of the characteristic function on parametric uncertainties. This allows to map a region $\boldsymbol{\Gamma}$ from the complex plane to the parameter space so as to examine stability or other important system characteristics.

[3] For a summary of different stability notions, see Appendix A.1

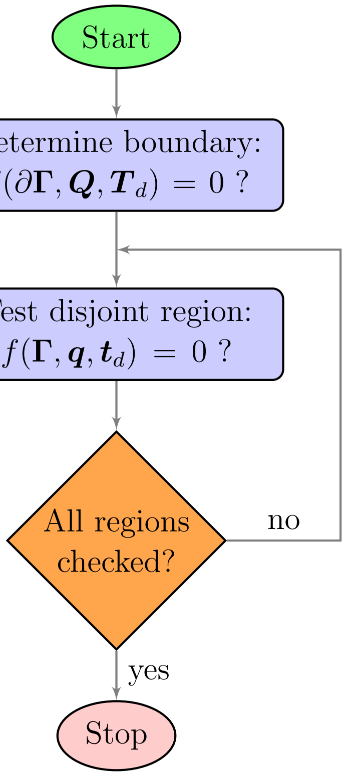

Start
Determine boundary:
$f(\partial\mathbf{\Gamma}, \boldsymbol{Q}, \boldsymbol{T}_d) = 0$?
Test disjoint region:
$f(\mathbf{\Gamma}, \boldsymbol{q}, \boldsymbol{t}_d) = 0$?
All regions checked?
no
yes
Stop

Figure 3.1: Flow-chart of overall algorithm used for robust stability analysis

3.3 Overall Algorithm

The overall algorithm distinguishes regions in $\boldsymbol{Q} \times \boldsymbol{T}_d$ for which no eigenvalue of the system matrix is in a given $\mathbf{\Gamma}$-region from regions in $\boldsymbol{Q} \times \boldsymbol{T}_d$ for which at least one Eigenvalue is in the $\mathbf{\Gamma}$-region. On the one hand, this allows an analysis of asymptotic stability. In this case $\mathbf{\Gamma}$ is the closed right half-plane, i.e., $\mathbf{\Gamma} = \{\boldsymbol{s} \mid \mathrm{Re}(\boldsymbol{s}) \geq 0\}$. On the other hand, desired system characteristics, e.g., minimum damping may be taken into account.

A flow chart of the overall algorithm is depicted in Fig. 3.1. As inputs a system matrix $\boldsymbol{A}(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)$ as well as bounded sets of system parameters $\boldsymbol{Q}$ and time delays $\boldsymbol{T}_d$ are required (see Section 3.2). Moreover, an appropriate region $\mathbf{\Gamma}$ in the complex plane (see Section 3.4) must be given. This $\mathbf{\Gamma}$ region will generally be unbounded, e.g., the closed right half-plane. The method developed in Section 3.5 is used to transform the unbounded $\mathbf{\Gamma}$ region to bounded $\mathbf{\Gamma}$ regions and associated boundaries $\partial\mathbf{\Gamma}$. Finally, the resolution of the boundary mapping must be set to a desired value for each parameter in $\boldsymbol{Q} \times \boldsymbol{T}_d$.

Using these inputs the actual stability analysis is performed. First, the boundaries $\partial\mathbf{\Gamma}$ are mapped to the parameter space using the branch and bound algorithm given in Section 3.6 and zero exclusion/inclusion is evaluated for the characteristic function $f(\partial\mathbf{\Gamma}, \boldsymbol{Q}, \boldsymbol{T}_d)$. Thereby, boundaries $\mathcal{B}$ are determined in the $\boldsymbol{Q} \times \boldsymbol{T}_d$ space for which one eigenvalue of the characteristic function passes $\partial\mathbf{\Gamma}$.

Due to the *zero-exclusion principle* which is introduced in detail in the following section, a change of the number of eigenvalues in the $\mathbf{\Gamma}$-region can only occur on the boundary $\mathcal{B}$. Therefore, all disjoint regions in $\boldsymbol{Q} \times \boldsymbol{T}_d$ not on the boundary $\mathcal{B}$ must either have no eigenvalue in the $\mathbf{\Gamma}$-region or at least one Eigenvalue in the $\mathbf{\Gamma}$-region, and it is sufficient to check whether there is an Eigenvalue in the $\boldsymbol{\Gamma}$-region for one fixed parametrization $\boldsymbol{q} \in \boldsymbol{Q}$ and time delay $\boldsymbol{t}_d \in \boldsymbol{T}_d$ in each disjoint region in $\boldsymbol{Q} \times \boldsymbol{T}_d$. The same branch and bound algorithm used for boundary mapping and given in Section 3.6 is used to evaluate whether an eigenvalue lies in the region $\boldsymbol{\Gamma}$. This is done by evaluating zero exclusion/inclusion for the value set $f(\boldsymbol{\Gamma}, \boldsymbol{q}, \boldsymbol{t}_d)$ for one value of $\boldsymbol{q}$ and $\boldsymbol{t}_d$ in each disjoint region.

This concludes the overall algorithm for robust stability analysis of time-delay systems. Depending on the resolution which is chosen for the different parameters in $\boldsymbol{Q} \times \boldsymbol{T}_d$ the algorithm either simply checks whether the system is stable for the complete set of interval parameters $\boldsymbol{Q} \times \boldsymbol{T}_d$ or maps stable and unstable regions to a low-dimensional sub-space of $\boldsymbol{Q} \times \boldsymbol{T}_d$[4].

In the following section, the *zero-exclusion principle* is introduced, different $\boldsymbol{\Gamma}$-regions are presented, and implications of mapping these regions to the parameter space are discussed. Then, the different steps of the overall algorithm are introduced in detail.

3.4 Regions in the Complex Plane

Roots of a polynomial (and also a quasi-polynomial with time delays) vary *continuously* with changes of the variables [22]. Therefore, instead of mapping a two-dimensional $\boldsymbol{\Gamma}$-region from the complex plane to the parameter space we may consider mapping the one-dimensional boundary $\partial\boldsymbol{\Gamma}$ of the region to the parameter space. The underlying concept is generally referred to as *zero-exclusion principle* and is introduced in the following. Then, different $\boldsymbol{\Gamma}$-regions are introduced and the implications of mapping these regions from the complex plane to the parameter space are discussed.

3.4.1 Zero-Exclusion Principle

The *zero-exclusion principle* has been known in the field of robust control for more than half a century. The method is described in a textbook by *Zadeh and Desoer* [54, Chapter 9.17] published in 1963 and a result by *Frazer and Duncan* [55] which is very close to the *zero-exclusion principle* was published in 1929. Recent textbooks on robust control mentioning the *zero-exclusion principle* include [1,6,56] where it is used, e.g., to prove such well known results as *Kharitonov's Theorem* [23] and the *Edge Theorem* [24,57]. For a more in-depth historical review of the *zero exclusion principle*, see [58].

The *zero exclusion principle* in its original form is defined for delay-free systems and therefore considers an uncertain polynomial

$$p(\boldsymbol{s}, \boldsymbol{q}) = \sum_{i=0}^{n} a_i(\boldsymbol{q}) \boldsymbol{s}^i \tag{3.11}$$

with uncertain parameters $\boldsymbol{q} \in \boldsymbol{Q}$ and coefficients $a_i(\boldsymbol{q})$ which smoothly depend on $\boldsymbol{q}$. Then, the coefficients a_i vary continuously with changes in the parameters $\boldsymbol{q}$, and the

[4] The resolution of the calculated stability boundaries is directly specified in the parameter space. In practice, these stability boundaries must be treated as unstable by the system designer. Thus, when mapping stable and unstable regions to a low-dimensional parameter space, the resolution with which the boundaries are determined coincides with the conservatism of the stability analysis.

following *zero-exclusion principle* holds:

Theorem 3.1. *A characteristic polynomial $p(\boldsymbol{s}, \boldsymbol{q})$ is robustly stable for all parameters $\boldsymbol{q} \in \boldsymbol{Q}$ if and only if*

- *the characteristic polynomial $p(\boldsymbol{s}, \boldsymbol{q})$ is stable for one set of parameters $\boldsymbol{q} \in \boldsymbol{Q}$,*
- *$0 \notin p(j\omega, \boldsymbol{Q})$ for all $\omega \in \mathbb{R}$, i.e., zero is excluded from the value set of the characteristic equation for all $\boldsymbol{s}$ on the imaginary axis.*

Proof. This follows directly from Lemma 3.1 and 3.2 and the fact that zeros of the polynomial p vary continuously with changes of $\boldsymbol{s}$ or changes of $\boldsymbol{q}$. □

The extension of the *zero-exclusion principle* to time-delay systems requires some additional assumptions which are taken from [6, Section 4.4] and are based on the results on root chains in [22]. For the class of systems considered in this thesis, LTI RFDEs with polynomial dependence of the coefficients of the characteristic function, these assumptions always hold:

Assumption 3.1. Every member of the uncertain quasi-polynomial f has a non-zero principal term, i.e.

$$\deg(p_0) = n \geq \deg(p_i),\ 1 \leq i \leq m. \tag{3.12}$$

This is necessary as a quasi-polynomial without a principal term cannot have all roots in the open left half-plane, see [6, Section 4.3]. For the case of LTI RFDEs which is considered in this thesis this assumption always holds as

$$\deg(p_0) = n > \deg(p_i),\ 1 \leq i \leq m. \tag{3.13}$$

Assumption 3.2. All time delays $t_{d,i}$ are positive for $i > 0$., i.e., we only consider causal systems. For the class of systems considered in this thesis this assumption is always true.

Assumption 3.3. An $R > 0$ and $\epsilon > 0$ must exist so that for all $\boldsymbol{q} \in \boldsymbol{Q}$ and $\boldsymbol{t}_d \in \boldsymbol{T}_d$ the quasi-polynomial $a_n(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)$ has either no zeros at all or no zeros of magnitude greater than R with real part greater than $-\epsilon$.

According to [6, Section 4.4], a sufficient condition for this assumption to hold is

$$\deg(p_0) = n > \deg(p_i),\ 0 < i \leq m \tag{3.14}$$

which corresponds to a system where all *root chains* are of *retarded type* which implies that for $\omega \to \infty$ the roots (of which there are an infinite number) all have a negative real part[5]. For LTI RFDEs which are considered in this thesis, this assumption always holds.

Assumption 3.4. The set $\boldsymbol{a} \times \boldsymbol{t}_d$ with the so-called coefficient vector

$$\boldsymbol{a} = (a_{00}, a_{01}, \ldots, a_{0m}, \ldots, a_{n0}, a_{n1}, \ldots, a_{nm}), \tag{3.15}$$

consisting of all coefficients a_{ji} in f, is compact and pathwise connected.

[5] For a brief summary on root-chains of time-delay systems and their properties, see Appendix A.2. For an in-depth analysis of this subject, see [22].

In our case this is always true: The set $\boldsymbol{Q}$ is an interval box which is compact and pathwise connected. The coefficients a_{ji} depend on the parameters $\boldsymbol{q} \in \boldsymbol{Q}$ polynomially. Therefore, as polynomials preserve compactness and pathwise connectedness, the coefficients a_{ji} are compact and pathwise connected. As $\boldsymbol{t}_d \in \boldsymbol{T}_d$ is an interval box and thereby compact and pathwise connected as well, the assumption holds.

Then, the *zero-exclusion principle* extended to time-delay systems with quasi-polynomials as characteristic function may be written as:

Theorem 3.2. *A characteristic function $f(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)$ satisfying the four assumptions is asymptotically stable for all parameters $\boldsymbol{q} \in \boldsymbol{Q}$ and time delays $\boldsymbol{t}_d \in \boldsymbol{T}_d$ if and only if*

- *$f(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)$ is asymptotically stable for one set of parameters $\boldsymbol{q} \in \boldsymbol{Q}$ and time delays $\boldsymbol{t}_d \in \boldsymbol{T}_d$, i.e., there are no roots in the closed right half-plane for one specific parametrization.*
- *$0 \notin f(j\omega, \boldsymbol{Q}, \boldsymbol{T}_d)$ for all $\omega \in \mathbb{R}$, i.e., zero is excluded from the value set of the characteristic function for all $\boldsymbol{s}$ on the imaginary axis.*

Proof. Due to Assumptions 3.1 to 3.4 the roots of the quasi-polynomial f vary continuously with changes of $\boldsymbol{q}$, and $\boldsymbol{t}_d$ (a detailed proof of this is given in [6, Section 4.4.2]). This implies that the number of roots in the right-half plane can only change if a root crosses the imaginary axis, i.e., there is a root on the imaginary axis for some parameter $\boldsymbol{q} \in \boldsymbol{Q}$ and time delay $\boldsymbol{t}_d \in \boldsymbol{T}_d$. Due to the first condition there is no root in the right half-plane for *one* parameter $\boldsymbol{q} \in \boldsymbol{Q}$ and time delay $\boldsymbol{t}_d \in \boldsymbol{T}_d$ and due to the second condition there is no root on the imaginary axis for *any* parameter $\boldsymbol{q} \in \boldsymbol{Q}$ and time delay $\boldsymbol{t}_d \in \boldsymbol{T}_d$, see Lemma 3.2. Therefore, there is no root in the right half-plane for *any* parameter $\boldsymbol{q} \in \boldsymbol{Q}$ and time delay $\boldsymbol{t}_d \in \boldsymbol{T}_d$. This a sufficient and necessary condition for asymptotic stability. □

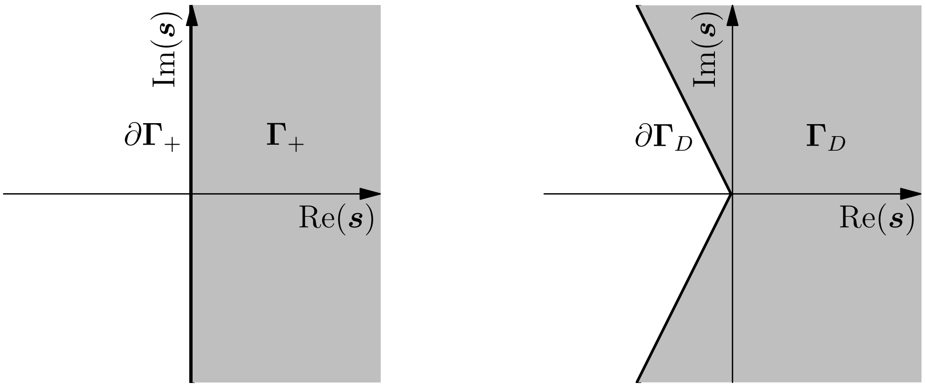
The extension of the *zero-exclusion principle* to different $\boldsymbol{\Gamma}$-regions is straightforward. In this case, we must assure that there are no roots within the $\boldsymbol{\Gamma}$-region for *one* parametrization and no roots on the boundary $\partial\boldsymbol{\Gamma}$ for *any* parametrization.

Corollary 3.1. A characteristic function satisfying the four assumptions given above has no roots in the two-dimensional region $\boldsymbol{\Gamma}$ for any parameter $\boldsymbol{q} \in \boldsymbol{Q}$ and time delay $\boldsymbol{t}_d \in \boldsymbol{T}_d$ if and only if

- the characteristic function has no roots in $\boldsymbol{\Gamma}$ for one specific parametrization $\boldsymbol{q} \in \boldsymbol{Q}$ and one specific time delay $\boldsymbol{t}_d \in \boldsymbol{T}_d$ and
- the characteristic function has no roots on the boundary $\partial\boldsymbol{\Gamma}$ for any parameter $\boldsymbol{Q}$ and time delay $\boldsymbol{T}_d$.

Similarly to the *zero-exclusion principle* a *zero-inclusion principle* can be formulated[6].

[6] The author is not aware of any prior work in which this has been published although it is equally straightforward. This may be due to the fact that practical applicability requires an inner approximation of the value set which is not usually available.



(a) Asymptotic stability

(b) Minimum damping

Figure 3.2: $\boldsymbol{\Gamma}$-regions used for system analysis. The $\boldsymbol{\Gamma}$-regions are shown in gray while the boundary is depicted as bold black line. To assure a certain system characteristic, no eigenvalues are allowed to lie within the respective $\boldsymbol{\Gamma}$-region.

Corollary 3.2. A characteristic function satisfying the four assumptions given above has a root in the two-dimensional region $\boldsymbol{\Gamma}$ for any parameter $\boldsymbol{q} \in \boldsymbol{Q}$ and time delay $\boldsymbol{t}_d \in \boldsymbol{T}_d$ if[7]

- the characteristic function has a root in $\boldsymbol{\Gamma}$ for one specific parametrization $\boldsymbol{q} \in \boldsymbol{Q}$ and one specific time delay $\boldsymbol{t}_d \in \boldsymbol{T}_d$ and
- the characteristic function has no roots on the boundary $\partial\boldsymbol{\Gamma}$ for any parameter $\boldsymbol{Q}$ and time delay $\boldsymbol{T}_d$.

From Corollary 3.1 and Corollary 3.2 it follows that mapping the boundary $\partial\boldsymbol{\Gamma}$ to the parameter space splits the parameter space into different regions which either have at least one eigenvalue in $\boldsymbol{\Gamma}$ or have no Eigenvalue in $\boldsymbol{\Gamma}$. These two cases (at least one eigenvalue / no Eigenvalue) must be discerned by a subsequent check for each disjoint region in the parameter space. In the following sections, different $\boldsymbol{\Gamma}$-regions are introduced. By mapping these regions to the parameter space, system parameters which result in different system properties (asymptotic stability, a minimum damping, or a minimum decay rate) can be determined.

3.4.2 Asymptotic Stability

A system is asymptotically stable if there is no eigenvalue of the system matrix in the closed right half-plane given by

$$\boldsymbol{\Gamma}_+ = [0, +\infty) + j(-\infty, +\infty). \tag{3.16}$$

[7] This is a sufficient but not necessary condition, as one root may lie in $\boldsymbol{\Gamma}$ and another root may pass the boundary $\partial\boldsymbol{\Gamma}$. In other words: a boundary in the parameter space not only separates regions with no roots in $\boldsymbol{\Gamma}$ from regions with roots in $\boldsymbol{\Gamma}$ but also separates two regions with a different number of roots in $\boldsymbol{\Gamma}$, even if both regions have roots in $\boldsymbol{\Gamma}$.

If there is at least one eigenvalue of the system matrix in $\mathbf{\Gamma}_+$, then the system is either unstable or marginally stable. See Fig. 3.2a for an illustration of the $\mathbf{\Gamma}_+$-region. The corresponding boundary $\partial\mathbf{\Gamma}_+$ is the imaginary axis

$$\partial\mathbf{\Gamma}_+ = 0 + j(-\infty, +\infty). \tag{3.17}$$

3.4.3 Minimum Damping

Instead of only checking for asymptotic stability other $\mathbf{\Gamma}$-regions may be mapped from the complex plane to the parameter space. A minimum system damping, e.g., may be enforced by mapping

$$\mathbf{\Omega}_D = (-\infty, +\infty) \tag{3.18}$$

$$\mathbf{\Gamma}_D = [-\delta|\mathbf{\Omega}_D|, +\infty) + j\mathbf{\Omega}_D \tag{3.19}$$

from the complex plane to the parameter space (see Fig. 3.2b). The corresponding boundary $\partial\mathbf{\Gamma}_D$ is

$$\partial\mathbf{\Gamma}_D = -\delta|\mathbf{\Omega}_D| + j\mathbf{\Omega}_D. \tag{3.20}$$

3.4.4 Minimum Decay Rate

The boundary for asymptotic stability $\partial\mathbf{\Gamma}_+$ or minimum damping $\partial\mathbf{\Gamma}_D$ may be shifted in direction of the negative real axis. Thereby, the real part of the eigenvalues is constrained to some negative value for all frequencies. This results in a system with given minimum decay rate.

Such a constraint can easily be taken into account by performing a coordinate transformation on the characteristic function. If $f(\mathbf{\Gamma}, \boldsymbol{Q}, \boldsymbol{T}_d)$ is the original characteristic function, then evaluating $f(\mathbf{\Gamma} - \epsilon, \boldsymbol{Q}, \boldsymbol{T}_d)$ shifts the region gamma by $-\epsilon$ along the real axis.

3.5 Transformation of Unbounded Regions

The regions $\mathbf{\Gamma}$ as well as the boundaries $\partial\mathbf{\Gamma}$ are unbounded intervals as at least one boundary is infinity. An evaluation of the value set for unbounded intervals results in an unbounded value set which consists of Bernstein coefficients with infinite magnitude as well as an interval remainder of infinite size. Unfortunately, the zero-exclusion and zero-inclusion checks introduced in Section 3.6.1 cannot be applied to unbounded value sets[8].

The problem of unbounded regions is not specific to the value set approach but arises in many stability analysis methods which rely on a frequency sweep. A frequently used workaround to this problem is to replace ±∞ with $\pm c$, with c a large finite constant, e.g., in the *Parameter Space Approach* in [1, Chapter 2-4]. This is problematic as not all of $\mathbf{\Gamma}$ or $\partial\mathbf{\Gamma}$ is checked.

[8] The zero-inclusion check and zero-exclusion check in Section 3.6.1 are based on checking whether the negative interval remainder is included in or excluded from a convex hull of Bernstein coefficients. Neither the inclusion-check nor the exclusion-check can succeed for an unbounded value set as we cannot determine whether an interval remainder of infinite size is included in or excluded from a convex hull with Bernstein coefficients of infinite magnitude.

An alternative approach which has been developed in the context of the zero exclusion principle and value set evaluation is to determine upper and lower bounds for the frequency ω for which a crossing of the imaginary axis is possible [21]. Then, the subsequent value set evaluation can be carried out on this reduced frequency range. This approach is however not applicable to systems with time delay. Moreover, no method is known to us to determine these limits when searching for roots in a two-dimensional region, e.g., the right half-plane.

We propose a method to circumvent the problems associated with $\pm\infty$ or large finite boundaries $\pm c$. Instead of simply evaluating the value set $f(\boldsymbol{s}, \boldsymbol{Q}, \boldsymbol{T}_d)$ and checking for zero inclusion and zero exclusion we also evaluate the value set of $\bar{f}(\boldsymbol{s}, \boldsymbol{Q}, \boldsymbol{T}_d)$ which is defined as[9]

$$\begin{aligned} \bar{f}(\boldsymbol{s}, \boldsymbol{Q}, \boldsymbol{T}_d) &:= f(\boldsymbol{s}^{-1}, \boldsymbol{Q}, \boldsymbol{T}_d)\boldsymbol{s}^n && (3.21) \\ &= \sum_{i=0}^{n} a_i(\boldsymbol{s}^{-1}, \boldsymbol{Q}, \boldsymbol{T}_d)\boldsymbol{s}^{-i}\boldsymbol{s}^n = \sum_{i=0}^{n} a_i(\boldsymbol{s}^{-1}, \boldsymbol{Q}, \boldsymbol{T}_d)\boldsymbol{s}^{(n-i)} && (3.22) \end{aligned}$$

where n is the system order as the following theorem holds.

Theorem 3.3. *When examining systems without time delay or asymptotic stability of time-delay systems the value set of $f(\boldsymbol{s}^{-1}, \boldsymbol{Q}, \boldsymbol{T}_d)$ includes zero if and only if the value set of $\bar{f}(\boldsymbol{s}, \boldsymbol{Q}, \boldsymbol{T}_d)$ includes zero, i.e.,*

$$f(\boldsymbol{s}^{-1}, \boldsymbol{Q}, \boldsymbol{T}_d) = 0 \Leftrightarrow \bar{f}(\boldsymbol{s}, \boldsymbol{Q}, \boldsymbol{T}_d) = 0 \qquad (3.23)$$

Proof. For any choice of $\boldsymbol{s} \neq 0$, it is clear that

$$f(\boldsymbol{s}^{-1}, \boldsymbol{Q}, \boldsymbol{T}_d) = 0 \Leftrightarrow \bar{f}(\boldsymbol{s}, \boldsymbol{Q}, \boldsymbol{T}_d) = f(\boldsymbol{s}^{-1}, \boldsymbol{Q}, \boldsymbol{T}_d)\boldsymbol{s}^n = 0$$

as

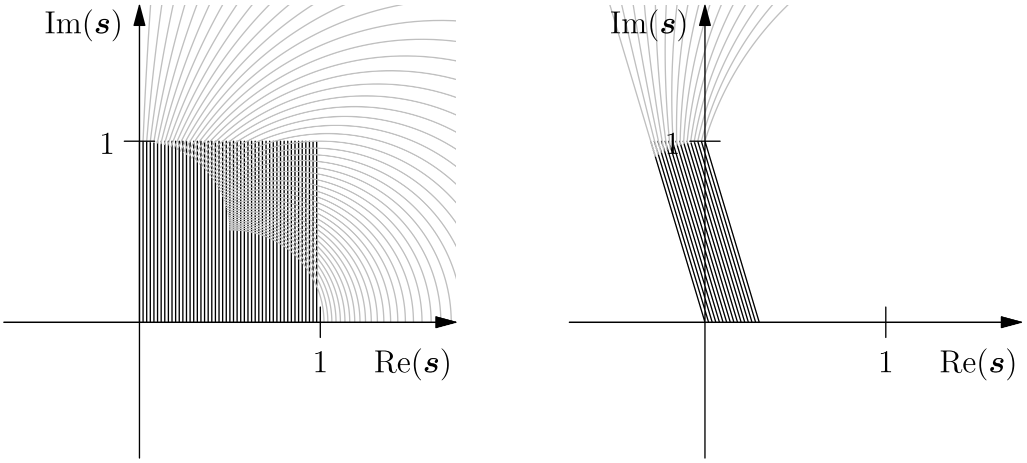
$$\boldsymbol{s} \neq 0 \Leftrightarrow \boldsymbol{s}^n \neq 0.$$

For $\boldsymbol{s} \to 0$ the evaluation of $\bar{f}(\boldsymbol{s}, \boldsymbol{Q}, \boldsymbol{T}_d) = 0$ results in

$$\begin{aligned} \lim_{\boldsymbol{s}\to 0} \bar{f}(\boldsymbol{s}, \boldsymbol{Q}, \boldsymbol{T}_d) = \lim_{\boldsymbol{s}\to 0} f(\boldsymbol{s}^{-1}, \boldsymbol{Q}, \boldsymbol{T}_d)\boldsymbol{s}^n &= \lim_{\boldsymbol{s}\to 0} \sum_{i=0}^{n} a_i(\boldsymbol{s}^{-1}, \boldsymbol{Q}, \boldsymbol{T}_d)\boldsymbol{s}^{(n-i)} && (3.24) \\ &= \lim_{\boldsymbol{s}\to 0} a_n(\boldsymbol{s}^{-1}, \boldsymbol{Q}, \boldsymbol{T}_d) = 0, && (3.25) \end{aligned}$$

which corresponds to the so-called *infinite root-boundary* as, e.g., defined in [1]. Note that for the case with time delay this theorem only holds when examining asymptotic stability as all terms a_i which depend on the time-delay term would become unbounded for $\boldsymbol{s} \to 0$ if $\boldsymbol{s}$ is not constrained to the right half-plane. This is explained in more detail in Section 3.5.3. □

[9] In the delay-free case this corresponds to the *non-conjugate* reciprocal polynomial.



(a) black: $\boldsymbol{\Gamma}_1$, gray: $(\boldsymbol{\Gamma}_1^{-1})^*$ **(b)** black: $\boldsymbol{\Gamma}_2$, gray: $(\boldsymbol{\Gamma}_2^{-1})^*$

Figure 3.3: Representation of unbounded regions $\boldsymbol{\Gamma}_+$ and $\boldsymbol{\Gamma}_D$ by bounded value sets $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$ and the complex conjugate of their inverse, where $\boldsymbol{\Gamma}_1 = \boldsymbol{\Sigma}_1 + j\boldsymbol{\Omega}_1$ and $\boldsymbol{\Gamma}_2 = \boldsymbol{\Sigma}_2 - \delta\boldsymbol{\Omega}_1 + j\boldsymbol{\Omega}_1$ with $\boldsymbol{\Omega}_1 = [0,1]$, $\boldsymbol{\Sigma}_1 = [0,1]$, $\boldsymbol{\Sigma}_2 = [0,\delta]$, and $\delta = 0.3$. The different lines represent different gridded values of $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$.

Using Theorem 3.3 the value set evaluation can be transformed from a check on *one unbounded region* to checks on *several bounded regions* as will be shown in the following. If the value set for *all* of the regions excludes zero, then there is no root in the complete region. Alternatively, if the value set for *at least one* of the regions includes zero, then there is a root somewhere in the complete region.

This method is illustrated here first for the imaginary axis $\partial\boldsymbol{\Gamma}_+$ and right half-plane $\boldsymbol{\Gamma}_+$. Subsequently, modifications necessary for the boundary $\partial\boldsymbol{\Gamma}_D$ and region $\boldsymbol{\Gamma}_D$ are given and the implications on time-delay terms are discussed. In all cases, it is sufficient to only consider regions with non-negative imaginary part as roots are either real or complex conjugate pairs [22].

3.5.1 Asymptotic Stability

For the case of asymptotic stability we consider the imaginary axis $\partial\boldsymbol{\Gamma}_+$. Two value sets are examined: $f(\partial\boldsymbol{\Gamma}_1, \boldsymbol{Q}, \boldsymbol{T}_d)$ and $\bar{f}(\partial\boldsymbol{\Gamma}_1, \boldsymbol{Q}, \boldsymbol{T}_d)$ with

$$\partial\boldsymbol{\Gamma}_1 = j\boldsymbol{\Omega}_1 \tag{3.26}$$

where

$$\boldsymbol{\Omega}_1 = [0,1]. \tag{3.27}$$

The first value set covers the region $\partial\boldsymbol{\Gamma}_1 = j[0,1]$ while the second value set covers the region $\partial\boldsymbol{\Gamma}_1^{-1} = j(-\infty,-1]$ without actually evaluating the characteristic function for the unbounded interval but instead evaluating the transformed characteristic function for the bounded region $\partial\boldsymbol{\Gamma}_1$. Thus, as roots are either real or complex conjugate pairs, the complex imaginary axis $\partial\boldsymbol{\Gamma}_+ = (-\infty,\infty)$ is covered by evaluating these two bounded value sets.

Therefore, if one of the value sets includes zero then there is a root on $\partial\mathbf{\Gamma}_+$. Likewise, if both value sets exclude zero there is no root on $\partial\mathbf{\Gamma}_+$.

Next, consider the right half-plane given by $\mathbf{\Gamma}_+$. Again, two value sets are examined: $f(\mathbf{\Gamma}_1, \boldsymbol{Q}, \boldsymbol{T}_d)$ and $\bar{f}(\mathbf{\Gamma}_1, \boldsymbol{Q}, \boldsymbol{T}_d)$ with

$$\mathbf{\Gamma}_1 = \mathbf{\Sigma}_1 + j\mathbf{\Omega}_1 \tag{3.28}$$

where

$$\mathbf{\Sigma}_1 = [0, 1] \tag{3.29}$$

and $\mathbf{\Omega}_1$ is given in (3.27). The first value set covers the region $\mathbf{\Gamma}_1 = [0, 1] + j[0, 1]$. The second value set covers the region $\mathbf{\Gamma}_1^{-1} = 1/([0, 1] + j[0, 1])$. The regions covered by $\mathbf{\Gamma}_1$ and $(\mathbf{\Gamma}_1^{-1})^*$ are depicted in Fig. 3.3a. Due to symmetry reasons the two regions cover the complete right half-plane[10]. Therefore, if one of the value sets includes zero then there is a root on $\mathbf{\Gamma}_+$. Likewise, if both value sets exclude zero there is no root on $\mathbf{\Gamma}_+$.

3.5.2 Minimum Damping

The problem becomes more complex for the boundary $\partial\mathbf{\Gamma}_D$ and region $\mathbf{\Gamma}_D$ which are considered next. The boundary $\partial\mathbf{\Gamma}_{D+}$ (the part of $\partial\mathbf{\Gamma}_D$ with non-negative imaginary part) is given by

$$\partial\mathbf{\Gamma}_{D+} = -\delta\mathbf{\Omega}_{D+} + j\mathbf{\Omega}_{D+} \tag{3.30}$$

with

$$\mathbf{\Omega}_{D+} = [0, +\infty) \tag{3.31}$$

and damping δ. Two value sets are examined: $f(\partial\mathbf{\Gamma}_2, \boldsymbol{Q}, \boldsymbol{T}_d)$ and $\bar{f}(\partial\mathbf{\Gamma}_2, \boldsymbol{Q}, \boldsymbol{T}_d)$ with

$$\partial\mathbf{\Gamma}_2 = -\delta\mathbf{\Omega}_1 + j\mathbf{\Omega}_1 \tag{3.32}$$

where $\mathbf{\Omega}_1$ is given in (3.27). The first value set covers the boundary $\partial\mathbf{\Gamma}_D$ for the imaginary value in the range $[0, 1]$ and thereby, as roots are either real or complex conjugate pairs, the range $[-1, 1]$. The rest of the boundary is covered by the second value set (with some overlap).

The region $\mathbf{\Gamma}_{D+}$ (the part of $\mathbf{\Gamma}_D$ with non-negative imaginary part) is

$$\mathbf{\Gamma}_{D+} = [-\delta\mathbf{\Omega}_{D+}, +\infty) + j\mathbf{\Omega}_{D+}. \tag{3.33}$$

This can be handled similarly as the right half-plane was handled in the previous section. In this case, four value sets are examined: $f(\mathbf{\Gamma}_1, \boldsymbol{Q}, \boldsymbol{T}_d)$, $\bar{f}(\mathbf{\Gamma}_1, \boldsymbol{Q}, \boldsymbol{T}_d)$, $f(\mathbf{\Gamma}_2, \boldsymbol{Q}, \boldsymbol{T}_d)$, and $\bar{f}(\mathbf{\Gamma}_2, \boldsymbol{Q}, \boldsymbol{T}_d)$ with $\mathbf{\Gamma}_1$ given in (3.28) and

$$\mathbf{\Gamma}_2 = \mathbf{\Sigma}_2 - \delta\mathbf{\Omega}_1 + j\mathbf{\Omega}_1 \tag{3.34}$$

[10] As can be seen in Fig. 3.3a some parts of the complex plane are covered by both value sets. This is not problematic for the algorithms used in the following sections.

where

$$\mathbf{\Sigma}_2 = [0, \delta] \tag{3.35}$$

and $\mathbf{\Omega}_1$ is given in (3.27). The regions covered by the two additional value sets $f(\mathbf{\Gamma}_2, \boldsymbol{Q}, \boldsymbol{T}_d)$ and $\bar{f}(\mathbf{\Gamma}_2, \boldsymbol{Q}, \boldsymbol{T}_d)$ are depicted in Fig. 3.3b. Together, the four value sets cover the complete region $\mathbf{\Gamma}_D$.

3.5.3 Time Delay

The transformation introduced in this section must also be applied to the time-delay terms, i.e., if f contained delay-terms with delays different from zero, then $\bar{f}$ contains terms $e^{-t_{d,i}\boldsymbol{s}^{-1}}$. The term $\boldsymbol{s}^{-1}$ is unbounded when $\boldsymbol{s}$ includes zero. In the following, a solution to this problem is presented for asymptotic stability checks.

The exponential function of a complex value x may be written as

$$e^x = e^{\mathrm{Re}(x)} e^{\mathrm{Im}(x)} = e^{\mathrm{Re}(x)} \left(\cos(\mathrm{Im}(x)) + j \sin(\mathrm{Im}(x))\right) \tag{3.36}$$

where the first factor specifies the magnitude and the second factor is a rotation around the origin. For the case of asymptotic stability, the real part of the exponent is non-positive, i.e., the maximum magnitude $e^{\mathrm{Re}(x)}$ is one. The imaginary part of the exponent is unbounded and includes infinity for $\boldsymbol{s}$ including zero, and therefore the Taylor Model of the complete exponential term also becomes unbounded. However, we know that the imaginary part of the exponential function can be represented as sin and cos which are both limited to the interval $[-1, 1]$. Therefore, the exponential term can be evaluated using interval arithmetic instead of Taylor Models in the case when $\boldsymbol{s}$ includes zero by replacing the second term with an interval box $[-1, 1] + j[-1, 1]$ and multiplying this interval box with the interval of the magnitude[11]. Thus, the largest possible interval of a time delay term for the case of asymptotic stability is $[-1, 1] + j[-1, 1]$.

This is unfortunately not the case for $\mathbf{\Gamma}_D$ regions with a minimum damping. In this case, the real part of $\mathbf{\Gamma}_D$ and $\partial\mathbf{\Gamma}_D$ is partly negative. E.g., for $\partial\mathbf{\Gamma}_D$ the real part goes towards $-\infty$ for $\Omega_D \to +\infty$, i.e., the magnitude of the exponential function becomes unbounded. Therefore, the transformation of unbounded regions to bounded regions is only possible when either considering minimum system damping *or* time delays. For the combination, i.e., determining parameters which result in a desired minimum damping for time-delay systems, it is not possible to transform the unbounded region to bounded ones. In this case we must, as with other frequency sweeping tests, replace ∞ by a large constant.

[11] In the actual implementation this replacement is only carried out when mapping complete $\mathbf{\Gamma}$ regions. In this case, the overapproximation of the Taylor Model is severe and this is the only strategy which has been identified to cope with this problem. However, replacing the Taylor Model with an interval box impairs the applicability of gradient estimation methods as the size of the interval box may be insensitive to parameter changes. This, in turn, reduces the convergence rate of the branch and bound algorithm. Therefore, this replacement is not carried out for boundary mapping.

3.6 Branch and Bound Algorithm

The general idea of the stability analysis method is to distinguish regions in the parameter space which have a root in $\mathbf{\Gamma}$ from regions which do not. This can either be used to determine stable regions in a low-dimensional parameter space or to simply check if a system is stable for a complete set of interval parameters. This section introduces the branch and bound algorithm used to map boundaries $\partial\mathbf{\Gamma}$ or regions $\mathbf{\Gamma}$ from the complex plane to the parameter space. A branch and bound algorithm is necessary for this due to two reasons.

On the one hand, the zero exclusion and zero inclusion check used for stability analysis and introduced in Section 3.6.1 are both sufficient but not necessary conditions, i.e., if both of these conditions are false for a given region in the parameter space we do not know whether this region includes or excludes zero. Then, we subdivide the region into two parts and separately reevaluate zero inclusion and exclusion for these two parts. This is done recursively until one of the checks succeeds which will be the case eventually, as the inner and outer approximation of the value set becomes more accurate with each subdivision, see Section 3.6.3. On the other hand, the stability analysis method can be used to graphically examine stability in the parameter space. Therefore, we want to determine regions with a desired resolution for plotting. In that case, we do not want to determine stability separately for each region with the desired resolution as this is computationally very expensive[12]. Instead, we start out by checking the complete region and only subdivide the region into smaller regions if necessary.

We assume that all regions $\mathbf{\Gamma}$ and boundaries $\partial\mathbf{\Gamma}$ that are used as input to this algorithm are bounded. Such bounded regions (and associated characteristic functions) are determined using the appropriate transformations given in Section 3.5. To simplify notation the *extended parameter set* $\tilde{\boldsymbol{Q}} = \boldsymbol{Q} \times \boldsymbol{T}_d \times \mathbf{\Gamma}$ or $\tilde{\boldsymbol{Q}} = \boldsymbol{Q} \times \boldsymbol{T}_d \times \partial\mathbf{\Gamma}$ is introduced which includes uncertain system parameters $\boldsymbol{Q}$, time delays $\boldsymbol{T}_d$, and the region/boundary in the complex plane $\mathbf{\Gamma}/\partial\mathbf{\Gamma}$. The value set of the characteristic function can then simply be written as $f(\tilde{\boldsymbol{Q}})$.

Using these definitions, the goal of the branch and bound algorithm is to distinguish regions in $\tilde{\boldsymbol{Q}}$ for which the values set $f(\tilde{\boldsymbol{Q}})$ excludes the point zero from regions for which the value set includes the point zero. Therefore, a desired resolution is specified by $\tilde{\boldsymbol{q}}_{\min}$ for each dimension of $\tilde{\boldsymbol{Q}}$, see Section 3.6.2. Then, the algorithm depicted in Fig. 3.4 and described in the rest of this section is executed. For details on computing the value set, Bernstein coefficients, and interval remainder, see Chapter 2.

The algorithm is initialized by setting the set of regions which should be checked to $\mathcal{Q} = \left\{\tilde{\boldsymbol{Q}}_0\right\}$ where $\tilde{\boldsymbol{Q}}_0$ corresponds to the initial parameter set. Then, the steps of the branch and bound algorithm are as follows. First, a region $\tilde{\boldsymbol{Q}}$ is popped from the set of regions $\mathcal{Q}$, i.e., $\tilde{\boldsymbol{Q}}$ is set to one region in the set $\mathcal{Q}$ and this region is removed from the set $\mathcal{Q}$. Then, the value set $f(\tilde{\boldsymbol{Q}})$ is computed for this region. This results in Bernstein coefficients of the polynomial part of the value set and a complex interval remainder.

Zero exclusion is then evaluated according to the algorithm presented in Section 3.6.1. If

[12] Consider the case where we want to determine stability in a 2D-plane with a resolution of 1/100th of the parameter-range in both dimensions, this would result in 10000 regions.

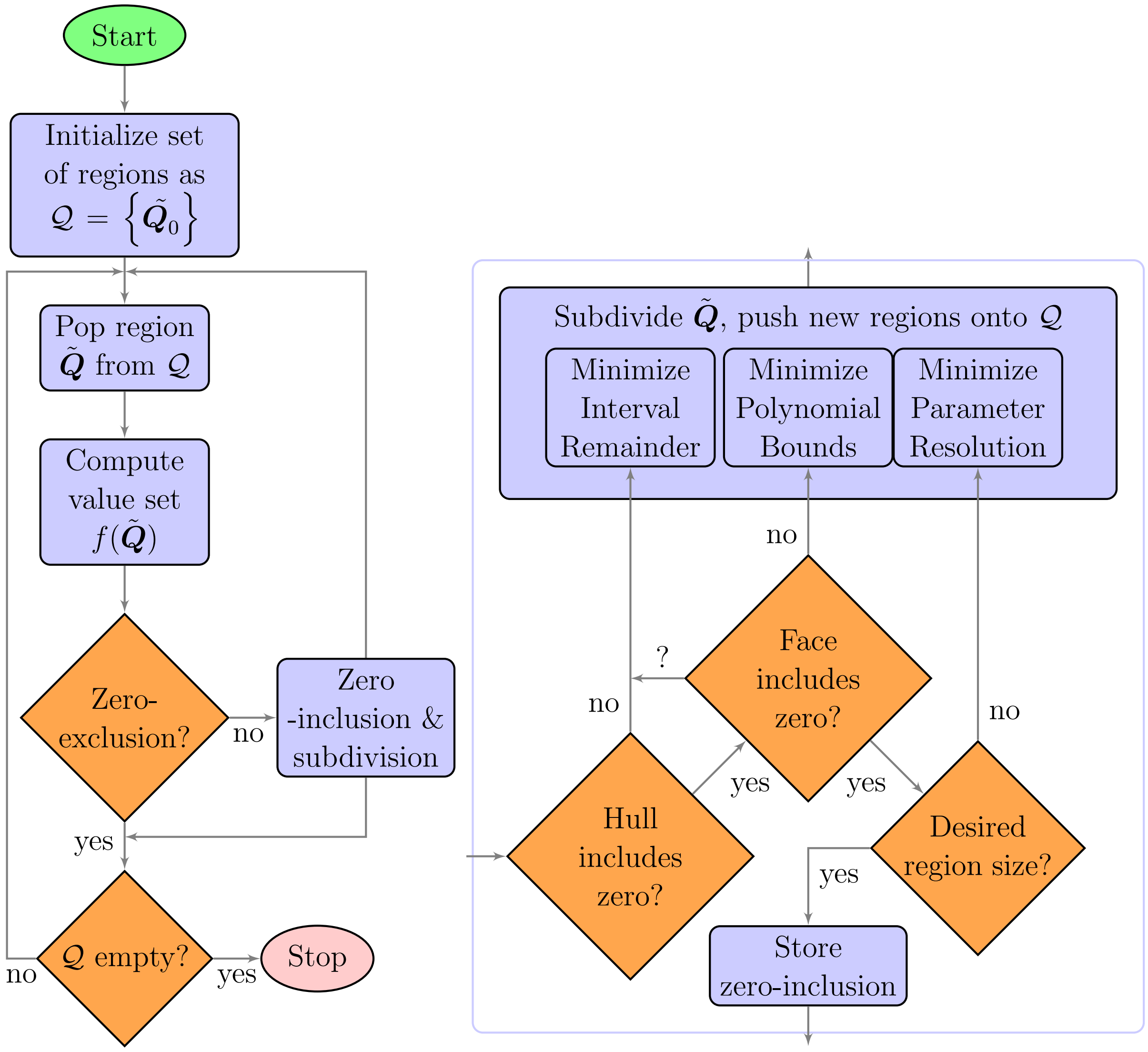

Start
Initialize set of regions as $\mathcal{Q} = \left\{\tilde{\boldsymbol{Q}}_0\right\}$
Pop region $\tilde{\boldsymbol{Q}}$ from $\mathcal{Q}$
Compute value set $f(\tilde{\boldsymbol{Q}})$
Zero-exclusion?
no
Zero -inclusion & subdivision
yes
$\mathcal{Q}$ empty?
no
yes
Stop
Subdivide $\tilde{\boldsymbol{Q}}$, push new regions onto $\mathcal{Q}$
Minimize Interval Remainder
Minimize Polynomial Bounds
Minimize Parameter Resolution
no
Face includes zero?
?
no
Hull includes zero?
yes
yes
no
Desired region size?
yes
Store zero-inclusion

Figure 3.4: Flow-chart of the branch & bound algorithm used for boundary mapping and region mapping. On the left, the complete algorithm is shown, while the details of the step *Zero-inclusion & subdivision* are shown on the right.

zero is excluded the algorithm checks whether there are any regions left in $\mathcal{Q}$. If there are, the algorithm restarts for the next region, i.e., a region is popped from $\mathcal{Q}$, the value set is evaluated, etc. If zero is not excluded, inclusion is evaluated using the algorithm presented in Section 3.6.1. This algorithm does not simply result in inclusion or no inclusion but provides more fine-grained results.

In a first step complete inclusion of the negative interval remainder in the convex hull of all Bernstein coefficients is checked as this is a necessary condition for the following tests to succeed. If this check is not successful the region $\tilde{\boldsymbol{Q}}$ is subdivided so as to minimize the size of the interval remainder (see Section 3.6.3) and the resulting regions are pushed onto $\mathcal{Q}$ (i.e., appended to the set $\mathcal{Q}$). Then, the algorithm restarts for the next region, i.e., a region is popped from $\mathcal{Q}$, the value set is evaluated, etc.

Otherwise, inclusion of the negative interval remainder in one of the sets of Bernstein coefficients corresponding to a face of the parameter set is checked, see Section 3.6.1. If it is completely excluded from all of these faces, the region is subdivided so as to minimize the bounds of the polynomial by using the Bernstein derivative, see Section 3.6.3. The resulting regions are pushed onto $\mathcal{Q}$ and the algorithm restarts for the next region, i.e., a region is popped from $\mathcal{Q}$, the value set is evaluated, etc. If it is not completely included in any face but partially included in some face, the region is subdivided according to the size of the interval remainder (see Section 3.6.3) and the resulting regions are pushed onto $\mathcal{Q}$. Then, the algorithm restarts for the next region, i.e., a region is popped from $\mathcal{Q}$, the value set is evaluated, etc.

If it is completely included in one of the faces a final check is performed. The size of the region is compared to the desired resolution $\tilde{\boldsymbol{q}}_{\min}$, see Section 3.6.3. If it is larger, then the region is subdivided according to the relative size of the region (see Section 3.6.3) and the resulting regions are pushed onto $\mathcal{Q}$. Then, the algorithm restarts for the next region, i.e., a region is popped from $\mathcal{Q}$, the value set is evaluated, etc. If it is smaller, then the inclusion check was successful. The region is marked as including zero. More precisely, the region is enlarged to the size $\tilde{\boldsymbol{q}}_{\min}$ and the complete *cell* is marked as including zero. In case the algorithm is used to map the boundary $\partial\boldsymbol{\Gamma}$ this means the stability boundary passes through this region. In case we are performing the stability check for disjoint regions this means that there is a zero in the region $\boldsymbol{\Gamma}$ for some point $\boldsymbol{q}$, $\boldsymbol{t}_d$, and the disjoint region is marked as unstable.

In the following section the algorithms used to determine zero exclusion and inclusion are introduced. Then, the desired resolution $\tilde{\boldsymbol{q}}_{\min}$ is explained and some exemplary uses are shown. Finally, the different subdivision algorithms used within this branch and bound algorithm are described.

3.6.1 Evaluation of Zero Exclusion and Inclusion

In this section efficient methods to check for *zero inclusion* (Lemma 3.1) and *zero exclusion* (Lemma 3.2) are introduced. The methods are based on the value set which is evaluated as introduced in Section 2.4.5, i.e., the characteristic function is evaluated using Taylor Models with interval remainder, the polynomial part is transformed into Bernstein form, and the convex hull of all Bernstein coefficients as well as the convex hull of different edges of the parameter-set are used to determine an inner and outer approximation of the value set. Then, an efficient method to check for inclusion/exclusion of zero in the Taylor Model $T(\tilde{\boldsymbol{Q}})$ is to check inclusion/exclusion of the negative interval remainder $-R(\tilde{\boldsymbol{Q}})$ in the value set of the polynomial part $\tilde{p}(\boldsymbol{U}, \tilde{\boldsymbol{Q}})$ as we will show in the following subsections.

The conditions for zero exclusion as well as zero inclusion introduced in the following are sufficient but not necessary conditions. This is due to the fact that we use an outer approximation of the value set to check for zero exclusion and an inner approximation of the value set to check for zero inclusion. Thus, if zero is *excluded* from the outer approximation it must be *excluded* from the exact value set. However, if zero is *not excluded* from the outer approximation it must not necessarily be *not excluded* from the exact value set. Likewise, if zero is *included* in the inner approximation it must be *included* in the exact value set. However, if zero is *not included* in the inner approximation it must not necessarily be *not included* in the exact value set. Hence, both conditions are sufficient but not necessary.

Sufficient Condition for Zero Exclusion

Exclusion of zero from the value set, i.e.,

$$\mathbf{0} \neq f(\tilde{\boldsymbol{q}}) \quad \forall\, \tilde{\boldsymbol{q}} \in \tilde{\boldsymbol{Q}} \tag{3.37}$$

is implied by exclusion of zero from the outer approximation of the value set, i.e.,

$$\Leftarrow \mathbf{0} \notin \tilde{p}(\boldsymbol{u}, \tilde{\boldsymbol{Q}}) + R(\tilde{\boldsymbol{Q}}) \quad \forall\, \boldsymbol{u} \in \boldsymbol{U} \tag{3.38}$$

which may be reformulated as

$$\mathbf{0} \cap \tilde{p}(\boldsymbol{u}, \tilde{\boldsymbol{Q}}) + R(\tilde{\boldsymbol{Q}}) = \{\} \quad \forall\, \boldsymbol{u} \in \boldsymbol{U} \tag{3.39}$$

$$\Leftrightarrow -R(\tilde{\boldsymbol{Q}}) \cap \tilde{p}(\boldsymbol{u}, \tilde{\boldsymbol{Q}}) = \{\} \quad \forall\, \boldsymbol{u} \in \boldsymbol{U} \tag{3.40}$$

$$\Leftrightarrow -R(\tilde{\boldsymbol{Q}}) \cap \tilde{p}(\boldsymbol{U}, \tilde{\boldsymbol{Q}}) = \{\}. \tag{3.41}$$

Thus, a sufficient (but not necessary) condition for zero exclusion from $f(\tilde{\boldsymbol{q}})$ is the exclusion of the negative interval remainder $-R(\tilde{\boldsymbol{Q}})$ from $\tilde{p}(\boldsymbol{U}, \tilde{\boldsymbol{Q}})$. Likewise, a sufficient (but not necessary) condition for the exclusion of the negative interval remainder $-R(\tilde{\boldsymbol{Q}})$ from $\tilde{p}(\boldsymbol{U}, \tilde{\boldsymbol{Q}})$ is the exclusion of $-R(\tilde{\boldsymbol{Q}})$ from the outer approximation of $\tilde{p}(\boldsymbol{U}, \tilde{\boldsymbol{Q}})$ which is given by the the convex hull $\boldsymbol{H}(\tilde{\boldsymbol{Q}})$ of the Bernstein coefficients (see Section 2.4.5). Thus, we may write

$$-R(\tilde{\boldsymbol{Q}}) \cap \boldsymbol{H}(\tilde{\boldsymbol{Q}}) = \{\} \tag{3.42}$$

$$\Rightarrow -R(\tilde{\boldsymbol{Q}}) \cap \tilde{p}(\boldsymbol{U}, \tilde{\boldsymbol{Q}}) = \{\} \tag{3.43}$$

$$\Rightarrow \mathbf{0} \neq f(\tilde{\boldsymbol{q}}) \quad \forall\, \tilde{\boldsymbol{q}} \in \tilde{\boldsymbol{Q}} \tag{3.44}$$

i.e., as the complete negative interval remainder $-R(\tilde{\boldsymbol{Q}})$ is excluded from the outer approximation of the value set of the polynomial, zero is excluded from the value set and (3.37) holds.

Sufficient Condition for Zero Inclusion

Analogously, inclusion of zero in the value set, i.e.,

$$\exists\, \tilde{\boldsymbol{q}} \in \tilde{\boldsymbol{Q}}: \quad f(\tilde{\boldsymbol{q}}) = \mathbf{0} \tag{3.45}$$

is implied by inclusion of zero in the inner approximation of the value set.

As explained in Section 2.4.5, the value sets of the faces of the parameter set are used as inner approximations of the value set. More specifically, outer approximations of the value sets of the edges of one face of the parameter set are determined. If none of these include zero but zero is surrounded by these outer approximations, then zero is included in the value set.

An efficient solution of this problem is as follows. First, a necessary condition for this sufficient condition to be true is checked. For zero to be included in any inner approximation of the value set, it must be included in the convex hull of all Bernstein coefficients, see [7]. This convex hull is given by

$$\boldsymbol{H}(\tilde{\boldsymbol{Q}}) = \mathrm{hull}_{I \leq N}(b_I(\tilde{\boldsymbol{Q}})). \tag{3.46}$$

Inclusion of zero in $\boldsymbol{H}(\tilde{\boldsymbol{Q}}) + R(\tilde{\boldsymbol{Q}})$, i.e.,

$$\mathbf{0} \in \boldsymbol{H}(\tilde{\boldsymbol{Q}}) + \boldsymbol{r} \quad \forall \boldsymbol{r} \in R(\tilde{\boldsymbol{Q}}) \tag{3.47}$$

may be checked by

$$-\boldsymbol{r} \in \boldsymbol{H}(\tilde{\boldsymbol{Q}}) \quad \forall \boldsymbol{r} \in R(\tilde{\boldsymbol{Q}}) \tag{3.48}$$

$$\Leftrightarrow -R(\tilde{\boldsymbol{Q}}) \cap \boldsymbol{H}(\tilde{\boldsymbol{Q}}) = -R(\tilde{\boldsymbol{Q}}) \tag{3.49}$$

If zero is not included in the convex hull, the following, computationally more expensive tests may be skipped.

Once this necessary condition[13] is evaluated as true the sufficient condition for zero inclusion is evaluated. Instead of enlarging the approximations of the convex hulls by the interval remainder as explained in Section 2.4.5 an alternative approach similar to the case of *zero exclusion* in the previous section may be used.

For each face of the parameter set the following check is carried out. First, evaluate whether zero is included in the quad which connects the value set of the four vertices of the face which is currently being evaluated. This may be tested, by checking whether the negative interval remainder is completely included in this quad. Note that for each of the vertices of the face, the value of the Bernstein coefficient is exact, i.e., the value of Bernstein coefficient is exactly the same as the value of the polynomial part of the Taylor Model evaluated for this parameter value (see Section 2.3.5).

Then, for each edge, exclusion of zero from the enlarged convex hull $\boldsymbol{H}_i(\tilde{\boldsymbol{Q}}) + R(\tilde{\boldsymbol{Q}})$ of the Bernstein coefficients, i.e.,

$$\mathbf{0} \notin \boldsymbol{H}_i(\tilde{\boldsymbol{Q}}) + R(\tilde{\boldsymbol{Q}}) \tag{3.50}$$

may be checked by evaluating

$$\mathbf{0} \cap \boldsymbol{H}_i(\tilde{\boldsymbol{Q}}) + R(\tilde{\boldsymbol{Q}}) = \{\} \tag{3.51}$$

$$\Leftrightarrow -R(\tilde{\boldsymbol{Q}}) \cap \boldsymbol{H}_i(\tilde{\boldsymbol{Q}}) = \{\}. \tag{3.52}$$

If this exclusion check succeeds for all edges of one face, then zero is not included in any of the edges. As zero is included in the quad connecting the corners of this face of the parameter set and not included in any of the edges of this face of the parameter set, zero must be included in the value set of the face of the parameter set and thereby zero must be included in the exact value set and (3.45) holds.

Otherwise, zero inclusion/exclusion is not known. Then, we want to subdivide the parameter set in such a way that a concrete result is obtained. Two different strategies are chosen depending on the situation: If the interval remainder is completely included in the outer approximation and not included at all in any inner approximation we subdivide the parameter set so as to minimize the bounds of the polynomial part. This can be done very efficiently using the Bernstein derivative, see next section. The reason for using this approach is that it is extremely fast (computationally) and by making the bounds on the polynomial part more and more accurate we will eventually arrive at one of three situations:

[13] *Necessary* for the sufficient condition to be true, not *necessary* for stability.

- The remainder is not included in the outer approximation (the outer approximation gets smaller)
- The remainder is included in the inner approximation (the inner approximation gets larger)
- The remainder is partly included in the outer approximation or partly included in the inner approximation (see following case)

Otherwise, the interval remainder is not completely included in the convex hull of all Bernstein coefficients or the interval remainder is partly included in the inner approximation, i.e., overlaps with the convex hull of one of the edges of the face of the parameter set. Then, the parameter set is subdivided so as to minimize the interval remainder. This is computationally more expensive than using the Bernstein derivative, however, in this case it is not sufficient to make the approximation of the polynomial part more accurate as this will generally not lead to one of the two concrete results (inclusion or exclusion). Moreover, when subdividing the parameter set so as to minimize the interval remainder the Bernstein coefficients are recomputed as well (see Section 3.6.3), i.e., in general, the polynomial part is approximated more accurately as well.

3.6.2 Desired Resolution

A desired resolution must be set for each parameter within the extended parameter set $\tilde{\boldsymbol{Q}}$. In essence, this specifies the resolution with which stability is examined with respect to this parameter.

The resolution is set as fraction of the width of the original interval for each parameter in the vector $\tilde{\boldsymbol{q}}_{\min} \in \mathbb{R}^{(l+m+2)}$. Within the branch and bound algorithm the desired resolution is used to determine whether a search is terminated if zero is included in the value set. Only if all parameters have a size smaller than their desired resolution the termination condition is reached. Furthermore, if the branch and bound algorithm terminates with the condition that zero is included, instead of marking the current region as including zero the region is enlarged to the *cell* (of which the size is given by $\tilde{\boldsymbol{q}}_{\min}$) in which it lies.

Hence, the desired resolution $\tilde{\boldsymbol{q}}_{\min}$ determines the resolution of the stability check in each direction and is used to specify dimensions in which the stability regions should be gridded, e.g., for plotting. All dimensions of $\tilde{\boldsymbol{q}}_{\min}$ over which a plot is to be performed are set to a value larger than zero and smaller than one, where the value determines the resolution in this dimension. All dimensions of $\tilde{\boldsymbol{q}}_{\min}$ over which no plot is performed are set to one[14].

The following two examples show that the values of the desired resolution strongly influence the *type* of result of the overall algorithm. It is, e.g., possible to use the method for robust controller synthesis as well as robust stability analysis.

[14] In our examples, the values of $\tilde{\boldsymbol{q}}_{\min}$ corresponding to the $\boldsymbol{\Gamma}$-regions are generally set to one as we do not want to examine *for which value* of $\boldsymbol{\Gamma}$ there is a zero in the value set but instead want to detect if there is a zero in the value set *for any value* in $\boldsymbol{\Gamma}$.

Example 1 Consider the case of a system with 5 parameters $\boldsymbol{Q}$, no time delay $\boldsymbol{T}_d$, and the region $\boldsymbol{\Gamma}_+$. The desired resolution for the $\boldsymbol{\Gamma}$-region is set to one. If the desired resolution is set to 0.1 for two parameters q_1 and q_2 in $\boldsymbol{Q}$ and is set to one for the other parameters in $\boldsymbol{Q}$, then the stability check determines regions in the q_1/q_2-plane which are stable for all values of the other parameters in $\boldsymbol{Q}$ and all frequencies Ω. The stable regions are determined with a resolution of one tenth of the range of the parameters q_1/q_2, i.e., the stability boundary (which we must consider as unstable) is overapproximated at most by one tenth of the range of the parameters q_1/q_2. If q_1 and q_2 are controller parameters and the other parameters are plant parameters, this case represents an example for robust controller synthesis.

Example 2 Consider the case where all values in $\tilde{\boldsymbol{q}}_{\min}$ are set to one. Then, the stability check determines if there is any parameter within $\tilde{\boldsymbol{Q}}$ for which the system is unstable. This can be used for a robust stability check once a controller parametrization has been chosen. This is similar to, e.g., a stability check according to Kharitonov, the Edge Theorem, etc. In contrast to these methods our stability check is however applicable to a much larger class of systems and, as stable regions are not mapped to the parameter space in this case, the results are non-conservative.

3.6.3 Subdivision

The branch and bound algorithm is used to subdivide the parameter set into two regions of equal size along a direction which coincides with one parameter in the extended parameter set[15]. Three different algorithms, which are used to select the direction in which the parameter set is subdivided, are introduced here. These algorithms are used to subdivide the parameter set based on different criteria. In the case where we can neither determine zero inclusion nor zero exclusion we minimize the polynomial bound or minimize the interval remainder depending on the result of the zero inclusion test. In the case where zero is included in the value set but the desired resolution is not yet reached we subdivide the parameter set in such a way that the desired resolution is reached as fast as possible.

Minimize Polynomial Bound

When the interval remainder is completely included in the convex hull of the Bernstein coefficients and completely excluded from all Bernstein patches when performing the edge-check, the bound of the polynomial, i.e., the size of the value set of the polynomial part, is minimized. In this case, a subdivision according to the maximum first partial derivative is performed which can be estimated efficiently using the Bernstein algorithm for derivative estimation in Section 2.3.6.

As this partial derivative is complex the norm is used. The subdivision direction is selected so as to subdivide the parameter set in the direction r_1 of the maximum norm of

[15] Regions which include/exclude zero might be found faster by choosing a different subdivision-point than the center. This possibility is not further explored here but could be an interesting topic for future research.

the maximum partial derivative estimation for each direction which is given by

$$r_1 = \underset{j=1,\ldots,l}{\arg\max}\, \tilde{I}_j^{(1)} \tag{3.53}$$

where $\tilde{I}_j^{(1)}$ is given in (2.33). When this algorithm is used to subdivide the parameter space no reevaluation of the Taylor Model is performed. Instead, the new Bernstein coefficients are determined using a Bernstein subdivision algorithm (see Section 2.3.6). Highly efficient implementations of the Bernstein subdivision algorithm are possible leading to very fast execution times compared to reevaluating the Taylor Model.

By using the Bernstein subdivision algorithm no reevaluation of the interval remainder is performed. Thus, the interval remainder of the two new regions is set to the interval remainder of the old region. This results in an overapproximation of the remainder compared to reevaluating the Taylor Model. This overapproximation may lead to a partial inclusion/exclusion of the interval remainder. As this triggers a reevaluation of the Taylor Model no conservatism is introduced by using the old interval remainder where possible.

Minimize Interval Remainder

If the interval remainder is not completely included in the convex hull of the Bernstein coefficients or the interval remainder is partly included in the inner approximation, i.e., overlaps with the convex hull of one of the edges of the face of the parameter set the following steps are taken.

If the Taylor Model has not been evaluated for this region, i.e., the interval remainder was actually determined for a larger region, then the Taylor Model is evaluated, the value set is computed from this Taylor Model and the zero exclusion/inclusion check is restarted for this new value set.

Otherwise, a subdivision according to the size of the interval remainder is performed, i.e., the goal of the subdivision is to reduce the size of the interval remainder so that either *zero exclusion* or *zero inclusion* is achieved.

Therefore, the region is subdivided once in each direction $j = 1, \ldots, l$ and the Taylor Models are evaluated which results in two polynomials $P_{j,1}$, $P_{j,2}$ and corresponding complex interval remainders $\boldsymbol{R_{j,1}}$, $\boldsymbol{R_{j,2}}$ for each direction j. The direction is then used, for which the average norm of the two complex interval remainders (of the two regions) is minimal, i.e.

$$r_2 = \underset{j=1,\ldots,l}{\arg\min} \sqrt{\operatorname{Re}(\boldsymbol{R_{j,1}})^2 + \operatorname{Im}(\boldsymbol{R_{j,1}})^2} + \sqrt{\operatorname{Re}(\boldsymbol{R_{j,2}})^2 + \operatorname{Im}(\boldsymbol{R_{j,2}})^2} \tag{3.54}$$

As the Taylor Models must be evaluated here anyway, Bernstein coefficients are calculated from the polynomial parts of the Taylor Models and the new interval remainders are used.

Minimize Parameter Resolution

If the value set *includes zero* but the desired resolution is not yet reached, the goal is to subdivide in the direction which will result in reaching the desired resolution as fast as possible. Whether the desired resolution is reached may be determined by

$$w_{\max} = \max_{j=1,\ldots,l} \frac{q_{w,j}}{q_{w0,j}\tilde{q}_{\min,j}} \tag{3.55}$$

where the width of the parameter set and original parameter set is given by

$$\boldsymbol{q_w} = \overline{\tilde{\boldsymbol{Q}}} - \underline{\tilde{\boldsymbol{Q}}} \tag{3.56}$$

$$\boldsymbol{q_{w0}} = \overline{\tilde{\boldsymbol{Q}}_{\boldsymbol{0}}} - \underline{\tilde{\boldsymbol{Q}}_{\boldsymbol{0}}}. \tag{3.57}$$

If $w_{\max} \leq 1$ the required resolution is reached. Otherwise, the direction r_3 is chosen as subdivision direction for which the interval width normalized by the width of the original parameter set $\tilde{\boldsymbol{Q}}_0$ and corresponding desired resolution is largest. This direction may be determined as

$$r_3 = \underset{j=1,\ldots,l}{\arg\max} \frac{q_{w,j}}{q_{w0,j}\tilde{q}_{\min,j}}. \tag{3.58}$$

3.7 Example for Robust Stability Analysis

In this section an example taken from [1] is used to illustrate the boundary mapping algorithm introduced in this chapter. The example consists of a crane with uncertain load and rope length. In [1] the linearized system model is presented and a feedback controller is developed. Controller parameters to robustly stabilize the system are then determined using the parameter-space approach. In the following, first, the system model is shortly summarized. Then, our $\boldsymbol{\Gamma}$-region mapping algorithm is used to determine parameters for robustly stabilizing the system. In a next step, parameters are determined which assure a given system damping. This illustrates mapping of different $\boldsymbol{\Gamma}$-regions. Finally, a time delay is introduced in the control loop to illustrate the applicability of the proposed method to time-delay systems.

3.7.1 System Model

The examined system is illustrated in Fig. 3.5. The linearized system dynamics of this system are presented in [1] as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u \tag{3.59}$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & m_L g/m_C & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -(m_L + m_C)g/(m_C l) & 0 \end{bmatrix} \tag{3.60}$$

$$\boldsymbol{b}^T = \begin{bmatrix} 0 & 1/m_C & 0 & -1/m_C l \end{bmatrix} \tag{3.61}$$

where g is the gravitational constant, l is the length of the rope, m_L is the mass of the load, and m_C is the mass of the crab. A state feedback controller

$$u = k_1(v - x_1) - k_2 x_2 - k_3 x_3 - k_4 x_4 \tag{3.62}$$

with gains k_i and new input v is considered in [1]. Together with the reasonable assumptions $l > 0$ and $m_C > 0$ this results in the following characteristic function:

$$f(\boldsymbol{s}) = a_0 + a_1\boldsymbol{s} + a_2\boldsymbol{s}^2 + a_3\boldsymbol{s}^3 + a_4\boldsymbol{s}^4 \tag{3.63}$$

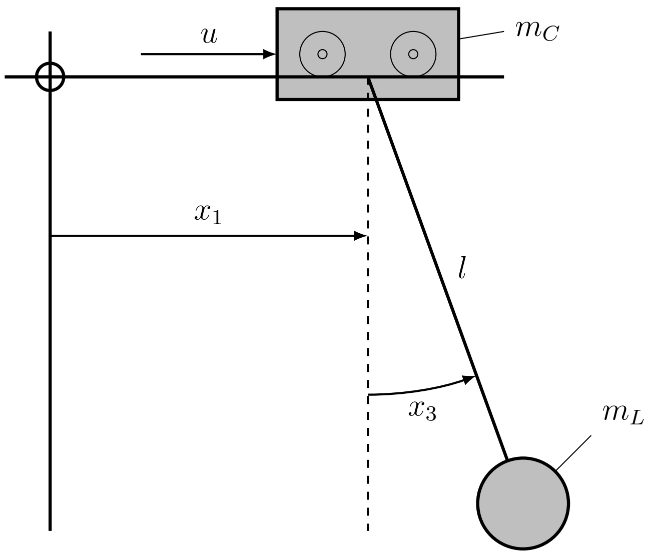


Figure 3.5: Crane

with

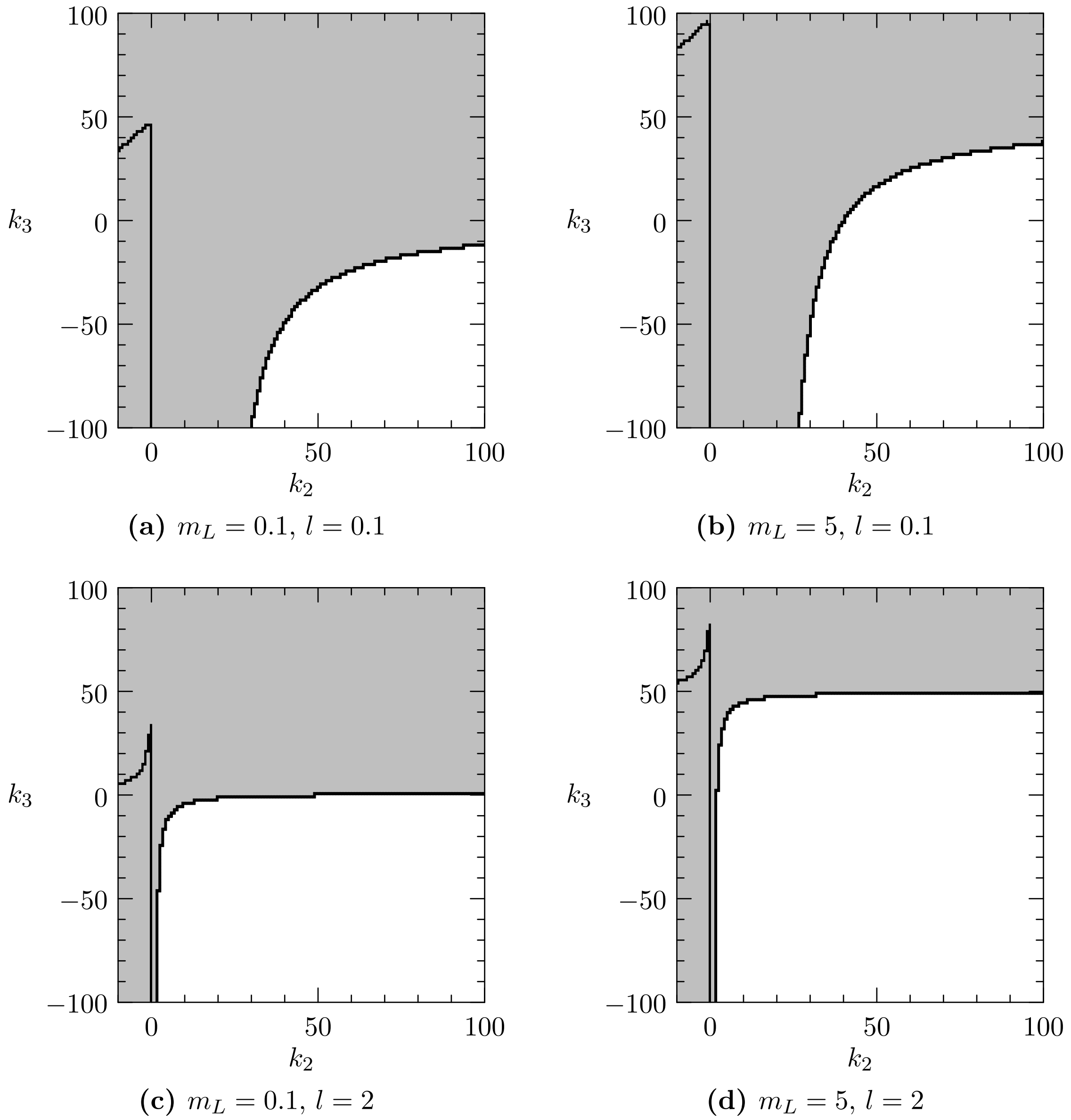
$$
\begin{aligned}
a_0 &= k_1 g \\
a_1 &= k_2 g \\
a_2 &= (m_L + m_C)g + k_1 l - k_3 \\
a_3 &= k_2 l - k_4 \\
a_4 &= l m_C
\end{aligned}
$$

The characteristic function (3.63) is used as the basis for the analysis in the following sections. Thereby, different parameters are set to fixed values while others are assumed to be uncertain and summarized in $\boldsymbol{q} \in \boldsymbol{Q}$.

3.7.2 Hurwitz Stability

As a first example of applying the boundary mapping approach introduced in this chapter, Hurwitz stability of the crane with state feedback is examined. As in [1], let $m_C = 5$, $g = 10$, $k_1 = 1$, $k_4 = 2$ and set the uncertain parameter vector as $\boldsymbol{q}^T = [k_2,\ k_3]$. Then, the Γ_+-region is mapped to the $k_2 \times k_3$-plane using the region mapping algorithm from Section 3.3. Thereby, stable regions in the $k_2 \times k_3$-plane are determined for different values of m_L and l as depicted in Fig. 3.6. The considered parameter range in Fig. 3.6 is $k_2 = [-10,\ 100]$ and $k_3 = [-100,\ 100]$ and the resolution of these two dimensions is set to $1/128$ of the complete parameter range by choosing $\tilde{\boldsymbol{q}}_{\min,1} = \tilde{\boldsymbol{q}}_{\min,2} = 1/128$. For the case $m_L = 5$, $l = 2$ (see Fig. 3.6d) the results are identical to those in [1, Fig. 2.1], for the other parameter values no results are given in [1]. The computation time for each of the cases depicted in Fig. 3.6 is below one second[16].

[16] Computations were carried out on an Intel Core2 Quad Q9550 running at 2.83 GHz with 8 GB of RAM. As operating system Ubuntu 14.04 with Linux kernel 3.13.0 was used. To compile the software a GCC 4.8.4 was used. Our C++ implementation (available for download at http://www.github.com/schauss/glob_stab) is multi-threaded and the four processor cores are fully utilized.



(a) $m_L = 0.1$, $l = 0.1$

(b) $m_L = 5$, $l = 0.1$

(c) $m_L = 0.1$, $l = 2$

(d) $m_L = 5$, $l = 2$

Figure 3.6: Analysis of Hurwitz stability for crane example. Four different combinations of load mass m_L and rope length l are depicted. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both k_2 and k_3.

The power of the method introduced here in contrast with the *Parameter-Space Approach* from [1] becomes evident when more than two uncertain parameters are considered. This is only possible with the *Parameter-Space Approach* by gridding some of the parameters[17]. With the stability analysis method introduced in this chapter the solution is

[17] This means stability boundaries are plotted for a large number of different parameter values for $q_3 \dots q_l$ in the q_1/q_2 plane. Theoretically, an infinite number of boundaries would have to be examined, i.e., we would have to grid each of the parameters $q_3 \dots q_l$ infinitely fine. In practice, a relatively small grid

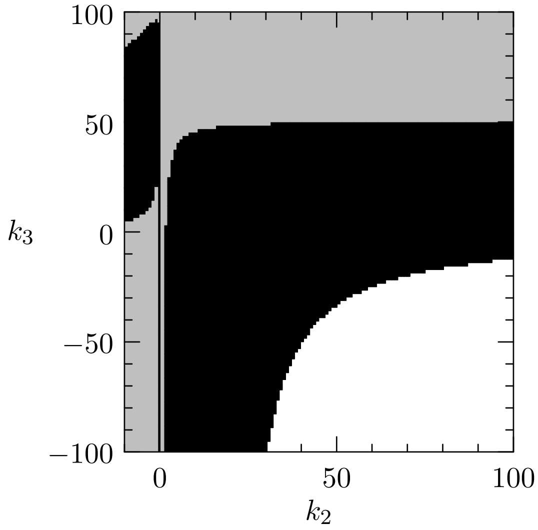


Figure 3.7: Analysis of Hurwitz stability for crane example. The load mass m_L and rope length l are uncertain ($m_L = [0.1,\ 5]$, $l = [0.1,\ 2]$). Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both k_2 and k_3. For this simple example the stable region coincides with the solution at the corner $m_L = 0.1$, $l = 0.1$ of the interval box (see Fig. 3.6a).

straightforward. In Fig. 3.7 the mass of the load m_L and length of the rope l are considered as uncertain, within the intervals $m_L = [0.1,\ 5]$ and $l = [0.1,\ 2]$. Again, stable regions are determined within the $k_2 \times k_3$-plane. The considered parameter range in Fig. 3.7 is $k_2 = [-10,\ 100]$ and $k_3 = [-100,\ 100]$ and the resolution of these dimensions is set to 1/128 of the complete parameter range. The parameters m_L and l are not gridded, i.e., regions marked as stable are stable for *all* parameters within the given intervals. The computation time for the region depicted in Fig. 3.7 is roughly 18 seconds.

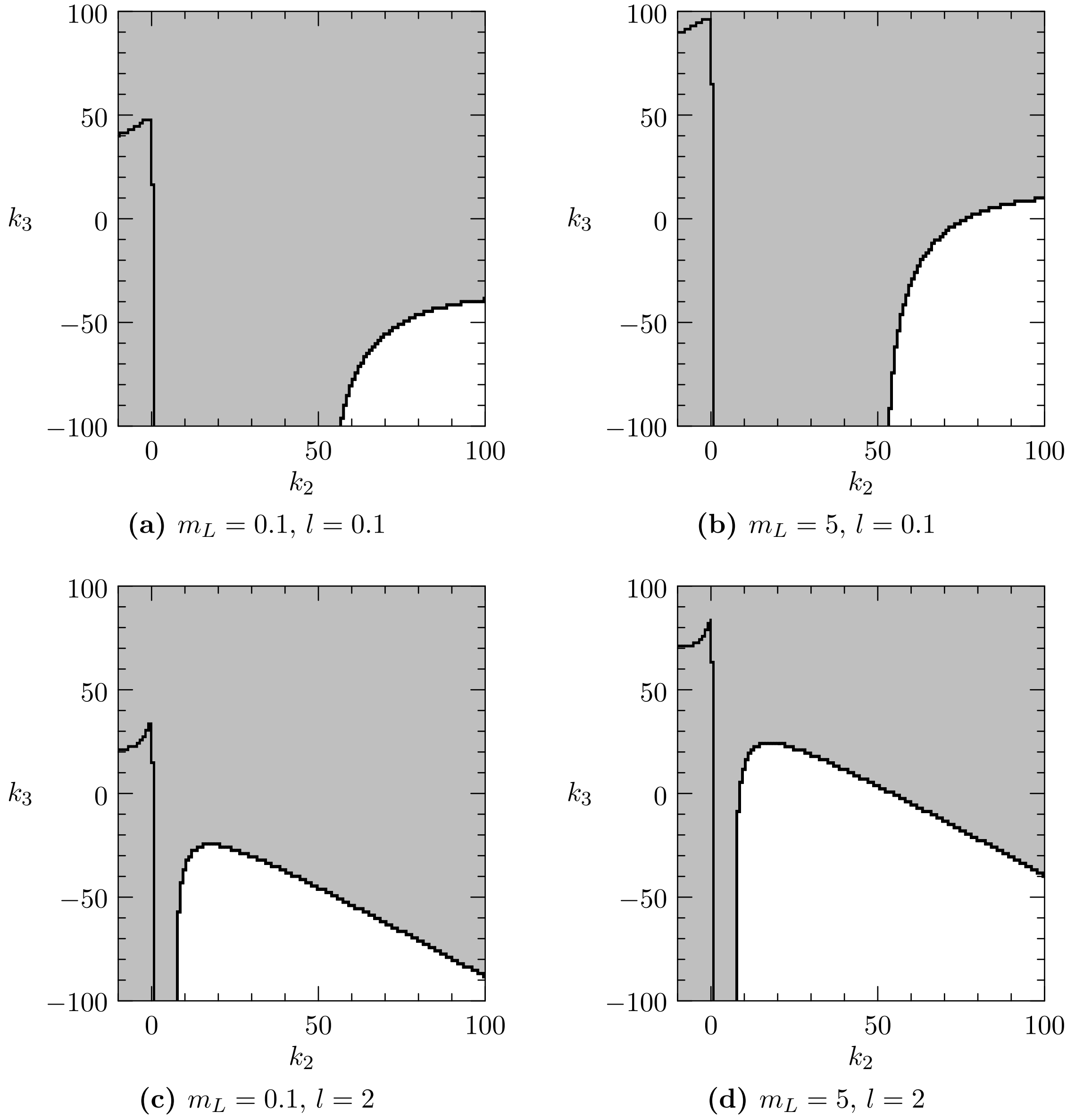
In this case, simply determining the stable regions for the four corners of the interval box and choosing the smallest would have been sufficient as the stable region coincides with the case $m_L = 0.1$, $l = 0.1$ (see Fig. 3.6a). However, checking stability for the corners of the interval-box is not sufficient in general.

3.7.3 Minimum Damping

Next, regions in the parameter space which result in a minimum damping of $\delta = 0.1$ are determined. Therefore, the same computations as in the previous section are carried out, but a damped region $\mathbf{\Gamma}_D$ is mapped to the parameter space instead of the right half-plane. Apart from the damping δ the same parameter values are chosen as in Section 3.7.2.

Regions which achieve this minimum damping and regions which do not are depicted in Fig. 3.8 for the same values of m_L and l as in Fig. 3.6. The regions for which the resulting

must be chosen as, e.g., in [1, Figure 3.21]. This quickly leads to a huge number of boundaries as the total number of boundaries grows exponentially for each additional parameter l.



(a) $m_L = 0.1$, $l = 0.1$

(b) $m_L = 5$, $l = 0.1$

(c) $m_L = 0.1$, $l = 2$

(d) $m_L = 5$, $l = 2$

Figure 3.8: Regions with minimum damping $\delta = 0.1$ for crane example. Four different combinations of load mass m_L and rope length l are depicted. Gray regions contain an eigenvalue in $\mathbf{\Gamma}_D$ for all parameters in this region while black regions contain an Eigenvalue on the boundary $\boldsymbol{\partial}\Gamma_D$ for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in $\mathbf{\Gamma}_D$ and the system therefore achieves the required minimum damping. The resolution $\tilde{q}_{\min}$ is 1/128 in both k_2 and k_3.

system is damped, with $\delta > 0.1$, are considerably smaller than the Hurwitz stable regions in Fig. 3.6. Computation times are not affected strongly by considering a minimum damping for these simple two-dimensional cases, and remain below one second for each case.

In Fig. 3.9 the mass of the load m_L and length of the rope l are again chosen to be uncertain, within the intervals $m_L = [0.1,\ 5]$ and $l = [0.1,\ 2]$. Again, the region for which the resulting system is damped, with $\delta > 0.1$, is considerably smaller than the Hurwitz stable region in Fig. 3.7. The computation time is approximately 22 seconds, which is

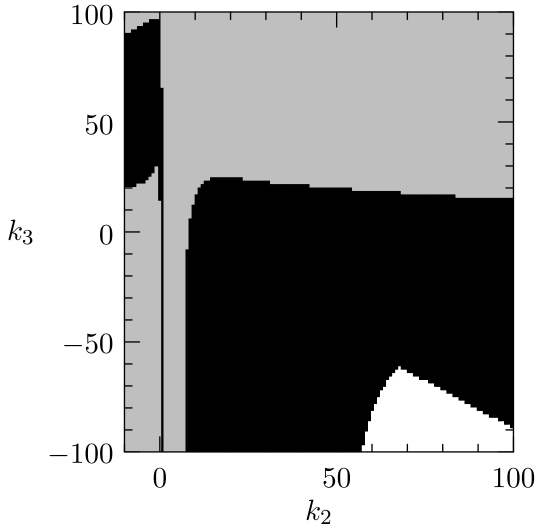


Figure 3.9: Regions with minimum damping $\delta = 0.1$ for crane example. The load mass m_L and rope length l are uncertain ($m_L = [0.1,\ 5]$, $l = [0.1,\ 2]$). Gray regions contain an eigenvalue in $\mathbf{\Gamma}_D$ for all parameters in this region while black regions contain an Eigenvalue on the boundary $\boldsymbol{\partial}\Gamma_D$ for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in $\mathbf{\Gamma}_D$ and the system therefore achieves the required minimum damping. The resolution $\tilde{q}_{\min}$ is 1/128 in both k_2 and k_3.

slightly larger than when computing Hurwitz stable regions.

3.7.4 Time Delay

In this section we illustrate the handling of time delay with the proposed method. We consider a scenario, where the rope angle x_3 and therefore also the rope speed x_4 can only be measured with a delay t_d. This could, e.g., be caused by using a camera system and image processing algorithm to measure this angle.

The modified feedback controller becomes

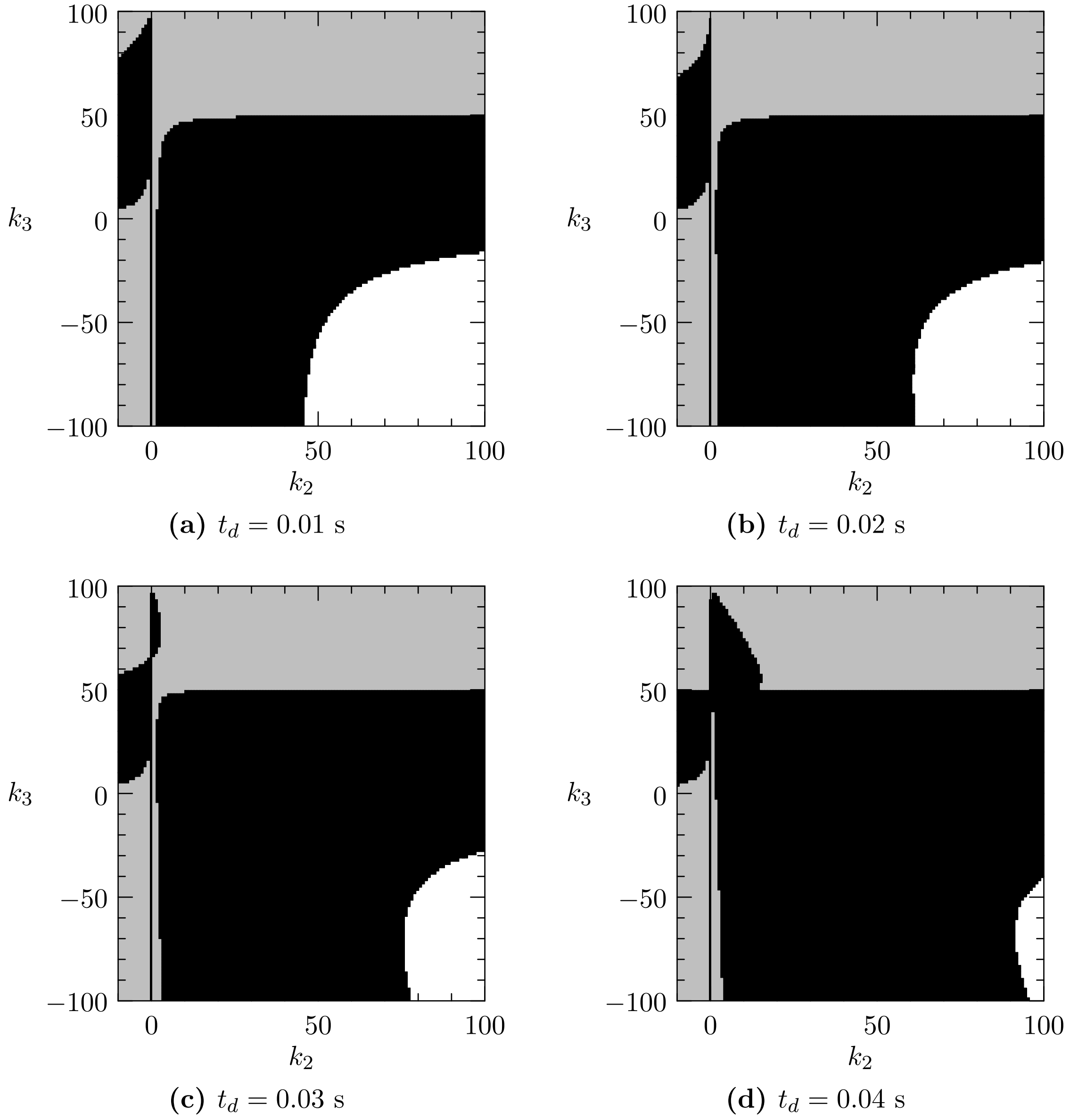
$$u_{td} = k_1(v - x_1) - k_2x_2 - (k_3x_3 + k_4x_4)e^{-t_d\boldsymbol{s}} \tag{3.64}$$

which results in the following characteristic function:

$$f_{td}(\boldsymbol{s}) = a_{0,td} + a_{1,td}\boldsymbol{s} + a_{2,td}\boldsymbol{s}^2 + a_{3,td}\boldsymbol{s}^3 + a_{4,td}\boldsymbol{s}^4 \tag{3.65}$$

with

$$\begin{aligned}
a_{0,td} &= a_0 = k_1g \\
a_{1,td} &= a_1 = k_2g \\
a_{2,td} &= (m_L + m_C)g + k_1l - k_3e^{-t_d\boldsymbol{s}} \\
a_{3,td} &= k_2l - k_4e^{-t_d\boldsymbol{s}} \\
a_{4,td} &= a_4 = lm_C
\end{aligned}$$



(a) $t_d = 0.01$ s

(b) $t_d = 0.02$ s

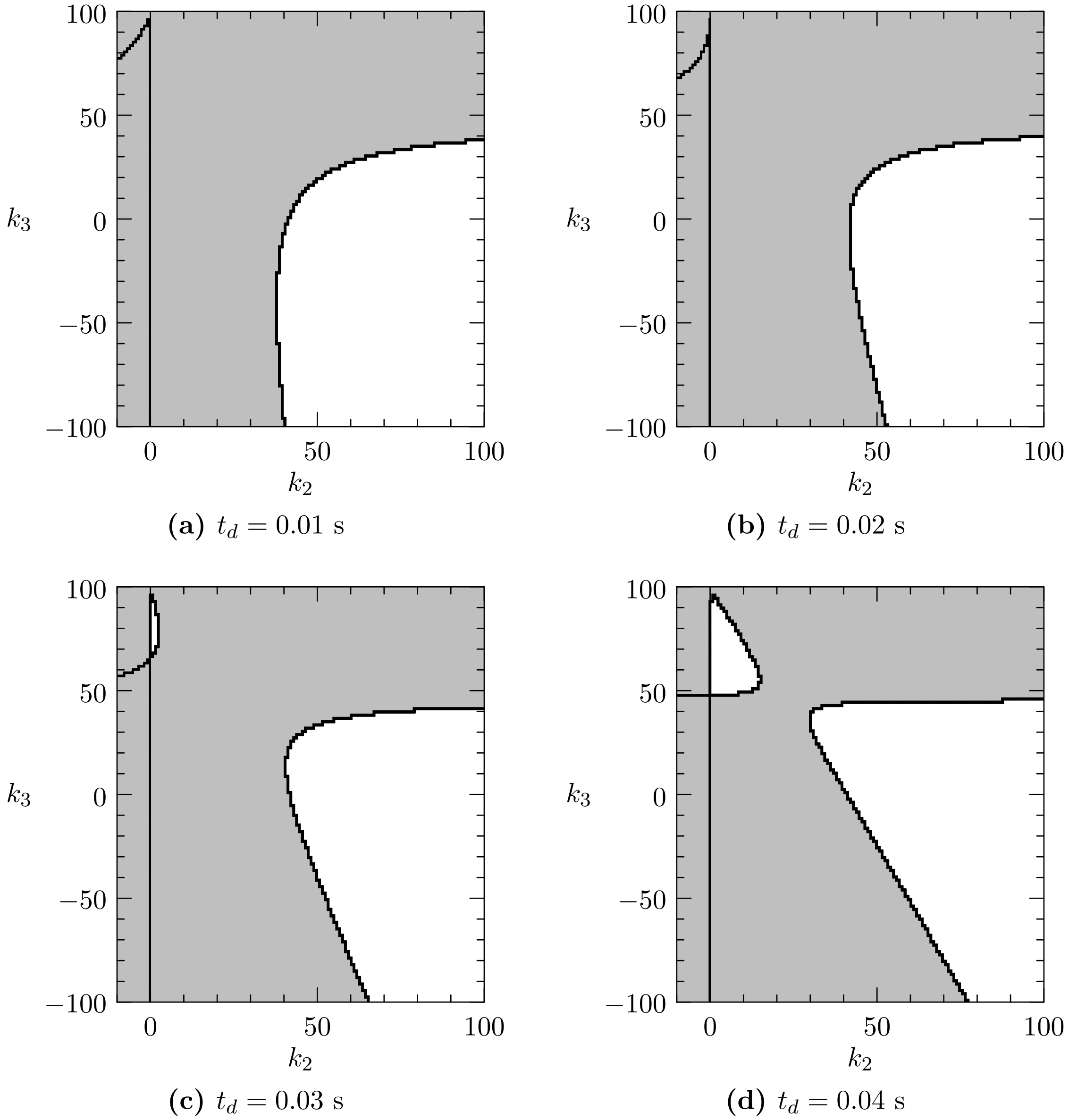
(c) $t_d = 0.03$ s

(d) $t_d = 0.04$ s

Figure 3.10: Analysis of Hurwitz stability for crane example with time delay. The load mass m_L and rope length l are uncertain ($m_L = [0.1,\ 5]$, $l = [0.1,\ 2]$). Stable regions for four different time delays t_d are depicted. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both k_2 and k_3.

For the following analysis, the same parameter values are chosen as in Section 3.7.2 for the delay-free case. Stable and unstable regions in the $k_2 \times k_3$-plane are depicted for different values of t_d in Fig. 3.10. The effect of time delay on stable regions in the parameter space is evident.

Due to the higher complexity of this problem the computation time for the results in Fig. 3.10 rises. For the different cases, computation times of 46, 55, 53, and 75 seconds are



(a) $t_d = 0.01$ s

(b) $t_d = 0.02$ s

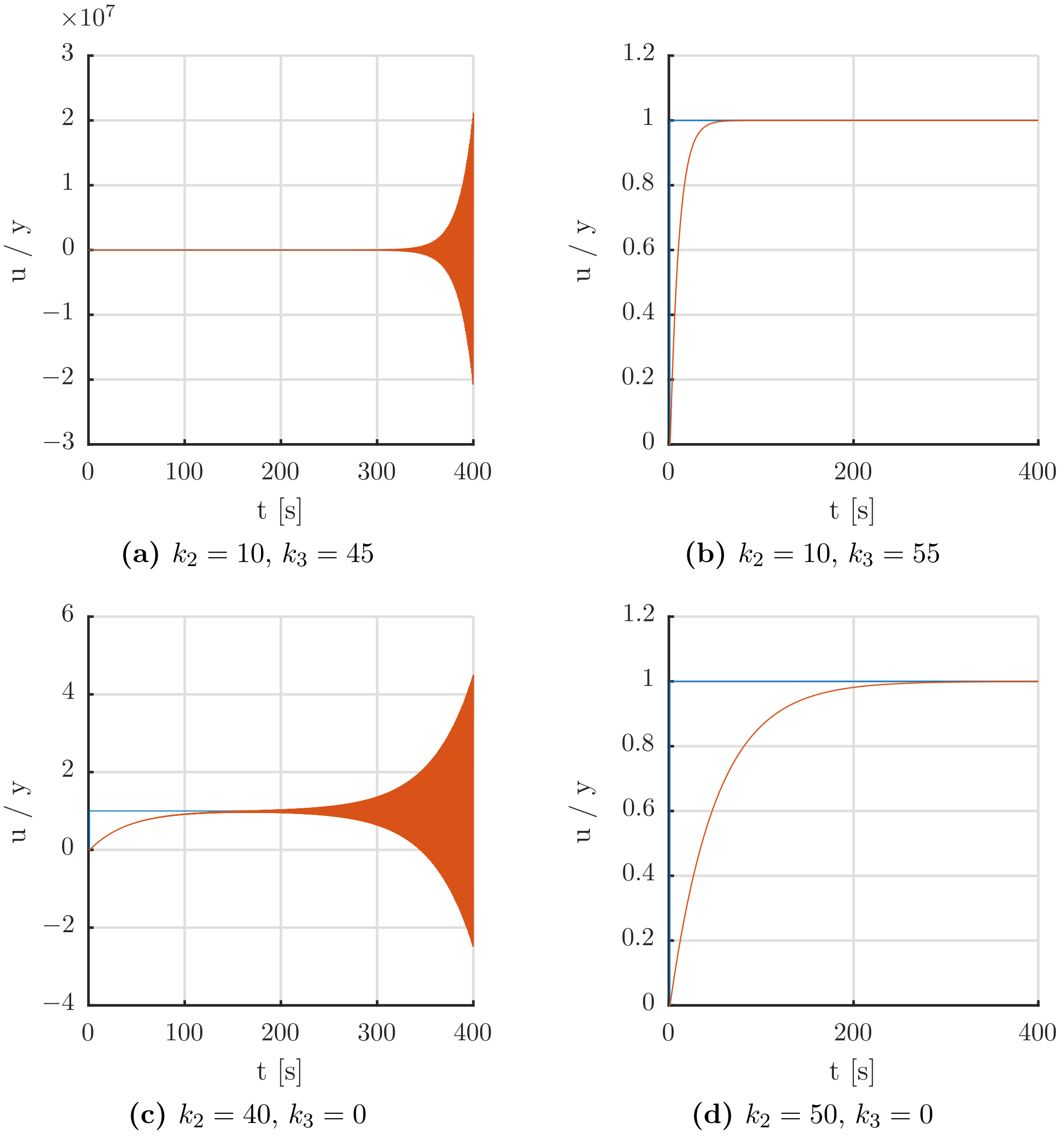
(c) $t_d = 0.03$ s

(d) $t_d = 0.04$ s

Figure 3.11: Analysis of Hurwitz stability for crane example. The load mass m_L and rope length l are set to fixed values ($m_L = 5$, $l = 0.1$). Stable regions for four different time delays t_d are depicted. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both k_2 and k_3.

required compared to 18 seconds for the case without time delay in Fig. 3.7.

Such an analysis could not have been carried out with the parameter-space approach from [1]. On the one hand four uncertain parameters are considered of which only two are plotted which is only possible by gridding two parameters when using the parameter space approach. On the other hand, more importantly, the parameter space approach cannot be used to examine systems with time delay.



(a) $k_2 = 10$, $k_3 = 45$

(b) $k_2 = 10$, $k_3 = 55$

(c) $k_2 = 40$, $k_3 = 0$

(d) $k_2 = 50$, $k_3 = 0$

Figure 3.12: Simulation results using Simulink for crane example with a time delay $t_d = 0.04$ s, load mass $m_L = 5$, and rope length $l = 0.1$. This corresponds to the parametrization in Fig. 3.11d. The reference signal u (blue) changes from 0 to 1 after 1 second while large differences are observable in the system output (red) for the different parametrizations of k_2/k_3. For parameters which were shown to be stable in Fig. 3.11d the simulation results are stable while an oscillation with growing amplitude can be observed in the other cases.

The next example examines the effect of time delay for a fixed mass $m_L = 5$ and rope length $l = 0.1$. Fig. 3.11 shows stable and unstable regions in the $k_2 \times k_3$-plane. This example illustrates that time delay in the control loop can lead to an enlarged stable region in the parameter space in comparison with the delay-free case, e.g., when considering the parameters $k_2 = 10$, $k_3 = 60$. Again, computation time rises compared to the case without time delay. Here, computation times of 17, 18, 29, and 32 seconds are required (compared to less than one second without time delay).

For the case with time delay no comparison results exist as no time delay is considered in [1]. Therefore, a simulation was carried out using Simulink[18]. The results of a step response for four different combinations of k_2 and k_3 are given in Fig. 3.12. In all cases, the load mass is fixed to $m_L = 5$, the rope length is fixed to $l = 0.1$ and a time delay of $t_d = 0.04$ s is considered, which corresponds to the parametrization in Fig. 3.11d. The results of the simulation in Fig. 3.12 are as expected. For parameters where the stability analysis indicates a stable system the simulation results are stable. Likewise, for parameters where the stability analysis indicates an unstable system the simulation results are unstable.

3.8 Examples for Stability Analysis with Incommensurate Delays

The previous section demonstrated how our method can be applied to robust stability analysis of a system without time delay or with one time delay. In this section, the examples from [34] are taken. In that publication, different methods which can cope with incommensurate time delays are compared for nine examples by determining computation time and maximum time delay which the respective method classifies as stable. Several LMI-based methods as well as some methods based on linear-programming and sum-of-squares techniques are compared. In addition, one method [43] is considered, which non-conservatively maps stability boundaries to the delay-space by solving a quadratic eigenvalue problem. However, the problem is solved for a finite number of values from an interval which would actually have to be checked completely, i.e., the results are not rigorous[19]. Moreover, it is not clear how the method could be extended for systems with parametric uncertainties.

The system matrices of the examples from [34] are given in Table 3.1. No parametric uncertainties are present in any of the examples. In our algorithm, we will however treat the time delay(s) as uncertain so as to determine the time delay at which the system becomes unstable. For all examples with one or two time delays the results determined with our method are identical to the *exact* results obtained with the method from [43] within the chosen resolution of our algorithm while for the example with three time delays the accuracy of the results is discussed explicitly. Note that we do *not* compare computation times of our results to the ones published in [34] due to the difficulties discussed in Section 3.9.8 where we also point out the necessity of establishing a collection of suitable benchmark examples to evaluate robust stability analysis methods for time-delay systems.

3.8.1 One time delay

Three different examples are given in [34] for systems with one time delay. The first example is a second-order system with one delay. This example represents a special case where an eigenvalue touches the imaginary axis for a delay of $t_{d,1} = \pi$ s but doesn't cross

[18] We used Simulink 2015b, part of Matlab 2015b by Mathworks, www.mathworks.com. The simulation results were calculated using the variable-step-size continuous-time solver *ode23t* with standard parametrization. This solver is well suited for systems with time delay.

[19] This means that unstable system may falsely be characterized as stable.

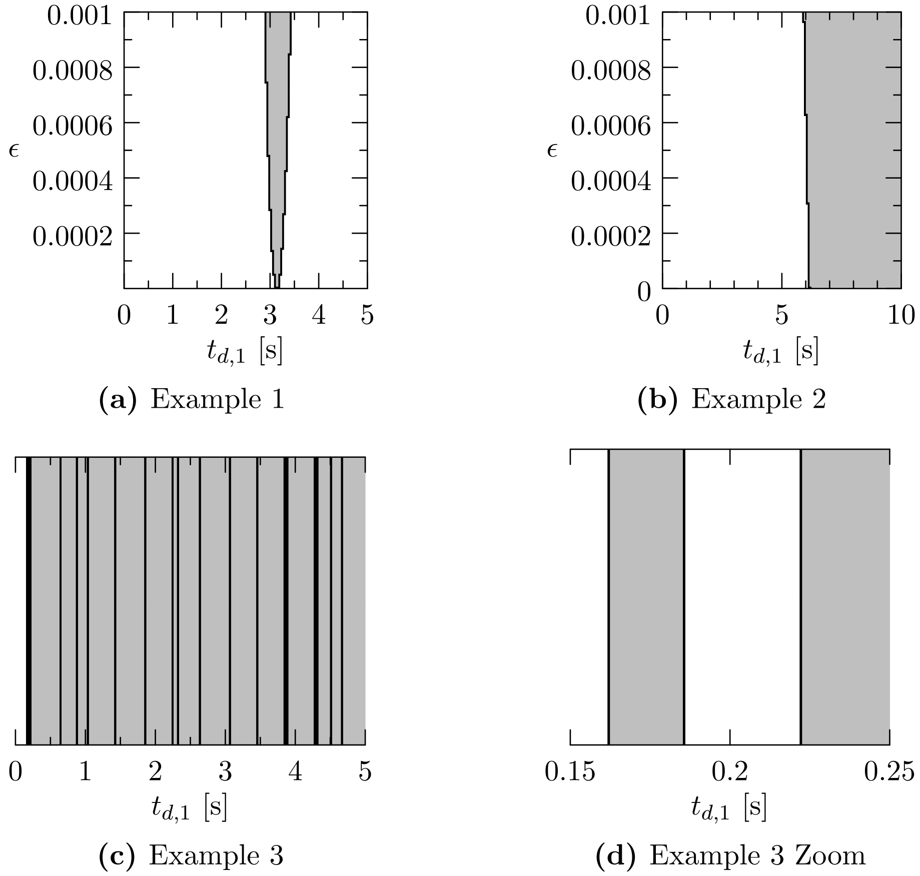
Table 3.1: System matrices for the time-delay examples taken from [34].

Example	A_0	A_1	A_2	A_3
1	$\begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$		
2	$\begin{bmatrix} -2 & 0 \\ 0 & -0.9 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ -1 & -1 \end{bmatrix}$		
3	$\begin{bmatrix} -1 & 13.5 & -1 \\ -3 & -1 & -2 \\ -2 & -1 & -4 \end{bmatrix}$	$\begin{bmatrix} -5.9 & 7.1 & -70.3 \\ 2 & -1 & 5 \\ 2 & 0 & 6 \end{bmatrix}$		
4	0.5	-0.9	-1.5	
5	0	-1	-2	
6	$\begin{bmatrix} 0 & 1 \\ -1\frac{49}{256} & -\frac{7}{8} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ \frac{1}{5} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ -\frac{4}{5} & 0 \end{bmatrix}$	
7	$\begin{bmatrix} -1 & 13.5 & -1 \\ -3 & -1 & -2 \\ -2 & -1 & -4 \end{bmatrix}$	$\begin{bmatrix} -5.9 & 0 & 0 \\ 2 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 7.1 & -70.3 \\ 0 & -1 & 5 \\ 0 & 0 & 6 \end{bmatrix}$	
8	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2 & -3 & -5 & -2 \end{bmatrix}$	$\begin{bmatrix} -\frac{9}{200} & \frac{1.5}{200} & \frac{1}{4} & 0 \\ \frac{1}{200} & \frac{1}{200} & \frac{1}{20} & 0 \\ 0 & 0 & 0 & \frac{1}{2000} \\ -2 & -\frac{1}{2} & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{3}{80} & 0 & \frac{3}{40} & \frac{1}{8} \\ 0 & \frac{1}{20} & \frac{1}{20} & 0 \\ \frac{1}{20} & \frac{1}{20} & 0 & 0 \\ 0 & -2.5 & 0 & -1 \end{bmatrix}$	
9	-2	-4	-3	-1

the imaginary axis to the right half-plane. In this case, our stability analysis method does not converge, see Section 3.9.3 for a discussion. It is however possible to shift the stability boundary to the left half-plane by a small value ϵ using the transformation from Section 3.4.4. Then, the results depicted in Fig. 3.13a are achieved for different values of ϵ. The stability boundary converges to $t_{d,1} = \pi$ s for $\epsilon \rightarrow 0$. The overall calculation time is 21 s.

The second example again considers a second-order system. This example does not suffer from the same difficulty as example one, i.e., an eigenvalue passes the imaginary axis when the system becomes unstable, and shifting the stability boundary into the left half-plane is not necessary. Nevertheless, results are computed depending on the two parameters $t_{d,1}$ and ϵ to demonstrate the effect of shifting the stability boundary into the left half-plane. The results shown in Fig. 3.13b are in line with the results achieved in [34] for $\epsilon = 0$. The overall calculation time is 615 s.

In contrast to the first two examples the third example considers a third-order system. This example is used to demonstrate the ability of our approach to calculate results of near arbitrary precision. A calculation with a range of $t_{d,1} = [0,\ 5]$ s with a resolution of 1/128 (i.e., 0.039 s) yields a stability boundary in the range 0.1562 s to 0.1953 s and stability for smaller time delays, see Fig. 3.13c. This calculated range includes the exact solution 0.1624 s given in [34]. In this case, calculation took 496 s



(a) Example 1

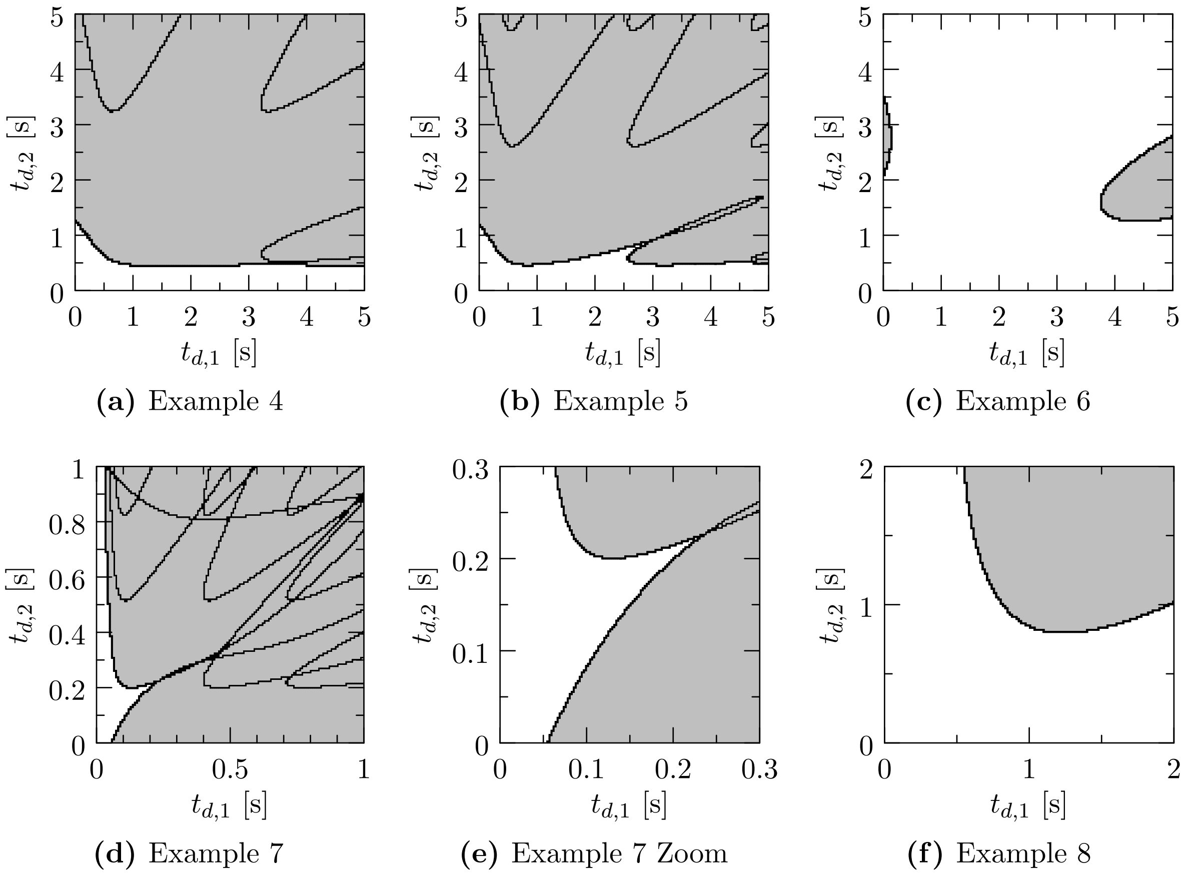
(b) Example 2

(c) Example 3

(d) Example 3 Zoom

Figure 3.13: Analysis of stability for different examples with one time delay. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in $t_{d,1}$ and ϵ.

If more exact results are desired, this can be achieved using two different methods. On the one hand, we can simply restart the algorithm using a smaller desired resolution. This does however lead to longer computation times and more memory consumption. On the other hand, we can restart the calculation using a smaller range of interest. E.g., if we choose $t_{d,1} = [0.15,\ 0.25]$ s with a desired resolution of 1/128 (i.e., 0.00078 s) this yields the range 0.1617 s to 0.1625 s for the lower stability boundary, see Fig. 3.13d. When using this smaller range of interest it also becomes clear that there is an additional stable region with $t_{d,1}$ around 0.2 s. If we repeat the stability analysis again using a starting range of $t_{d,1} = [0.1617,\ 0.1625]$ s the result shows that the bottom stability boundary is in the range 0.16234 s to 0.16235 s. The computation times for the smaller time ranges reduce significantly (11 s for $t_{d,1} = [0.15,\ 0.25]$ and 6 s for $t_{d,1} = [0.1617,\ 0.1625]$ s), although the relative resolution with respect to the considered interval $t_{d,1}$ does not change.



(a) Example 4 **(b)** Example 5 **(c)** Example 6

(d) Example 7 **(e)** Example 7 Zoom **(f)** Example 8

Figure 3.14: Analysis of stability for different examples with two time delays. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in $t_{d,1}$ and $t_{d,2}$.

3.8.2 Two incommensurate time delays

Five different examples are considered for systems with two time delays. In contrast to the results in [34] we do not calculate one upper limit for both delays but instead examine stability in the $t_{d,1}/t_{d,2}$-plane[20]. This gives deeper insights into the stability properties of the systems than computing one upper limit which is valid for both time delays.

Example four considers a first-order system with two time delays. Although a system of order one seems trivial, stability analysis of such a system with two delays is a complex problem as is shown in [34]. Using our algorithm, the results shown in Fig. 3.14a are achieved. For a small $t_{d,1}$ a slightly larger $t_{d,2}$ is allowed than for a bigger $t_{d,1}$, while for a small $t_{d,2}$ a time delay $t_{d,1}$ within a large range is admissible. The results are calculated in 217 s.

[20] The results from [34] can be determined from these plots by finding the largest square region around $t_{d,1} = t_{d,2} = 0$ which does not intersect any stability boundary. By doing this we can see that the results shown here correspond to the exact results given in [34] (within the chosen resolution).

The results for example five, another first-order system with two delays, are quite similar. However, the stable regions in the delay space $t_{d,1}/t_{d,2}$ become a little more complex as is shown in Fig. 3.14b. Again, for a small $t_{d,2}$ a rather large range for $t_{d,1}$ is admissible, while in this case, for a $t_{d,1}$ around 2.5 s a slightly larger $t_{d,2}$ is allowed than for other values of $t_{d,1}$ (except for very small delays). Calculation takes rather long in this case and completes after 1087 s.

In example six a second-order system with two delays is examined. In [34] an upper bound for the delays of roughly 2.1 s is given. Actually, as is shown in Fig. 3.14c, much larger delays can also lead to a stable system as long as the delays are present in both delay-channels. These results are calculated in 35 s.

Example seven considers a third-order system with two delays. No usable results are shown for this system in [34] except for the results generated by the algorithm from [43] which is not rigorous. In our case, a smaller upper limit of 1 s is chosen for $t_{d,1}$ and $t_{d,2}$ as the stability analysis otherwise takes extremely long. The results are depicted in Fig. 3.14d which is calculated in 2637 s. The reason why the calculation takes so long is evident from that figure: there is a very large number of boundaries where one eigenvalue crosses the imaginary axis. It can be assumed that the number of boundaries becomes even larger when considering a larger range of time delays. Consequently, stability for an even smaller range of time delays (see Fig. 3.14e) is evaluated much faster within 191 s.

The final example with two time delays, example eight, considers a system of order four. Again, only [43] generates usable results in [34] which are however not rigorous. With our algorithm, we consider an upper bound of $t_{d,1} = t_{d,2} = 2$ s. and determine the results depicted in Fig. 3.14f within a calculation time of 34 s

3.8.3 Three incommensurate time delays

Finally, one example with three time delays is considered. The stability analysis is extremely complex in that case as is indicated by the results given in [34] and by the computation times for our algorithm. Here, we choose to determine the upper bound allowed for all three delays as is done in [34][21]. To achieve this result each delay $t_{d,i}$ is defined as an uncertain parameter $\alpha_i = [0,\ 1]$ multiplied by the same time delay τ, i.e.,

$$t_{d,i} = \alpha_i \tau \tag{3.66}$$

for $i = 1, 2, 3$. When considering the starting range $\tau = [0,\ 5]$ with a resolution of 1/128 the lowest stability boundary is found in the range $\tau = [0.2344,\ 0.2734]$ within a calculation time of 10852 s. By constraining the search to the interval $\tau = [0.2,\ 0.3]$, again with a resolution of 1/128 we achieve the result of $\tau = [0.2351,\ 0.2360]$ for the lowest stability boundary[22]. Stability evaluation on this small parameter range is very fast and is performed within 22 s.

[21] Determining the allowed delay in the three-dimensional delay-space was not feasible, i.e., the algorithm had not terminated after several hours.

[22] In [34] an admissible time delay of 0.238 s is determined using the *exact* method from [43]. This is *larger* than the value determined with our method and shows the danger of evaluating a stability condition which must be checked for an infinite range, at a number of discrete points: the stable region may be overapproximated if the actual stability boundary is between the discretization points.

3.9 Discussion

This chapter presented a newly developed stability analysis algorithm for LTI systems with time delay and parametric uncertainties. The algorithm is based on work by *Zettler, Garloff et al.* in [7, 21] and extends their algorithm as follows:

- time delay systems (also with incommensurate delays) are examined by incorporating Taylor Models,
- stable regions may be determined in a low-dimensional parameter space (e.g., two parameters for plotting) while considering additional[23] interval parameters (i.e., parameters not known exactly but constrained to an interval), which is of interest especially for robust controller synthesis,
- desired minimum damping may be specified,
- and frequency sweeps up to infinite frequency are enabled by a system transformation.

This results in an efficient, versatile stability analysis algorithm. The classes of systems which can be examined using this approach are LTI systems of retarded type

- with incommensurate time delays and parametric uncertainties,
- with commensurate time delays and parametric uncertainties,
- without time delay but with parametric uncertainties,
- with incommensurate time delays and without parametric uncertainties,
- with commensurate time delays and without parametric uncertainties,
- without time delay and without parametric uncertainties,

where the first is the most general case. In the following, several aspects of the proposed stability analysis algorithm are discussed.

3.9.1 Computational Complexity

The algorithm may be applied to systems of relatively high order (around 10 to 20) with several uncertain parameters (around 5 to 10). Most of the examples in Chapter 4, e.g., have five uncertain parameters of which two are gridded while three are constrained to intervals and the system with highest order which is considered is of order 12.

The overall computational complexity is dominated by the branch and bound algorithm which has exponential complexity with respect to the number of dimensions in $\tilde{\boldsymbol{Q}}$. Determining the effect of the number of states and delays on computation time is unfortunately not easy as the complexity of evaluating the Taylor Model depends on the exact characteristic function, although a rising number of states and delays usually leads to longer computation times.

[23] The number of parameters which can be considered is not limited in theory. In the examples in Chapter 4 we generally consider five parameters of which two are gridded for plotting and the other three are constrained to an interval.

In practice, computation times strongly depend on the exact problem, parameter ranges, etc. as the number of necessary subdivisions has by far the greatest effect on computation time. On the one hand, a large region can be ignored if zero exclusion is determined, and if zero inclusion is determined it is sufficient to subdivide in the direction of the parameters for which $\tilde{\boldsymbol{q}}_{\min} < 1$, i.e., the dimension of the problem is reduced. On the other hand, it can take very long to determine all boundaries in the parameter space for problems with many stability crossings. Computation times are also strongly affected by the number of delays, i.e., incommensurate time delays generally lead to a very long computation time as can be seen for the examples considered in Section 3.8. More specifically, this has a large effect on the time it takes to check whether disjoint regions are stable whereas the effect on boundary mapping is minimal.

3.9.2 Conservatism

The stability analysis algorithm can either be used to determine whether a system is stable for a set of interval parameters or determine regions in the parameter space which are stable, regions which are not stable, and regions on the stability boundary. For the former case the results of the algorithm are non-conservative, while for the latter minimum conservatism is introduced due to the limited resolution with which the boundary mapping is performed.

If a region is on the boundary this means that at least one eigenvalue passes the stability boundary for some parameter value within this region. Therefore, we must assume that the boundary belongs to the unstable region when interpreting the results. This introduces some minimal conservatism to the boundary mapping, i.e. the size of the unstable region is overapproximated. However, in contrast to other stability analysis methods the conservatism, i.e., the maximum amount of overapproximation is tunable by specifying the desired resolution $\tilde{\boldsymbol{q}}_{\min}$ in the parameter space. Reducing the resolution and thus conservatism generally leads to longer computation times. Alternatively, a less conservative result can be determined by choosing a smaller starting range for the parameter of interest, see example three in Section 3.8.1 and example nine in Section 3.8.3.

3.9.3 Marginal Stability

For a marginally stable system at least one eigenvalue lies exactly on the imaginary axis while no Eigenvalue is right of the imaginary axis (see Section A.1). For the case of systems with uncertain time delay, marginal stability may also arise for some time delay while the system is stable for larger or smaller time delays, see example one in Section 3.8.1 for an example. In such a case, with increasing time delay the real part of an eigenvalue with negative real part increases until it is zero and with further increasing time delay it decreases again. Thereby, the eigenvalue touches the imaginary axis but never passes over the imaginary axis.

Whether there is an eigenvalue on the imaginary axis or an Eigenvalue touches the imaginary axis for some value of an uncertain time delay: both of these cases are problematic for the proposed algorithm. As the value set is overapproximated when checking for zero exclusion and underapproximated when checking for zero inclusion, the algorithm never terminates. See example one in Section 3.8.1 for a case where this occurs.

Unfortunately, it is not actually possible to distinguish cases when the system may be marginally stable. Instead, this can currently only be assumed, if the algorithm does not converge for a very long time. In such a case it is possible to examine stability using the proposed algorithm by shifting the stability boundary into the left half-plane by some small value. This results in some conservatism which may be interpreted as a stability reserve.

3.9.4 Marginal Stability for Time-Delay Systems

For time-delay systems, one special case of marginal stability is considered where an eigenvalue touches the point $\boldsymbol{s} = 0$ due to the time-delay term, i.e., $a_0 = 0$ for $\boldsymbol{s} = 0$ and $a_0 \neq 0$ for $\boldsymbol{s} = j\omega$ with $\omega \in \mathbb{R}\backslash\{0\}$[24]. This case can occur, e.g., in systems which can be simplified in the case without time delay. The teleoperation system with position-based admittance controller and force-force exchange (see Section 4.5.5) is one example where this occurs.

In this case, the stability analysis is carried out as follows in our implementation:

- The special case is detected by evaluating a_0 for $\boldsymbol{s} = 0$.
- If this is zero then the subsequent boundary mapping is carried out for $\partial\boldsymbol{\Gamma} = [\epsilon, \infty]$ with $\epsilon > 0$ a small positive constant.
- The same is then done for the final stability test for the whole region $\boldsymbol{\Gamma}$, i.e., a small region around $\boldsymbol{s} = 0$ is excluded from this final test.
- The stability analysis can then be carried out as usual but the overall system is marked as marginally stable.

It must be noted that using this approach it cannot be guaranteed that there is no eigenvalue in the small region around $\boldsymbol{s} = 0$ which is excluded from the boundary mapping and final stability check[25].

3.9.5 Convergence

There are some edge cases where the proposed algorithm does unfortunately not converge, in addition to the case of marginal stability, which is discussed in Section 3.9.3 and Section 3.9.4. Similarly to the case of marginal stability, if an eigenvalue exactly lies on the boundary $\partial\boldsymbol{\Gamma}$ for the minimum or maximum value of the considered parameter range, then the algorithm can never determine zero-inclusion or exclusion for this point. In this case convergence can be achieved by minimally changing the upper or lower bound of the parameter range. This could also be done automatically which is however a topic of future research.

[24] In literature on delay-independent stability such a system has recently been referred to as *weakly stable* but not *strongly stable*, see [59].

[25] In our practical implementation the value $\epsilon = 1e^{-10}$ is chosen which is a very small value. Even so, theoretically there could be an additional eigenvalue in this small region and the system may be unstable.

This problem can also occur in other cases, e.g., if an eigenvalue lies on the boundary $\partial\mathbf{\Gamma}$ for the mid-point of the considered parameter range. Then, after subdividing the starting interval once in this direction, the same problem occurs. In this case the problem could be circumvented by subdividing the same interval at different random points near the mid-point. As the computational cost increases when taking this measure, this would however require a good heuristic to decide when this is necessary which is a possible topic of future research.

3.9.6 BIBO-Stability

Instead of examining the characteristic equation the poles of the transfer function may be examined. The $\mathbf{\Gamma}$-mapping method introduced in this chapter is applicable to this case. Then, the method determines regions in the parameter space for which there is no pole of the transfer function in a specifiable region $\mathbf{\Gamma}$ in the complex plane. If the right half-plane, i.e. $\mathbf{\Gamma} = \mathbf{\Gamma}_+$ is chosen, this results in determining Bounded Input Bounded Output (BIBO) stable regions in the parameter space.

Special care has to be taken to pole-zero cancellations. If these arise independent of $\boldsymbol{s}$ one may proceed as for the case without uncertainties, i.e. if the pole is in the left half-plane it may be canceled, otherwise not. In some cases pole-zero cancellation may arise depending on $\boldsymbol{s}$ due to the time delay. The pole-zero cancellation for a time-delay system in the case of $s = 0$ is discussed in [60]. This special case is comparable to the case of marginal stability with an eigenvalue at $\boldsymbol{s} = 0$ discussed in the previous section. It is not further considered here but may be an interesting topic for future research.

3.9.7 Mapping of unbounded regions

In this chapter, a method to map unbounded regions to the parameter space by transforming such regions to several bounded regions is introduced. This is used, e.g., to mapping the imaginary axis to the parameter space when considering Hurwitz stability. The transformation introduced here may be seen as a general solution which is applicable to many different stability analysis methods which rely on a frequency sweep. An open research question which remains in this context is whether there is a better way of dealing with time delay terms than proposed here.

3.9.8 Comparability of Results

When evaluating the examples in Section 3.7 and Section 3.8 it became evident that it is quite difficult to compare different robust stability analysis algorithms for time-delay systems. Different algorithms offer a different accuracy which is often also influenced by user-controller parameters, and calculation times strongly depend on hardware and implementation. Moreover, comparing calculation times between algorithms with different accuracy is problematic. Finally, there is a lack of standardized benchmark examples which makes the selection of *good* examples difficult.

Investigating solutions to these problems is an interesting topic for future research and could greatly facilitate further development in this field. It would, e.g., be interesting to use a metric for the calculation time which does not directly depend on the used hardware and makes algorithms comparable, e.g., the number of floating-point operations. Moreover, a large collection of standard benchmark examples could be established which could then be used to compare existing methods and evaluate new methods. Such a collection should cover a large class of time-delay systems with different types of uncertainties and should include practically relevant examples as well as some special cases.

3.10 Summary and Future Work

In this chapter a novel approach to the stability analysis of LTI systems of retarded type with incommensurate time delays and parametric uncertainties was introduced. The overall algorithm maps $\boldsymbol{\Gamma}$-regions from the complex plane to the parameter space $\boldsymbol{Q} \times \boldsymbol{T}_d$ by using a branch and bound algorithm which, in turn, uses efficient checks for zero-inclusion and zero-exclusion in the non-linear holomorphic function that are based on evaluating inclusion and exclusion of the negative interval remainder of the Taylor Model in the polynomial part of the Taylor Model. The algorithm was demonstrated using a simple example for robust stability analysis in Section 3.7 and a number of time-delay problems from literature in Section 3.8.

The method allows for an intuitive stability analysis and controller design in the parameter space, similar to the well-known *Parameter-Space Approach* by *Ackermann* [1]. Like the *Parameter-Space Approach*, our method can handle non-affine dependencies of the characteristic equation on the uncertain parameters, i.e., the coefficients of the characteristic equation may depend polynomially on the uncertain parameters. The key difference is that our method is also applicable to time-delay systems and even systems with incommensurate delays can be examined which is a very challenging problem. Moreover, a larger number of uncertain parameters can be considered in comparison with [1] as we can map stable regions to a low-dimensional parameter space while additional parameters are not known exactly but instead constrained to intervals. This property is especially useful for controller design where we can determine controller parameters which result in a stable system for a whole range of plant parameters. Or we could determine suitable gain scheduling of one controller parameter, depending on one plant parameter, while other parameters are not known exactly. In comparison with other robust stability analysis methods for time-delay systems, the actual stability check used in our method is not conservative, i.e., the results are non-conservative if stability is evaluated for a fixed parametrization or set of interval parameters, and the only conservatism when mapping stable regions to the parameter space, is due to the chosen resolution. This unique property makes an application of our newly developed method to various time-delay systems especially interesting. A C++ implementation of the presented algorithm is available for download at `http://www.github.com/schauss/glob_stab`.

An application of the method to different teleoperation architectures together with a newly developed transparency analysis method is the topic of Chapter 4. An application to other time-delay systems, be it with or without parametric uncertainties, could be a topic of future research. In addition, convergence for the edge cases mentioned in the discussion could be further evaluated and strategies to assure convergence in all cases could be developed. Moreover, the mapping of unbounded regions to several bounded regions could be further improved for the case of time delay. Specifically, simply replacing the time-delay term by an interval may lead to a slow convergence rate and there may be better solutions, e.g., symbolically determining the partial derivatives of the function and evaluating these partial derivatives. Finally, the method could be extended towards an automatic parameter tuning by incorporating it in a global optimization method. This possible future extension of our method is described in more detail in Chapter 5.

4 Stability and Transparency Analysis of Teleoperation Systems

Summary. *In this chapter we analyze the stability and transparency of teleoperation systems which are modeled as linear time-invariant systems with parametric uncertainties and time delay in the communication channel. Our robust stability analysis method is combined with a novel transparency analysis method which results in an intuitive analysis and design of teleoperation systems in the parameter space. The main contributions of this chapter are*

- *a parameter-space stability analysis of several well-known teleoperation architectures with time delay in the communication channel,*
- *a novel method for parameter-space transparency analysis of teleoperation systems,*
- *a parameter-space transparency analysis of these architectures which especially focuses on evaluating the effects of time delay on transparency.*

The method developed in the previous chapter can be used for the robust stability analysis and controller design of time-delay systems. In this chapter, the method is applied to teleoperation systems which represent a typical application area for such a method. A teleoperation system allows a human operator to perceive and interact with a remote environment by using a robotic system. Thereby, cognitive skills of a human operator can be *transferred* to the robotic system which is generally called teleoperator or telerobot in this context. Teleoperation has been a field of research for several decades. In this thesis, we focus on haptic teleoperation which is often also referred to as telemanipulation or bilateral force-reflecting teleoperation. For a literature review of this field, see [5].

A haptic teleoperation system enables an operator to physically interact with a remote environment. In essence, a robotic system, which is referred to as teleoperator in the following, is controlled by the operator using a haptic interface. Idealized, the functionality of a haptic teleoperation system can be described as follows: the operator moves the haptic interface, this movement is measured and imposed on the teleoperator, the teleoperator measures the interaction force with the remote environment, and this force is applied to the haptic interface and thereby felt by the operator. In practice, many different control architectures, which exchange different types of information between haptic interface and teleoperator, can be used that all result in a similar overall system behavior. The different architectures do however result in different stability and transparency properties which must both be taken into account when designing teleoperation systems. On the one hand, the system should display the remote environment to the operator as if he were directly interacting with it, i.e., the system should be transparent. On the other hand, the overall system must be robustly stable when interacting with arbitrary environments. A perfectly transparent teleoperation system is only marginally stable (see [4]) and can therefore not be realized in practice. Therefore, an optimal compromise is sought which consists of a system which is robustly stable for all remote environments of interest and, at the same time, is as transparent as possible. This goal is pursued in this chapter by combining the stability analysis method developed in Chapter 3 with a novel transparency analysis method which we first presented in [61]. In the following, we shortly describe why these tools are suitable for the stability and transparency analysis of teleoperation systems.

The stability analysis of teleoperation systems is a challenging topic which has been a field of active research for many decades. Difficulties in this context mainly arise from two aspects which are inherent to most practical teleoperation systems. First, the remote environment is often located at a large distance from the operator. This implies time delays in the communication channel which must be taken into account in the stability analysis. Second, a teleoperation system is generally used to interact with different objects in the remote environment which exhibit different dynamics. Moreover, the dynamics of the operator interacting with the system are not known exactly. This results in uncertainties in the overall system model which must be considered in the stability analysis. One approach to cope with these uncertainties is to assume a model of the environment and a model of the operator which are both subject to parametric uncertainties. Then, the stability analysis method developed in Chapter 3 is ideally suited to examine stability of the overall system subject to parametric uncertainties and time delay.

In comparison with stability analysis methods, much less research effort has gone into the transparency analysis of teleoperation systems for several reasons. On the one hand, transparency of teleoperation systems is a secondary goal compared to stability which is absolutely necessary. On the other hand, stability analysis methods developed in the field of teleoperation are, in general, also applicable to other domains which makes research in this field very interesting. Therefore, in general, relatively simple measures are used to quantify the transparency of teleoperation systems which, e.g., often only consider one or two fixed environments, usually the extreme cases of free space and stiff contact. In contrast, our approach determines the distortion of the environment impedance caused by the teleoperation system for a whole range of remote environments. The impedance displayed to the operator is approximated by a simple, physically interpretable mechanical impedance, e.g., a mass-spring-damper system, and parameters of this approximated displayed impedance are determined for a whole range of parameters of the environment impedance. This results in an intuitive graphical representation of transparency which offers insights into the range of environments for which a system is sufficiently transparent.

The rest of this chapter is structured as follows: First, a short state of the art of stability and transparency analysis of teleoperation systems is presented which mainly focuses on closely related work. Then, a general controller architecture which is widely used in literature, the four-channel architecture, is introduced together with models of the haptic interface, teleoperator, operator, and environment in Section 4.2. This is followed by a short description of how stability analysis using the method from Chapter 3 is applied to teleoperation systems in Section 4.3. Next, a novel method for parameter-space transparency analysis is introduced in Section 4.4. Stability and transparency of several common teleoperation setups is examined in the main part of the chapter in Section 4.5. The chapter ends with a summary of the most important results.

4.1 State of the Art

A good overview of early research in the field of teleoperation can be found in the well-known book [2] by *Sheridan* published in 1992. Around this time there was a large interest in bilateral force-reflecting teleoperation and several fundamental papers concerning stability and transparency were published. These papers lay much of the theoretical groundwork on which most newer publications build. *Hannaford* introduces the hybrid two-port model in [62], which is used for stability and transparency analysis of teleoperation systems since. In [63], *Anderson and Spong* introduce the scattering transformation to teleoperation which passivates the communication channel and enables stable teleoperation in the presence of time delays. A very similar result using a different formulation is achieved by *Niemeyer and Slotine*, who introduce wave variables to teleoperation in [64]. In 1992, *Lawrence* first formulates the four-channel architecture in which forces and velocities are transmitted from master to slave and vice versa, determines the hybrid two-port model of this general architecture, and formulates stability and transparency objectives in [4, 65]. *Lawrence* comes to the conclusion that force as well as velocity information is necessary to cancel the device dynamics of master and slave and enable transparent force-reflecting teleoperation. The same conclusion is reached by *Yokokohji and Yoshikawa* in [66] where a very similar framework is introduced. Building on these fundamental results, many practical teleoperation systems have been developed in the following years while theoretical research continues in many different directions. For more information on the broad field of bilateral force-reflecting teleoperation, the literature review in [5] is recommended. The rest of this section concentrates on work closely related to this chapter, i.e., prior work on stability and transparency analysis of classic two-channel and four-channel teleoperation architectures.

4.1.1 Stability Analysis

A common stability analysis method for teleoperation systems consists of decomposing the overall system into several subsystems (one-ports and two-ports) connected in cascade structure. Passivity of each subsystem then results in passivity of the complete teleoperation system which, in turn, implies stability, see, e.g., [64]. This approach is widely spread in literature, especially for systems with time delay [5]. However, passivity of each subsystem can only be shown if some simplifying assumptions are made, e.g. actuator dynamics are neglected and humans are modeled as passive systems. Moreover, passivity of each subsystem is also not actually necessary and generally leads to conservative results [67, 68] as the overall system can be passive if some subsystems inject energy while other subsystems extract energy. In [4], this conservatism is reduced by modeling the teleoperation system consisting of master, slave, communication channel, and all controllers as *one* two-port and examining stability using *Llewellyns absolute stability criterion* [69] which states: a two-port is absolutely stable if and only if, the one-ports resulting from terminating the two-port with an arbitrary, passive impedance on either side are passive.

The *absolute stability criterion* seems ideal for teleoperation systems where the operator and environment are generally considered as unknown but passive impedances[1]. However, in reality the impedance of the environment and, even more so, the impedance of the human arm are limited to a small subset of all possible passive impedances. Therefore, [70, 71] *Haddadi & Hashtrudi-Zaad* developed a graphical method for stability analysis in the scattering domain which is applicable to systems with time delay and allows imposing bounds on either the environment impedance or the human arm impedance. However, only one of the two impedances can be limited and it is not clear how limits on the inertia, stiffness, and damping of a mass-spring-damper environment can be taken into account. Moreover, a systematic approach to controller design using this method has not been published.

Even less conservative results can be achieved if models for the human impedance and environment impedance are considered, and the model parameters are constrained to a reasonable range. Then, a stability analysis and controller design for the complete closed-loop system can be performed. For the case without time delay this has, e.g., been done in [72] and [73] using the *Parameter-Space Approach* from [1]. In this chapter, we use the stability analysis method from Chapter 3 to perform the same sort of non-conservative stability analysis and controller design in the parameter space. The main novelties in contrast to [72] and [73] is that we can consider the time delay in the communication channel which is inherent in many teleoperation systems and a larger number of uncertain parameters can be taken into account[2].

4.1.2 Transparency Analysis

Systematic transparency analysis of teleoperation systems has been investigated for more than 20 years, since it is an important design objective. In [66], *Yokokohji and Yoshikawa* defined transparency based on the frequency-dependent position and force error. An evaluation of this transparency measure therefore requires a model of the environment as well as the human operator. Then, transparency for one environment impedance *and* operator impedance can be determined[3].

The impedance transmitted to the operator, which was introduced in the context of

[1] The operator is not actually passive as he can inject energy into the system. For stability analysis the operator is however generally assumed to consist of a passive impedance and an additional *exogenous force* which constitutes the voluntary force applied by the operator. This *exogenous force* is bounded and limited to a low bandwidth. Furthermore, it is assumed that the *exogenous force* is *state-independent* and the operator does not destabilize the system on purpose. This is a reasonable assumption if we neglect cases with very large time delay where an operator might destabilize the system *by mistake* due to the fact that he cannot predict the dynamics of the system

[2] More than two parameters can only be considered with the Parameter-Space Approach by gridding the additional parameters. Therefore, it is difficult to consider more than three or four uncertain parameters.

[3] In [66] the two cases of free space and rigid contact are evaluated for one fixed human operator impedance.

the transparency-optimized four-channel architecture by *Lawrence* in [4], is more useful in practice as it does not depend on the operator impedance. It is used in [68] to examine the minimum impedance and maximum impedance which can be displayed by different teleoperation systems[4].

The impedance transmitted to the operator represents an interesting starting point for transparency analysis. However, it is difficult to interpret the transparency of a system by analyzing Bode plots of the minimum impedance (which represents the impedance displayed in free space) and the maximum impedance (which represents the impedance in rigid contact) displayed to the operator as, e.g., done in [68].

Therefore, an alternative presentation of the impedance transmitted to the operator is proposed in this chapter, which we first presented in [61]. The main idea is to approximate the impedance transmitted to the operator by a low-order impedance model, e.g., a mass-spring-damper system. On the one hand, the parameters of such an impedance model can easily be interpreted as they represent well-known physical properties. On the other hand, this allows examining the transparency for a whole range of environment parameters, by graphically analyzing the relationship between environment parameters and parameters of the approximated transmitted impedance. Differences between environment parameters and parameters of the approximated transmitted impedance then represent the distortion caused by the teleoperation system.

A similar approach to transparency analysis is published in [75] by *Hirche* et al. where an idealized teleoperation system with wave variables in the communication channel is considered, and the effect of time delay and wave damping on transparency is analyzed. More specifically, analytic expressions are derived which approximate the effect of time delay and wave damping on transparency. This offers some very interesting insights but limits the applicability of this approach to a small class of systems while our method is applicable to a wide range of teleoperation architectures.

4.2 System Model

In this section we first summarize the modeling assumptions that are made throughout this chapter. Then, a general teleoperation architecture, the so-called four-channel architecture, is briefly explained and models for the human, environment, haptic interface, and teleoperator are introduced.

4.2.1 Modeling Assumptions

In general, teleoperation systems are highly complex non-linear systems. Consider, e.g., the multi-user teleoperation system presented in [3] as an example. The overall system consists of two teleoperators mounted on mobile bases. Each teleoperator has two redundant arms with seven Degrees of Freedom (DoF) and is mounted on a mobile base which allows moving through the remote environment. The haptic interfaces on master side consist of two stationary 10-DoF hyper-redundant haptic interfaces and two 7-DoF redundant

[4] Actually, the minimum impedance and the so-called *Z-width*, i.e., the difference between minimum and maximum impedance as defined by [74] is examined

haptic interfaces mounted on a mobile base. The end-effectors of all haptic interfaces and teleoperator arms are equipped with 6-DoF force-torque sensors. Digital controllers with limited sampling rate are used to generate the control signals which are applied by motors with intrinsic actuator dynamics acting on gears with non-linear elasticities and friction. As the geometry of teleoperator arms and haptic interfaces is not identical, high-level Cartesian controllers are combined with elaborate redundancy resolution schemes which optimize, e.g., manipulability of the end-effectors. Environment dynamics are unknown, non-linear, and may change instantaneously on impact. In addition, as different arms of the two teleoperators might interact over a common object, passivity of the remote environment is not guaranteed. Operator dynamics are unknown and non-linear, and the operators are not necessarily passive but can inject an arbitrary amount of energy into the system. A large distance between the operator site and remote environment results in time delay in the communication channel, generally a packet switched network which implies time-varying delay, packet loss, and possibly limited bandwidth.

As a stability and transparency analysis of such a complex teleoperation system is not possible in practice, it is common practice to make some simplifying assumptions. The assumptions used in this chapter are:

- We assume local computed-torque controllers which effectively decouple the multiple DoF of the overall system. Therefore, the different DoF can be considered to be independent and stability can be evaluated for each DoF separately. Moreover, the haptic interface and teleoperator are modeled as damped masses as it is assumed that the computed-torque controller compensates for non-linear effects. These assumptions are very common and can be achieved in practice, e.g., by applying an adaptive trajectory controller [76] or a more advanced adaptive approach which imposes virtual dynamics on a complex robotic system as presented in [77] and applied to teleoperation for the first time in [78].

- The remote environment is modeled as a passive mass-spring-damper system, as, e.g., done in [66, 73, 79]

- The human is also modeled as mass-spring-damper system and an additional exogenous force with limited bandwidth which can be seen as the voluntary force the human applies to the system. This exogenous force is neglected in the stability analysis as it is assumed that it is *state-independent* and bounded and therefore has no effect on stability. This is the same model as, e.g., chosen in [66, 72, 73, 79, 80].

- Actuator dynamics and sensor dynamics are taken into account and modeled as first-order low-pass filters as is done, e.g., in [72, 73].

- We assume a stiff coupling between operator and haptic interface, as well as between remote environment and teleoperator. This is common practice, see, e.g., [72, 73, 79].

- A communication channel with time delay is considered and non-linear effects like packet loss are neglected as, e.g., in [4, 64, 68].

- Finally, the overall system is considered to be time-invariant for the sake of stability analysis as, e.g., in [72, 73, 79]. This is a reasonable assumption as the time delay generally does not vary much when considering modern network or internet connections and the dynamic properties of the human operator certainly vary but generally at a slow rate. Likewise, the dynamic properties of the remote environment vary slowly if the special case of impact is ignored[5]. Therefore, instead of considering time-varying parameters for operator and environment we consider the parameters to be unknown but constrained to an interval. A thorough analysis of several different teleoperation architectures for the delay-free case which considers unknown parameters constrained to intervals can be found in [72] and [73].

Using these assumptions the teleoperation system can be modeled as linear time-invariant one-DoF system with time delay and parametric uncertainties, and the stability analysis method from Chapter 3 can be applied to the system. It must be noted that the results obtained in this chapter only hold if these assumptions are true. However, the modeling assumptions used here are very common in literature on teleoperation, and stability analysis results using these assumptions have been shown to closely represent experimental results, e.g., for the delay-free case in [72, 73][6].

4.2.2 Teleoperation Architecture

To generally describe a teleoperation system, we briefly introduce the four-channel architecture with local force-control loops from [68] here, which is depicted in Fig. 4.1. It is based on the well-known work by *Lawrence* [4]. We choose this architecture as it can represent many different interconnections between the system components.

The blocks in the four-channel architecture describe the following system components. The impedance characterizing the master and slave system is denoted as $Z_m(\boldsymbol{s})$ and $Z_s(\boldsymbol{s})$ respectively, while $Z_h(\boldsymbol{s})$ and $Z_e(\boldsymbol{s})$ represent the impedance of human operator and environment, and $C_{1..6}(\boldsymbol{s})$, $C_m(\boldsymbol{s})$, and $C_s(\boldsymbol{s})$ are different filters/controllers. In all communication channels between master and slave side there is a time delay T_d. The teleoperation system has two ports on master side, the velocity $V_h(\boldsymbol{s})$ and the force $F_h(\boldsymbol{s})$. It also has two ports on slave side, the velocity $V_e(\boldsymbol{s})$ and the force $F_e(\boldsymbol{s})$. In addition to the human impedance $Z_h(\boldsymbol{s})$, which represents the dynamics of the human arm, the human applies an exogenous force $F_h^*(\boldsymbol{s})$ on master side. In contrast, the exogenous force $F_e^*(\boldsymbol{s})$ on slave side is considered to be zero, i.e., the force flowing into the system on slave side is completely determined by the environment impedance $Z_e(\boldsymbol{s})$.

[5] Impact analysis for teleoperation systems is a complex problem which can only be analyzed in practice by modeling the overall system as hybrid system with different contact states and explicitly examining switches between free-space movement and contact. This goes beyond the scope of this thesis and is not further considered here.

[6] The assumption of a linear mass-spring-damper model for human operator and remote environment is one of two common assumptions made in literature: either they are modeled as linear mass-spring-damper systems, as is done here, or they are modeled as passive but otherwise unknown impedance. Both of these approaches have advantages and disadvantages: using a linear mass-spring-damper model generally leads to less conservative results, while considering a completely unknown passive impedance generally leads to quite conservative results which might be more *safe* in some circumstances, e.g., if the mass-spring-damper models for human and environment do not hold.

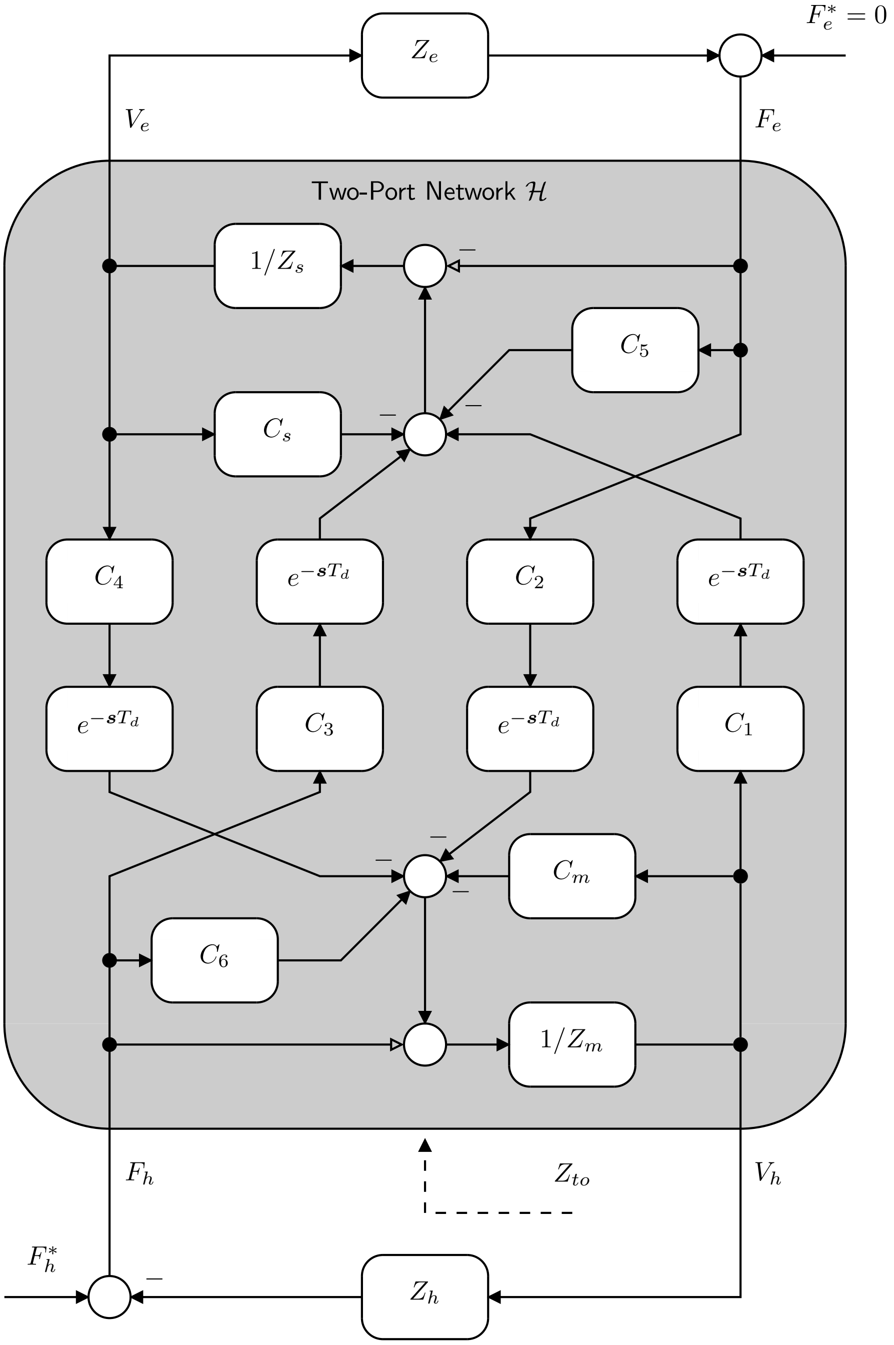


Figure 4.1: General Four-Channel Architecture of a Teleoperation System.

Following the presentation in [68], a two-port network $\mathcal{H}$ describing the complete teleoperation system can be determined as

$$\begin{bmatrix} F_h(\boldsymbol{s}) \\ -V_e(\boldsymbol{s}) \end{bmatrix} = \mathcal{H}(\boldsymbol{s}) \begin{bmatrix} V_h(\boldsymbol{s}) \\ F_e(\boldsymbol{s}) \end{bmatrix} = \begin{bmatrix} H_{11}(\boldsymbol{s}) & H_{12}(\boldsymbol{s}) \\ H_{21}(\boldsymbol{s}) & H_{22}(\boldsymbol{s}) \end{bmatrix} \begin{bmatrix} V_h(\boldsymbol{s}) \\ F_e(\boldsymbol{s}) \end{bmatrix} \tag{4.1}$$

with

$$H_{11}(\boldsymbol{s}) = \frac{(Z_m(\boldsymbol{s}) + C_m(\boldsymbol{s}))\,(Z_s(\boldsymbol{s}) + C_s(\boldsymbol{s})) + C_1(\boldsymbol{s})\,C_4(\boldsymbol{s})\,e^{-2\boldsymbol{s}T_d}}{H_{den}(\boldsymbol{s})} \tag{4.2}$$

$$H_{12}(\boldsymbol{s}) = \frac{C_2(\boldsymbol{s})\,(Z_s(\boldsymbol{s}) + C_s(\boldsymbol{s}))\,e^{-\boldsymbol{s}T_d} - (1 + C_5(\boldsymbol{s}))\,C_4(\boldsymbol{s})\,e^{-\boldsymbol{s}T_d}}{H_{den}(\boldsymbol{s})} \tag{4.3}$$

$$H_{21}(\boldsymbol{s}) = -\frac{(Z_m(\boldsymbol{s}) + C_m(\boldsymbol{s}))\,C_3(\boldsymbol{s})\,e^{-\boldsymbol{s}T_d} + C_1(\boldsymbol{s})\,(1 + C_6(\boldsymbol{s}))\,e^{-\boldsymbol{s}T_d}}{H_{den}(\boldsymbol{s})} \tag{4.4}$$

$$H_{22}(\boldsymbol{s}) = \frac{(1 + C_5(\boldsymbol{s}))\,(1 + C_6(\boldsymbol{s})) - C_2(\boldsymbol{s})\,C_3(\boldsymbol{s})\,e^{-2\boldsymbol{s}T_d}}{H_{den}(\boldsymbol{s})} \tag{4.5}$$

where

$$H_{den}(\boldsymbol{s}) = (1 + C_6(\boldsymbol{s}))\,(Z_s(\boldsymbol{s}) + C_s(\boldsymbol{s})) - C_3(\boldsymbol{s})\,C_4(\boldsymbol{s})\,e^{-2\boldsymbol{s}T_d}. \tag{4.6}$$

In the following we will also use the determinant of $\mathcal{H}$ given by

$$\Delta H(\boldsymbol{s}) = H_{11}(\boldsymbol{s})\ H_{22}(\boldsymbol{s}) - H_{12}(\boldsymbol{s})\ H_{21}(\boldsymbol{s}). \tag{4.7}$$

The four channel architecture as presented here is widely used in literature on teleoperation. It can be used to represent a large variety of interconnections between master and slave by choosing $C_{1..6}$, C_m, and C_s appropriately. This is due to the fact that an arbitrary combination of master velocity, master force, slave velocity, and slave force can be used to determine the actuator force on master side as well as slave side. The actuator force applied to the master device is given by

$$\tilde{F}_m(\boldsymbol{s}) = -C_m(\boldsymbol{s})V_h(\boldsymbol{s}) + C_6(\boldsymbol{s})F_h(\boldsymbol{s}) - e^{-\boldsymbol{s}T_d}(C_4(\boldsymbol{s})V_e(\boldsymbol{s}) + C_2(\boldsymbol{s})F_e(\boldsymbol{s})) \tag{4.8}$$

while the actuator force applied to the slave device is

$$\tilde{F}_s(\boldsymbol{s}) = e^{-\boldsymbol{s}T_d}(C_1(\boldsymbol{s})V_h(\boldsymbol{s}) + C_3(\boldsymbol{s})F_h(\boldsymbol{s})) - C_s(\boldsymbol{s})V_e(\boldsymbol{s}) - C_5(\boldsymbol{s})F_e(\boldsymbol{s}). \tag{4.9}$$

Thus, we can use this architecture to, e.g., represent classical two-channel architectures, two-channel architectures with local position-based admittance controllers, transparency-optimized three-channel or four-channel architectures, and many more[7].

The four-channel architecture is used as basis to formulate the transparency analysis method presented in Section 4.4. Therefore, the controllers $C_{1..6}$, C_m, and C_s which allow a representation using this four-channel architecture are given for all examples in Section 4.5. In addition to these controller models for human, environment, master (haptic interface), and slave (teleoperator) are required to completely determine the dynamics of a teleoperation system. These models are introduced in the following subsections.

[7] It must be noted that some architectures which can be represented very easily if we don't constrain ourselves to the four-channel architecture become quite complex when formulated using the four-channel architecture, e.g., the position-based admittance controller with force-exchange in Section 4.5.5.

4.2.3 Human

The human is modeled as passive impedance (mass-spring-damper system) and an additional exogenous force $f_h^\star / F_h^\star(\boldsymbol{s})$ which is assumed to be zero for the sake of stability analysis, i.e.

$$f_h = f_h^\star - (m_h + m_{em})\ddot{x}_m - b_h\dot{x}_m - c_h x_m, \tag{4.10}$$

in the time domain or, equivalently, in the frequency domain

$$Z_h(\boldsymbol{s}) = \left.\frac{-F_h(\boldsymbol{s})}{V_h(\boldsymbol{s})}\right|_{F_h^\star(\boldsymbol{s})=0} = \frac{(m_h + m_{em})\boldsymbol{s}^2 + b_h\boldsymbol{s} + c_h}{\boldsymbol{s}}, \tag{4.11}$$

where x_m is the position, $\dot{x}_m/V_h(\boldsymbol{s})$ is the velocity, and and $\ddot{x}_m$ is the acceleration of the end-effector, $f_h/F_h(\boldsymbol{s})$ is the interaction force at the end-effector, and m_h, b_h, and c_h are the inertia, damping, and stiffness used to model the human arm, while m_{em} is the inertia of the end-effector of the haptic interface[8]. The measured human force $\tilde{f}_h/\tilde{F}_h(\boldsymbol{s})$ is determined by low-pass filtering the actual force $f_h/F_h(\boldsymbol{s})$ to account for sensor dynamics, i.e.

$$\dot{\tilde{f}}_h = \frac{f_h - \tilde{f}_h}{\tau_h} \tag{4.12}$$

which can be written in the frequency domain as

$$G_h(\boldsymbol{s}) = \frac{\tilde{F}_h(\boldsymbol{s})}{F_h(\boldsymbol{s})} = \frac{1}{1 + \tau_h\boldsymbol{s}}. \tag{4.13}$$

4.2.4 Environment

The environment is modeled as passive mass-spring-damper system, i.e.

$$f_e = (m_e + m_{es})\ddot{x}_m + b_e\dot{x}_s + c_e x_s, \tag{4.14}$$

in the time domain or, equivalently, in the frequency domain

$$Z_e(\boldsymbol{s}) = \frac{F_e(\boldsymbol{s})}{V_e(\boldsymbol{s})} = \frac{(m_e + m_{es})\boldsymbol{s}^2 + b_e\boldsymbol{s} + c_e}{\boldsymbol{s}}, \tag{4.15}$$

where x_s is the position, $\dot{x}_s/V_e(\boldsymbol{s})$ is the velocity, and and $\ddot{x}_s$ is the acceleration of the end-effector, $f_e/F_e(\boldsymbol{s})$ is the interaction force at the end-effector, and m_e, b_e, and c_e are the inertia, viscous damping, and stiffness of the environment, while m_{es} is the inertia of the end-effector of the teleoperator[9]. Note the opposite sign in comparison with the model of the human operator.

[8] We assume a stiff coupling between end-effector and human arm and can therefore add the end-effector inertia to the inertia of the human arm.

[9] As we assume a stiff coupling between end-effector and environment the end-effector inertia can be added to the inertia of the environment.

This linear, passive environment model is used throughout this chapter. Using a linear model is necessary when performing a stability and transparency analysis in the frequency domain and is common within literature on haptic teleoperation[10].

The measured environment force $\tilde{f}_e/\tilde{F}_e(\boldsymbol{s})$ is determined by low-pass filtering the actual force $f_e/F_e(\boldsymbol{s})$ to account for sensor dynamics, i.e.

$$\dot{\tilde{f}}_e = \frac{f_e - \tilde{f}_e}{\tau_e} \tag{4.16}$$

which can be written in the frequency domain as

$$G_e(\boldsymbol{s}) = \frac{\tilde{F}_e(\boldsymbol{s})}{F_e(\boldsymbol{s})} = \frac{1}{1+\tau_e \boldsymbol{s}}. \tag{4.17}$$

4.2.5 Haptic Interface (Master)

The haptic interface is modeled as damped mass on which the force of the human $f_h/F_h(\boldsymbol{s})$ as well as the actuator force $\tilde{f}_m/\tilde{F}_m(\boldsymbol{s})$ acts. This can be written as

$$\tilde{f}_m + f_h = m_m \ddot{x}_m + b_m \dot{x}_m, \tag{4.18}$$

in the time domain or, equivalently, in the frequency domain

$$Z_m(\boldsymbol{s}) = \frac{\tilde{F}_m(\boldsymbol{s}) + F_h}{V_h(\boldsymbol{s})} = m_m \boldsymbol{s} + b_m, \tag{4.19}$$

where x_m is the position, $\dot{x}_m/V_h(\boldsymbol{s})$ is the velocity, and and $\ddot{x}_m$ is the acceleration of the end-effector, and m_m and b_m are the inertia and damping used to model the haptic interface. The actuator force $\tilde{f}_m/\tilde{F}_m(\boldsymbol{s})$ acting at the master device is determined by low-pass filtering a desired actuator force $f_m/F_m(\boldsymbol{s})$ to account for actuator dynamics, i.e.

$$\dot{\tilde{f}}_m = \frac{f_m - \tilde{f}_m}{\tau_m} \tag{4.20}$$

which can be written in the frequency domain as

$$G_m(\boldsymbol{s}) = \frac{\tilde{F}_m(\boldsymbol{s})}{F_m(\boldsymbol{s})} = \frac{1}{1+\tau_m \boldsymbol{s}}. \tag{4.21}$$

4.2.6 Teleoperator (Slave)

The teleoperator is modeled as damped mass on which the force of the environment $f_e/F_e(\boldsymbol{s})$ as well as the actuator force $\tilde{f}_s/\tilde{F}_s(\boldsymbol{s})$ acts. This can be written as

$$\tilde{f}_s - f_e = m_s \ddot{x}_s + b_s \dot{x}_s, \tag{4.22}$$

[10] Mainly when examining transient responses on impact it is common to use non-linear environment models instead of linear ones. This is not a topic of this thesis.

in the time domain or, equivalently, in the frequency domain

$$Z_s(\boldsymbol{s}) = \frac{\tilde{F}_s(\boldsymbol{s}) - F_e(\boldsymbol{s})}{V_e(\boldsymbol{s})} = m_s\boldsymbol{s} + b_s, \tag{4.23}$$

where x_s is the position, $\dot{x}_s/V_e(\boldsymbol{s})$ is the velocity, and and $\ddot{x}_s$ is the acceleration of the end-effector, and m_s and b_s are the inertia and damping used to model the teleoperator. Note the opposite sign of the environment force f_e in comparison with the model of the master.

The actuator force $\tilde{f}_s/\tilde{F}_s(\boldsymbol{s})$ acting at the slave is determined by low-pass filtering a desired actuator force $f_s/F_s(\boldsymbol{s})$ to account for actuator dynamics, i.e.

$$\dot{\tilde{f}}_s = \frac{f_s - \tilde{f}_s}{\tau_s} \tag{4.24}$$

which can be written in the frequency domain as

$$G_s(\boldsymbol{s}) = \frac{\tilde{F}_s(\boldsymbol{s})}{F_s(\boldsymbol{s})} = \frac{1}{1 + \tau_s\boldsymbol{s}}. \tag{4.25}$$

4.3 Parameter-Space Stability Analysis

The robust stability analysis method for time-delay systems developed in Chapter 3 is used in this chapter to analyze stability of the different teleoperation architectures in Section 4.5. Therefore, a state-space representation of the complete system is determined from the dynamic equations of human (4.10), environment (4.14), haptic interface (4.18), and teleoperator (4.22), the filters (4.12) and (4.16) representing the force sensor dynamics and low-pass filtering of the measured force, the actuator filters (4.20) and (4.24) representing the actuator dynamics, as well as the controllers which represent the interconnection between master and slave[11].

In all cases, this results in a state-space equation of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{q}, T_d)\boldsymbol{x} = \boldsymbol{A}_0(\boldsymbol{q})\boldsymbol{x} + \boldsymbol{A}_1(\boldsymbol{q})e^{-T_d\boldsymbol{s}}\boldsymbol{x} \tag{4.26}$$

which represents a simpler version of (3.1) limited to the case of one time delay. In all examples, the system state $\boldsymbol{x} \in \mathbb{R}^n$ includes the position and velocity of master and slave as well as the filtered values of the sensor and actuator filters[12]. In some examples, there are additional states, e.g., integrated transmitted velocity for the wave-variable example in Section 4.5.3 or the positions and velocities of the virtual admittances on master and slave side for the position-based admittance control architecture with force exchange in

[11] The equations for the controllers are given in Section 4.5. They include the communication channel between master and slave which consists of the time delay between master and slave and, for the example with wave variables, the wave variable transformations.

[12] In case of the simplified force-position architecture without sensor and actuator filters considered in Section 4.5.2 the system state only contains the master and slave velocities and positions.

Section 4.5.5. The parameter vector $\boldsymbol{q} \in \mathbb{R}^l$ contains the parameter vectors

$$\boldsymbol{q}_h = [m_h,\ b_h,\ c_h]^T \tag{4.27}$$

$$\boldsymbol{q}_e = [m_e,\ b_e,\ c_e]^T \tag{4.28}$$

$$\boldsymbol{q}_m = [m_m,\ m_{em},\ b_m,\ \tau_m,\ \tau_h]^T \tag{4.29}$$

$$\boldsymbol{q}_s = [m_s,\ m_{es},\ b_s,\ \tau_s,\ \tau_e]^T \tag{4.30}$$

related to the models of human ($\boldsymbol{q}_h$), environment ($\boldsymbol{q}_e$), haptic interface ($\boldsymbol{q}_m$), and teleoperator ($\boldsymbol{q}_s$) as well as the specific controller parameters which are different for each considered example in Section 4.5. Note that some of these parameters are set to fixed values while an interval with given upper and lower bound is considered for other parameters[13].

4.4 Parameter-Space Transparency Analysis

In this section our novel transparency analysis method which we first presented in [61] is introduced. The general idea is as follows: a physical model consisting of a mass, spring, and damper and represented by the impedance $\hat{Z}_{to}$ is fitted to the impedance transmitted to the operator Z_{to} by a numerical optimization. This system identification is performed for a range of environment parameters which allows for a graphical transparency analysis in the parameter space, by relating parameters of the approximated impedance transmitted to the operator to parameters of the model of the environment.

It must be noted that the transparency analysis method developed here is based on a linear environment model while real environments are often non-linear in practice. However, non-linear environment models can often be represented as linear models where the model parameters vary, e.g., in the case of a Hunt-Crossley Model [81] depending on penetration depth and speed. Therefore, our results are also applicable to non-linear environment models by making sure the range of environment parameters we consider covers the variation of the parameters caused by the non-linearity of the environment model.

4.4.1 Transparency Transfer Function

The impedance of the environment transmitted to the operator Z_{to} over the teleoperation system can be determined from the general four channel architecture (see Section 4.2.2) as

$$Z_{to}(\boldsymbol{s}) = \left.\frac{F_h(\boldsymbol{s})}{V_h(\boldsymbol{s})}\right|_{F_e^*(\boldsymbol{s})=0} = \frac{H_{11}(\boldsymbol{s}) + \Delta H(\boldsymbol{s})\, Z_e(\boldsymbol{s})}{1 + H_{22}(\boldsymbol{s})\, Z_e(\boldsymbol{s})} \text{ with } Z_e(\boldsymbol{s}) = \left.\frac{F_e(\boldsymbol{s})}{V_e(\boldsymbol{s})}\right|_{F_e^*(\boldsymbol{s})=0}. \tag{4.31}$$

This function, introduced in [4] and extended to the case with local force-controllers in [68], is often also referred to as transparency transfer function. It represents the impedance that is displayed to the operator when interacting with the haptic interface and consists of a combination of the environment impedance $Z_e(\boldsymbol{s})$ and distortion effects caused by the teleoperation system. For a perfectly transparent teleoperation system the relation

$$Z_{to}(\boldsymbol{s}) = Z_e(\boldsymbol{s}) \tag{4.32}$$

[13] All device-specific parameters in $\boldsymbol{q}_m$ and $\boldsymbol{q}_s$ are set to fixed values while intervals are considered for the parameters $\boldsymbol{q}_h$ characterizing the human impedance. For all other parameters we partly consider intervals and partly consider fixed values in the examples in Section 4.5.

must hold. This is only possible in theory and only for a system without time delay and actuator dynamics which incorporates noise-free acceleration measurement on master side and slave side [4, 68].

As *perfect* transparency is not possible the difference between environment impedance $Z_e(\boldsymbol{s})$ and impedance transmitted to the operator $Z_{to}(\boldsymbol{s})$ is examined in the following. This is a common approach to transparency analysis. However, generally only the extreme cases of free space and rigid contact are evaluated and Bode plots of the transmitted impedance are used to present the results, e.g., in [68]. This is neither very intuitive nor does it give insights into the transparency of the system for environment parameters *between* the two extreme cases. Therefore, we develop a graphical transparency analysis method in the following sections which produces intuitively interpretable results and allows a transparency analysis over a whole range of environment impedances.

4.4.2 System Identification

The method is based on identifying parameters of a low-order model which is used to approximate the transmitted impedance. The approximated transmitted impedance $\hat{Z}_{to}(\boldsymbol{s})$ is defined here as

$$\hat{Z}_{to}(\boldsymbol{s}, m_{to}, b_{to}, c_{to}) = \frac{m_{to}\boldsymbol{s}^2 + b_{to}\boldsymbol{s} + c_{to}}{\boldsymbol{s}} \tag{4.33}$$

where m_{to}, b_{to}, and c_{to} are the mass, viscous damping, and stiffness displayed to the operator[14]. The approximated transmitted impedance $\hat{Z}_{to}(\boldsymbol{s})$ is then identified from the exact transmitted impedance $Z_{to}(\boldsymbol{s})$ which can numerically be evaluated for a given environment impedance $Z_e(\boldsymbol{s})$. Therefore, the identification error Z_{error} defined as

$$Z_{\text{error}}(m_{to}, b_{to}, c_{to}) = \\ \frac{1}{N}\sum_{i=1}^{N}\left|\log_{10}(Z_{to}(j\omega[i]) - \log_{10}(\hat{Z}_{to}(j\omega[i], m_{to}, b_{to}, c_{to})\right|^2$$

is evaluated over a range of frequencies $\omega[i]$ with $i = 1 \ldots N$ and the optimization problem

$$\min_{m_{to}, b_{to}, c_{to}} \; Z_{\text{error}}(m_{to}, b_{to}, c_{to}) \tag{4.34}$$

is solved. The logarithmic least-squares minimization used here is, e.g., proposed in [82] to solve the numerical problems other estimators have in low gain areas of functions with large dynamic range. In [83] a comparison with other cost functions yields superior results compared to other deterministic approaches.

Various other, generally more complex methods exist to identify system models from frequency-response data. These methods often offer superior noise robustness in comparison with least-squares minimization which is however not of interest in our case: as the

[14] We choose this very simple model here as it is most easy to interpret. However, a slightly more complex model could also be used to more accurately approximate the actually transmitted impedance (e.g. two masses coupled by a spring-damper).

frequency-response is computed from a high-order model and does not originate from measured data, no noise robustness is necessary. Also, persistent excitation, which is an issue when identifying models from measured data, is easily assured here, as the identification is performed in the frequency domain over a large number of frequencies. The transparency analysis results in Section 4.5, e.g., are determined using $N = 500$ frequencies. This corresponds to identifying a system from measurement data, which is excited by 500 sine-waves of different frequency.

The system identification presented here, fits a second-order system to the frequency response of a system of higher order. Clearly, no perfect fit can be achieved over a large frequency range, which is however also not necessary in practice due to the limited range of haptic kinesthetic interaction as, e.g., determined in [84]. In addition to the frequency range, the weighting of different frequencies is of importance as a good fit at low frequencies (representative for damping and stiffness) is desired in our case. This is achieved here by a logarithmic spacing of the frequencies $\omega[i]$ as proposed e.g. in [82]. Therefore, $\omega[i]$ is defined as

$$\omega[i] = 10^{\log_{10}(\omega_-)+\frac{i-1}{n-1}(\log_{10}(\omega_+)-\log_{10}(\omega_-))}, \tag{4.35}$$

where ω_- and ω_+ are the lower and upper bound of the frequency range which are chosen as

$$\omega_- = 0.001\,\frac{1}{\mathrm{s}}\,, \quad \omega_+ = 20.0\,\frac{1}{\mathrm{s}} \tag{4.36}$$

here. Different frequency ranges result in slightly different identification results, especially for the identified inertia, as is shown in [61]. In that paper we experimentally evaluate the results determined for different frequency ranges. We conclude that an upper bound of $\omega_+ = 20.0$ results in superior identification results, for the hardware setup used in Section 4.5, in comparison with a larger upper bound of $\omega_+ = 200.0$, which would completely cover the range of haptic interaction given in [84].

It must be noted, that the numerical optimization used to solve the minimization problem is not necessarily convex and the algorithm could converge to a local minimum. A common strategy to prevent this is to start the minimization from various random values and choose the best result[15].

4.4.3 Graphical Analysis

After identifying the transmitted impedance over a range of environments the transparency of the teleoperation system can be examined graphically. Therefore, mass, damping, and stiffness of the approximated transmitted impedance $\hat{Z}_{to}(\boldsymbol{s})$ are plotted with respect to the parameters of the actual environment $Z_e(\boldsymbol{s})$. In general, a two-dimensional (spring-damper) or three-dimensional (mass-spring-damper) parameter space would have to be mapped to a three-dimensional parameter space which is difficult to analyze graphically.

[15] For the transparency analysis in Section 4.5 the parameter values of the environment impedance are chosen as initial values. The identification error is always within the expected range which makes local minima very unlikely. For the examples in [61] we evaluated other initial values over a large range which all lead to the same results. This suggests that the optimization problem is convex in a relatively large range around the global minimum.

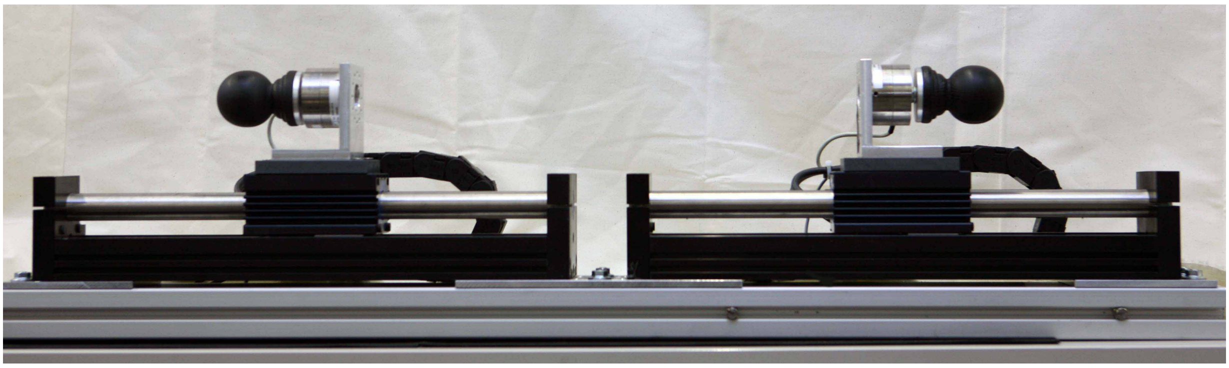


Figure 4.2: One-DoF linear axis teleoperation system.

Therefore, the influence of one parameter of the environment on all parameters of the transmitted impedance, is examined, and the fixed environment parameter is set to zero.

4.5 Examples

In this section robust stability and transparency of different teleoperation architectures is examined. Therefore, we first summarize the model parameters used for human, environment, haptic interface, and teleoperator. Then, the different teleoperation architectures are introduced, a robust stability analysis is performed, and transparency is examined for different time delays. As architectures some widely used classic teleoperation architectures are selected: the two channel force-position architecture (Section 4.5.2), the two-channel force-position architecture with wave-variables in the communication channel (Section 4.5.3), and the transparency optimized four-channel architecture (Section 4.5.4). In addition, a position-based admittance control with force-force exchange is considered which is widely used at our institute (Section 4.5.5).

For the first example, the two channel force-position architecture, the stability analysis and controller design is shown in great detail so as to thoroughly introduce the different possibilities our stability analysis method offers. Moreover, this example is used to illustrate the effect of sensor and actuator filters on stability. For the remaining examples the stability analysis is reduced to a few relevant plots and we only consider the complete system with sensor and actuator filters.

4.5.1 Model Parameters

The same model parameters are chosen for the different examples in this chapter. These correspond to an actual hardware setup which is depicted in Fig. 4.2 and used for basic research at the Chair of Automatic Control Engineering, TU München.

The setup consists of two identical one-DoF linear axis (*Copley Controls, Thrusttube 2504* equipped with force sensor *Burster, model 8524*). One of these linear axis is used as haptic interface while the other is used as teleoperator. The parameters of this teleoperation system are given in Table 4.1. The actuator time constants τ_m and τ_s are taken from the datasheet of the Thrusttube, while the force sensor time constants τ_h and τ_e are set to

Table 4.1: Parameters of haptic interface and teleoperator.

Parameter	Value	Unit	Description
m_m, m_s	2.386	kg	inertia
m_{em}, m_{es}	0.112	kg	end-effector inertia
b_m, b_s	20.0	Ns/m	damping
τ_m, τ_s	0.00065	s	actuator time constant
τ_h, τ_e	0.0032	s	force sensor time constant

Table 4.2: Parameters of human and environment.

Parameter	Lower bound	Upper bound	Unit	Description
m_h	0.1	5.0	kg	human impedance inertia
b_h	0.1	6.0	Ns/m	human impedance damping
c_h	0.1	40.0	N/m	human impedance stiffness
m_e	0.0	0.0	kg	environment inertia
b_e	0.0	100.0	Ns/m	environment damping
c_e	0.0	10.0	kN/m	environment stiffness

the value of a digital low-pass filter with a cut-off frequency of 50 Hz which is necessary to suppress the measurement noise of the force sensor. The inertia of the haptic interface and teleoperator as well as the end-effectors are measured, and the damping is chosen so as to represent the friction as closely as possible.

This admittance-type teleoperation system is suitable for interaction with very stiff environments. Therefore, a large range is considered for the environment stiffness and damping. To simplify the presentation of the results in the following sections we examine an interaction with a pure spring-damper environment and therefore set the environment inertia m_e to zero[16]. As the haptic interface can be firmly gripped by the human a large range is also considered for the stiffness, damping, and mass of the human[17]. These parameter-sets are summarized in Table 4.2.

[16] As we consider a stiff coupling between end-effector and environment the end-effector inertia can be considered as part of the environment inertia, so the *combined* inertia in the remote environment is m_{es}.

[17] Different values for the range of the impedance of the human arm can be found in different publications in literature, see, e.g., [80, 85]. We use the impedance range from [73] which is similar to [80] but considers a larger upper bound for the inertia. In contrast to both [73] and [80] a three-dimensional interval-box is considered for the different parameters of the human arm impedance, i.e., the value of the three parameters is independent, whereas in [73] and [80] the three parameters are set to a common factor α multiplied by different upper bounds, which restricts the impedance to one line within the three-dimensional interval-box.

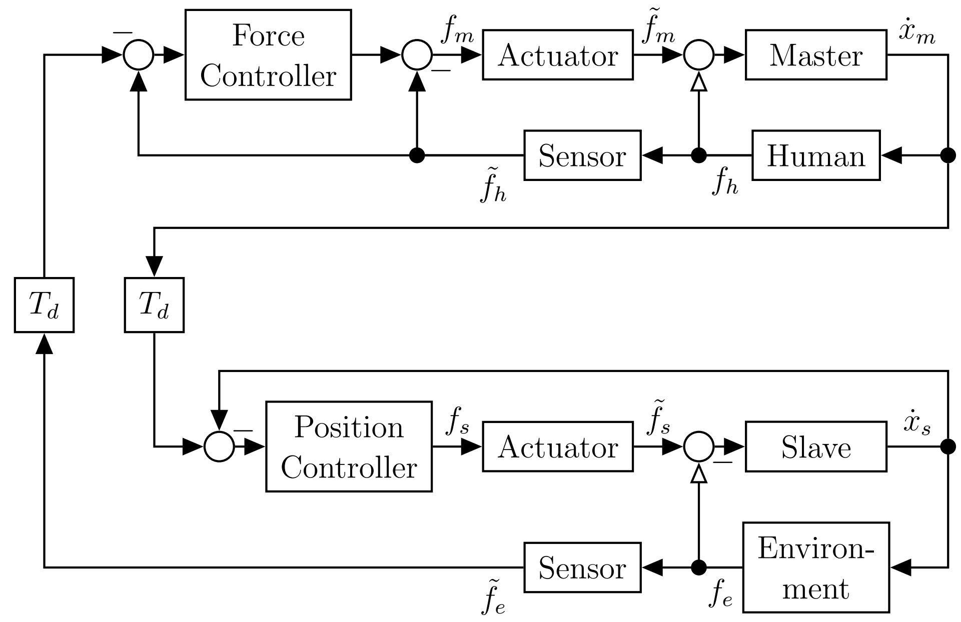


Figure 4.3: Force-position architecture

4.5.2 Force-Position Architecture (FP)

In this section a stability and transparency analysis is performed for a two-channel force-position architecture. First, the system model of the overall teleoperation system including human and environment is presented. Then, a stability analysis and controller design as well as a transparency analysis is carried out and results are discussed.

System Model

Within the force-position architecture forces are transmitted from the slave side to the master side while positions are transmitted from the master side to the slave side. A proportional force controller together with an external force compensation generates the desired actuator force on master side, i.e.

$$f_m = K_f(\tilde{f}_h - \tilde{f}_e e^{-T_d \boldsymbol{s}}) - \tilde{f}_h, \tag{4.37}$$

where K_f is the gain of the force controller and T_d is the one-way time delay between slave and master. A proportional-derivative (PD) position controller generates the desired actuator force on slave side, i.e.

$$f_s = K_p(x_m e^{-T_d \boldsymbol{s}} - x_s) + K_d(\dot{x}_m e^{-T_d \boldsymbol{s}} - \dot{x}_s), \tag{4.38}$$

where K_p and K_d are the proportional and derivative gain of the PD-controller and T_d is the one-way time delay between master and slave. The overall system is depicted in Fig. 4.3.

For stability analysis, a state-space representation of the system is determined as described in Section 4.3 by taking the models of human, environment, haptic interface, and teleoperator as well as the controller equations (4.37, 4.38) into account. This results in a system with system state $\boldsymbol{x} \in \mathbb{R}^8$ and parameter vector $\boldsymbol{q} \in \mathbb{R}^{19}$ given by

$$\boldsymbol{x} = [\dot{x}_m,\ x_m,\ \dot{x}_s,\ x_s,\ \tilde{f}_m,\ \tilde{f}_s,\ \tilde{f}_h,\ \tilde{f}_e]^T \tag{4.39}$$

$$\boldsymbol{q} = [\boldsymbol{q}_h^T,\ \boldsymbol{q}_e^T,\ \boldsymbol{q}_m^T,\ \boldsymbol{q}_s^T,\ K_p,\ K_d,\ K_f]^T \tag{4.40}$$

$$\tag{4.41}$$

where $\boldsymbol{q}_h$, $\boldsymbol{q}_e$, $\boldsymbol{q}_m$, and $\boldsymbol{q}_s$ are given in Section 4.3[18].

For transparency analysis using the method introduced in Section 4.4 the controllers $C_{1..6}$, C_m, and C_s of the four-channel architecture in Section 4.2.2 are determined from (4.37) and (4.38) as

$$C_s(\boldsymbol{s}) = G_s\,(K_p/\boldsymbol{s} + K_d) \tag{4.42}$$

$$C_1(\boldsymbol{s}) = G_s\,(K_p/\boldsymbol{s} + K_d) \tag{4.43}$$

$$C_2(\boldsymbol{s}) = G_m K_f G_e \tag{4.44}$$

$$C_6(\boldsymbol{s}) = G_m(K_f - 1)G_h \tag{4.45}$$

and $C_m(\boldsymbol{s}) = C_3(\boldsymbol{s}) = C_4(\boldsymbol{s}) = C_5(\boldsymbol{s}) = 0$. To simplify the transparency analysis the force sensor filters $G_h(\boldsymbol{s})$ and $G_e(\boldsymbol{s})$ and the filters $G_m(\boldsymbol{s})$ and $G_s(\boldsymbol{s})$ representing the actuator dynamics are all set to one, i.e., these filters are not considered in the transparency analysis. As the bandwidth used for system identification (see (4.36)) is well below the cut-off frequency of these filters, the effect of this simplification on the identification results is negligible.

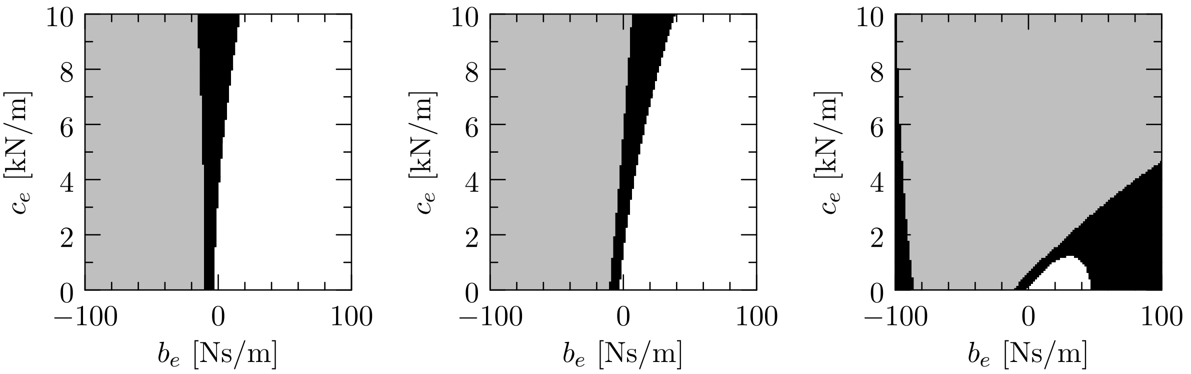
Stability Analysis Without Sensor and Actuator Filters

In this section a stability analysis and controller design is carried out for the simplified system without sensor and actuator filters. We carry out this analysis in great detail so as to demonstrate the methodology and possibilities which arise from our novel stability analysis method. For all results shown in this section, the impedance of the operator is set to the three-dimensional interval box given in Table 4.2, i.e., we consider a whole range of values for inertia, damping, and stiffness of the human arm.

We start out with a nominal parametrization which has been selected heuristically. The parameters selected are $K_p = 70$ kN/m, $K_d = 500$ Ns/m, $K_f = 5$. The resulting stability regions in the environment plane b_e/c_e for different time delays between 0 ms and 10 ms are depicted in Fig. 4.4. Actually we would like to design a controller which is stable for $b_e = [0,\ 100]$ Ns/m and $c_e = [0,\ 10]$ kN/m. Clearly, this is not the case for the chosen parametrization for any time delay. Moreover, the size of the stable parameter set shrinks drastically for larger time delay.

Therefore, we determine two sets of controller parameters which stabilize the system for different time delays. One set for a one-way time delay of 1 ms and one set for a one-way time delay of 10 ms. For comparison we also show stability analysis results for

[18] In case of the simplified system which does not take actuator and sensor dynamics into account the filter time-constants are set to $\tau_m = \tau_s = \tau_h = \tau_e = 0$ which allows reducing the system order from eight to four and the number of system parameters from 19 to 15



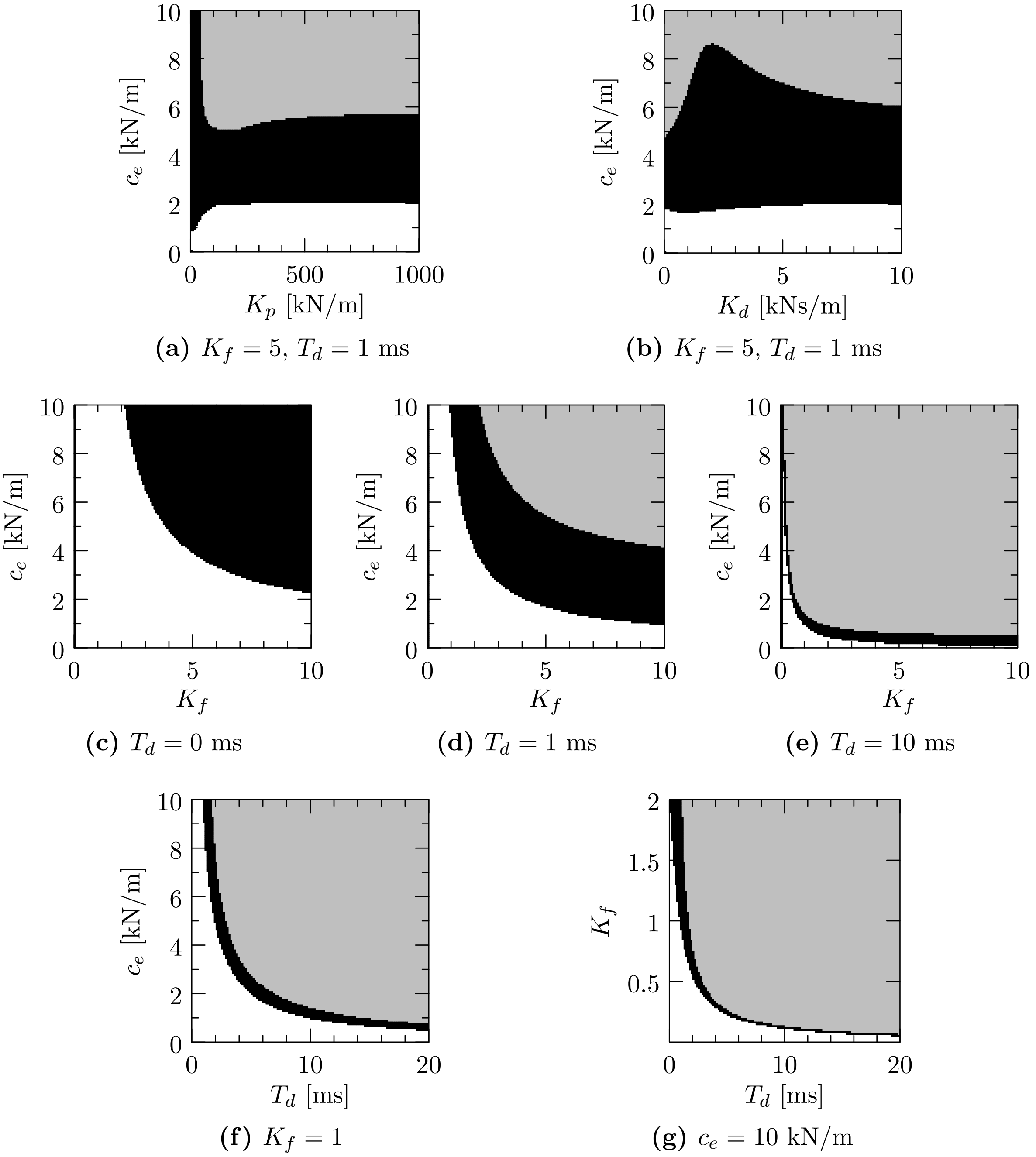
(a) $T_d = 0$ ms **(b)** $T_d = 1$ ms **(c)** $T_d = 10$ ms

Figure 4.4: Stability analysis of simplified force-position architecture for different time delays. Controller parameters are chosen heuristically. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both b_e and c_e.

the delay-free case. The corresponding plots are all given in Fig. 4.5 and are described in detail in the following. We start by examining the effect of different controller parameters on the stable region. Therefore, we set the environment damping b_e to zero and examine how changes to different controller parameters affect the stable region. This analysis (for $T_d = 1$ ms) is depicted in Fig. 4.5a to Fig. 4.5b for the position controller gains.

Clearly, changes of K_p and K_d over a large range only have a slight influence on the stable region (for $b_e = 0$ Ns/m). Therefore, these values are not further considered, and we instead examine stability in the K_f/c_e-plane for different time delays in Fig. 4.5c to Fig. 4.5e. The results can be summarized as follows: the smaller the force controller gain K_f, the larger the environment stiffness c_e can be, while still resulting in an asymptotically stable system. This relation holds for all time delays. However, with growing time delay the environment stiffness c_e which can be displayed for a given force controller gain K_f is reduced which is also shown in the following two plots. In Fig. 4.5f stability is analyzed in the T_d/c_e-plane for a fixed controller gain $K_f = 1$, while in Fig. 4.5g stability is analyzed in the T_d/K_f-plane for a fixed environment with damping $b_e = 0$ Ns/m and stiffness $c_e = 10$ kN/m.

The controller design based on Fig. 4.5 results in two different K_f-values for the two time delays 1 ms and 10 ms. For both of these values, a stability analysis in the environment plane is given in Fig. 4.6 for different time delays. For a one-way time delay of $T_d = 1$ ms the value $K_f = 1$ is chosen. The stability analysis in the environment plane (b_e/c_e) is depicted in Fig. 4.6a to Fig. 4.6c. For $T_d = 0$ ms and $T_d = 1$ ms the system is stable for the considered range of environment stiffness (0 kN/m $\leq c_e \leq$ 10 kN/m) and damping (0 Ns/m $\leq b_e \leq$ 100 Ns/m). However, for larger time delays only part of this environment range remains stable. For a one-way time delay of $T_d = 10$ ms the value $K_f = 0.1$ is chosen. The corresponding stability analysis is shown in Fig. 4.6d to Fig. 4.6f. Here, the complete plotted parameter range (including negative damping) is stable for $T_d = 0$ ms



(a) $K_f = 5$, $T_d = 1$ ms

(b) $K_f = 5$, $T_d = 1$ ms

(c) $T_d = 0$ ms

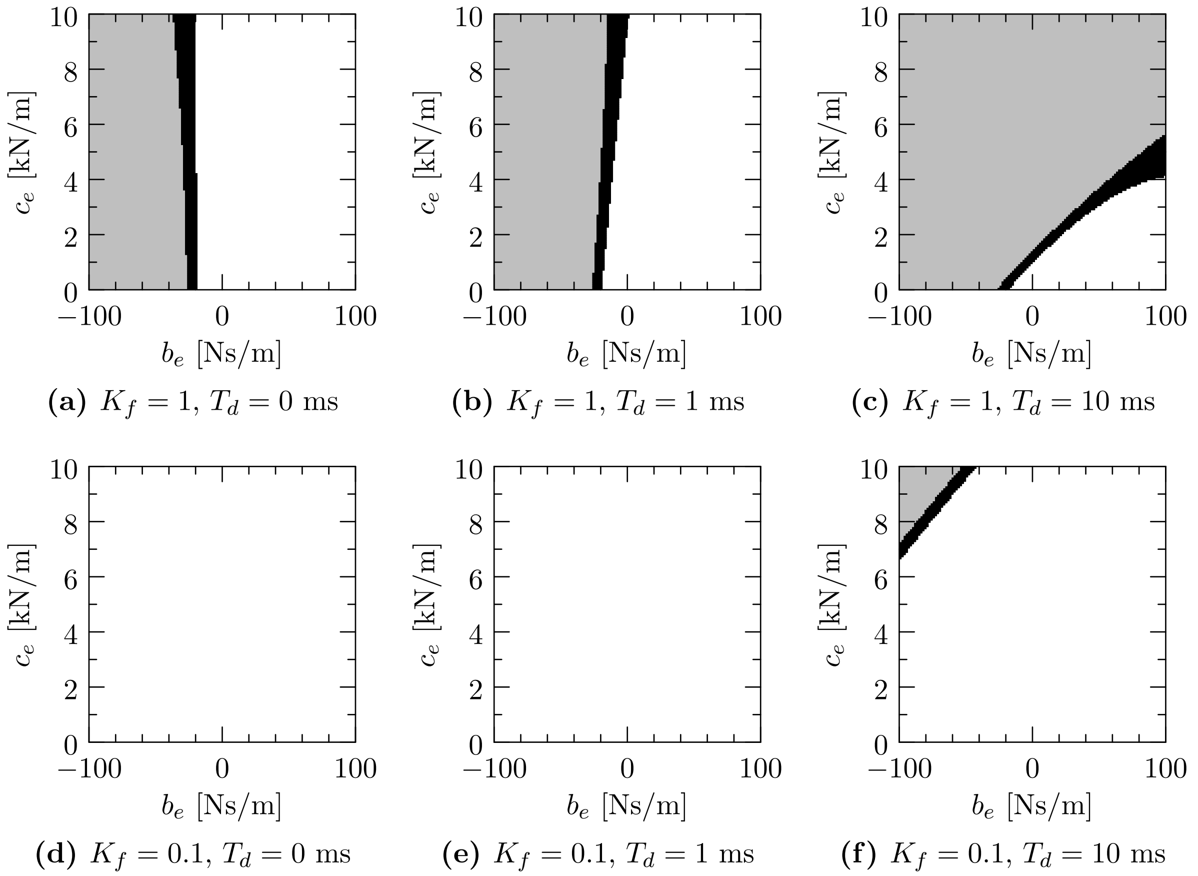
(d) $T_d = 1$ ms

(e) $T_d = 10$ ms

(f) $K_f = 1$

(g) $c_e = 10$ kN/m

Figure 4.5: Controller design for simplified force-position architecture. In the top row, stability with respect to environment stiffness c_e is examined for changes to the position controller for one time delay (1 ms). In the second row, stability is examined in the K_f/c_e-plane for different time delays. Finally, in the third row, the relation between time delay, force controller gain, and environment stiffness is examined. The parameters that are not varied or given in the caption of the subfigure are set to $b_e = 0$ Ns/m, $K_p = 70$ kN/m, and $K_d = 500$ Ns/m. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in each axis.



(a) $K_f = 1$, $T_d = 0$ ms **(b)** $K_f = 1$, $T_d = 1$ ms **(c)** $K_f = 1$, $T_d = 10$ ms

(d) $K_f = 0.1$, $T_d = 0$ ms **(e)** $K_f = 0.1$, $T_d = 1$ ms **(f)** $K_f = 0.1$, $T_d = 10$ ms

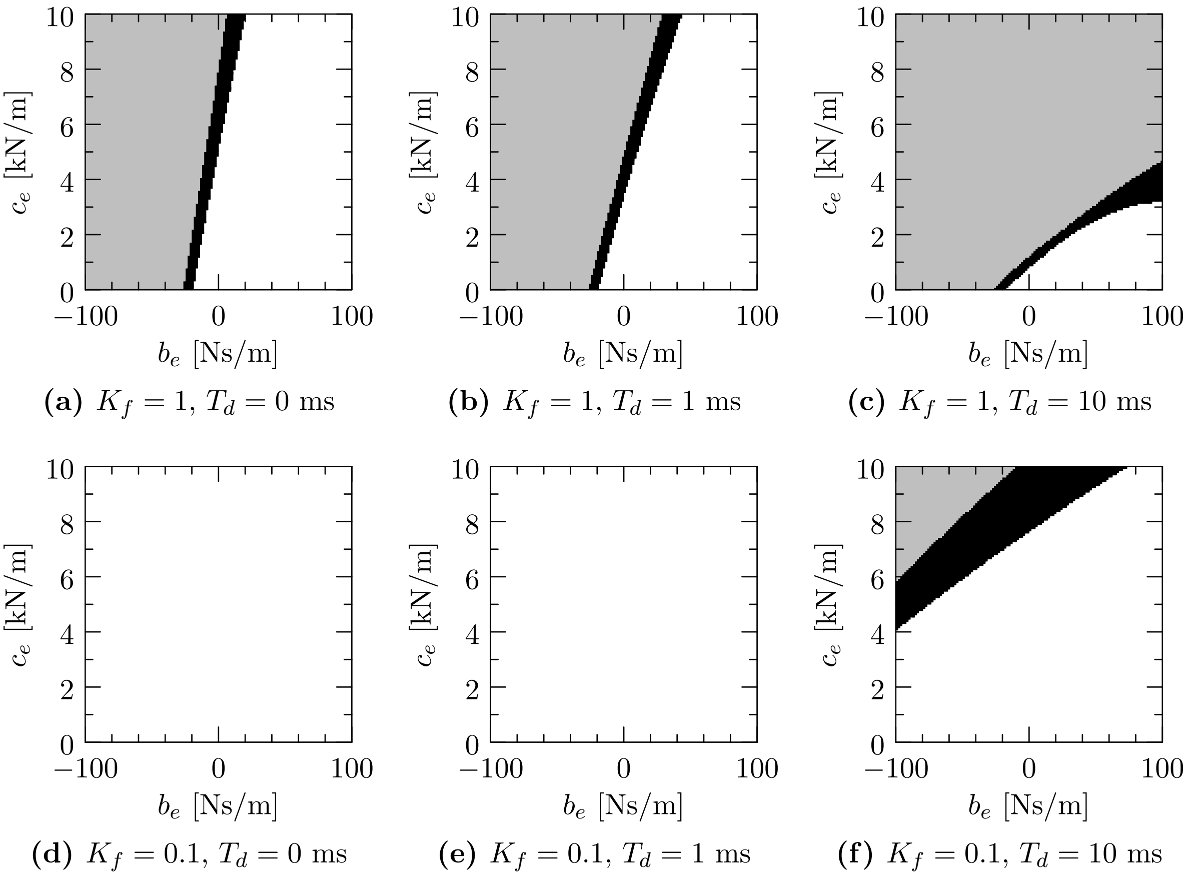
Figure 4.6: Stability analysis of simplified force-position architecture for $K_f = 1$ and $K_f = 0.1$ for different time delays. The remaining position controller parameters are $K_p = 70$ kN/m and $K_d = 500$ Ns/m. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both b_e and c_e.

and $T_d = 1$ ms. For $T_d = 10$ ms the system is stable for the desired environment range.

Stability Analysis With Sensor and Actuator Filters

The stability analysis in the previous section considered the simplified system without sensor and actuator filters. Here, the more complex system with sensor and actuator filters is considered. The same thorough analysis is carried out as for the simplified system so as to examine the effect of sensor and actuator filters on stability. For all results shown in this section, the impedance of the operator is set to the three-dimensional interval box given in Table 4.2, i.e., we consider a whole range of values for inertia, damping, and stiffness of the human arm.

We start out by examining stability using the final parametrization of the case without filters, i.e., using $K_p = 70$ kN/m, $K_d = 500$ Ns/m and two different values $K_f = 1$ and $K_f = 0.1$ for the force controller. The resulting stability analysis in the environment plane



(a) $K_f = 1$, $T_d = 0$ ms **(b)** $K_f = 1$, $T_d = 1$ ms **(c)** $K_f = 1$, $T_d = 10$ ms

(d) $K_f = 0.1$, $T_d = 0$ ms **(e)** $K_f = 0.1$, $T_d = 1$ ms **(f)** $K_f = 0.1$, $T_d = 10$ ms

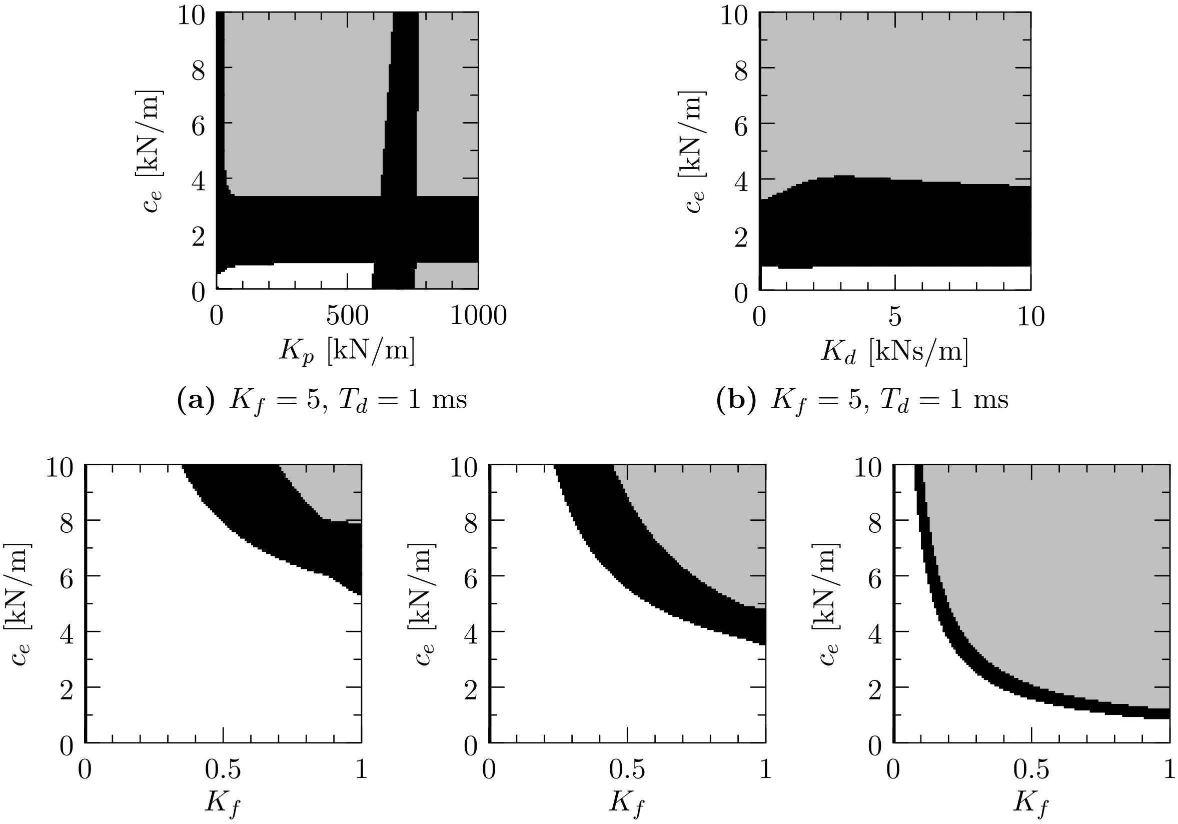
Figure 4.7: Stability analysis of force-position architecture with sensor and actuator filters for different time delays. Controller parameters are taken from the final results for the case without filters, see Fig. 4.6. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both b_e and c_e.

is depicted in Fig. 4.7 for different time delays. The size of the stable regions is considerably smaller in this case compared to the case without sensor and actuator filters in Fig. 4.6.

In the following, we again design controllers for different time delays which stabilizes the system for $b_e = [0,\ 100]$ Ns/m and $c_e = [0,\ 10]$ kN/m and show stability analysis results for the delay-free case for comparison. The corresponding plots are all given in Fig. 4.8 and are described in detail in the following. We start by examining the effect of different controller parameters on the stable region. Therefore, we set the environment damping b_e to zero and examine how changes to the position controller parameters affect stability. This analysis (for $T_d = 1$ ms) is depicted in Fig. 4.8a to Fig. 4.8b[19].

Although there is now an upper limit on K_p, in contrast to the case without filters, changes of K_p and K_d over a large range only have a slight influence on stability. Therefore,

[19] Note that we choose $K_f = 5$ here so as to allow a comparison with the results for the case without filters depicted in Fig. 4.5a to Fig. 4.5b.



(a) $K_f = 5$, $T_d = 1$ ms

(b) $K_f = 5$, $T_d = 1$ ms

(c) $T_d = 0$ ms

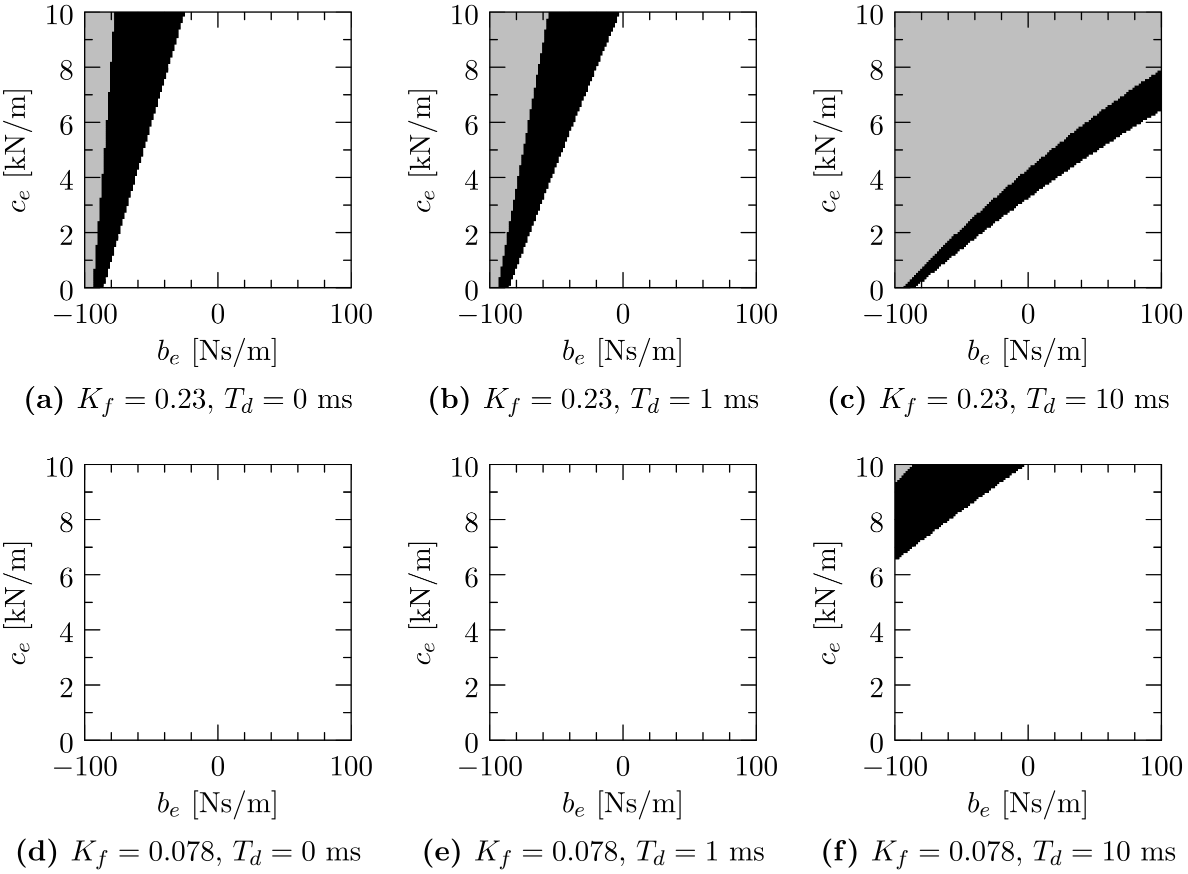
(d) $T_d = 1$ ms

(e) $T_d = 10$ ms

Figure 4.8: Controller design for force-position architecture with sensor and actuator filters. In the top row, stability with respect to environment stiffness c_e is examined for changes to the position controller parameters for one time delay (1 ms) while the second row considers changes of K_f for different time delays. The parameters that are not varied or given in the caption of the subfigure are set to $b_e = 0$ Ns/m, $K_p = 70$ kN/m, and $K_d = 500$ Ns/m. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in each axis.

these values are not further considered, and we instead examine stability in the K_f/c_e-plane for different time delays in Fig. 4.8c to Fig. 4.8e. The results are very similar to the case without filters and can be summarized as follows: the smaller the force controller gain K_f, the larger the environment stiffness c_e can be, while still resulting in an asymptotically stable system. This relation holds for all time delays and with growing time delay the environment stiffness c_e which can be displayed for a given force controller gain K_f is reduced. In contrast to the case without filters, the admissible controller gain K_f for a given time delay and environment stiffness is significantly smaller.

Based on these results we again choose two values for K_f which stabilize the system for the two time delays 1 ms and 10 ms and analyze stability in the environment plane in Fig. 4.9. For a one-way time delay of $T_d = 1$ ms the value $K_f = 0.23$ is chosen which is a much smaller value than is possible without filters ($K_f = 1$). The stability analysis



(a) $K_f = 0.23$, $T_d = 0$ ms **(b)** $K_f = 0.23$, $T_d = 1$ ms **(c)** $K_f = 0.23$, $T_d = 10$ ms

(d) $K_f = 0.078$, $T_d = 0$ ms **(e)** $K_f = 0.078$, $T_d = 1$ ms **(f)** $K_f = 0.078$, $T_d = 10$ ms

Figure 4.9: Stability analysis of force-position architecture with sensor and actuator filters for $K_f = 0.23$ and $K_f = 0.078$ for different time delays. The remaining position controller parameters are $K_p = 70$ kN/m and $K_d = 500$ Ns/m. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both b_e and c_e.

in the environment plane (b_e/c_e) is depicted in Fig. 4.9a to Fig. 4.9c. The system is stable for $T_d = 0$ ms and $T_d = 1$ ms in the considered range of environment stiffness (0 kN/m $\leq c_e \leq$ 10 kN/m) and damping (0 Ns/m $\leq b_e \leq$ 100 Ns/m). For larger time delays, only part of the considered environment range is stable. For a one-way time delay of $T_d = 10$ ms the value $K_f = 0.078$ is chosen which is slightly smaller than is possible without filters ($K_f = 0.1$). The corresponding stability analysis is shown in Fig. 4.9d to Fig. 4.9f. Here, the complete plotted parameter range (including negative damping) is stable for $T_d = 0$ ms and $T_d = 1$ ms while the system is stable for the desired environment range for $T_d = 10$ ms.

Transparency Analysis

In this section, the transparency of the force-position architecture is analyzed. Transparency is evaluated for a smaller range of environment stiffness than we considered for

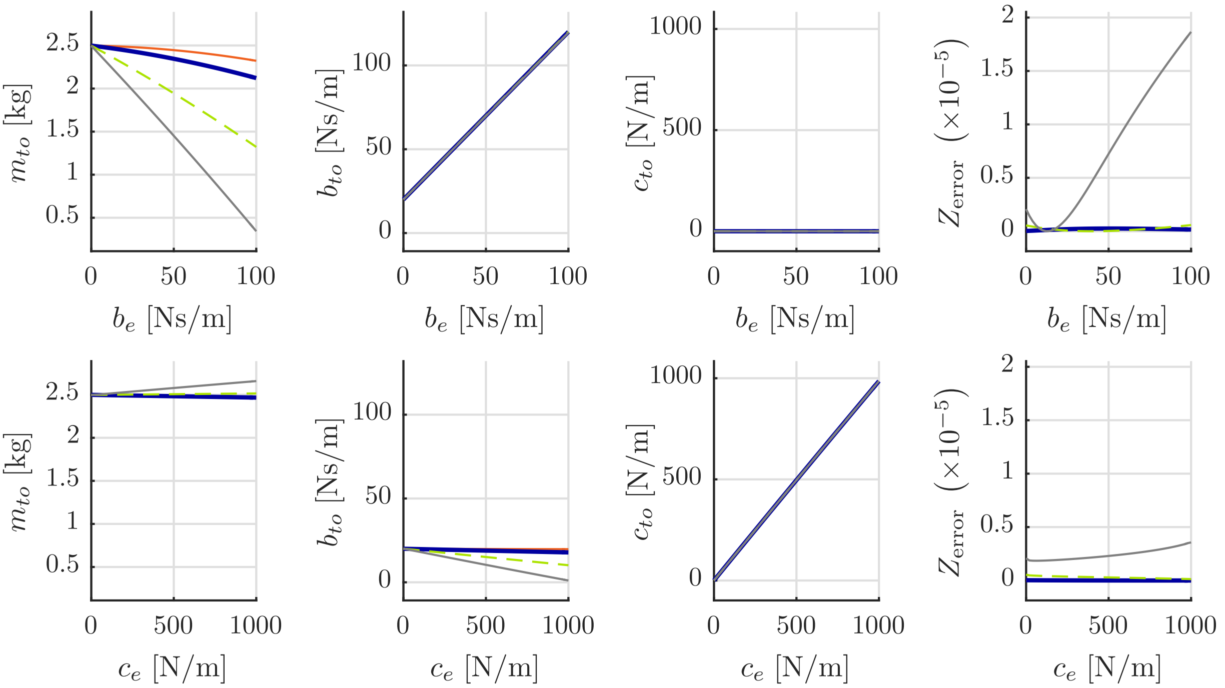


Figure 4.10: Transparency analysis of force-position architecture for $K_f = 1$ for different time delays (red, solid: no delay; blue, bold: $T_d = 1$ ms; green, dashed: $T_d = 5$ ms; gray, bold: $T_d = 10$ ms). System identification is performed for 51 values of the variable environment parameter.

the stability analysis, the rational being that an undistorted presentation of the environment impedance is of interest especially when interacting with soft environments or moving in free space. This is due to the fact that the absolute just noticeable difference of impedances increases with decreasing compliance according to *Weber's law* as has been experimentally shown for teleoperation systems, e.g., in [86]. As parameter sets we choose the same values as determined in the previous section. We select two different values for the force scaling K_f to demonstrate the effect.

First, we choose a value of $K_f = 1$. According to the previous section this is only stable for the complete environment range for our system in the case with no filters and a time delay of up to $T_d = 1$ ms. Even for the smaller environment range considered for transparency analysis the system with filters is only stable up to $T_d = 1$ ms, see Fig. 4.7. Nevertheless, the results in Fig. 4.10 show the transparency of the system for a whole range of different time delays ranging from 0 ms to 10 ms. For the case with no time delay the remote environment together with the device dynamics of the haptic interface are displayed to the operator[20]. For increasing time delay these results are only slightly impaired: for rising environment damping less of the inertia of the haptic interface is displayed while for rising environment stiffness less damping is displayed to the operator. It must be pointed out again that some of these results are only theoretically interesting as the closed-loop system including the operator is actually not stable for all considered time delays[21].

[20] Actually, the damping and stiffness are slightly reduced as they are limited due to the stiffness of the position control loop on slave side.

[21] As described in Section 4.4 the transparency analysis considers the system without the operator dy-

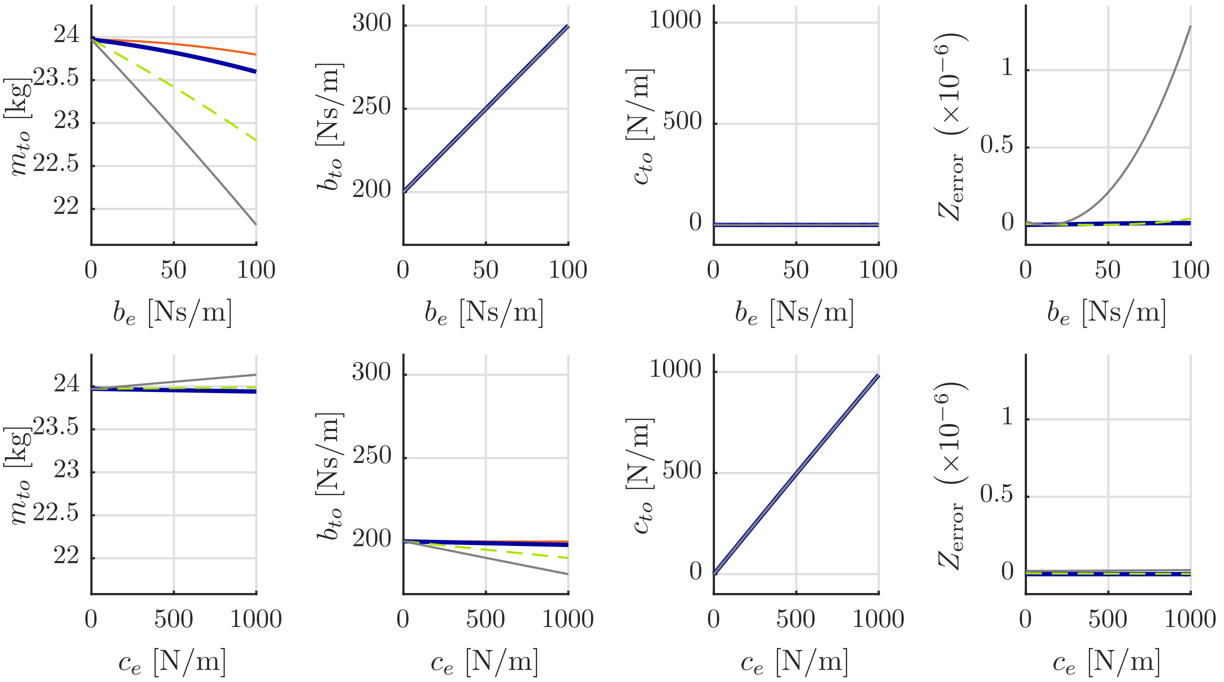


Figure 4.11: Transparency analysis of force-position architecture for $K_f = 0.1$ for different time delays (red, solid: no delay; blue, bold: $T_d = 1$ ms; green, dashed: $T_d = 5$ ms; gray, bold: $T_d = 10$ ms). System identification is performed for 51 values of the variable environment parameter.

Next, we consider a value of $K_f = 0.1$ which would allow for a stable interaction with environments up to a stiffness of $c_e = 10$ kN/m for a time delay of $T_d = 1$ ms and up to a stiffness of $c_e \approx 7$ kN/m for a time delay of $T_d = 10$ ms, see Fig. 4.7. The results are shown in Fig. 4.11. Using this parametrization the inertia and damping of the haptic interface are amplified by a factor of 10. Apart from this effect, the transparency properties are identical to the case with $K_f = 1$. A rising environment damping results in nearly the same rise of displayed environment damping and the environment stiffness and the stiffness displayed to the operator are nearly identical.

Discussion

In this section we carried out a stability analysis for a force-position architecture in great detail, which demonstrates the possibilities our method offers and shows how it can be used for controller design. The stability analysis was done once for a system without sensor and actuator filters and once for a system with sensor and actuator filters. The position controller gains K_p and K_d only have little effect on stability[22], although there is an upper bound on the value of K_p for the case with sensor and actuator filters. The limiting factor for stability is the force controller gain K_f which must be chosen appropriately. The maximum admissible range of K_f is limited by the time delay T_d: larger time delay results

namics.

[22] In practice, there is always an upper limit on controller gains due to the implementation on a digital controller with limited sampling rate!

in a smaller upper limit for the force controller gain. In addition, for the case with filters a smaller gain K_f must be chosen.

For the delay-free case, the stable regions in the environment plane are similar to previous results in [72], where stability of a different admittance-type teleoperation system was analyzed. The general *shape* of the stable regions in the environment plane is very similar, although the numeric values differ due to the different teleoperation system and the fact that the ratio between the three parameters of the human arm impedance was fixed in [72]. Moreover, the effect of the sensor filters on stability is comparable.

The transparency analysis yields the following results: Depending on the gain K_f of the force controller the impedance of the haptic interface is more or less perceivable. A larger gain K_f results in a more transparent system, while a smaller gain K_f results in a less transparent system. The perceived device dynamics are even amplified for $K_f < 1$[23], e.g., by a factor of 10 for $K_f = 0.1$. Another effect which is not shown here but becomes clear when analyzing transparency for a larger environment range is that the displayable impedance is limited by the position-controller on slave side: actually a series connection of the spring-damper of the PD-controller with the damped inertia of the teleoperator, coupled to the environment, is displayed to the operator.

When considering a fixed controller parametrization, the effect of time delay on transparency is actually rather small for this architecture. For large time delay, environment damping leads to a reduction of the displayed inertia while environment stiffness leads to a reduction of the displayed damping. These effects are however not very pronounced. Especially in the case of rising environment damping this damping probably masks the perception of the change in inertia.

Summarizing these results, the main impact of time delay for this architecture is a reduced upper limit of the force controller gain K_f which, in turn, leads to an amplification of the perceived dynamics of the haptic interface. Therefore, this architecture is suitable only for very small time delays. The instability due to time delay, which requires a reduction of the force controller gain, is caused by the non-passive nature of the communication channel. A common solution to this problem is to passivate the communication channel by using wave variables which is examined in the following section.

4.5.3 Force-Position Architecture with Wave Variables (FP-W)

In this section a stability and transparency analysis is performed for a two-channel force-position architecture with wave variables in the communication channel. First, the system model of the overall teleoperation system including human and environment is presented. Then, the different analysis steps are carried out and results are discussed.

System Model

In this architecture, in contrast to the simple force-position architecture (see 4.5.2), wave variables are used to transmit information from master side to slave side and vice versa. Thereby, passivity of the communication channel is assured at the cost of a *softer* system.

[23] Here, we only show results for $K_f = 1$ and $K_f = 0.1$ but it can be shown that this holds for all $K_f < 1$.

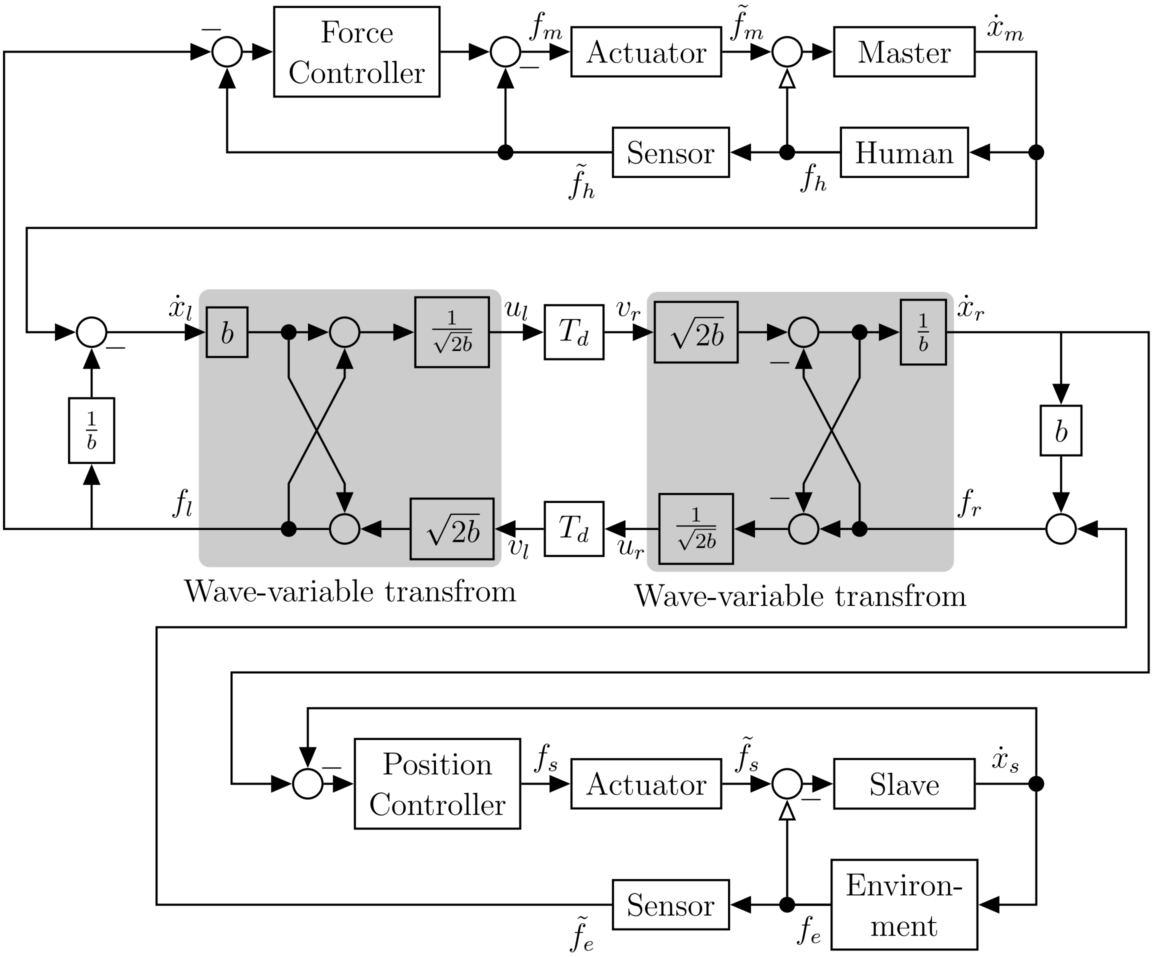


Figure 4.12: Force-position architecture with wave variables in the communication channel

This approach was first introduced in [64]. In contrast to [64] we consider a system with force sensors on master and slave side. This is necessary due to the fact that we consider an admittance-type haptic interface and an admittance-type teleoperator. If this setup were used without force sensors this would lead to a very poor performance due to the large device dynamics and friction effects. Unfortunately, wave variables do not necessarily ensure passivity when used in combination with a feedback of the measured environment force, see, e.g., [87].

The overall system is depicted in Fig. 4.12. The communication channel in the wave domain is described by the four equations

$$u_l = \frac{f_l + b\dot{x}_l}{\sqrt{2b}} \tag{4.46}$$

$$v_r = e^{-T_d \boldsymbol{s}} u_l = \frac{f_r + b\dot{x}_r}{\sqrt{2b}} \tag{4.47}$$

$$u_r = \frac{f_r - b\dot{x}_r}{\sqrt{2b}} \tag{4.48}$$

$$v_l = e^{-T_d \boldsymbol{s}} u_r = \frac{f_l - b\dot{x}_l}{\sqrt{2b}} \tag{4.49}$$

where f_l and f_r are the forces on master side and slave side, $\dot{x}_l$ and $\dot{x}_r$ are the velocities on master side and slave side, u_l,v_r,u_r,v_l are the four wave variables, and b is the wave damping. Communication using wave variables generally leads to wave reflections which result in vibrations on master and slave side. These wave reflections can be of a large magnitude and render the teleoperation system unusable. Therefore, different approaches to reduce this effect have been investigated, e.g., in [64, 88]. Here, we add termination elements to either side of the communication channel to match the impedance of the channel and cancel all wave reflections. This is achieved by setting

$$f_r = \tilde{f}_e + b\dot{x}_r \tag{4.50}$$

$$\dot{x}_l = \dot{x}_m - \frac{1}{b} f_l \tag{4.51}$$

as is first discussed in [64]. In this case, the communication channel with termination described by equations (4.46) to (4.51) can be simplified to

$$f_l = \frac{1}{2}\tilde{f}_e e^{-T_d \boldsymbol{s}} + \frac{b}{2}\dot{x}_m \tag{4.52}$$

$$\dot{x}_r = \frac{1}{2}\dot{x}_m e^{-T_d \boldsymbol{s}} - \frac{1}{2b}\tilde{f}_e \tag{4.53}$$

as shown in [64][24]. The actual controllers are then modified to use the transmitted velocity and force, i.e.

$$f_m = K_f(\tilde{f}_h - f_l) - \tilde{f}_h, \tag{4.54}$$

where K_f is the gain of the force controller. A proportional-derivative (PD) position controller generates the desired actuator force on slave side, i.e.

$$f_s = K_p(x_r - x_s) + K_d(\dot{x}_r - \dot{x}_s), \tag{4.55}$$

where K_p and K_d are the proportional and derivative gain of the PD-controller and x_r is the integral of the transmitted master velocity $\dot{x}_r$.

For stability analysis, a state-space representation of the system is determined as described in Section 4.3 by taking the models of human, environment, haptic interface, and teleoperator as well as the controller equations (4.52 - 4.55) into account. This results in a system with system state $\boldsymbol{x} \in \mathbb{R}^9$ and parameter vector $\boldsymbol{q} \in \mathbb{R}^{20}$ given by

$$\boldsymbol{x} = [\dot{x}_m,\ x_m,\ \dot{x}_s,\ x_s,\ \tilde{f}_m,\ \tilde{f}_s,\ \tilde{f}_h,\ \tilde{f}_e,\ x_r]^T \tag{4.56}$$

$$\boldsymbol{q} = [\boldsymbol{q}_h^T,\ \boldsymbol{q}_e^T,\ \boldsymbol{q}_m^T,\ \boldsymbol{q}_s^T,\ K_p,\ K_d,\ K_f,\ b]^T \tag{4.57}$$

$$\tag{4.58}$$

where $\boldsymbol{q}_h$, $\boldsymbol{q}_e$, $\boldsymbol{q}_m$, and $\boldsymbol{q}_s$ are given in Section 4.3. Note, that due to the wave variables an additional integrator must be added on slave side resulting in an additional system state in comparison with the case without wave variables. In addition, the stability analysis is more involved due to the more complex interconnection.

[24] It must be noted that adding the termination elements to the communication channel introduces scaling and results in a significant position drift [64]. Different approaches to compensate for a position drift exist, see, e.g., [5] for an overview. Moreover, it is also possible to prevent wave reflections completely, at least for the case of free-space motion, without adding scaling or significant drift by using impedance controllers as shown in [88].

For transparency analysis using the method introduced in Section 4.4 the controllers $C_{1..6}$, C_m, and C_s of the four-channel architecture in Section 4.2.2 must be determined from equations (4.52 - 4.55) which results in

$$C_m(\boldsymbol{s}) = \frac{b}{2} \tag{4.59}$$

$$C_s(\boldsymbol{s}) = G_s\left(K_p/\boldsymbol{s} + K_d\right) \tag{4.60}$$

$$C_1(\boldsymbol{s}) = \frac{1}{2}G_s\left(K_p/\boldsymbol{s} + K_d\right) \tag{4.61}$$

$$C_2(\boldsymbol{s}) = \frac{1}{2}G_m K_f G_e \tag{4.62}$$

$$C_5(\boldsymbol{s}) = \frac{1}{2b} \tag{4.63}$$

$$C_6(\boldsymbol{s}) = G_m(K_f - 1)G_h \tag{4.64}$$

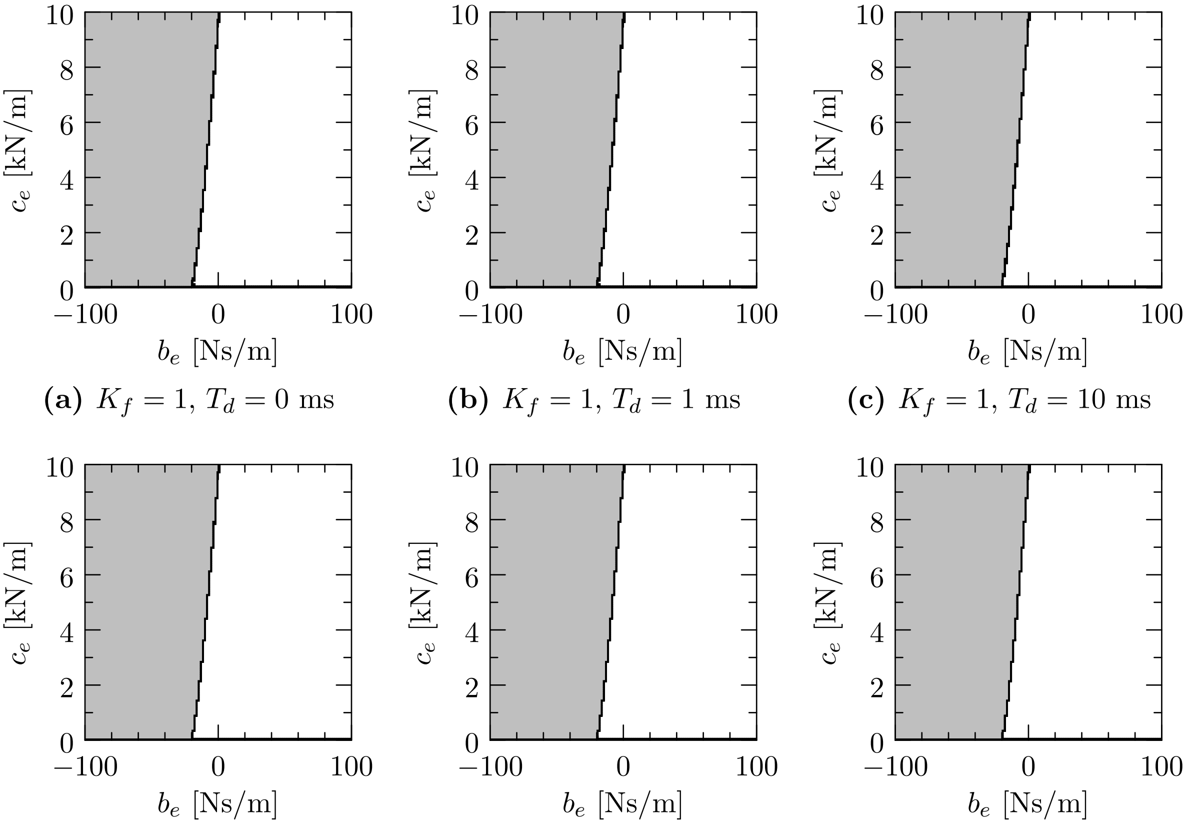
and $C_3(\boldsymbol{s}) = C_4(\boldsymbol{s}) = 0$. To simplify the transparency analysis the force sensor filters $G_h(\boldsymbol{s})$ and $G_e(\boldsymbol{s})$ and the filters $G_m(\boldsymbol{s})$ and $G_s(\boldsymbol{s})$ representing the actuator dynamics are all set to one, i.e., these filters are not considered in the transparency analysis. As the bandwidth used for system identification (see (4.36)) is well below the cut-off frequency of these filters, the effect of this simplification on the identification results is negligible.

Stability Analysis

In this section we consider the force-position architecture with wave variables in the communication channel. Our goal is to determine the effect of wave variables on stability, as it is well known that the overall system must not necessarily be stable for the case with environment force feedback. Therefore, the same parameters are considered here as for the case with sensor and actuator filters in the previous section. For all results, the impedance of the operator is set to the three-dimensional interval box given in Table 4.2, i.e., we consider a whole range of values for inertia, damping, and stiffness of the human arm.

We start out with the parametrization $K_p = 70$ kN/m, $K_d = 500$ Ns/m, and two different values $K_f = 1$ and $K_f = 0.1$ for the force controller gain. As value for the wave damping we choose $b = 10$ Ns/m. The resulting stability analysis in the environment plane is depicted in Fig. 4.13 for different time delays. As we expect from a system with wave variables in the communication channel, the time delay does not have any significant effect on stability. In this case, the force controller gain does not have any significant effect on stability either which we did not expect.

In all six cases, the system is stable for the most part of the environment range of interest (0 kN/m $\leq c_e \leq$ 10 kN/m and 0 Ns/m $\leq b_e \leq$ 100 Ns/m) with the exception of a small region with small environment damping and large stiffness. Therefore, we pursue a controller parametrization in Fig. 4.14 which stabilizes the complete region of interest. The admissible range of environment stiffness for $b_e = 0\ Ns/m$ is examined for different controller parametrizations in Fig. 4.14a to Fig. 4.14c. It turns out that we can stabilize the complete environment range of interest by slightly increasing one of the position controller gains K_p or K_d. In contrast, the force controller gain K_f only has little effect.



(a) $K_f = 1$, $T_d = 0$ ms **(b)** $K_f = 1$, $T_d = 1$ ms **(c)** $K_f = 1$, $T_d = 10$ ms

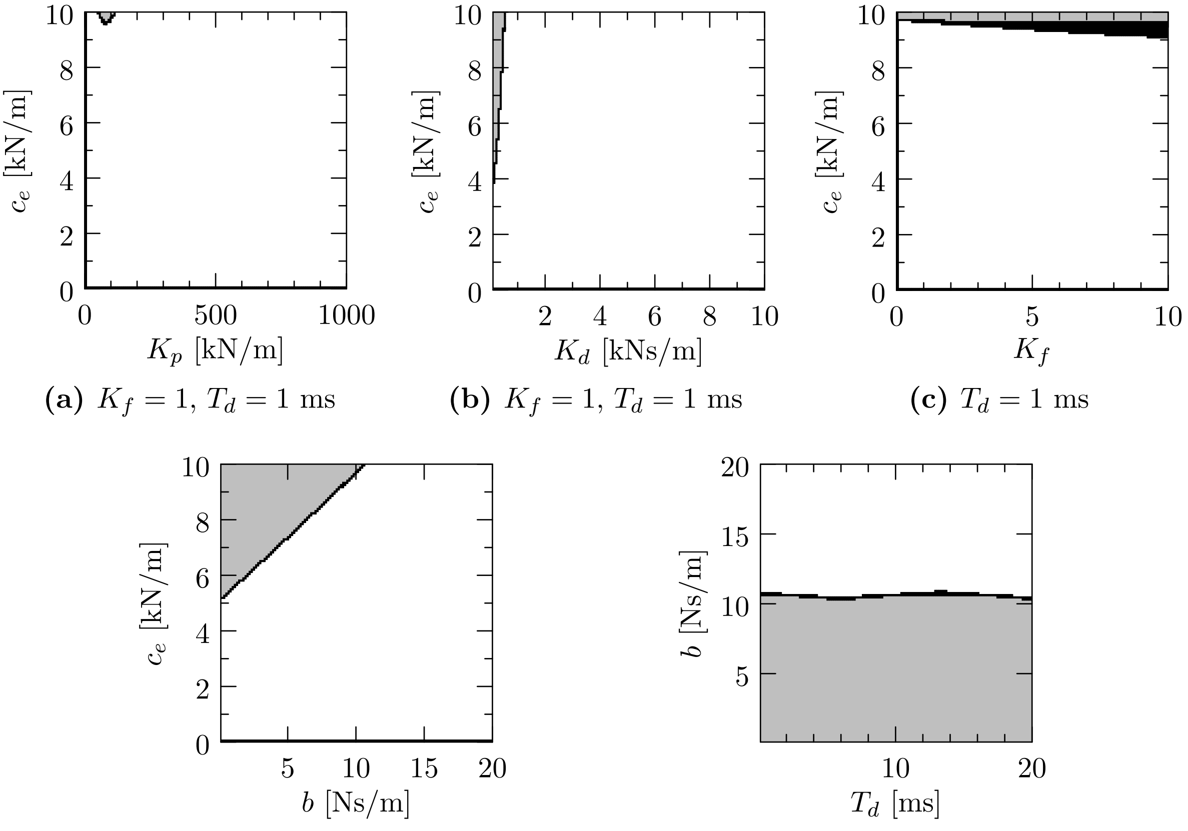
(d) $K_f = 0.1$, $T_d = 0$ ms **(e)** $K_f = 0.1$, $T_d = 1$ ms **(f)** $K_f = 0.1$, $T_d = 10$ ms

Figure 4.13: Stability analysis of force-position architecture with wave variables for different time delays. Controller parameters are taken from the case without wave variables, see Fig. 4.7. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both b_e and c_e.

Alternatively, we can also slightly increase the wave damping b so as to stabilize the complete environment range of interest as is shown in Fig. 4.14d and Fig. 4.14e. This is due to the impedance matching elements which introduce an additional damping proportional to b[25]. These results indicate that by, e.g., choosing $b = 15$ Ns/m the system is stable for 0 kN/m $\leq c_e \leq$ 10 kN/m and $b_e = 0$ Ns/m for a time delay of $T_d = 10$ ms and that for all time delays up to $T_d = 20$ ms the system is stable for $c_e = 10$ kN/m and $b_e = 0$ Ns/m.

Finally, we verify that for $b = 15$ Ns/m the complete environment range of interest is stable for all considered time delays in Fig. 4.15.

[25] If wave variables are used without impedance matching, i.e., the wave damping b is only present in the wave transformations of the communication channel, wave reflections are influenced by the wave damping b but not the stability of the system [64].



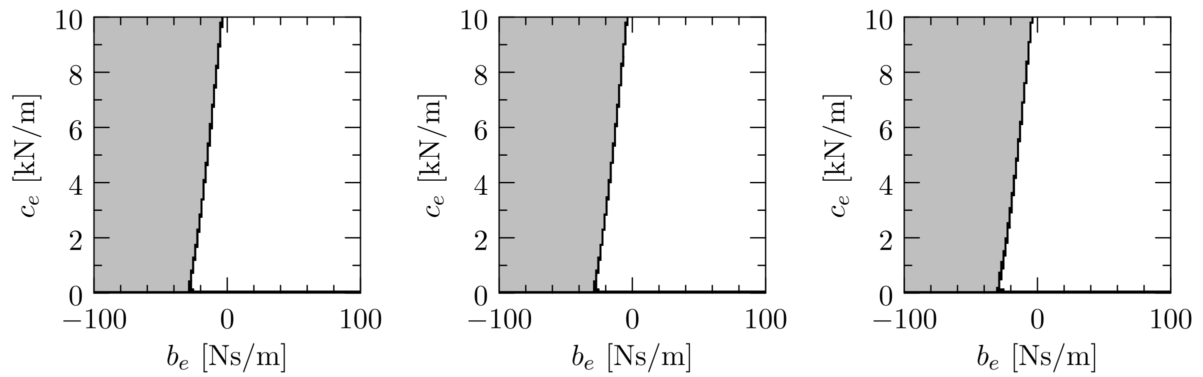
(a) $K_f = 1$, $T_d = 1$ ms **(b)** $K_f = 1$, $T_d = 1$ ms **(c)** $T_d = 1$ ms

(d) $K_f = 1$, $T_d = 10$ ms **(e)** $K_f = 1$, $c_e = 10$ kN/m

Figure 4.14: Controller design for force-position architecture with wave variables. In the top row, stability with respect to environment stiffness c_e is examined for changes to different controller parameters for one time delay (1 ms). In the second row, stability with respect to the wave damping b is examined. The parameters that are not varied or given in the caption of the subfigure are set to $b = 10$ Ns/m, $b_e = 0$ Ns/m, $K_p = 70$ kN/m, and $K_d = 500$ Ns/m. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in each axis.

Transparency Analysis

The parameters $b = 15$ Ns/m, $K_f = 1$, $K_p = 70$ kN/m, and $K_d = 500$ Ns/m for which the stability analysis is shown in Fig. 4.15 are also used for transparency analysis. The results of the transparency analysis in the parameter space are depicted in Fig. 4.16. Two main effects with respect to the distortion of the remote environment are observable in this case. On the one hand, there is a linear relationship between environment damping and damping displayed to the operator, as well as environment stiffness and stiffness displayed to the operator. A little *crosstalk* is observable, i.e., environment stiffness has an effect on displayed damping and environment damping has an effect on displayed inertia. However, these effects are much less pronounced than for the force-position architecture in Fig. 4.10



(a) $K_f = 1$, $T_d = 0$ ms (b) $K_f = 1$, $T_d = 1$ ms (c) $K_f = 1$, $T_d = 10$ ms

Figure 4.15: Stability analysis of force-position architecture with wave variables for $b = 15$ Ns/m, $K_f = 1$, $K_p = 70$ kN/m, and $K_d = 500$ Ns/m for different time delays. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both b_e and c_e.

or Fig. 4.11. On the other hand, there is a scaling of the environment dynamics by a factor of 1/4, e.g., for an environment stiffness of 1 kN/m a stiffness of 250 N/m is displayed.

Apart from these two effects there is a *nominal* impedance which is displayed in free space which consists of the inertia of the haptic interface and a significant damping which is larger than the inherent damping of the haptic interface. This *nominal* impedance is independent of the time delay.

Discussion

In this section we performed a stability and transparency analysis of a two-channel force-position architecture with wave variables in the communication channel which takes sensor and actuator dynamics into account. Wave reflections were prevented by adding termination elements on either side of the communication channel as described by *Niemeyer and Slotin* in [64]. This so-called impedance matching suppresses *all* wave reflections but also introduces significant scaling and position drift into the system as shown in [64].

The main results of our stability and transparency analysis are as follows. By introducing wave variables, stability of the system is not anymore affected by the amount of time delay, at least for the considered range of time delays, and the system can easily be stabilized for all considered time delays by appropriately choosing the controller parameters. Moreover, the termination elements which are added on either side of the communication channel introduce additional damping into the system. Therefore, the value of the wave impedance b has an effect on stability. This is not generally mentioned in the context of wave variables as it is assumed that the human, master, slave, and environment are passive and it is therefore only necessary to passivate the communication channel. In our case this assumption is not true due to the sensor and actuator filter which are necessary to accurately model admittance-type devices.

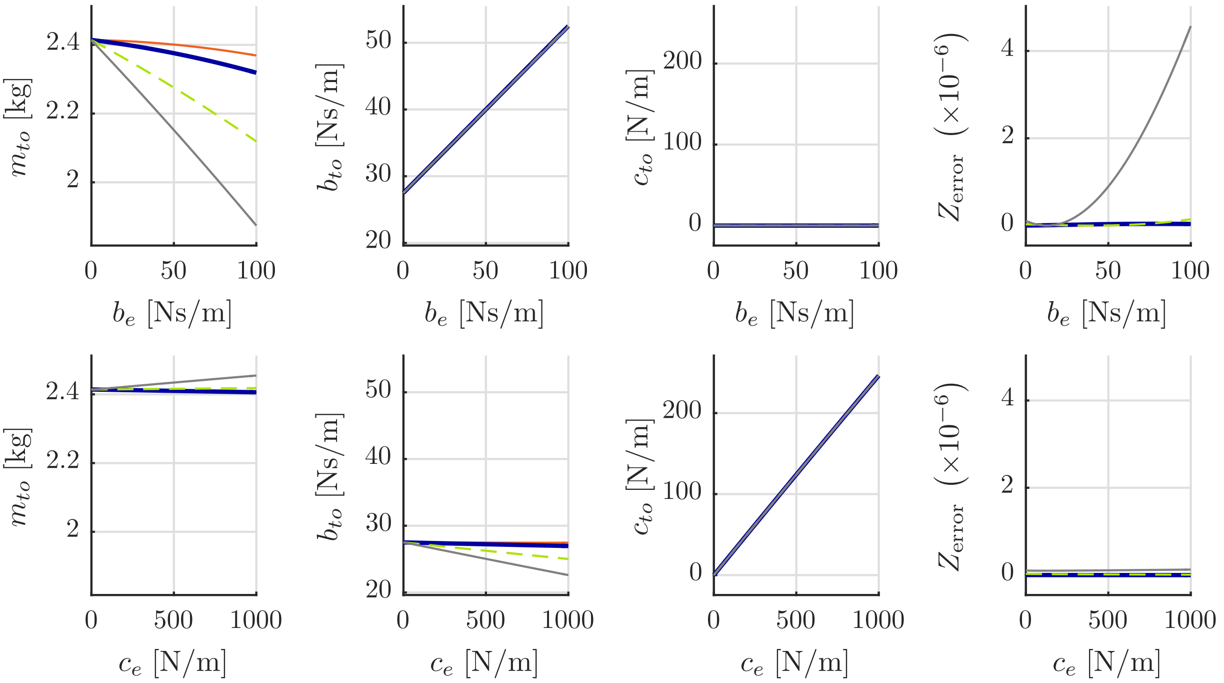


Figure 4.16: Transparency analysis of force-position architecture with wave variables and matched impedance on master and slave side for different time delays (red, solid: no delay; blue, bold: $T_d = 1$ ms; green, dashed: $T_d = 5$ ms; gray, bold: $T_d = 10$ ms). System identification is performed for 51 values of the variable environment parameter.

The transparency analysis considered controller parameters that stabilize the system for the environment range of interest. Several results were observable: On the one hand, the inertia of the haptic interface and significant damping is displayed to the operator. The damping is larger than the damping inherent to the haptic interface but a lot smaller than the damping which is displayed if $K_f = 0.1$ is used in the force-position architecture without wave variables, as is necessary for all but the smallest time delays, see Section 4.5.2. On the other hand, the scaling of forces transmitted from slave to master (by a factor of 1/2) and velocities transmitted from master to slave (also by a factor of 1/2) results in a scaling of the displayed impedance in comparison with the environment impedance by a factor of 1/4. These scaling effects are the main problem when considering this simple form of impedance matching, especially the scaling of the transmitted velocity as it leads to a significant position drift between master and slave device. Therefore, several alternative wave-variable-based architectures have been explored in literature, e.g., in [64, 88–92]. Examining stability and transparency of these different approaches goes beyond the scope of this thesis.

Summarizing these results, although wave variables mitigate the destabilizing effect of time delay they are not applicable in practice in this simple form due to the scaling effects. Instead, one of the more complex approaches based on wave variables could be selected. Alternatively, we examine in the following sections whether the transparency-optimized four-channel architecture can be tuned so as to provide stable results for small time delays or whether a position-based admittance controller with force exchange is suitable in this case.

4.5.4 Four-Channel Architecture (4C)

In this section the transparency-optimized four-channel architecture is examined which was first introduced in [4]. In this architecture, forces as well as velocities are transmitted from master to slave and vice versa. In theory this architecture results in an ideally transparent teleoperation system which is however only marginally absolutely stable, even for the case with no delays in the communication channel [68]. Therefore, even small delays or filters (e.g., force sensor filters) can impair stability, see [70].

In the rest of this section, stability and transparency of the transparency-optimized four-channel architecture is analyzed with respect to different values of time delay. First, the system model is presented in the following section. Then, the robust stability analysis is performed, transparency is evaluated for stable parameter sets, and the results are discussed.

System Model

The model of the general four-channel architecture as well as the resulting equations are introduced in Section 4.2.2. To arrive at a transparency-optimized four-channel architecture the controllers $C_{1..6}$, C_m, and C_s must be chosen. The local controllers C_5, C_6, C_m, and C_s can be chosen quite freely. The controllers $C_{1..4}$ in the communication channel must then be selected based on these local controllers so as to assure optimal transparency.

First, local force controllers are chosen. As the system is symmetric we choose the same parametrization for both sides. We select a simple P-controller with compensation of local force which is also used on master side for the FP-architecture in Section 4.5.2, i.e.,

$$C_5(\boldsymbol{s}) = G_s(K_f - 1)G_e \tag{4.65}$$

$$C_6(\boldsymbol{s}) = G_m(K_f - 1)G_h \tag{4.66}$$

where K_f is the gain of the force controller, the force sensor filter $G_h(\boldsymbol{s})$ on master side is given in (4.13), the force sensor filter $G_e(\boldsymbol{s})$ on slave side is given in (4.17), the actuator force filter $G_m(\boldsymbol{s})$ on master side is given in (4.21), and the actuator force filter G_s on slave side is given in (4.25).

Then, local position controllers for haptic interface and teleoperator are chosen. We select the PD-controller from Section 4.5.2 with the same parametrization for both sides, i.e.,

$$C_m(\boldsymbol{s}) = G_m\left(K_p/\boldsymbol{s} + K_d\right) \tag{4.67}$$

$$C_s(\boldsymbol{s}) = G_s\left(K_p/\boldsymbol{s} + K_d\right) \tag{4.68}$$

where K_p and K_d are the proportional and derivative gain of the PD-controller.

The rest of the controllers in Fig. 4.1 should then be selected according to

$$C_1(\boldsymbol{s}) = C_s(\boldsymbol{s}) + Z_s(\boldsymbol{s}) \tag{4.69}$$

$$C_2(\boldsymbol{s}) = 1 + C_6(\boldsymbol{s}) \tag{4.70}$$

$$C_3(\boldsymbol{s}) = 1 + C_5(\boldsymbol{s}) \tag{4.71}$$

$$C_4(\boldsymbol{s}) = -C_m(\boldsymbol{s}) - Z_m(\boldsymbol{s}) \tag{4.72}$$

$$\tag{4.73}$$

which would result in a complete compensation of the dynamics of the haptic interface and teleoperator. This selection is however not possible in practice. On the one hand, this would require an acceleration measurement which is quite noisy in practice. Therefore, instead of compensating the complete device dynamics $Z_m(\boldsymbol{s})$ and $Z_s(\boldsymbol{s})$, we only compensate the device damping b_m/b_s and leave the device inertia m_m/m_s uncompensated. On the other hand, we use the local force controllers $C_5(\boldsymbol{s})$ and $C_6(\boldsymbol{s})$ to model sensor and actuator dynamics in addition to the actual controller. Therefore, exactly achieving the transparency objective for $C_2(\boldsymbol{s})/C_3(\boldsymbol{s})$ is not possible in practice. The resulting, *practically possible* transparency-optimal controller parameters are

$$C_1(\boldsymbol{s}) = C_s(\boldsymbol{s}) + b_s \tag{4.74}$$

$$C_2(\boldsymbol{s}) = G_m(\boldsymbol{s})K_f G_e(\boldsymbol{s}) \tag{4.75}$$

$$C_3(\boldsymbol{s}) = G_s(\boldsymbol{s})K_f G_h(\boldsymbol{s}) \tag{4.76}$$

$$C_4(\boldsymbol{s}) = -C_m(\boldsymbol{s}) - b_m \tag{4.77}$$

$$\tag{4.78}$$

where in our case, due to the system symmetry $G_m(\boldsymbol{s}) = G_s(\boldsymbol{s})$ and $G_h(\boldsymbol{s}) = G_e(\boldsymbol{s})$ and therefore

$$C_2(\boldsymbol{s}) = G_m(\boldsymbol{s})G_e(\boldsymbol{s}) + C_6(\boldsymbol{s}) \tag{4.79}$$

$$C_3(\boldsymbol{s}) = G_s(\boldsymbol{s})G_h(\boldsymbol{s}) + C_5(\boldsymbol{s}) \tag{4.80}$$

$$\tag{4.81}$$

which is very close to the optimal case for the frequency range of interest.

The four-channel controllers given here can directly be used for a transparency analysis using the method from Section 4.4. To simplify the transparency analysis the force sensor filters $G_h(\boldsymbol{s})$ and $G_e(\boldsymbol{s})$ and the filters $G_m(\boldsymbol{s})$ and $G_s(\boldsymbol{s})$ representing the actuator dynamics are all set to one, i.e., these filters are not considered in the transparency analysis. As the bandwidth used for system identification (see (4.36)) is well below the cut-off frequency of these filters, the effect of this simplification on the identification results is negligible.

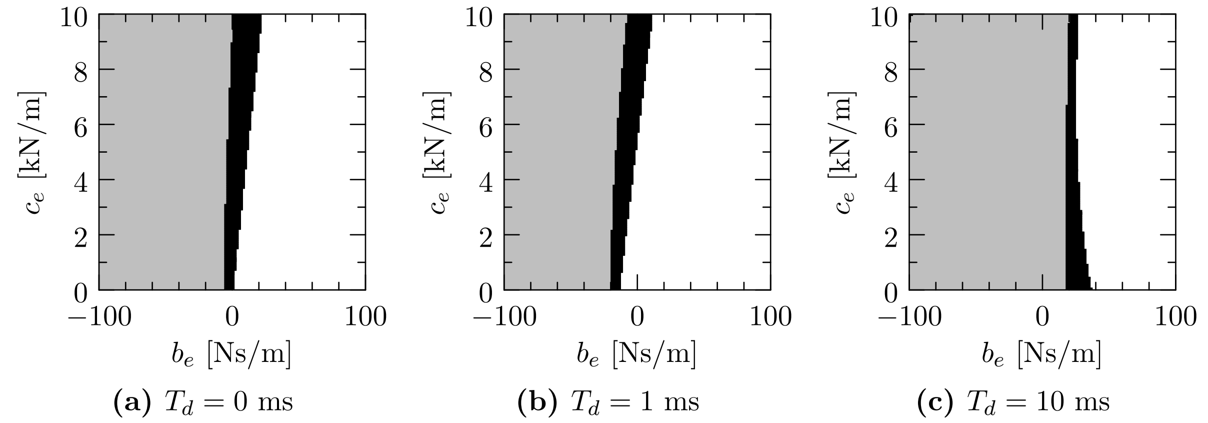
For stability analysis, a state-space representation of the system is determined as described in Section 4.3 by taking the models of human, environment, haptic interface, and teleoperator as well as the controller equations (4.65 - 4.68, 4.74 - 4.77) into account. This results in a system with system state $\boldsymbol{x} \in \mathbb{R}^8$ and parameter vector $\boldsymbol{q} \in \mathbb{R}^{19}$ given by

$$\boldsymbol{x} = [\dot{x}_m,\ x_m,\ \dot{x}_s,\ x_s,\ \tilde{f}_m,\ \tilde{f}_s,\ \tilde{f}_h,\ \tilde{f}_e]^T \tag{4.82}$$

$$\boldsymbol{q} = [\boldsymbol{q}_h^T,\ \boldsymbol{q}_e^T,\ \boldsymbol{q}_m^T,\ \boldsymbol{q}_s^T,\ K_p,\ K_d,\ K_f]^T \tag{4.83}$$

$$\tag{4.84}$$

where $\boldsymbol{q}_h$, $\boldsymbol{q}_e$, $\boldsymbol{q}_m$, and $\boldsymbol{q}_s$ are given in Section 4.3[26].



(a) $T_d = 0$ ms **(b)** $T_d = 1$ ms **(c)** $T_d = 10$ ms

Figure 4.17: Stability analysis of transparency optimized four-channel architecture for different time delays. Controller parameters are chosen heuristically. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both b_e and c_e.

Stability Analysis

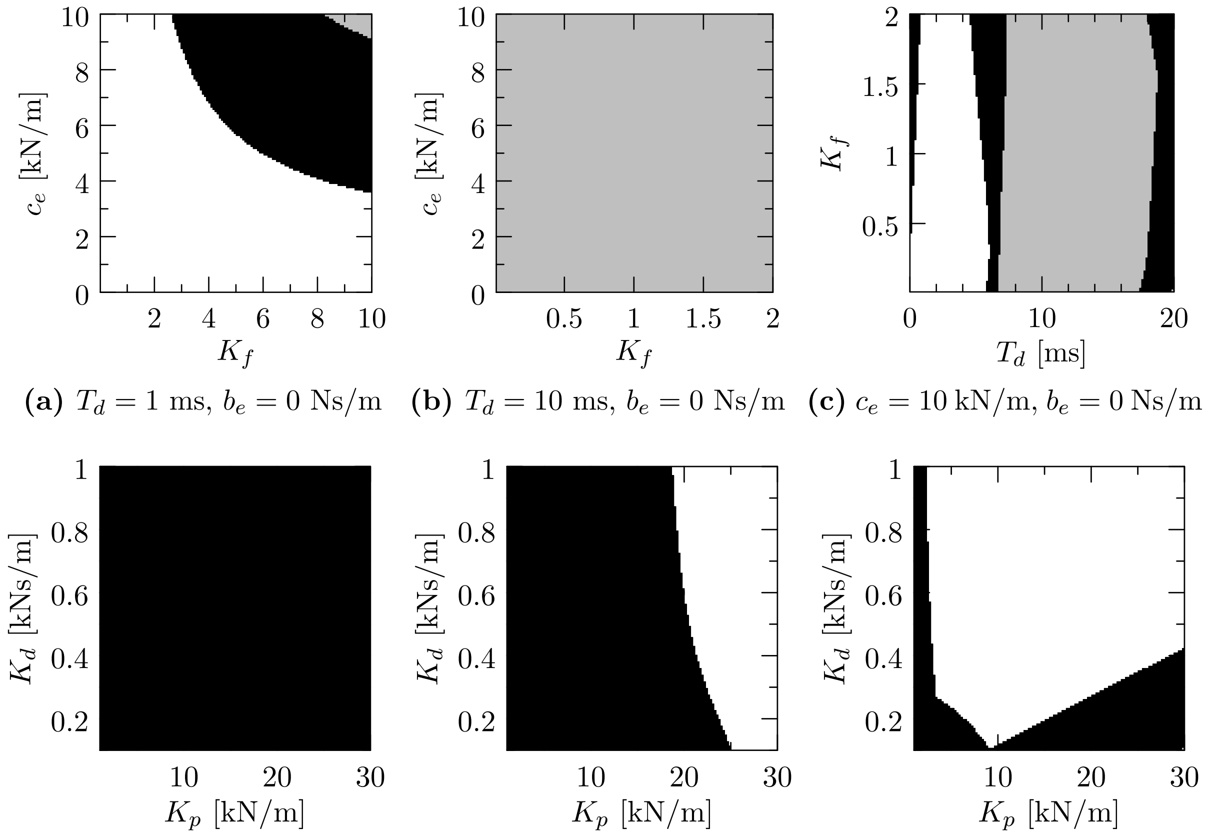
In this section, stability of the complete system with sensor and actuator filters is examined. For all results shown in this section, the impedance of the operator is set to the three-dimensional interval box given in Table 4.2, i.e., we consider a whole range of values for inertia, damping, and stiffness of the human arm.

We start out by examining stability for a heuristically chosen parametrization of $K_p = 70$ kN/m, $K_d = 500$ Ns/m, and $K_f = 1$ which we take from the force-position architecture in Section 4.5.2. Resulting stable and unstable regions in the environment plane b_e/c_e are shown in Fig. 4.17. The system is not stable for the complete environment range $b_e = [0,\ 100]$ Ns/m and $c_e = [0,\ 10]$ kN/m for any time delay. Interestingly, for free space ($b_e = 0$ Ns/m, $c_e = 0$ kN/m) the system is on the stability boundary for the delay-free case while a small negative damping is allowed for the case with $T_d = 1$ ms.

Therefore, we examine the effect of different controller parameters on stability in Fig. 4.18. As for the two-channel force-position architecture, the force controller gain K_f has a large impact on stability for a time delay of $T_d = 1$ ms as shown in Fig. 4.18a. However, even reducing the force controller gain to relatively small values does not stabilize the system for all time delays of interest as shown in Fig. 4.18b for a time delay of $T_d = 10$ ms. The effect becomes even more clear in Fig. 4.18c where stability is examined in the T_d/K_f-plane: there is an upper limit on the admissible time delay for all considered values of K_f and there is a small upper bound for the value of K_f for the delay-free case.

Consequently, we examine the effect of the other controller parameters on stability. We choose $K_f = 1$ (which stabilizes the system for $T_d = 1$ ms) and examine stability in the

[26] Note that the state vector as well as the system parameters are identical to the force-position architecture in Section 4.5.2. The stability analysis is however more complex due to the more complex system interconnection.



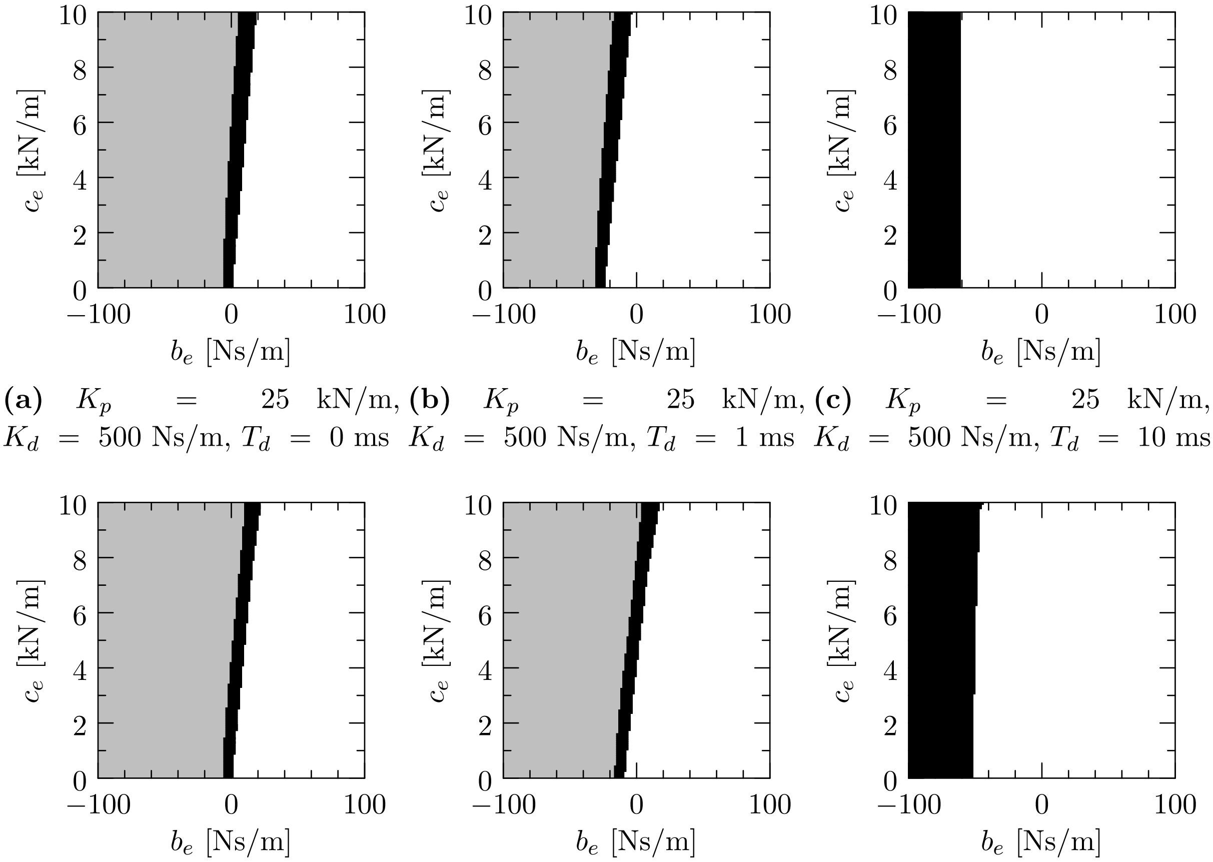
(a) $T_d = 1$ ms, $b_e = 0$ Ns/m **(b)** $T_d = 10$ ms, $b_e = 0$ Ns/m **(c)** $c_e = 10\,$kN/m, $b_e = 0\,$Ns/m

(d) $T_d = 0$ ms **(e)** $T_d = 1$ ms **(f)** $T_d = 10$ ms

Figure 4.18: Controller design for transparency optimized four-channel architecture. In the top row, stability with respect to environment stiffness c_e is examined for changes to K_f for a fixed environment damping $b_e = 0$ Ns/m and position controller gains $K_p = 70$ kN/m, and $K_d = 500$ Ns/m. In the second row, stable regions in the K_p/K_d-plane which are stable for the complete environment range of interest, i.e., $b_e = [0,\ 100]$ Ns/m and $c_e = [0,\ 10]$ kN/m for different time delays. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in each axis.

K_p/K_d-plane for the complete environment range of interest in Fig. 4.18d to Fig. 4.18f[27]. For the delay-free case no stable parameters can be found in the considered range of K_p and K_d for the chosen force controller gain, see Fig. 4.18d. By intersecting the white regions of the two plots in Fig. 4.18e and Fig. 4.18f a parametrization which yields stable results for the environment range $b_e = [0,\ 100]$ Ns/m, $c_e = [0,\ 10]$ kN/m for time delays of 1 ms and

[27] We choose smaller upper limits for K_p and K_d in this case in comparison with the other system architectures as small values of K_p and K_d are beneficial for transparency when time delay is present in the communication channel. This is shown in the subsequent transparency analysis. According to the transparency analysis it would also be beneficial to increase the force controller gain K_f. However, e.g., for $K_f = 2$, no parametrization that is stable for the environment range could be found for a reasonable range of K_p and K_d.



(a) $K_p = 25$ kN/m, $K_d = 500$ Ns/m, $T_d = 0$ ms **(b)** $K_p = 25$ kN/m, $K_d = 500$ Ns/m, $T_d = 1$ ms **(c)** $K_p = 25$ kN/m, $K_d = 500$ Ns/m, $T_d = 10$ ms

(d) $K_p = 10$ kN/m, $K_d = 200$ Ns/m, $T_d = 0$ ms **(e)** $K_p = 10$ kN/m, $K_d = 200$ Ns/m, $T_d = 1$ ms **(f)** $K_p = 10$ kN/m, $K_d = 200$ Ns/m, $T_d = 10$ ms

Figure 4.19: Stability analysis of transparency optimized four-channel architecture for $K_f = 1$ and different values for K_p and K_d. Gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (white) there is no eigenvalue in the closed right half-plane and the system is therefore asymptotically stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both b_e and c_e.

10 ms is found, e.g., for $K_p = 25$ kN/m, $K_d = 500$ Ns/m, $K_f = 1$. If we limit ourselves to the case with $T_d = 10$ ms we could, e.g., also choose $K_p = 10$ kN/m, $K_d = 200$ Ns/m, $K_f = 1$.

Finally, we verify the controller design for the different parameter choices by performing a stability analysis in the environment plane b_e/c_e in Fig. 4.19[28]. The system is stable for the whole environment range of interest for the parametrization $K_p = 25$ kN/m, $K_d = 500$ Ns/m, $K_f = 1$ for time delays of 1 ms and 10 ms while it is stable for time delays of $T_d = 10$ ms using the parametrization $K_p = 10$ kN/m, $K_d = 200$ Ns/m, $K_f = 1$. Notably, for the three values of time delay considered here, the amount of permissible *negative* environment damping grows for both parametrizations for rising time delay.

[28] This is not really necessary in this case as the design process assures a stable system for the compete environment range of interest.

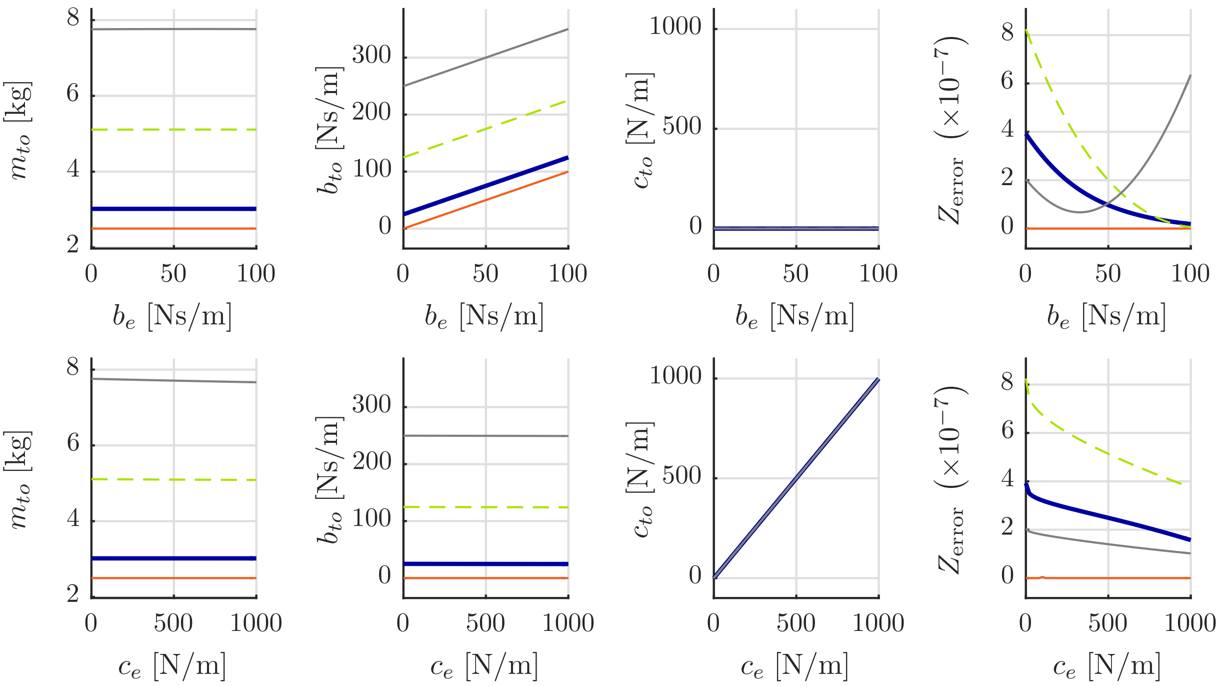


Figure 4.20: Transparency analysis of four-channel architecture with $K_p = 25$ kN/m, $K_d = 500$ Ns/m, $K_f = 1$ for different time delays (red, solid: no delay; blue, bold: $T_d = 1$ ms; green, dashed: $T_d = 5$ ms; gray, bold: $T_d = 10$ ms). System identification is performed for 51 values of the variable environment parameter.

For the parametrization $K_p = 25$ kN/m, $K_d = 500$ Ns/m, $K_f = 1$ we perform one final stability analysis considering the environment range $b_e = [0,\ 100]$ Ns/m, $c_e = [0,\ 10]$ kN/m, as well as the continuous range of time delays $T_d = [1,\ 10]$ ms which confirms that the system is stable in the complete six-dimensional interval box[29].

In the following section, transparency of the system is analyzed for both sets of parameters considered in Fig. 4.19.

Transparency Analysis

In Fig. 4.20 the parameter-space transparency analysis of the transparency-optimized four-channel architecture is shown for a parametrization of $K_p = 25$ kN/m, $K_d = 500$ Ns/m, $K_f = 1$ which is stable for environments in the range $b_e = [0,\ 100]$ Ns/m and $c_e = [0,\ 10]$ kN/m for time delays in the range $T_d = [1,\ 10]$ ms . For the case with no time delay (which is actually not stable) the results are as expected: the inertia of the haptic interface is displayed together with the environment damping and stiffness. For rising time delay the results become less and less transparent. More specifically, there is a linear relationship between time delay in the communication channel and displayed inertia and damping, e.g., for a time delay of 10 ms an additional damping of 250 Ns/s is displayed to the operator.

[29] The interval box consists of three dimensions for the human impedance, two dimensions for the environment, as well as one dimension for the time delay. The result is not depicted as we choose to simply *check* stability for the *complete* interval-box.

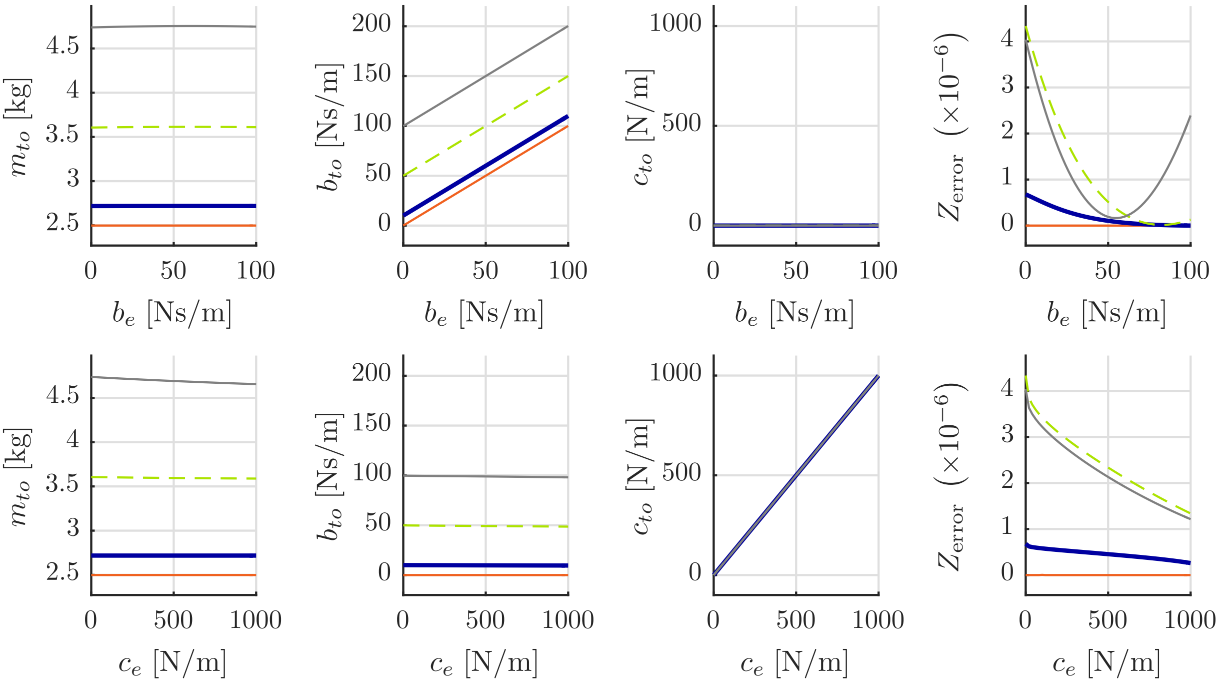


Figure 4.21: Transparency analysis of four-channel architecture with $K_p = 10$ kN/m, $K_d = 200$ Ns/m, $K_f = 1$ for different time delays (red, solid: no delay; blue, bold: $T_d = 1$ ms; green, dashed: $T_d = 5$ ms; gray, bold: $T_d = 10$ ms). System identification is performed for 51 values of the variable environment parameter.

In Fig. 4.21 we examine if reducing the position-controller gains to $K_p = 10$ kN/m and $K_d = 200$ Ns/m improves transparency. Indeed, it does, as the time-delay dependent damping is reduced by a factor of 2.5 (just as K_d is reduced by a factor of 2.5).

Finally, the effect of increasing the force controller gain is analyzed in Fig. 4.22 where we use the parameters $K_p = 10$ kN/m, $K_d = 200$ Ns/m, and $K_f = 2$. By doing this, the inertia displayed to the operator is reduced by a factor of 2, as well as the inertia and damping due to time delay. It must however be noted that this parametrization is not stable for the complete environment range $b_e = [0,\ 100]$ Ns/m and $c_e = [0,\ 10]$ kN/m.

Discussion

In this section stability and transparency of the transparency-optimized four-channel architecture was evaluated.

To our best knowledge, this is the first time that stability has been investigated non-conservatively in the parameter space for different time delays while also taking sensor and actuator dynamics into account by considering appropriate filters. The results show that it is possible to parametrize the system in such a way that it is stable for a wide range of environment and operator impedances for one-way time delays between 1 ms and 10 ms. Moreover, for any time delay, increasing the position-controller gains enlarges the range of environments which result in a stable system. Finally, our analysis shows that for the same set of controller parameters the system is actually not stable for the delay-free case for a range of environments with low damping values where it is stable in the case with

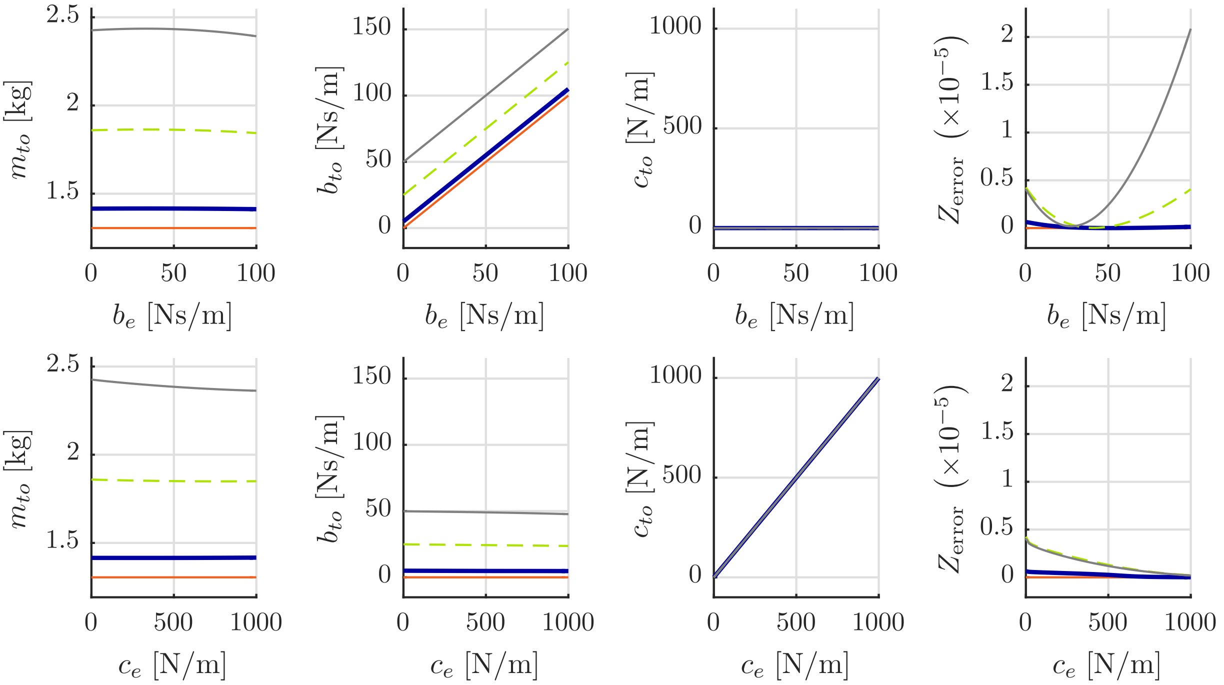


Figure 4.22: Transparency analysis of four-channel architecture with $K_p = 10$ kN/m, $K_d = 200$ Ns/m, $K_f = 2$ for different time delays (red, solid: no delay; blue, bold: $T_d = 1$ ms; green, dashed: $T_d = 5$ ms; gray, bold: $T_d = 10$ ms). System identification is performed for 51 values of the variable environment parameter.

time delays between 1 ms and 10 ms[30].

These results are very interesting in comparison with the results obtained using the absolute stability criterion, which is usually used to analyze stability of this teleoperation architecture. The absolute stability criterion checks whether the system is stable when *any* passive operator interacts with *any* passive environment using the teleoperation system. It does not take actuator or sensor dynamics into account and yields that the system is marginally stable for the delay-free case and minimal time delay might destabilize the system. Our approach considers *specific models* with limited parameter ranges for human and environment and takes actuator and sensor dynamics into account. For the delay-free case, our method shows that the system might become unstable even for passive operator and environment impedances due to the sensor and actuator dynamics. For the case with time delays, our approach offers less conservative results than the absolute stability criterion but relies on the assumption that the models and parameter ranges we consider for human and environment are a sufficient representation of the actual situation.

Once stable parameters were determined we performed a transparency analysis which yielded three interesting results. First, increasing time delay leads to an increase of the damping and inertia displayed to the operator. Second, the damping and inertia displayed to the operator due to time delay can be reduced by increasing the force controller gain K_f or reducing the position-controller gains K_p (to reduce the displayed damping) and

[30] This is due to the sensor and actuator filters. For the case without these filters, which is not shown here, the system is stable for all positive values of environment damping and stiffness and is not stable for minimal negative damping.

K_d (to reduce the displayed inertia). Finally, apart from the time-delay dependent *offset*, the transparency properties of the four-channel architecture are retained, i.e., an increase of environment damping or stiffness leads to the same increase of displayed damping or stiffness.

Summarizing these results we can conclude that the transparency-optimized four-channel architecture is suitable for many real-world teleoperation systems with small but non-negligible time delay as long as the modeling assumptions made in Section 4.2.1 hold.

4.5.5 Position-Based Admittance Control with Force Exchange (FaFa)

The position-based admittance control with force exchange (FaFa) is an architecture which has been used for many different teleoperation setups at the Chair of Automatic Control Engineering (LSR). This is mainly due to two reasons: On the one hand, it is easily applicable to very complex systems due to a limited number of parameters. On the other hand, it is transparent with respect to environment stiffness if no time delay is present and more robust than a four-channel architecture.

The general idea of this system architecture is to feed the sum of the interaction forces measured on master side and slave side into a virtual admittance (more specifically a mass-damper system) on master and slave side. The position and velocity of the virtual admittance on master side is then imposed on the haptic interface using a high-gain position control loop. Likewise, the position and velocity of the virtual admittance on slave side is imposed on the teleoperator.

Setups for which this architecture has been used include, e.g., one-DoF linear devices used for basic research [73], four-DoF telemanipulators and haptic interfaces [93], anthropomorphic manipulators coupled to hyper-redundant haptic interfaces [3, 73], etc.

In the following section, the system model is presented in detail. Then, the robust stability analysis as well as a parameter-space transparency analysis are performed. Finally, the results are discussed.

System Model

The position-based admittance controller with force exchange is a symmetric control architecture which is depicted in Fig. 4.23. Measured forces are exchanged in both directions and applied to an identical[31] virtual admittance

$$Y_d(\boldsymbol{s}) = \frac{1}{m_d\boldsymbol{s} + b_d} \tag{4.85}$$

with virtual inertia m_d and virtual damping b_d on either side, i.e.

$$\tilde{f}_h - \tilde{f}_e e^{-T_d\boldsymbol{s}} = m_d\ddot{x}_{dm} + b_d\dot{x}_{dm} \tag{4.86}$$

$$\tilde{f}_h e^{-T_d\boldsymbol{s}} - \tilde{f}_e = m_d\ddot{x}_{ds} + b_d\dot{x}_{ds}, \tag{4.87}$$

[31] If a virtual admittance with different parameters were chosen on master and slave side this would result in a position drift and velocity drift.

where T_d is the one-way time delay between slave and master. A proportional-derivative (PD) position controller then drives the haptic interface and teleoperator to the position of the respective virtual admittance, i.e.

$$f_m = K_p(x_{dm} - x_m) + K_d(\dot{x}_{dm} - \dot{x}_m) \tag{4.88}$$

$$f_s = K_p(x_{ds} - x_s) + K_d(\dot{x}_{ds} - \dot{x}_s), \tag{4.89}$$

where K_p and K_d are the proportional and derivative gain of the PD-controller[32].

The effect of position-based admittance control with force exchange is as follows. Using a virtual admittance and high-gain PD control the actual dynamics of haptic interface as well as teleoperator are canceled and the behavior of the virtual admittance is imposed on both devices. Then, the sum of the interaction forces between teleoperator and remote environment and between haptic interface and human is fed into each virtual admittance. As the admittance has a filtering effect on the force measurements this implies that it is not necessary to use such a low cut-off frequency for the force sensor filter as for the other architectures in this chapter[33]. Instead, the force sensor filter time constants are set to $\tau_h = \tau_e = 0.1\pi$ ms which corresponds to a cut-off frequency of 500 Hz.

In the case without time delay the inputs to the two virtual admittance controllers are identical. This means that the outputs of the two virtual admittance controllers are identical as well and the only remaining error between haptic interface and teleoperator is due to the non-perfect position control loop. We could remove one of the virtual admittance blocks and simply use the same desired position and velocity for haptic interface and teleoperator as, e.g., done in some of our previous work [3, 93–96]. When considering the case with time delay, this simplification is not possible, as the inputs to the two virtual admittance blocks are different, and stability of the overall system would be greatly impaired by removing one of the admittance controllers.

For stability analysis, a state-space representation of the system is determined as described in Section 4.3 by taking the models of human, environment, haptic interface, and teleoperator as well as the controller equations (4.86 - 4.89) into account. This results in a system with system state $\boldsymbol{x} \in \mathbb{R}^{12}$ and parameter vector $\boldsymbol{q} \in \mathbb{R}^{20}$ given by

$$\boldsymbol{x} = [\dot{x}_m,\ x_m,\ \dot{x}_s,\ x_s,\ \dot{x}_{dm},\ x_{dm},\ \dot{x}_{ds},\ x_{ds},\ \tilde{f}_m,\ \tilde{f}_s,\ \tilde{f}_h,\ \tilde{f}_e]^T \tag{4.90}$$

$$\boldsymbol{q} = [\boldsymbol{q}_h^T,\ \boldsymbol{q}_e^T,\ \boldsymbol{q}_m^T,\ \boldsymbol{q}_s^T,\ K_p,\ K_d,\ m_d,\ b_d]^T \tag{4.91}$$

where $\boldsymbol{q}_h$, $\boldsymbol{q}_e$, $\boldsymbol{q}_m$, and $\boldsymbol{q}_s$ are given in Section 4.3.

For transparency analysis using the method introduced in Section 4.4 the controllers $C_{1..6}$, C_m, and C_s of the four-channel architecture in Section 4.2.2 are determined from

[32] Generally, different gains for the position controller can be used on master and slave sides. Moreover, the PD-controller can also be replaced by more elaborate local controllers, e.g., a computed torque controller.

[33] For all other architectures considered in this chapter, the measured force is multiplied by a proportional gain and then *directly* applied to an actuator.

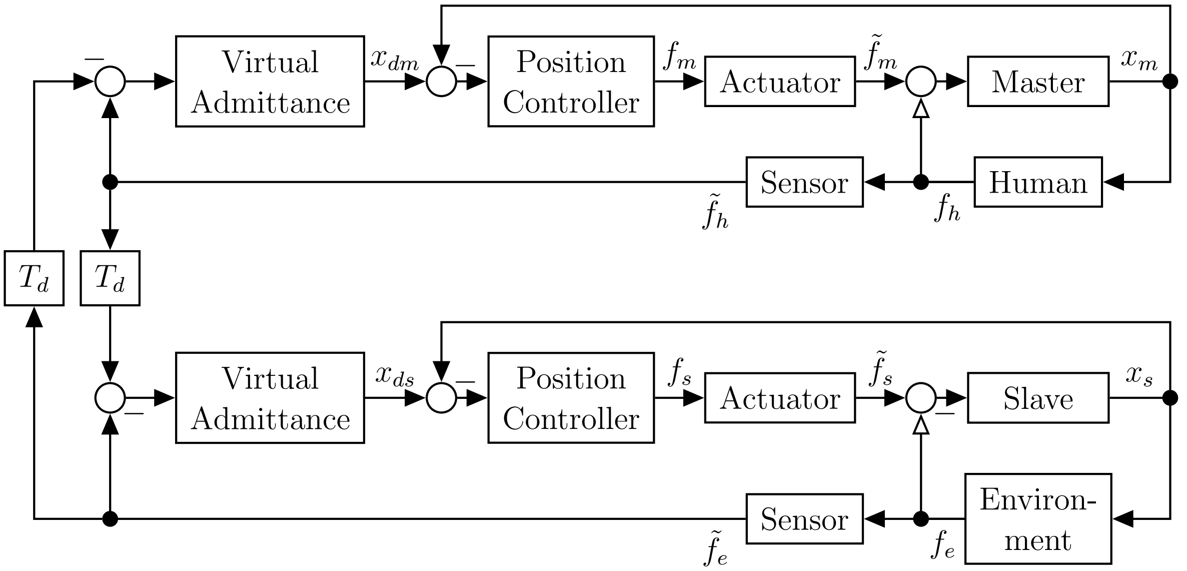


Figure 4.23: FaFa architecture

(4.85), (4.88), and (4.89) as

$$C_m(\boldsymbol{s}) = G_m\left(K_p/\boldsymbol{s} + K_d\right) \tag{4.92}$$

$$C_s(\boldsymbol{s}) = G_s\left(K_p/\boldsymbol{s} + K_d\right) \tag{4.93}$$

$$C_2(\boldsymbol{s}) = G_m\left(K_p/\boldsymbol{s} + K_d\right) Y_d(\boldsymbol{s})G_e \tag{4.94}$$

$$C_3(\boldsymbol{s}) = G_s\left(K_p/\boldsymbol{s} + K_d\right) Y_d(\boldsymbol{s})G_h \tag{4.95}$$

$$C_5(\boldsymbol{s}) = G_s\left(K_p/\boldsymbol{s} + K_d\right) Y_d(\boldsymbol{s})G_e \tag{4.96}$$

$$C_6(\boldsymbol{s}) = G_m\left(K_p/\boldsymbol{s} + K_d\right) Y_d(\boldsymbol{s})G_h \tag{4.97}$$

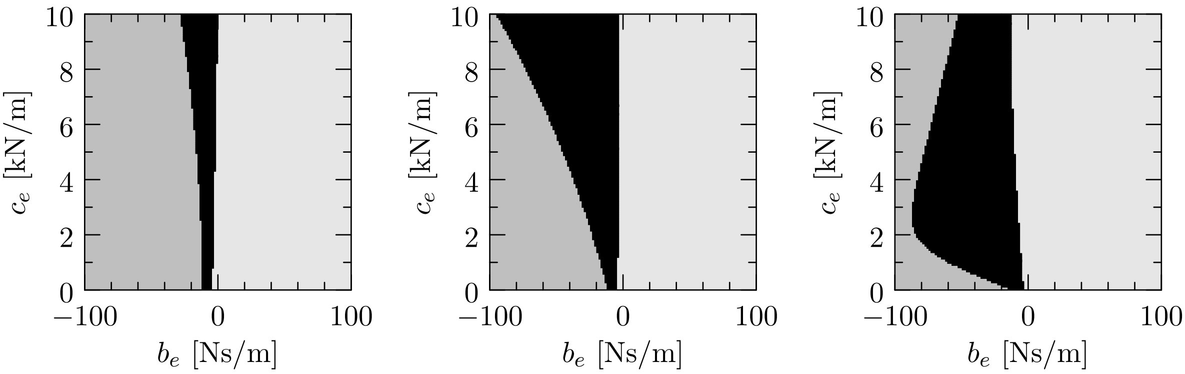
and $C_1(\boldsymbol{s}) = C_4(\boldsymbol{s}) = 0$. To simplify the transparency analysis the force sensor filters $G_h(\boldsymbol{s})$ and $G_e(\boldsymbol{s})$ and the filters $G_m(\boldsymbol{s})$ and $G_s(\boldsymbol{s})$ representing the actuator dynamics are all set to one, i.e., these filters are not considered in the transparency analysis. As the bandwidth used for system identification (see (4.36)) is well below the cut-off frequency of these filters, the effect of this simplification on the identification results is negligible.

Stability Analysis

In this section the complete system including sensor and actuator filters is considered. For all results shown in this section, the impedance of the operator is set to the three-dimensional interval box given in Table 4.2, i.e., we consider a whole range of values for inertia, damping, and stiffness of the human arm.

Before starting the actual stability analysis, it is pointed out that the system is not asymptotically stable for any considered parametrization. Instead, the system is at most marginally stable. This is due to the fact that the coefficient $a_0(\boldsymbol{s}, \boldsymbol{q}, T_d) = 0$ for $\boldsymbol{s} = 0$. As the condition $\boldsymbol{s} = 0$ does not depend on any parameters $\boldsymbol{q}$ this implies that the system is at most marginally stable[34]. Marginal stability with an eigenvalue at $\boldsymbol{s} = 0$ is indicated

[34] In the case of marginal stability for $\boldsymbol{s} = 0$, a very small region $\boldsymbol{s} = \pm 10^{-10} \pm 10^{-10} j$ around $\boldsymbol{s} = 0$ is



(a) $T_d = 0$ ms **(b)** $T_d = 1$ ms **(c)** $T_d = 10$ ms

Figure 4.24: Stability analysis of position-based admittance control architecture with force exchange and sensor and actuator filters for different time delays. Controller parameters are chosen heuristically. Dark gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (light gray) there is one eigenvalue at $\boldsymbol{s} = 0$ and the system is therefore marginally stable. The resolution $\tilde{q}_{\min}$ is 1/128 in both b_e and c_e.

by light gray regions in the stability analysis plots. When describing the results this is considered equivalent to stability for sake of a simpler formulation, especially when comparing the results to other architectures.

As for the other considered architectures, we start out with a nominal parametrization of $K_p = 70$ kN/m, $K_d = 500$ Ns/m, $m_d = 2$ kg, and $b_d = 5$ Ns/m and show the stability analysis results in the environment plane b_e/c_e for different time delays in Fig. 4.24. The system is stable for $b_e = [0,\ 100]$ Ns/m and $c_e = [0,\ 10]$ kN/m, i.e., for the complete environment range of interest.

Next, we examine the range in which we can tune the controller parameters while retaining stability for the region of interest. In this case, we do not further consider the parameters of the inner-loop PD-controller and, instead, only evaluate the effect of virtual inertia m_d and virtual damping b_d. The results are shown in Fig. 4.25. In Fig. 4.25a to Fig. 4.25f we examine the stable range of environment stiffness c_e for a fixed environment damping of $b_e = 0$ Ns/m. The results indicate, that there is a lower bound on the admissible damping b_d and inertia m_d for small time delays while there is a lower bound on the inertia m_d for larger time delays. A stability analysis in the m_d/b_d-plane for a fixed environment impedance $c_e = 10$ kN/m, $b_e = 0$ Ns/m shown in Fig. 4.25g to Fig. 4.25i yields that the heuristically chosen values $m_d = 2$ kg and $b_d = 5$ Ns/m are close to the minimal

excluded from the subsequent stability analysis, see Section 3.9.4. Therefore, it is theoretically possible that an additional root crosses the imaginary axis near $\boldsymbol{s} = 0$ (for boundary mapping) or that there is a root with positive real part near $\boldsymbol{s} = 0$ (when checking stability of disjoint regions). The first case would result in a stability boundary not being identified while the second case could result in an unstable region falsely being considered stable. As the region around $\boldsymbol{s} = 0$ is very small both of these cases are unlikely in practice.

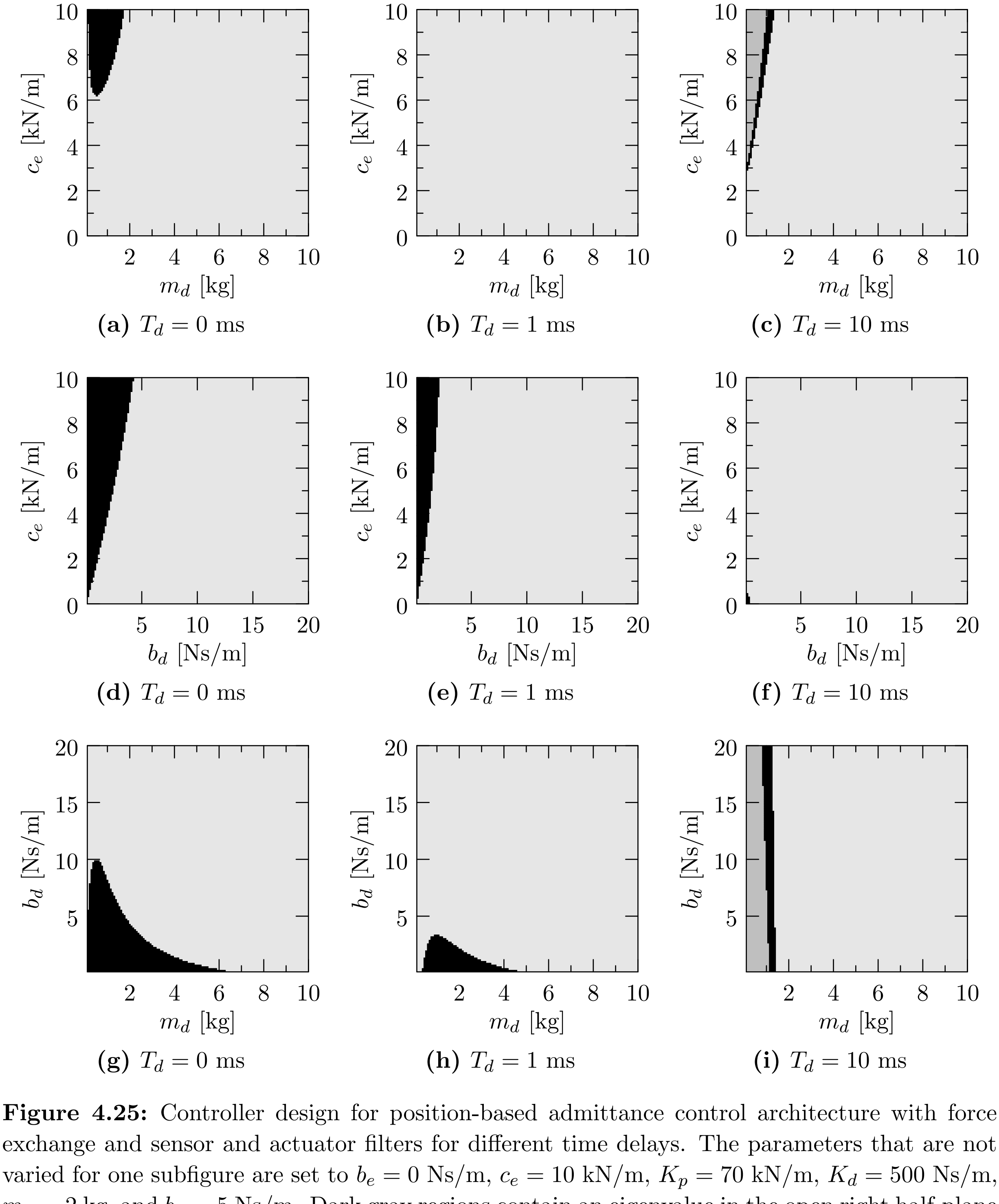


Figure 4.25: Controller design for position-based admittance control architecture with force exchange and sensor and actuator filters for different time delays. The parameters that are not varied for one subfigure are set to $b_e = 0$ Ns/m, $c_e = 10$ kN/m, $K_p = 70$ kN/m, $K_d = 500$ Ns/m, $m_d = 2$ kg, and $b_d = 5$ Ns/m. Dark gray regions contain an eigenvalue in the open right half-plane for all parameters in this region while black regions contain an Eigenvalue on the imaginary axis for at least one parametrization within this region. For the rest of the parameter space (light gray) there is one eigenvalue at $\boldsymbol{s} = 0$ and the system is therefore marginally stable. The resolution $\tilde{q}_{\min}$ is 1/128 in each axis.

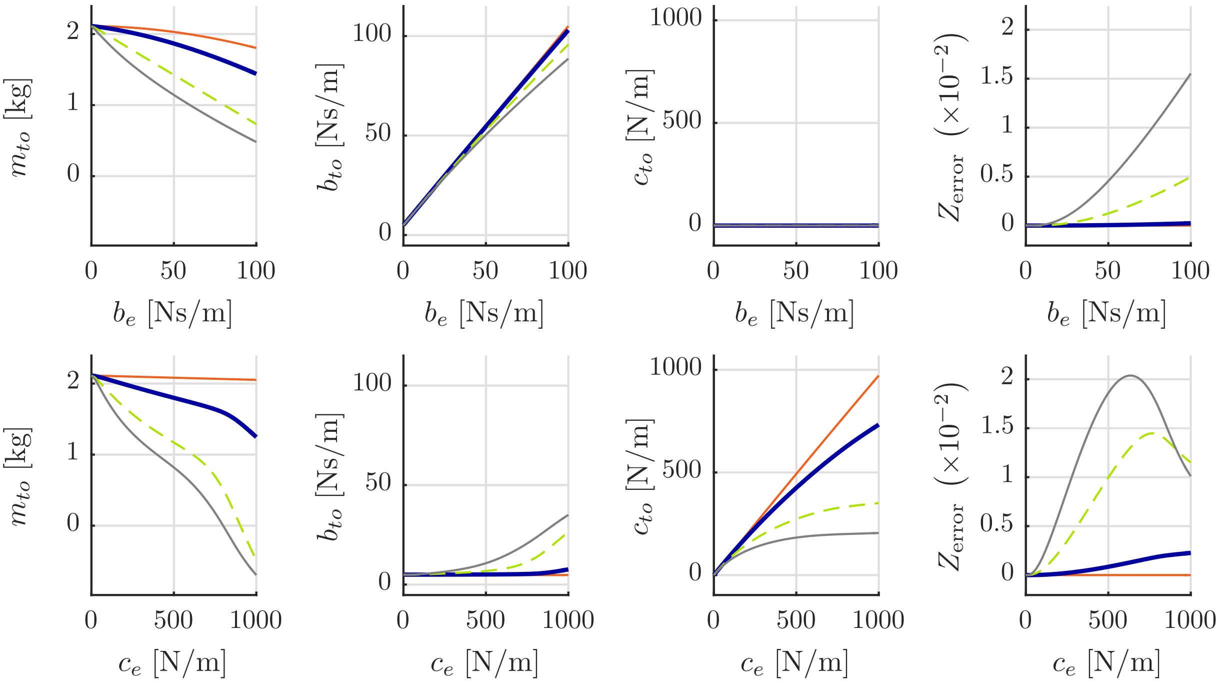


Figure 4.26: Transparency analysis of position-based admittance control architecture with force exchange (FaFa) for different time delays (red, solid: no delay; blue, bold: $T_d = 1$ ms; green, dashed: $T_d = 5$ ms; gray, bold: $T_d = 10$ ms). System identification is performed for 51 values of the variable environment parameter.

combination of admissible values for the considered time delays[35]. Therefore, transparency of the system is examined for this parametrization in the following.

Transparency Analysis

Fig. 4.26 depicts the parameter-space transparency analysis for the position-based admittance control architecture with force exchanged and a parametrization of $K_p = 70$ kN/m, $K_d = 500$ Ns/m, $m_d = 2$ kg, and $b_d = 5$ Ns/m. For the delay-free case the results indicate that the system displays the environment damping and stiffness to the operator in addition to the damping b_d and inertia m_d of the virtual admittance. Large damping or stiffness-values are however slightly attenuated. This is caused by the high but limited stiffness of the inner position-control loops. Moreover, the inertia displayed to the operator is slightly reduced for rising environment damping and stiffness.

For rising time delay there is a significant effect when considering interaction with a spring-environment: the maximum stiffness that can be displayed to the operator is greatly reduced with rising time delay. At the same time, the displayed damping rises[36]. When

[35] We choose fixed parameters for the environment impedance here as examining stability in seven dimensions (two for m_d and b_d and five for the parameters of human arm and environment impedance) was not computationally feasible. Stability for the complete environment range of interest is assured by the stability analysis in the environment plane which was shown in Fig. 4.24.

[36] Moreover, the inertia displayed to the operator is reduced. However, the reduction of inertia is likely not perceivable due to the masking effect caused by the rising damping. See, e.g., [97], for more information on masking effects between different impedance parameters.

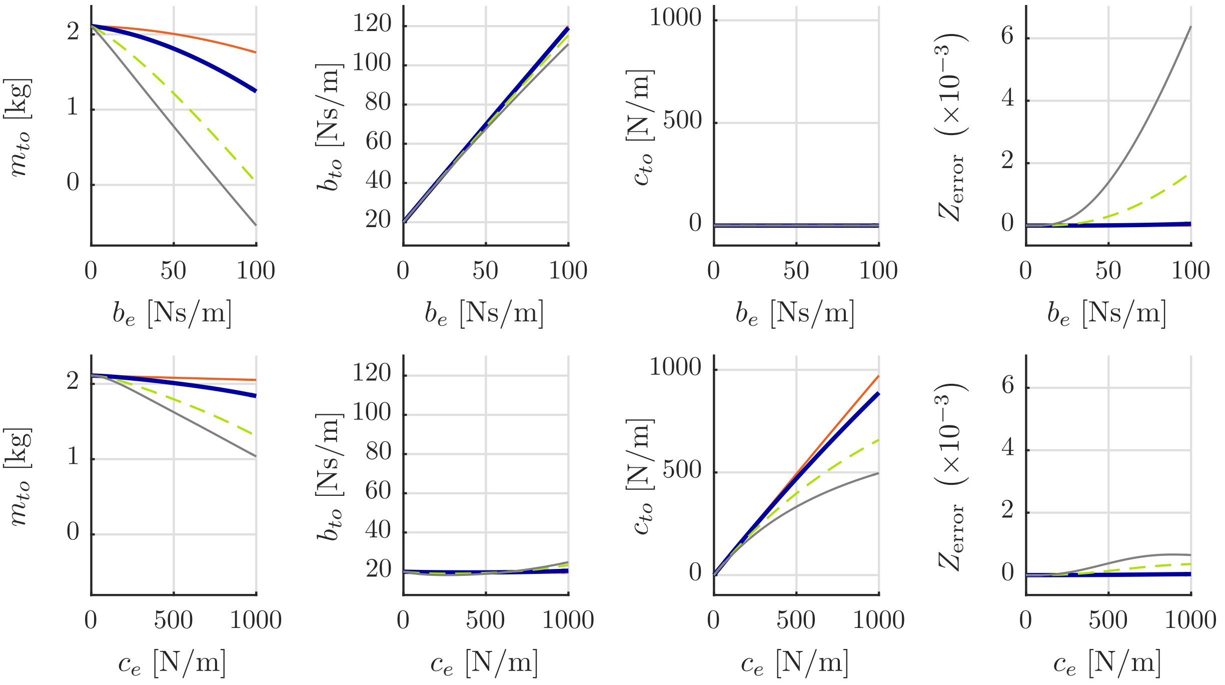


Figure 4.27: Transparency analysis of position-based admittance control architecture with force exchange (FaFa) and large virtual damping for different time delays (red, solid: no delay; blue, bold: $T_d = 1$ ms; green, dashed: $T_d = 5$ ms; gray, bold: $T_d = 10$ ms). System identification is performed for 51 values of the variable environment parameter.

considering interaction with a pure damped environment, a much smaller distortion due to time delay is observable, although the displayed damping as well as the displayed inertia are slightly reduced.

The transparency analysis results for an increased virtual damping of $b_d = 20$ Ns/m are shown in Fig. 4.27. All other controller parameters are left as for the previous analysis. As the larger virtual damping is displayed to the operator the results are inferior to the results with smaller virtual damping for the delay-free case and/or free space. Especially when interacting with stiff environments the increase of virtual damping is however beneficial as a larger maximum stiffness is displayed to the operator while there is less influence on displayed inertia and damping.

Discussion

In this section stability and transparency of the position-based admittance control architecture with force exchange was examined.

For the stability analysis we considered filters representing force sensor dynamics and actuator dynamics. Then, stability was analyzed in the environment plane and admissible controller parameters were determined for a range of different time delays. It turns out that there are lower bounds on the admissible values for virtual damping and inertia, as for the case without time delay examined in [73]. In contrast to [73] we examined how these bounds change for different time delays and determined a set of parameters which is stable for all considered time delays. Moreover, the ratio between the three parameters of the human arm impedance was fixed in [73] while we considered a complete three-dimensional interval box here. To the best of our knowledge, no non-conservative stability analysis results have been reported in literature so far for this complex system when considering time delay in the communication channel.

Using controller parameters which stabilize the system for the complete environment range of interest, transparency of the system was analyzed using our novel parameter-space transparency analysis method. Good transparency results are achieved for cases with negligible time delay, i.e., the environment dynamics in combination with the dynamics of the virtual admittance are displayed to the operator. For rising time delay the maximum stiffness that can be displayed to the operator is reduced while a stiffness dependent damping is observable. This effect might explain the robustness of this architecture to time delay in the communication channel. In essence, when interacting with a stiff environment the system becomes softer while increasing its damping for rising time delay. At the same time, rising time delay has no effect on the impedance displayed to the operator when moving in free space.

A nice property which has been pointed out for the delay-free case in [73] and is shown to hold for the case with time delay here is that it is possible to increase the virtual damping without affecting stability, if a minimum value is chosen for the virtual inertia. We make use of this property here and increase the virtual damping by a factor of four for our final transparency analysis. This results in a less transparent system in free space as the larger virtual damping is displayed to the operator. At the same time, transparency when interacting with stiff environments is increased, as the maximum displayable stiffness is increased.

Summarizing these results, it can be shown that the position-based admittance controller with force exchange offers nice transparency properties for negligible time delay while retaining stability for rising time delay (within the limits examined here). For rising time delay, the maximum stiffness that is displayed to the operator is reduced. The size of this effect depends on the chosen virtual damping.

4.6 Summary and Future Work

In this chapter, we analyzed the stability and transparency of a teleoperation system for different well-known teleoperation architectures. The system was modeled as a linear time-invariant system with parametric uncertainties and time delay in the communication channel, and the environment as well as the operator were modeled as linear mass-spring-damper systems, see Section 4.2.1 for details. These modeling assumptions are common in literature on teleoperation, and stability analysis results using these assumptions have been shown to closely represent experimental results for the delay-free case, e.g., in [72,73]. However, it must be noted that the results obtained in this chapter and summarized in the following paragraphs only hold if these assumptions are true.

For stability analysis, the method developed in Chapter 3 was used, which is ideally suited for the stability analysis of such a system. For transparency analysis, a new method was introduced which can be used to graphically assess the transparency of teleoperation systems. Unlike other approaches, not only free-space motion and hard contact are considered, but a whole range of environments. This allows a designer not only to determine *if* a system is sufficiently transparent, but *in which range* of environments a system is sufficiently transparent.

The results for the different teleoperation architectures are as follows. We first examined two examples for the *force-position architecture*: a simplified system where sensor and actuator dynamics are neglected and a more complex system which models sensor and actuator dynamics using first-order low-pass filters. In both cases, the force controller gain has by far the largest effect on stability and reducing the force controller gain stabilizes the system. When considering the same range of parameters for the human operator and remote environment, the upper bound on the force controller gain is smaller in the case with sensor and actuator dynamics than in the case without sensor and actuator dynamics. As for stability, transparency of the force-position architecture is largely determined by the size of the force controller gain. Unfortunately, a *smaller* force controller gain, which is beneficial for stability, leads to a *less transparent* system: for a force-controller gain of one the environment dynamics as well as the dynamics of the haptic interface are displayed to the operator. A larger force-controller gain attenuates the displayed dynamics of the haptic interface while a smaller force-controller gain amplifies the displayed dynamics of the haptic interface. Time delay does not directly influence transparency. However, for increasing time delay the range of admissible values for the force controller gain is reduced. Therefore, if the system is parametrized so as to be stable for a larger time delay, transparency is impaired for *all* time delays. As this effect is rather pronounced the system is not suitable for any non-negligible time delay.

The second teleoperation architecture we examined was the force-position architecture with wave variables in the communication channel. Termination elements were used to match the master and slave to the impedance of the communication channel. This results in a significant scaling and position drift, as shown in [64]. Our analysis showed that stability is nearly independent of delay for this architecture (at least for the delay values considered in our analysis). Moreover, the value of the wave damping has an influence on system stability, due to the fact that we incorporate sensor and actuator dynamics into our system model and the measured (and filtered) environment force is used as force feedback instead of the force generated by the controller on the slave side. The scaling effect caused by impedance matching obviously has a big impact on system transparency. In this case, a quarter of the environment dynamics is displayed to the operator together with the inertia and damping of the haptic interface and additional damping due to the damped wave. Time delay does not have a significant effect on transparency for the considered time delays. Nevertheless, a practical application of this architecture is not feasible, especially due to the large position drift caused by impedance matching by means of terminating the communication channel.

The third teleoperation architecture we considered is the transparency-optimized four-channel architecture. When appropriately parametrized the system is stable for a one-way time delay between 1 ms and 10 ms considered here. At the same time system transparency is near ideal for the case with negligible time delay. Then, the environment dynamics are only distorted by the additional inertia of the haptic interface. It must, however, be noted that the system is not stable for the complete environment range for the delay-free case when considering sensor and actuator dynamics. Transparency deteriorates with increasing time delay due to substantial time-delay dependent additional damping and inertia. In consequence, the transparency-optimized four-channel architecture should mainly be used for setups with very small time delays. If parametrized correctly, system stability is however robust against larger delays although transparency deteriorates.

Finally, we considered the position-based admittance control architecture with force exchange. For this architecture, parameters which stabilize the system for all time delays considered can be found very easily. Moreover, the system transparency is good for the case with no time delay, as only the inertia and damping of the virtual admittance are displayed to the operator, in addition to the otherwise nearly undistorted environment impedance. For increasing time delay, free-space transparency is unaffected, while interaction with a stiff environment becomes softer. The upper limit for the stiffness displayed to the operator directly depends on the size of the damping in the virtual admittance: a larger damping leads to an increased stiffness displayed to the operator. In consequence, this architecture is suitable for setups with small time delay if it is acceptable that the maximum displayable stiffness is reduced. Moreover, the architecture could easily be extended to incorporate a time-delay dependent damping term in the virtual admittance, which would lead to improved stiffness in the case with time delay.

Summarizing these results, it can be concluded that for all systems considered some novel insights could be gained using the stability analysis method developed in this thesis. These stability analysis results are complemented by intuitively interpretable transparency analysis results. Especially the transparency-optimized four-channel architecture and the position-based admittance control architecture with force exchange are interesting for teleoperation setups with small time delay as they offer good transparency for very small time delays (1 ms) and are robustly stable for larger time delays (up to 10 ms) at the cost of transparency.

Future work could focus on evaluating more complex teleoperation architectures using these tools. As one example, a combination of the transparency-optimized four-channel architecture with local position-based admittance controllers could be examined. Or, the wave-variable approach could be combined with local admittance or impedance controllers similar to [88]. As another example, stability of multi-user teleoperation systems (see, e.g., [3]) could be examined. Such systems could also incorporate control architectures especially developed to improve the interaction of multiple teleoperators, e.g., the virtual damping method introduced in [93, 95].

Another possible topic of future research could be the automatic parametrization of teleoperation systems with given control architecture so as to maximize transparency. This calls for a quantitative transparency measure that can be used as a cost function in a numerical optimization, as the method developed here still requires interpretation by the system designer. To bypass this human interpretation, the method could be combined with models of human haptic perception. Determining models of human haptic perception in dynamic environments is a current work in progress, see e.g. [98]. By embedding the resulting transparency measure in an optimization framework which minimizes this measure while assuring robust stability of the system over the whole environment range of interest, a fully automatic parametrization of teleoperation systems could be envisioned.

A possible future extension of our stability analysis method towards optimal parametrization of linear time-invariant systems with parametric uncertainties is therefore proposed in the following chapter. The proposed method is not limited to teleoperation systems or transparency analysis and considers a number of frequency-dependent cost functions in the optimization.

5 Towards Optimal Parametrization of LTI Systems

Summary. *In this chapter a novel solution for the optimal parametrization of linear time-invariant systems with time delays and parametric uncertainties is outlined. Therefore, a constrained optimization problem is formulated in which the cost function is related to the system performance while constraints assure robust stability and impose performance constraints. The optimization problem is solved in two steps by first determining the set of admissible parameters which assure robust stability and satisfy all performance constraints, and then minimizing the cost function within this set of admissible parameters. The main contributions of this chapter are*

- *a global optimization scheme for optimal parametrization of linear time-invariant systems with parametric uncertainties and time delay,*
- *a detailed description of how this optimization scheme could be implemented in practice and a thorough discussion of possible difficulties.*

In this thesis a stability analysis method for linear time-invariant systems with time delays and parametric uncertainties was introduced in Chapter 3. On the one hand, this method can be used for parameter-space stability analysis, e.g., to analyze the robust stability of a system. On the other hand, the method can be used for controller design by determining controller parameters which stabilize a given system for a range of plant parameters. This allows for an intuitive system design in the parameter space.

In this chapter an extension of the stability analysis method introduced in Chapter 3 is proposed, which allows for an optimal parametrization of LTI systems. First, performance criteria are defined for a general feedback system. Then, the desired system performance is specified by constraining the performance criteria. Two different possibilities to specify desired system performance are considered: we can either constrain the *worst-case performance* for a whole range of possible system parameters or the *nominal performance* for a specific parametrization. The performance constraints are mapped into the parameter space together with the stability boundary. By taking a number of different performance constraints into account a set of parameters can be determined which fulfills all desired specifications. The system designer can then manually choose a parametrization from this set of admissible parameters. Alternatively, a cost function may be defined which represents a weighted sum of different performance criteria. An optimization algorithm may then be used to find a parametrization which optimizes this cost function while assuring robust stability and possibly additional performance constraints. Depending on the given system dynamics and cost functions the optimization problem is not necessarily convex. Therefore, an algorithm which determines the global optimum within a given parameter range is required. An efficient algorithm for global optimization is the combination of Taylor Models and Bernstein Polynomials within a branch and bound algorithm. This approach is proposed here, as it can easily be combined with the stability analysis method presented in this thesis and offers a rigorous computation of the optimal solution.

The rest of the chapter is structured as follows. First, a brief overview of state of the art methods for automatic controller parametrization is presented. Then, a mathematical problem formulation is given and the general idea of the proposed algorithm is outlined. Next, a general system model is introduced and performance criteria for this system model are specified in Section 5.3. In Section 5.4 we describe a possible solution to the constrained optimization problem which first evaluates performance constraints and stability and then optimizes system performance. The chapter closes with a thorough discussion of problems which may be encountered when implementing the proposed algorithms.

5.1 State of the Art

The topic of this chapter is the optimal parametrization of a fixed structure controller for a linear time-invariant system with parametric uncertainties and possibly time delay.

A closely related problem which has attracted a much larger number of researchers in the past decades is that of finding an optimal controller (linear, but apart from this of arbitrary form) which results in optimal performance while assuring robust stability. The two most common methods which have been established to solve such problems are H_∞ control and μ synthesis. A thorough introduction to these methods can be found in [29].

H_∞ control generally leads to conservative results when considering systems with structured uncertainties [6]. This is mainly due to the fact that the *structure* of the uncertainty is not taken into account. Another approach to optimal design, μ synthesis, is based on the structured singular value μ. This approach is closely related to H_∞ control but, as the name suggests, takes the *structure* of the uncertainty into account. Theoretically the structured singular value offers an exact stability check for systems with uncertainties and time delays. In practice, an exact calculation of the structured singular value is not possible (see [6] and [29]) and especially for systems with multiple delays the structured singular value generally represents a very conservative stability criterion, see [34]. Moreover, when using the structured singular value μ to design a controller using μ synthesis a second difficulty arises due to the iterative method which is generally used to solve this problem, the so-called DK-iteration: as the DK-iteration results in a non-convex optimization problem it cannot be guaranteed that the globally optimal solution is found [29, Section 11.4].

Closely related to H_∞ control is H_∞ loop shaping. The goal of H_∞ loop shaping is to design a controller which results in a open-loop transfer function which can be specified by the designer and at the same time ensures robust stability. Different approaches to H_∞ loop shaping exist, for one example see [29, Chapter 18] where the designer specifies two compensators which together with the plant result in the desired system response. An H_∞ controller which achieves this response while robustly stabilizing the system is then synthesized if possible[1]. However, this approach to H_∞ loop shaping suffers from the same drawbacks as H_∞ controller synthesis, namely conservatism with respect to structured uncertainties.

The actual goal in this chapter is to find a parametrization for a fixed structure controller which optimizes a cost function and fulfills given performance constraints while guaranteeing robust stability. Recently, a promising new method which solves this problem was introduced in [99]. Essentially, this is a new μ-synthesis method which makes use of a dynamic inner approximation to solve the non-smooth optimization problem. In contrast to classical μ synthesis (with DK-iteration, see [29, Section 11.4]) this new method results in much less conservative results. However, up to now, no examples considering time delay are available and an evaluation of the conservatism in comparison with non-μ-techniques has not been published.

A different approach is pursued in [1, Sections 5.3-5.4], where *Ackermann et al.* introduce frequency response magnitude specifications and then map these specifications to the parameter space. This can be used to find a parametrization for a fixed structure controller which fulfills some given performance constraints. Our approach is very similar, there are however two main differences. On the one hand, we use Taylor Models and Bernstein Polynomials to map the constraints into parameter space. This allows us to consider more than two parameters without gridding the results and, more importantly, to consider time-delay systems. On the other hand, we combine performance constraints with a performance optimization.

5.2 Problem Formulation and General Idea

In this section a constrained optimization problem is formulated. The solution to this problem results in an optimal parametrization of a Linear Time-Invariant (LTI) system which can include parametric uncertainties and one or several incommensurate time delays. Robust stability of the system is imposed by a constraint which is evaluated using the method developed in Chapter 3.

As cost function a weighted sum of performance metrics is used. Moreover, constraints on performance metrics may be imposed using the same metrics. This allows an exact specification of the desired system dynamics, e.g. by tracking error, disturbance rejection, etc. For details on different performance metrics and weighting functions, see Section 5.3.

A general formulation of the constrained optimization problem is

$$\arg\min_{\boldsymbol{p}\in\boldsymbol{P}} g(\omega, \boldsymbol{p}, \bar{\boldsymbol{q}}, \boldsymbol{Q}, \bar{\boldsymbol{t}}_d, \boldsymbol{T}_d) \quad \forall\, 0 \le \omega \le \omega_{\max} \tag{5.1}$$

with

$$\begin{aligned} & f(\boldsymbol{s}, [\boldsymbol{q}^T \boldsymbol{p}^T]^T, \boldsymbol{t}_d) \quad s\text{table} \quad \forall\, \boldsymbol{q} \in \boldsymbol{Q},\ \boldsymbol{t}_d \in \boldsymbol{T}_d & (5.2)\\ \Leftrightarrow\quad & f(\boldsymbol{\gamma}, [\boldsymbol{q}^T \boldsymbol{p}^T]^T, \boldsymbol{t}_d) \neq 0 \quad \forall\, \boldsymbol{\gamma} \in \boldsymbol{\Gamma},\ \boldsymbol{q} \in \boldsymbol{Q},\ \boldsymbol{t}_d \in \boldsymbol{T}_d & (5.3)\\ & \boldsymbol{c}(\omega, \boldsymbol{p}, \bar{\boldsymbol{q}}, \boldsymbol{Q}, \bar{\boldsymbol{t}}_d, \boldsymbol{T}_d) < \boldsymbol{0} \quad \forall\, 0 \le \omega \le \omega_{\max} & (5.4)\end{aligned}$$

[1] This may however not be possible for the chosen compensators. The algorithm then simply states to choose *different* compensators without giving any indication how to choose feasible compensators.

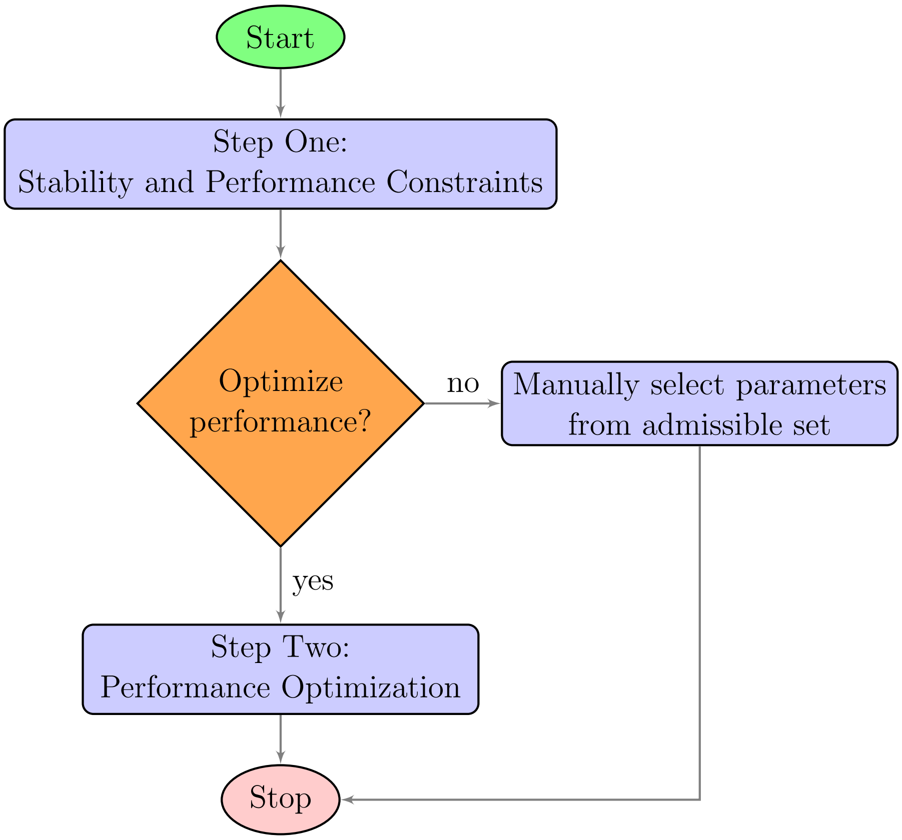

Start
Step One:
Stability and Performance Constraints
Optimize performance?
no
Manually select parameters from admissible set
yes
Step Two:
Performance Optimization
Stop

Figure 5.1: Flow-chart of overall algorithm used for performance optimization

where the parameter space consists of l uncertain parameters $\boldsymbol{q} \in \boldsymbol{Q} \in [\mathbb{R}]^l$, o tunable parameters $\boldsymbol{p} \in \boldsymbol{P} \in [\mathbb{R}]^o$, and m time delays $\boldsymbol{t}_d \in \boldsymbol{T}_d \in [\mathbb{R}]^m$. Moreover, the cost function $g \in \mathbb{R}^{m \times l \times o \times 1} \to \mathbb{R}$, the r performance constraints $\boldsymbol{c} \in \mathbb{R}^{m \times l \times o \times 1} \to \mathbb{R}^r$, and the characteristic function $f \in \mathbb{R}^{m \times l \times o \times 2} \to \mathbb{C}$ (see (3.6)) depend on nominal parameters $\bar{\boldsymbol{q}}$ and nominal time delays $\bar{\boldsymbol{t}}_d$, as well as the frequency ω, the Laplace operator $\boldsymbol{s}$, the desired $\boldsymbol{\Gamma}$-region used for stability analysis (see Section 3.4), and $\omega_{\max}$ which represents an upper bound for the frequency[2]. This formulation of the optimization problem allows to consider two different types of frequency-dependent performance metrics: nominal performance and worst-case performance.

The overall solution of the constrained optimization problem is determined in two steps as depicted in Fig. 5.1 and described in Section 5.4. First the set of admissible parameters is determined which satisfies all constraints (stability and performance), see Section 5.4.1. Then, the optimal parametrization within this set may be determined using a global optimization algorithm, see Section 5.4.2 Alternatively, the optimization step can be omitted and the system designer can manually select a parametrization from the set of admissible parameters. This corresponds to $\mathcal{B}$-stability mapping in [1, Section 5.4] with the exceptions that time delay and additional uncertain parameters can be considered without gridding.

[2] This indicates that it is not necessary to evaluate the performance constraints or cost function up to an infinite frequency. Instead, a reasonable upper bound above the system bandwidth can be chosen. This is a difference in comparison with stability analysis, where an analysis up to infinite frequencies is necessary to assure stability.

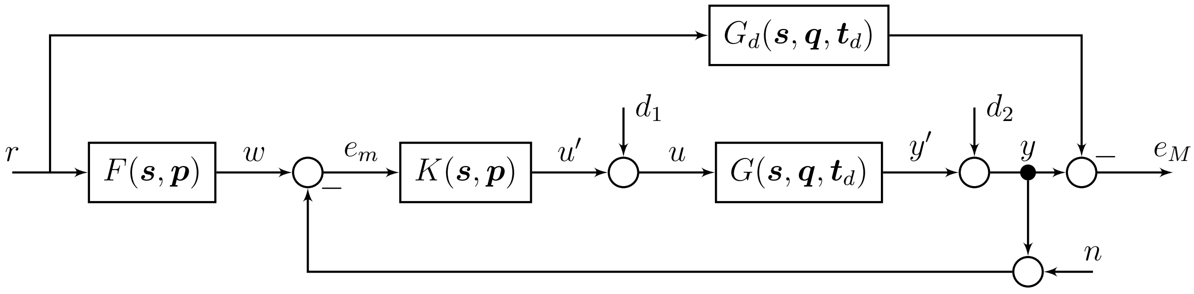

$G_d(\boldsymbol{s},\boldsymbol{q},\boldsymbol{t}_d)$
d_1
d_2
r
$F(\boldsymbol{s},\boldsymbol{p})$
w
e_m
$K(\boldsymbol{s},\boldsymbol{p})$
u'
u
$G(\boldsymbol{s},\boldsymbol{q},\boldsymbol{t}_d)$
y'
y
e_M
n
$-$
$-$

Figure 5.2: Single-loop feedback system (Fig. 5.32 from [1])

5.3 Performance Specification

A common approach to performance specification is chosen in this thesis which is used, e.g., in H_∞-control, see [29, Section 5.5]. Therefore, a general feedback system is introduced which can be used to represent a large class of systems. For this feedback system, different transfer functions are determined which, e.g., specify the tracking error, disturbance rejection, etc. In the following sections these transfer functions will then be used when specifying performance constraints and for performance optimization. Thereby, it is, e.g., possible to minimize the tracking error for low frequencies while maximizing sensor noise rejection for high frequencies. The rest of this section closely follows the presentation in [1, Section 5.3] from which the approach and notation for frequency response magnitude specification is adopted.

A general system model with parametric uncertainties and possibly time delay is given in Fig. 5.2 where G is the uncertain plant, K is the controller which should be parametrized, F is a prefilter (which may be a unit gain if it is not used), and G_d is a desired closed-loop transfer function. The inputs to the feedback system are the reference input r, plant input disturbance d_1, plant output disturbance d_2, and sensor noise n. Other important signals in this feedback system are the filtered reference input w, measured tracking error e_m, controller output u', actuator signal u, plant output y', and controlled signal y. In addition the *actual* tracking error

$$e = w - y \tag{5.5}$$

between filtered reference input w and controlled signal y is of interest[3]. Finally, if a desired transfer function G_d is considered, the model matching error is given by e_M .

The relation between filtered reference input w and the noise inputs(d_1, d_2, and n) and the signals of interest e (tracking error), u (actuator signal), and y (controlled signal) can be summarized as

$$\begin{bmatrix} e \\ u \\ y \end{bmatrix} = \begin{bmatrix} S & T & -S_G & -S \\ S_K & -S_K & S & -S_K \\ T & -T & S_G & S \end{bmatrix} \begin{bmatrix} w \\ n \\ d_1 \\ d_2 \end{bmatrix} \tag{5.6}$$

[3] The *actual* tracking error e is not shown in Fig. 5.2 and can not be measured in practice as a noise-free measurement is not possible. Instead, the *measured* tracking error e_m which includes measurement noise is used as controller input. However, when optimizing tracking performance we try to optimize the *actual* tracking error e.

where the so-called sensitivity functions S, T, S_G, and S_K all depend on the frequency ω, tunable parameters $\boldsymbol{p} \in \boldsymbol{P}$, uncertain parameters $\boldsymbol{q} \in \boldsymbol{Q}$, and time delays $\boldsymbol{t}_d \in \boldsymbol{T}_d$[4].

In the following the different sensitivity functions are introduced and corresponding desirable performance criteria are given.

Sensitivity Function / Tracking Error

The tracking error is characterized by the transfer function from the filtered reference input to the tracking error between filtered reference input and plant output. This transfer function is the *sensitivity function*

$$S(\boldsymbol{s}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d) = \frac{1}{1 + K(\boldsymbol{s}, \boldsymbol{p})G(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)} \tag{5.7}$$

$$= \frac{1}{1 + L(\boldsymbol{s}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d)}. \tag{5.8}$$

Generally, a small tracking error is especially important for low frequency input changes, i.e.,

$$|S(j\omega, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d)| \ll 1 \quad \forall\, 0 \leq \omega \leq \omega_S. \tag{5.9}$$

As is evident from (5.6) this implies a rejection of the output disturbance d_2.

Complementary Sensitivity Function / Sensor Noise Rejection

Sensor noise rejection which is given by the *complementary sensitivity function*

$$T(\boldsymbol{s}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d) = \frac{K(\boldsymbol{s}, \boldsymbol{p})G(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)}{1 + K(\boldsymbol{s}, \boldsymbol{p})G(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)} \tag{5.10}$$

$$= K(\boldsymbol{s}, \boldsymbol{p})G(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)S(\boldsymbol{s}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d) \tag{5.11}$$

$$= 1 - S(\boldsymbol{s}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d) \tag{5.12}$$

is another important goal in system design. As the characteristics of the sensor noise are often well known they can be taken into account when specifying the corresponding constraint. Generally, sensor noise rejection is especially of interest for high frequencies, i.e.,

$$|T(j\omega, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d)| \ll 1 \quad \forall\, \omega_T \leq \omega < \infty. \tag{5.13}$$

Input Sensitivity Function / Actuator Limits

Actuator limits may be considered using the *input sensitivity function*

$$S_K(\boldsymbol{s}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d) = \frac{K(\boldsymbol{s}, \boldsymbol{p})}{1 + K(\boldsymbol{s}, \boldsymbol{p})G(\boldsymbol{s}, \boldsymbol{q}, \boldsymbol{t}_d)} \tag{5.14}$$

$$= K(\boldsymbol{s}, \boldsymbol{p})S(\boldsymbol{s}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d) \tag{5.15}$$

which is the transfer function from the filtered reference input to the actuator output. By limiting this function we can specify frequency-dependent limits on the actuator output for a reference input of given bandwidth and magnitude, i.e.,

$$|S_K(j\omega, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d)| \quad \text{sufficiently small} \quad \forall\, 0 \leq \omega \leq \omega_S. \tag{5.16}$$

[4] These dependencies are omitted in (5.6) for the sake of brevity.

Output Sensitivity Function / Disturbance Rejection

Disturbance rejection is specified by the *output sensitivity function*

$$\begin{aligned} S_G(\boldsymbol{s},\boldsymbol{p},\boldsymbol{q},\boldsymbol{t}_d) &= \frac{G(\boldsymbol{s},\boldsymbol{q},\boldsymbol{t}_d)}{1+K(\boldsymbol{s},\boldsymbol{p})G(\boldsymbol{s},\boldsymbol{q},\boldsymbol{t}_d)} \qquad (5.17)\\ &= G(\boldsymbol{s},\boldsymbol{q},\boldsymbol{t}_d)S(\boldsymbol{s},\boldsymbol{p},\boldsymbol{q},\boldsymbol{t}_d) \qquad (5.18)\end{aligned}$$

which is the transfer function from the input disturbance d_1 to the tracking error. As with the tracking error, a disturbance rejection is especially of interest for low frequencies, i.e.,

$$|S_G(j\omega,\boldsymbol{p},\boldsymbol{q},\boldsymbol{t}_d)| \ll 1 \quad \forall\, 0 \le \omega \le \omega_S. \qquad (5.19)$$

Model Matching

Finally, a further possibility to specify system performance arises in the case where the input signal can or should not exactly correspond to the output signal. E.g., one might want to impose a certain system behavior on the closed-loop system (including filter F) specified by damping and critical frequency.

Therefore, the model matching error e_M can be considered which is given by

$$e_M = E(\boldsymbol{s},\boldsymbol{p},\boldsymbol{q},\boldsymbol{t}_d)\, r \qquad (5.20)$$

with

$$\begin{aligned} E(\boldsymbol{s},\boldsymbol{p},\boldsymbol{q},\boldsymbol{t}_d) &= G_d(\boldsymbol{s},\boldsymbol{q},\boldsymbol{t}_d) - H(\boldsymbol{s},\boldsymbol{p},\boldsymbol{q},\boldsymbol{t}_d) \qquad (5.21)\\ H(\boldsymbol{s},\boldsymbol{p},\boldsymbol{q},\boldsymbol{t}_d) &= \frac{F(\boldsymbol{s},\boldsymbol{p})K(\boldsymbol{s},\boldsymbol{p},)G(\boldsymbol{s},\boldsymbol{q},\boldsymbol{t}_d)}{1+K(\boldsymbol{s},\boldsymbol{p},)G(\boldsymbol{s},\boldsymbol{q},\boldsymbol{t}_d)}. \qquad (5.22)\end{aligned}$$

The model matching error may then be limited to a desired frequency-dependent value, i.e.,

$$|E(j\omega,\boldsymbol{p},\boldsymbol{q},\boldsymbol{t}_d)| \quad \text{sufficiently small} \quad \forall\, 0 \le \omega \le \omega_M. \qquad (5.23)$$

5.4 Two-Step Solution of the Constrained Optimization Problem

In this section a possible solution of the constrained optimization problem is presented which consists of two steps as depicted in Fig. 5.1. The two steps, first assuring stability and performance constraints, and then optimizing performance within a set of admissible parameters, are presented in this section.

In both steps, we can consider two different performance metrics: nominal performance or worst-case performance. If nominal performance is considered then the performance criteria are evaluated for one nominal parametrization. If worst-case performance is considered then the performance criteria are evaluated for the whole range of parameters and the worst case is constrained/optimized. One interesting scenario is, e.g., to optimize nominal performance while constraining worst-case performance.

We can decide whether to constrain/optimize nominal performance or worst-case performance by selecting $\boldsymbol{q}$ and $\boldsymbol{t}_d$ appropriately, i.e., nominal performance can be considered by simply replacing some or all uncertain parameters $\boldsymbol{Q}$ or time delays $\boldsymbol{T}_d$ by nominal parameters $\bar{\boldsymbol{q}}$ and nominal time delays $\bar{\boldsymbol{t}}_d$. By only replacing some uncertain parameters with nominal parameters or only replacing uncertain parameters with nominal parameters in some sensitivity functions or some constraints a mixed solution may also be determined which constrains/optimizes some criteria for the nominal case and other criteria for the worst case. Note that for the stability analysis we must *always* consider the whole range of parameters $\boldsymbol{Q} \times \boldsymbol{T}_d$.

5.4.1 Step One: Stability and Performance Constraints

First, the set of admissible parameters is determined for which the system is robustly stable and performance criteria are constrained to given values (frequency dependent). This may be interesting in many cases, e.g., when considering actuator limits or when a specified tracking error must be achieved with minimal actuator action or maximum disturbance rejection. Moreover, we could optimize some performance criteria for the nominal case while constraining them to a certain range for the worst case.

Formulation of Performance Constraints

The optimization problem formulated in Section 5.2 considers r performance constraints $\boldsymbol{c}(\omega, \boldsymbol{p}, \bar{\boldsymbol{q}}, \boldsymbol{Q}, \bar{\boldsymbol{t}}_d, \boldsymbol{T}_d) < 0$. As an example, to constrain the tracking error to a value smaller than $C_t(\omega)$ we set

$$c_1(\omega, \boldsymbol{p}, \bar{\boldsymbol{q}}, \boldsymbol{Q}, \bar{\boldsymbol{t}}_d, \boldsymbol{T}_d) = |S(j\omega, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{t}_d)| - C_t(\omega) \tag{5.24}$$

where the sensitivity function S depends on tunable parameters $\boldsymbol{p}$, uncertain parameters $\boldsymbol{q}$, and time delays $\boldsymbol{t}_d$. We can consider the cases of nominal performance or worst-case performance by selecting $\boldsymbol{q}$ and $\boldsymbol{t}_d$ appropriately, i.e., nominal performance can be considered by simply replacing some or all uncertain parameters $\boldsymbol{Q}$ or time delays $\boldsymbol{T}_d$ by nominal parameters $\bar{\boldsymbol{q}}$ and nominal time delays $\bar{\boldsymbol{t}}_d$.

Set of Admissible Parameters

In this section the set of admissible parameters $\boldsymbol{P}_a$ is determined which fulfills all performance constraints and assures robust stability.

For each of the r performance constraints c_i in $\boldsymbol{c}$ the parameter set $\boldsymbol{P}_i \subseteq \boldsymbol{P}$ is determined which satisfies the constraint $c_i < 0$:

$$\boldsymbol{P}_i = \left\{\boldsymbol{p} \in \boldsymbol{P} \mid c_i(\omega, \boldsymbol{p}, \bar{\boldsymbol{q}}, \boldsymbol{Q}, \bar{\boldsymbol{t}}_d, \boldsymbol{T}_d) < \boldsymbol{0} \; \forall \; 0 \leq \omega \leq \omega_{\max}\right\}. \tag{5.25}$$

Performance constraints can be evaluated using a similar method as for stability analysis in Chapter 3. First, the boundary on which the performance constraint is zero is mapped to the parameter space. Then, disjoint regions are checked to discern whether the performance constraint is violated in this region or not. For both of these steps a simplified version of the branch and bound algorithm in Section 3.6 can be used as we must only evaluate a real function instead of a complex function[5]. The set of parameters $\boldsymbol{P}_c$ which satisfies all performance constraints is then given by

$$\boldsymbol{P}_c = \bigcap_{i=1}^{r} \boldsymbol{P}_i. \tag{5.26}$$

Finally, the parameter set $\boldsymbol{P}_s \subseteq \boldsymbol{P}$ is determined[6] which satisfies the stability constraints, i.e., no eigenvalues in $\boldsymbol{\Gamma}$ for all values of $\boldsymbol{Q}$ and $\boldsymbol{T}_d$:

$$\boldsymbol{P}_s = \left\{ \boldsymbol{p} \in \boldsymbol{P} \mid f(\gamma, [\boldsymbol{q}^T \boldsymbol{p}^T]^T, \boldsymbol{t}_d) \neq 0 \; \forall \; \gamma \in \boldsymbol{\Gamma}, \; \boldsymbol{q} \in \boldsymbol{Q}, \; \boldsymbol{t}_d \in \boldsymbol{T}_d \right\}. \tag{5.27}$$

This can be done using the robust stability analysis method from Chapter 3, i.e., by first determining boundaries between the stable and unstable regions and then checking stability of each disjoint region. The overall set of admissible parameters $\boldsymbol{P}_a$ is then given by

$$\boldsymbol{P}_a = \boldsymbol{P}_c \cap \boldsymbol{P}_s \tag{5.28}$$

These parameters may be used by the system designer to manually choose a suitable parametrization. Alternatively, a cost function can be minimized so as to optimize performance by automatically selecting parameters from the admissible parameter set. This is pursued in the following section.

5.4.2 Step Two: Performance Optimization

The performance criteria given in Section 5.3 can also be used to optimize system performance. Therefore, a weighted sum of different performance criteria may be used as cost function in the optimization, where *weighted sum* means that a frequency-dependent weighting function is defined for each considered performance criterion.

Cost Function

The sum of all performance criteria from Section 5.3 multiplied with their respective frequency-dependent weighting functions is[7]

$$\begin{aligned} &g(\omega, \boldsymbol{p}, \bar{\boldsymbol{q}}, \boldsymbol{Q}, \bar{\boldsymbol{t}}_d, \boldsymbol{T}_d) \\ &= W_t(\omega)|S(j\omega)| + W_d(\omega)|S_G(j\omega)| + W_m(\omega)|E(j\omega)| \\ &\quad + W_n(\omega)|T(j\omega)| + W_a(\omega)|S_K(j\omega)| \end{aligned}$$

[5] In this case, zero inclusion and exclusion is simply determined from the inner and outer approximation of the *real* Bernstein Polynomial (see 2.3.5) which is enlarged or shrunk respectively by taking the interval remainder into account.

[6] Stable parameters can also be determined using the set $\boldsymbol{P}_c$ of parameters satisfying the performance constraints as starting set, i.e., $\boldsymbol{P}_s \subseteq \boldsymbol{P}_c$. In general, this is computationally more efficient.

[7] In Section 5.3 a number of quite general performance criteria are given. For different applications other, more specific, performance criteria may be of interest. In the case of teleoperation systems, e.g., we could constrain or optimize transparency, i.e., the difference between the environment impedance and the impedance transmitted to the operator (see Section 4.4 for more details).

with tracking weight W_t, disturbance rejection weight W_d, model matching weight W_m, sensor noise rejection weight W_n, and actuator limit weight W_a. The frequency-dependent weighting functions (e.g., low-pass, band-pass, high-pass, or more complex transfer function) are used to specify which frequency range is of interest for the corresponding sensitivity function. Note that the sensitivity functions all depend on the parameters $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{t}_d$ which is not shown explicitly in the equation above. We can consider the cases of nominal performance or worst-case performance by selecting $\boldsymbol{q}$ and $\boldsymbol{t}_d$ appropriately, i.e., nominal performance can be considered by simply replacing some or all uncertain parameters $\boldsymbol{Q}$ or time delays $\boldsymbol{T}_d$ by nominal parameters $\bar{\boldsymbol{q}}$ and nominal time delays $\bar{\boldsymbol{t}}_d$.

Optimization

Once the sensitivity functions and weighted cost function have been determined the final goal is to optimize the parametrization within the admissible parameter set $\boldsymbol{P}_a$. Therefore, the global minimum of the cost function

$$\arg\min_{\boldsymbol{p}\in\boldsymbol{P}_a} g(\omega, \boldsymbol{p}, \bar{\boldsymbol{q}}, \boldsymbol{Q}, \bar{\boldsymbol{t}}_d, \boldsymbol{T}_d) \quad \forall\, 0 \leq \omega \leq \omega_{\max} \tag{5.29}$$

is determined. As all parameters in the admissible set $\boldsymbol{P}_a$ fulfill all constraints, no additional constraints have to be taken into account.

Different approaches could be used to solve the optimization problem. We propose to use the global optimization algorithm from [100] which is based on Taylor Models and the Bernstein transformation and is referred to as *Taylor-Bernstein Form.* The main reasons for choosing this algorithm are:

- Efficient global optimization algorithm for non-convex problems [100].
- Straightforward consideration of parametric uncertainties by minimizing the upper bound of g.
- Straightforward integration of performance constraints by only considering $\boldsymbol{p} \in \boldsymbol{P}_a$.

Moreover, the algorithm uses the same mathematical tools which are necessary to analyze stability and evaluate the performance constraints in Section 5.4.1. The additional complexity of implementing this global optimization algorithm is therefore relatively small.

5.5 Discussion

The algorithm to determine an optimal parametrization for LTI systems proposed in this chapter has not been implemented in practice yet. Therefore, this discussion section mainly focuses on difficulties which must be considered when implementing this method.

5.5.1 Necessary Extensions to Taylor-Bernstein Form

In existing global optimization algorithms using the Taylor-Bernstein Form the *uncertain* parameters are actually *unknown* parameters of which an optimal value should be found.

In our case these would correspond to the parameters $\boldsymbol{p} \in \boldsymbol{P} \in [\mathbb{R}]^o$. Therefore, to solve the optimization problem formulated in this chapter, existing algorithms for global optimization using the Taylor-Bernstein form must be extended so as to consider optimization variables as well as other uncertain parameters. Then, the upper bound of the cost function must be minimized. This should only require minimal changes to existing algorithms.

One additional extension to existing algorithms is necessary as the admissible parameter set in which we want to find the optimal solution is not necessarily an interval box: the optimization algorithm must be initialized using a whole *set* of interval boxes which together represent the admissible parameter set. As the optimization algorithm anyway consists of a branch and bound algorithm which performs the same operations on a number of interval boxes this is a trivial extension.

5.5.2 One-Step Solution of the Constrained Optimization Problem

For complex problems with a large number of tunable parameters the algorithm presented in this chapter determines the high-dimensional set of admissible parameters which can be computationally expensive. Alternatively, the solution can be computed in one step by directly solving the constrained optimization problem, i.e., instead of mapping stable regions and performance constraints to the parameter space and then optimizing performance, we directly solve the complete constrained optimization problem. On the one hand, this results in a more complex stability evaluation as it is not possible to use the two-step stability check, i.e., first determine stability boundaries and then check stability. Instead, the characteristic function must be evaluated for the complete region $\boldsymbol{\Gamma}$. On the other hand, this circumvents computing admissible regions in the parameter set $\boldsymbol{P}$ of dimension o, where the number of parameters o which should be optimized may be large. Therefore, depending on the given optimization problem this may be more or less efficient than the two-step solution proposed in Section 5.4. For this one-step solution, additional extensions to existing optimization algorithms are necessary as the Taylor-Bernstein Form has only been used for unconstrained global optimization up to now. Therefore, besides considering uncertain parameters in addition to optimization variables, constraints must be taken into account: *inequality constraints* due to the performance constraints and *binary constraints* due to stability.

Another approach which results in a one-step solution is to start out using the stability analysis algorithm from Chapter 3. Extending this algorithm by adding additional constraints is relatively straightforward. The only question in this context is how to decide on the subdivision direction. As the partial derivative as well as the interval remainder of each constraint will indicate a different subdivision direction a good heuristic to select one of these directions is necessary. Then, the open question is how to combine the stability analysis with a global optimization which is actually quite straightforward. For each region, bounds of the cost function are calculated in addition to checking stability and performance constraints. Then, within the branch and bound algorithm, the following cases must be discerned for each considered region:

- If the region is stable and all performance constraints are fulfilled we can proceed as for the unconstrained global optimization within this region and determine the parametrization which leads to the minimum value for the upper bound of the cost function.

- If the region is not stable or does not fulfill all performance constraints it is not considered further.
- Consider the case, where a region P_1 is stable and fulfills all performance constraints and has a value x for the upper bound of the cost function. Then, if the lower bound of the cost function in any other region P_2 is bigger than x, the region P_2 is not considered further as the global optimum cannot be in P_2.
- Otherwise, we proceed as in the case without performance optimization, i.e., we subdivide the region so as to determine if all constraints are fulfilled or one constraint is violated.

In practice some issues will probably arise when implementing this algorithm which are not easily foreseeable at the moment. The biggest difficulty is probably determining a robust and efficient heuristic to decide on the subdivision direction when checking performance constraints in addition to stability.

5.6 Summary and Future Work

In this chapter an extension of the stability analysis method presented in Chapter 3 was proposed which results in an automatic parameter tuning for time-delay systems with parametric uncertainties. Therefore, frequency-dependent sensitivity functions are determined for a general single-loop feedback system. Performance constraints are then imposed by mapping constraints on these sensitivity functions to the parameter space. Thereby, a set of parameters is determined which fulfills all performance constraints. Intersecting this parameter set with the set of robustly stable parameters which is determined using the method from Chapter 3 results in a set of admissible parameters which satisfy all performance constraints as well as robust stability. In a final step, the parametrization within this set of admissible parameters is determined which results in optimal system performance. Optimal system performance is specified using a cost function which consists of a sum of different sensitivity functions, where each sensitivity function is multiplied with an appropriate frequency-dependent weight.

The proposed algorithm has not yet been implemented in practice. Therefore, future work mainly consists of implementing and evaluating this algorithm. It is expected that the problem will be computationally intractable when considering more than two or three tunable parameters. This is caused by the fact that the set of admissible parameters has the same dimension as the number of tunable parameters. In this case the alternative one-step solution proposed in Section 5.5.2 may be a viable solution which would however require a significantly larger implementation effort.

6 Conclusions and Outlook

In this thesis we considered the stability and transparency analysis of haptic teleoperation systems. A haptic teleoperation system enables an operator to physically interact with a remote environment: the operator interacts with a haptic interface, the movements of this haptic interface are performed by a teleoperator that interacts with a remote environment, and the interaction force between teleoperator and remote environment is displayed to the operator. In this thesis, the teleoperation system was modeled as Linear Time-Invariant (LTI) system with time delay, due to the communication channel, and the operator and remote environment were both modeled as linear mass-spring-damper systems with unknown parameters. These modeling assumptions are common in literature on teleoperation and necessary for the application of the methods developed in this thesis. Using these assumptions, the overall system can be modeled as LTI time-delay system with parametric uncertainties. Two of the main goals that are generally pursued when designing teleoperation systems are robust stability when interacting with different remote environments and good transparency properties, i.e., the distortion introduced by the teleoperation system should be minimal.

Although the stability analysis of time-delay systems has been an active field of research for more than half a decade existing stability analysis methods are generally conservative, especially when considering systems with parametric uncertainties. Therefore, a new stability analysis method for LTI time-delay systems with parametric uncertainties was developed in Chapter 3. The method determines stable regions in the parameter space and is therefore suitable for stability analysis as well as controller design. It is based on a branch and bound algorithm which evaluates the zero-exclusion principle using the value set of the characteristic function of the time-delay system. The key to an efficient evaluation of zero exclusion and zero inclusion within the branch and bound algorithm is the novel method for evaluating the value set which was introduced in Chapter 2 and makes use of Taylor Models as well as the Bernstein form. The stability analysis method developed in Chapter 3 is not limited to teleoperation systems and can therefore be seen as the main contribution of this thesis. Essentially, the results are similar to the well-known *Parameter-Space Approach* by *Ackermann* [1] which allows for an intuitive stability analysis and controller design in the parameter space. Both our method and the *Parameter-Space Approach* can handle non-affine dependencies of the characteristic equation on the uncertain parameters, i.e., the coefficients of the characteristic equation may depend polynomially on the uncertain parameters. However, our newly developed method is applicable to a much larger class of systems. On the one hand, time-delay systems and even systems with incommensurate delays can be examined. On the other hand, a larger number of uncertain parameters can be considered as we can determine stable regions in a low-dimensional parameter space while additional parameters are constrained to intervals but otherwise unknown. This is especially interesting for controller design, where stable regions can be determined for one or two controller parameters while other plant parameters are not exactly known. In comparison with other robust stability analysis methods for time-delay systems, the main advantage of our method is that the results are not conservative when evaluating stability for a set of interval parameters. When mapping stability boundaries to the parameter space, the only conservatism is due to the specifiable resolution of the boundary mapping algorithm.

The second main contribution is a novel transparency analysis method for teleoperation systems which was introduced in Section 4.4. The method is based on the general four-channel architecture as a representation of the teleoperation system. Using this architecture and the transparency transfer function the environment impedance transmitted to the operator is determined. The transmitted impedance is calculated for a range of environments and a mass-spring-damper model is fit to the transmitted impedance. Thereby, we determine which mass, damping, and stiffness an operator feels when interacting with remote environments of different damping and stiffness. The result is an intuitive, easily interpretable *parameter-space transparency analysis*.

The third main contribution is a detailed analysis of time-delay dependent stability and transparency for different teleoperation architectures in Section 4.5. For this analysis, one specific haptic teleoperation system was considered and the following steps were carried out for the different teleoperation architectures. First, our stability analysis method was used to determine stable controller parameters for different time delays. In all cases, the operator impedance as well as environment impedance was assumed to be unknown but constrained to the same range of values. Once stable controller parameters had been determined we evaluated transparency for each of the architectures. The analysis showed that especially the transparency-optimized four-channel architecture and the position-based admittance control architecture with force exchange are interesting for teleoperation setups with small time delay as they offer good transparency for very small time delays (1 ms) and are robustly stable for larger time delays (up to 10 ms) at the cost of transparency. It must be noted that the analysis was only carried out for one specific teleoperation system and quantitative results of the stability analysis as well as transparency analysis depend directly on the system parameters. The qualitative results, however, are general and should hold for other teleoperation systems as long as the modeling assumptions, especially the linear mass-spring-damper models for human operator and environment, hold. Several aspects of this in-depth analysis of the stability and transparency of different teleoperation systems go beyond the state of the art and represent a contribution towards a deeper understanding of different teleoperation systems. First, we performed a parameter-space stability analysis which considered an unknown but bounded environment and operator impedance, time delay in the communication channel, and sensor and actuator filters. Before this work this has only been done for the delay-free case. Second, our transparency analysis method offered unique insights into the performance of different teleoperation architectures for different time delays.

The final contribution of this thesis is the outline of a method for optimal parametrization of linear time-invariant systems with time delays and parametric uncertainties in Chapter 5. Therefore, the stability analysis method from Chapter 3 is combined with performance constraints which are mapped to the parameter space alongside the stability constraints. Then, the global minimum of a cost function is determined from the set of feasible parameters for which all performance constraints as well as the stability constraint are fulfilled. The proposed formulation of performance constraints is similar to the frequency response magnitude specifications as presented by *Ackermann* in [1, Section 5.3] where a number of transfer functions which are defined for a general feedback system are constrained to a frequency-dependent value. Similarly, the cost function for the global optimization is defined as a weighted sum of these transfer functions (each weighted by a frequency-dependent function). Using the proposed method the globally optimal parametrization can be determined. Performance constraints as well as the cost function can depend on nominal parameters or uncertain interval parameters. Thereby, nominal performance or worst-case performance can be considered.

6.1 Outlook

The methods for stability and transparency analysis developed in this thesis can be further extended in a number of different directions. This is especially interesting for the stability analysis method as it is not only applicable to teleoperation systems but to a large class of time-delay systems. One possible future research direction could evaluate the convergence properties of the algorithm and develop strategies to assure convergence for the edge cases of marginal stability and a stability boundary on the lower or upper bound of a parameter. Another aspect where further improvement might be possible is the transformation of unbounded regions for the case with time delay. On the one hand, the transformation cannot be applied to arbitrary regions in the complex plane in this case, e.g., regions assuring a minimum damping. On the other hand, the time-delay term is replaced by an interval for infinite frequencies which may impair the convergence of the algorithm. Finally, simply applying the method to other time-delay systems would be very interesting.

Future work could also consider extensions to the transparency analysis method developed in this thesis. By combining the parameter identification of the transmitted impedance with psychophysical limits of human perception, namely the just noticeable difference of an impedance, a quantitative transparency measure could be derived which could, e.g., be used within a parameter optimization scheme. Furthermore, the effect of choosing a slightly more complex model for the approximated transmitted impedance could be evaluated for different models. The transparency analysis method could also be used to evaluate the transparency with which the operator *acts* on the environment in addition to the transparency with which the operator *perceives* the environment which is generally considered, e.g., also in this thesis. This *transparency of action* would characterize how transparently the impedance of the human arm is displayed to the environment. This would, e.g., indicate how accurately the compliance of the teleoperator can be controlled by the operator.

Another interesting topic of future research could be the evaluation of more complex teleoperation architectures, e.g., wave-variable approaches with local impedance controllers, using the methods developed in this thesis. For many of these architectures, either only conservative stability analysis results are available or a stability analysis in the presence of time delays and/or actuator and sensor dynamics was not possible to date. Moreover, there is often a lack of easily interpretable transparency analysis results. The stability and transparency analysis methods developed in this thesis could also be applied to multi-user teleoperation systems or bi-manual teleoperation systems, e.g., two teleoperation systems controlled by two operators (or by the two hands of one operator) interacting with a common object. The application of the stability analysis method is straightforward and should be computationally feasible in most cases. The most problematic scenario would probably be a multi-user setup with different time delays between different operator/teleoperator pairs, as incommensurate time delays generally lead to very long calculation times. Therefore, it might be necessary to work on improving the convergence of the stability analysis method for incommensurate time delays before examining such a complex case. The application of our transparency analysis method to multi-user teleoperation systems would require determining the transparency with which one operator perceives another operator in addition to the transparency with which the operator perceives the remote object. In this context it would also be necessary to evaluate what level of inter-operator transparency is necessary and what effect this transparency has on task performance.

Finally, the optimal parametrization method outlined in Chapter 5 could be implemented and evaluated. This would result in an automatic parametrization of fixed-structure LTI time-delay systems with parametric uncertainties. The number of parameters which could be simultaneously tuned using the two-step solution outlined in Chapter 5 would probably be limited to about two to four. A larger number of tunable parameters could probably be considered by implementing the one-step solution which is discussed in Section 5.5.2.

A Stability of LTI-Systems

In this appendix stability of LTI systems without time delay is first briefly summarized. Then, it is shown that these well known properties extend to systems with time delay.

A.1 Stability of LTI-Systems without Time Delay

This section summarizes some important stability notions for LTI systems.

Consider an LTI system with n states $\boldsymbol{x}$, input u, and output y given by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u \tag{A.1}$$

$$y = \boldsymbol{c}^T\boldsymbol{x} + du, \tag{A.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{nxn}$, $\boldsymbol{b} \in \mathbb{R}^n$, $\boldsymbol{c} \in \mathbb{R}^n$, and $d \in \mathbb{R}$.

The characteristic equation $p(\boldsymbol{s})$ of the system (A.1, (A.2)) is

$$p(\boldsymbol{s}) = \det(\boldsymbol{s}\boldsymbol{I} - \boldsymbol{A}). \tag{A.3}$$

The eigenvalues λ_i of the system matrix $\boldsymbol{A}$ correspond to the roots/zeros of p.

The transfer-function $G(\boldsymbol{s})$ of the system (A.1, (A.2)) is

$$G(\boldsymbol{s}) = \boldsymbol{c}^T(\boldsymbol{s}\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b} + d \tag{A.4}$$

$$= \frac{\sum_{j=0}^{o} b_j s^j}{\sum_{i=0}^{p} a_i s^i} = \frac{Z(\boldsymbol{s})}{N(\boldsymbol{s})}. \tag{A.5}$$

A.1.1 Asymptotic Stability

The system (A.1), (A.2) is asymptotically stable if and only if all eigenvalues λ_i are on the open left half-plane, i.e.

$$\mathrm{Re}(\lambda_i) < 0 \quad \forall 1 \leq i \leq n. \tag{A.6}$$

Thus, all roots/zeros of the characteristic equation must have a negative real part, i.e.

$$\mathrm{Re}(\boldsymbol{s}) < 0 \quad \forall p(\boldsymbol{s}) = 0. \tag{A.7}$$

A.1.2 Marginal Stability

The system (A.1), (A.2) is marginally stable if and only if all eigenvalues λ_i are on the closed left half-plane, i.e.

$$\mathrm{Re}(\lambda_i) \leq 0 \quad \forall\, 1 \leq i \leq n. \tag{A.8}$$

and all eigenvalues with real part *zero* are simple[1]. Thus, all roots/zeros of the characteristic equation must have a non-positive real part, i.e.

$$\mathrm{Re}(\boldsymbol{s}) \leq 0 \quad \forall p(\boldsymbol{s}) = 0 \tag{A.9}$$

and all roots that are *zero* must be simple:

$$\mathrm{Im}(\boldsymbol{s}_i) \neq \mathrm{Im}(\boldsymbol{s}_j) \quad \forall\, \mathrm{Re}(\boldsymbol{s}_i) = 0,\ \mathrm{Re}(\boldsymbol{s}_j) = 0,\ p(\boldsymbol{s}_i) = 0,\ p(\boldsymbol{s}_j) = 0,\ i \neq j \tag{A.10}$$

A.1.3 Instability

The system (A.1), (A.2) is not stable if and only if it is not asymptotically stable and it is not marginally stable.

A.2 Stability of LTI-Systems with Time Delay

As for the delay-free case, the roots of the characteristic function of a time-delay LTI system can be used to analyze stability. This was first shown in [22]. The key ideas which apply to the problem considered in this thesis are briefly summarized here.

A Linear Time-Invariant (LTI) system with n states $\boldsymbol{x}$, input u, and output y given by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{s}, \boldsymbol{t}_d)\boldsymbol{x} + \boldsymbol{b}u \tag{A.11}$$

$$y = \boldsymbol{c}^T\boldsymbol{x} + du \tag{A.12}$$

is considered, where

$$\boldsymbol{A}(\boldsymbol{s}, \boldsymbol{t}_d) = \sum_{i=0}^{m} \boldsymbol{A}_i e^{-t_{d,i}\boldsymbol{s}} \tag{A.13}$$

$$= \boldsymbol{A}_0 + \sum_{i=1}^{m} \boldsymbol{A}_i e^{-t_{d,i}\boldsymbol{s}} \tag{A.14}$$

and $\boldsymbol{A}_i \in \mathbb{R}^{n\times n}$, $\boldsymbol{b} \in \mathbb{R}^n$, $\boldsymbol{c} \in \mathbb{R}^n$, $d \in \mathbb{R}$. Furthermore, the system dynamics depend on the time delays $\boldsymbol{t}_d \in \mathbb{R}^{m+1}$ with

$$\boldsymbol{t}_d = [t_{d,0}\ t_{d,1}\ t_{d,2}\ \ldots\ t_{d,m}]^T \tag{A.15}$$

$$t_{d,0} = 0 \tag{A.16}$$

$$t_{d,i} > 0\ \forall\, 1 \leq i \leq m. \tag{A.17}$$

An equivalent representation of the system dynamics is

$$\dot{\boldsymbol{x}}(t) = \sum_{i=0}^{m} \boldsymbol{A}_i \boldsymbol{x}(t - t_{d,i}) + \boldsymbol{b}u(t) \tag{A.18}$$

$$= \boldsymbol{A}_0\boldsymbol{x}(t) + \sum_{i=1}^{m} \boldsymbol{A}_i \boldsymbol{x}(t - t_{d,i}) + \boldsymbol{b}u(t). \tag{A.19}$$

[1] The Eigenvectors corresponding to Eigenvalues which are *zero* must be different.

The system given in (A.11) is a so-called linear retarded functional differential equation. For stability analysis the autonomous system without input is considered, i.e., $u(t) = 0$. As for the delay-free case a characteristic function can be determined which is given by

$$p(\boldsymbol{s}, \boldsymbol{t}_d) = \det(\boldsymbol{sI} - \boldsymbol{A}(\boldsymbol{s}, \boldsymbol{t}_d)) = \sum_{i=0}^{m} \sum_{j=0}^{n} a_{ji} s^j e^{-t_{d,i}\boldsymbol{s}}. \tag{A.20}$$

In this case, the characteristic function is a so-called quasi-polynomial with an infinite number of roots due to the exponential function. The system (A.11) is asymptotically stable according to [22, Corollary 6.1] if and only if all roots of the characteristic function (A.20) have negative real parts. Similarly the system (A.11) is marginally stable according to [22, Corollary 6.2] if all roots of the characteristic function (A.20) have non-positive real parts and are simple.

These two corollaries represent the basis of frequency-domain stability analysis of linear retarded functional differential equations. They are based on the fact that for a system of the form (A.11), a bound of the value of the functional differential equation can be determined which is related to the initial condition and roots of the characteristic function [22, Theorem 6.7]. The proof of this theorem is rather involved. It is based on the uniqueness of the solution of the functional differential equation [22, Theorem 6.2] and relies on the fact that a system of the form (A.11) only has root chains of *retarded* type according to [22, Theorem 12.12][2]. For a more detailed discussion and the associated proofs the interested reader is referred to [22].

[2] For a time-delay system, roots of large magnitude are grouped in *chains* of either *retarded*, *neutral*, or *advanced* type. For a root chain of *retarded* type the roots of large magnitude have a large negative real part. For a root chain of *neutral* type, the real parts of roots of large magnitude approach a line parallel to the imaginary axis. Finally, for a root chain of *advanced* type, the real parts of roots of large magnitude have a large positive real part, and consequently, a time-delay system with root chains of *advanced* type cannot be stable.

Bibliography

[1] J. Ackermann and P. Blue, *Robust Control: The Parameter Space Approach.* Springer, 2002.

[2] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control.* MIT Press, 1992.

[3] M. Buss, A. Peer, T. Schauss, N. Stefanov, U. Unterhinninghofen, S. Behrendt, J. Leupold, M. Durkovic, and M. Sarkis, "Development of a Multi-modal Multi-user Telepresence and Teleaction System," *The International Journal of Robotics Research*, vol. 29, no. 10, pp. 1298–1316, 2009.

[4] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 624–637, 1993.

[5] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[6] K. Gu, V. L. Kharitonov, and J. Chen, *Stability of Time-Delay Systems.* Control Engineering, Birkhäuser, 2003.

[7] M. Zettler and J. Garloff, "Robustness analysis of polynomials with polynomial parameter dependency using Bernstein expansion," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 425–431, 1998.

[8] A. Gonchar and B. Shabat, "Analytic function," in *Encyclopedia of Mathematics*, Kluwer Academic Publishers, 2002.

[9] E. Chirka, "Preservation of domain, principle of," in *Encyclopedia of Mathematics*, Kluwer Academic Publishers, 2002.

[10] R. E. Moore, *Interval Arithmetic and Automatic Error Analysis in Digital Computing.* Ph.d., Stanford University, Stanford, CA, USA, 1962.

[11] R. E. Moore, *Interval Analysis.* Englwood Cliffs, NJ, USA: Prentice Hall, 1966.

[12] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis.* Philadelphia, PA, USA: Society for Industrial and Applied Mathematics (SIAM), 2009.

[13] J. Garloff, "The Bernstein Algorithm," *Interval Computations*, vol. 2, no. 6, pp. 154–168, 1993.

[14] R. T. Farouki, "The Bernstein polynomial basis: A centennial retrospective," *Computer Aided Geometric Design*, vol. 29, no. 6, pp. 379–419, 2012.

[15] R. T. Farouki and V. T. Rajan, “Algorithms for polynomials in Bernstein form,” *Computer Aided Geometric Design*, vol. 5, no. 1, pp. 1–26, 1988.

[16] R. Graham, “An efficient algorith for determining the convex hull of a finite planar set,” *Information Processing Letters*, vol. 1, no. 4, pp. 132–133, 1972.

[17] A. Neumaier, “Taylor Forms—Use and Limits,” *Reliable Computing*, vol. 9, no. 1, pp. 43–79, 2003.

[18] K. Makino and M. Berz, “Taylor models and other validated functional inclusion methods,” *International Journal of Pure and Applied Mathematics*, vol. 4, no. 4, pp. 379–456, 2003.

[19] M. Berz and G. Hoffstätter, “Computation and application of Taylor polynomials with interval remainder bounds,” *Reliable Computing*, vol. 4, pp. 83–97, 1998.

[20] J. Garloff, “Convergent bounds for the range of multivariate polynomials,” in *Interval Mathematics* (K. Nickel, ed.), vol. 212 of *Lecture Notes in Computer Science*, pp. 37–56, Springer Berlin / Heidelberg, 1986.

[21] J. Garloff, B. Graf, and M. Zettler, “Speeding up an algorithm for checking robust stability of polynomials,” in *2nd IFAC Symposium on Robust Control Design* (C. Banyasz, ed.), (Oxford), pp. 183–188, Elsevier, 1998.

[22] R. Bellman and K. Cooke, *Differential-Difference Equations.* New York: Academic Press, 1963.

[23] V. L. Kharitonov, “Asympotic stability of an equilibrium position of a family of systems of linear differntial equations,” *Differntia Uravnen (Russian)*, vol. 14, no. 11, pp. 1483–1485, 1978.

[24] A. C. Bartlett, C. V. Hollot, and H. Lin, “Root Locations of an Entire Polytope of Polynomials: It Suffices to Check the Edges,” in *American Control Conference (ACC)*, (Minneapolis, MN, USA), pp. 1611–1616, 1987.

[25] P. Misra, “On stabilization of systems with uncertain parameters: An interval arithmetic approach,” in *American Control Conference (ACC)*, (Pittsburg, PA, USA), pp. 657–658, 1989.

[26] G. Zames, “Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms, and Approximate Inverses,” *IEEE Transactions on Automatic Control*, vol. 26, pp. 301–320, 1981.

[27] M. G. Safonov, “Stability margins of diagonally perturbed multivariable feedback systems,” in *Conference on Decision and Control (CDC)*, pp. 251–256, IEEE, 1981.

[28] J. Doyle, “Analysis of feedback systems with structured uncertainties,” *IEE Proceedings D Control Theory and Applications*, vol. 129, no. 6, p. 242, 1982.

[29] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control.* Prentice Hall, Englwood Cliffs, New Jersey, 1996.

[30] S. Malan, M. Milanese, M. Taragna, and J. Garloff, "B3 algorithm for robust performances analysis in presence of mixed parametric and dynamic perturbations," in *Conference on Decision and Control (CDC)*, pp. 128–133, IEEE, 1992.

[31] F. Montagner and P. L. D. Peres, "A new LMI condition for the robust stability of linear time-varying systems," in *Conference on Decision and Control (CDC)*, no. December, (Maui, Hawaii, USA), pp. 6133–6138, IEEE, 2003.

[32] S.-I. Niculescu, *Delay effects on stability: A robust control approach*, vol. 269. Springer, 2001.

[33] R. Sipahi, S.-I. Niculescu, C. T. Abdallah, W. Michiels, and K. Gu, "Stability and Stabilization of Systems with Time Delay," *IEEE Control Systems*, vol. 31, no. 1, pp. 38–65, 2011.

[34] T. Haag, U. Münz, and F. Allgöwer, "Comparison of different stability conditions for linear time-delay systems with incommensurate delays," in *8th IFAC Workshop on Time-Delay Systems*, (Criaova, Romania), pp. 136–141, 2009.

[35] A. Thowsen, "An analytic stability test for a class of time-delay systems," *IEEE Transactions on Automatic Control*, vol. 26, no. 3, pp. 735–736, 1981.

[36] D. Hertz, E. I. Jury, and E. Zeheb, "Simplified analytic stability test for systems with commensurate time delays," *Control Theory and Applications, IEE Proceedings D*, vol. 131, no. 1, pp. 52–56, 1984.

[37] N. MacDonald, "Comments on a simplified analytical stability test for systems with delay," *Control Theory and Applications, IEE Proceedings D*, vol. 132, no. 5, pp. 237–238, 1985.

[38] E. I. Jury and E. Zeheb, "On stability test for a class of distributed parameter systems with delays," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 10, pp. 1027–1028, 1986.

[39] N. Olgac and R. Sipahi, "An exact method for the stability analysis of time-delayed linear time-invariant (LTI) systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 5, pp. 793–797, 2002.

[40] U. Münz, C. Ebenbauer, T. Haag, and F. Allgöwer, "Stability Analysis of Time-Delay Systems With Incommensurate Delays Using Positive Polynomials," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1019–1024, 2009.

[41] C. Ebenbauer and F. Allgöwer, "Stability Analysis for Time-Delay Systems using Rekasius's Substitution and Sum of Squares," in *Conference on Decision and Control (CDC)*, (San Diego, CA, USA), pp. 5376–5381, IEEE, 2006.

[42] U. Münz, C. Ebenbauer, and F. Allgöwer, “Stability of Networked Systems with Multiple Delays Using Linear Programming,” in *American Control Conference (ACC)*, (New York, NY, USA), pp. 5515–5520, IEEE, 2007.

[43] E. Jarlebring, “Computing the stability region in delay-space of a TDS using polynomial eigenproblems,” in *6th IFAC Workshop on Time Delay Systems*, (L’Aquila, Italy), pp. 296–301, 2006.

[44] K. Gu, S.-I. Niculescu, and J. Chen, “On stability crossing curves for general systems with two delays,” *Journal of Mathematical Analysis and Applications*, vol. 311, no. 1, pp. 231–253, 2005.

[45] K. Gu and M. Naghnaeian, “Stability Crossing Set for Systems With Three Delays,” *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 11–26, 2011.

[46] C. Knospe and M. Roozbehani, “Stability of linear systems with interval time delays excluding zero,” *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1271–1288, 2006.

[47] K. Gu, “Refined discretized Lyapunov functional method for systems with multiple delays,” in *Conference on Decision and Control (CDC)*, vol. 1, (Orlando, FL, USA), pp. 985–990, IEEE, 2001.

[48] M. Fu, A. Olbrot, and M. Polis, “Robust stability for time-delay systems: the edge theorem and graphical tests,” in *Conference on Decision and Control (CDC)*, (Austin, TX, USA), pp. 98–105, IEEE, 1988.

[49] V. L. Kharitonov and J. A. Torres-Muñoz, “Robust Stability of Multivariate Polynomials. Part 1: Small Coefficient Perturbations,” *Multidimensional Systems and Signal Processing*, vol. 10, no. 1, pp. 7–20, 1999.

[50] V. L. Kharitonov, J. A. Torres-Muñoz, and M. I. Ramirez-Sosa, “Robust Stability of Multivariate Polynomials, Part 2: Polytopic Coefficient Variations,” *Multidimensional Systems and Signal Processing*, vol. 10, no. 1, pp. 21–32, 1999.

[51] V. L. Kharitonov, M. I. Ramirez-Sosa, and J. A. Torres-Muñoz, “Robust Stability of Multivariate Polynomials, Part 3: Frequency Domain Approach,” *Multidimensional Systems and Signal Processing*, vol. 11, no. 3, pp. 213–231, 2000.

[52] N. Hohenbichler, “All stabilizing PID controllers for time delay systems,” *Automatica*, vol. 45, no. 11, pp. 2678–2684, 2009.

[53] F. Schrödel and D. Abel, “Expanding the parameter space approach to multi loop control with multi time delays,” in *European Control Conference (ECC)*, pp. 73–78, IEEE, 2014.

[54] L. A. Zadeh and C. A. Desoer, *Linear system theory: the state space approach.* McGraw-Hill series in system science, McGraw-Hill, 1963.

[55] R. A. Frazer and W. J. Duncan, "On the Criteria for the Stability of Small Motions," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 124, no. 795, pp. 642–654, 1929.

[56] S.-I. Niculescu, E. Verriest, L. Dugard, and J.-M. Dion, "Stability and robust stability of time-delay systems: A guided tour," in *Stability and Control of Time-delay Systems* (L. Dugard and E. Verriest, eds.), pp. 1–71, Springer Berlin / Heidelberg, 1998.

[57] M. Fu and B. R. Barmish, "Polytopes of polynomials with zeros in a prescribed set," *IEEE Transactions on Automatic Control*, vol. 34, no. 5, pp. 544–546, 1989.

[58] B. R. Barmish, "New tools for robustness analysis," in *Conference on Decision and ControlControl*, no. December, (Austin, TX, USA), pp. 1–6, IEEE, 1988.

[59] J. Chen, S.-I. Niculescu, and P. Fu, "Robust Stability of Quasi-Polynomials: Frequency-Sweeping Conditions and Vertex Tests," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1219–1234, 2008.

[60] J. Chiasson and C. T. Abdallah, "Robust stability of time delay systems: Theory," in *3rd IFAC Workshop on Time Delay Systems*, (Santa Fe, NM, USA), pp. 125–130, 2001.

[61] T. Schauss and A. Peer, "Parameter-space transparency analysis of teleoperation systems," in *Haptics Symposium (HAPTICS)*, (Vancouver, BC, Canada), pp. 111–116, IEEE, 2012.

[62] B. Hannaford, "A design framework for teleoperators with kinesthetic feedback," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 426–434, 1989.

[63] R. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," *IEEE Transactions on Automatic Control*, vol. 34, no. 5, pp. 494–501, 1989.

[64] G. Niemeyer and J.-J. E. Slotine, "Stable adaptive teleoperation," *IEEE Journal of Oceanic Engineering*, vol. 16, no. 1, pp. 152–162, 1991.

[65] D. A. Lawrence, "Designing teleoperator architectures for transparency," in *International Conference on Robotics and Automation (ICRA)*, (Nice, France), pp. 1406–1411, IEEE, 1992.

[66] Y. Yokokohji and T. Yoshikawa, "Bilateral Control of Master-Slave Manipulators for Ideal Kinesthetic Coupling - Formulation and Experiment," *IEEE Transactions on Robotics and Automation*, vol. I0, no. 5, pp. 605–620, 1994.

[67] R. J. Adams and B. Hannaford, "Stable haptic interaction with virtual environments," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 465–474, 1999.

[68] K. Hashtrudi-Zaad and S. E. Salcudean, “Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators,” *The International Journal of Robotics Research*, vol. 20, no. 6, pp. 419–445, 2001.

[69] F. B. Llewellyn, “Some fundamental properties of transmission systems,” *Proceedings of the IRE*, vol. 2, no. 1, pp. 271–283, 1952.

[70] A. Haddadi and K. Hashtrudi-Zaad, “Delay-Robust Transparent Bilateral Teleoperation Control Design,” in *International Conference on Intelligent Robots and Systems (IROS)*, no. 1, pp. 22–26, IEEE/RSJ, 2008.

[71] A. Haddadi and K. Hashtrudi-Zaad, “A new robust stability analysis and design tool for bilateral teleoperation control systems,” in *International Conference on Robotics and Automation (ICRA)*, pp. 663–670, 2008.

[72] A. Kron and G. Schmidt, “Stability and Performance Analysis of Kinesthetic Control Architectures for Bimanual Telepresence Systems,” *Journal of Intelligent and Robotic Systems*, vol. 46, no. 1, pp. 1–26, 2006.

[73] A. Peer, *Design and Control of Admittance-Type Telemanipulation Systems.* Ph.d., Technische Universität München, 2008.

[74] J. E. Colgate and J. M. Brown, “Factors affecting the Z-Width of a haptic display,” in *International Conference on Robotics and Automation (ICRA)*, (San Diego, CA, USA), pp. 3205–3210, IEEE Comput. Soc. Press, 1994.

[75] S. Hirche, A. Bauer, and M. Buss, “Transparency of haptic telepresence systems with constant time delay,” in *Conference on Control Applications (CCA)*, pp. 328–333, IEEE, 2005.

[76] J.-J. E. Slotine and W. Li, “On the Adaptive Control of Robot Manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.

[77] W.-H. Zhu, Y.-G. Xi, Z.-J. Zhang, Z. Bien, and J. De Schutter, “Virtual Decomposition Based Control for Generalized High Dimensional Robotic Systems with Complicated Structure,” *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 411–436, 1997.

[78] W.-H. Zhu and S. E. Salcudean, “Stability guaranteed teleoperation: an adaptive motion/force control approach,” *IEEE Transactions on Automatic Control*, vol. 45, no. 11, pp. 1951–1969, 2000.

[79] K. Hashtrudi-Zaad and S. E. Salcudean, “Bilateral parallel force/position teleoperation control,” *Journal of Robotic Systems*, vol. 19, no. 4, pp. 155–167, 2002.

[80] D. A. Lawrence and J. D. Chapel, “Performance trade-offs for hand controller design,” in *International Conference on Robotics and Automation (ICRA)*, (San Diego, CA, USA), pp. 3211–3216, IEEE, 1994.

[81] K. H. Hunt and F. R. E. Crossley, “Coefficient of Restitution Interpreted as Damping in Vibroimpact,” *Journal of Applied Mechanics*, vol. 42, no. 2, p. 440, 1975.

[82] M. D. Sidman, F. E. DeAngelis, and G. C. Verghese, “Parametric system identification on logarithmic frequency response data,” *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1065–1070, 1991.

[83] R. Pintelon, P. Guillaume, Y. Rolain, J. Schoukens, and H. Van Hamme, “Parametric identification of transfer functions in the frequency domain-a survey,” *IEEE Transactions on Automatic Control*, vol. 39, no. 11, pp. 2245–2260, 1994.

[84] T. L. Brooks, “Telerobotic Response Requirements,” in *International Conference on Systems, Man, and Cybernetics*, pp. 113–120, IEEE, 1990.

[85] N. Hogan, “Controlling impedance at the man/machine interface,” in *International Conference on Robotics and Automation (ICRA)*, pp. 1626–1631 vol.3, 1989.

[86] M. Kuschel, *Visual-Haptic Presence Systems: Utility Optimization, Compliance Perception, and Data Compression.* Ph.d., Technische Universität München, 2009.

[87] S. Hirche, *Haptic Telepresence in Packet Switched Communication Networks.* Ph.d., Technische Universität München, 2005.

[88] G. Niemeyer and J.-J. E. Slotine, “Telemanipulation with Time Delays,” *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 873–890, 2004.

[89] G. Niemeyer and J.-J. E. Slotine, “Designing force reflecting teleoperators with large time delays to appear as virtual tools,” in *International Conference on Robotics and Automation (ICRA)*, vol. 3, (Albuquerque, NM, USA), pp. 2212–2218, IEEE, 1997.

[90] G. Niemeyer and J.-J. E. Slotine, “Using wave variables for system analysis and robot control,” in *International Conference on Robotics and Automation (ICRA)*, vol. 2, (Albuquerque, NM, USA), pp. 1619–1625, IEEE, 1997.

[91] Yongqiang Ye and P. X. Liu, “Improving Trajectory Tracking in Wave-Variable-Based Teleoperation,” *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 2, pp. 321–326, 2010.

[92] H. Li and K. Kawashima, “Achieving Stable Tracking in Wave-Variable-Based Teleoperation,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 5, pp. 1574–1582, 2014.

[93] H. Tanaka, T. Schauss, K. Ohnishi, A. Peer, and M. Buss, “A Coordinating Controller for Improved Task Performance in Multi-user Teleoperation,” in *Haptics: Generating and Perceiving Tangible Sensations* (A. Kappers, J. van Erp, W. Bergmann Tiest, and F. van der Helm, eds.), vol. 6191 of *Lecture Notes in Computer Science*, pp. 155–160, Springer Berlin / Heidelberg, 2010.

[94] T. Schauss, U. Unterhinninghofen, and M. Buss, “Unlimited Workspace - Coupling a Mobile Haptic Interface with a Mobile Teleoperator,” in *Human Centered Robot Systems* (H. Ritter, G. Sagerer, R. Dillmann, and M. Buss, eds.), vol. 6 of *Cognitive Systems Monographs*, pp. 33–41, Springer Berlin Heidelberg, 2009.

[95] T. Schauss, R. Groten, A. Peer, and M. Buss, “Evaluation of a Coordinating Controller for Improved Task Performance in Multi-user Teleoperation,” in *Haptics: Generating and Perceiving Tangible Sensations* (A. Kappers, J. van Erp, W. Bergmann Tiest, and F. van der Helm, eds.), vol. 6191 of *Lecture Notes in Computer Science*, pp. 240–247, Springer Berlin / Heidelberg, 2010.

[96] T. Schauss, C. Passenberg, N. Stefanov, D. Feth, I. Vittorias, A. Peer, S. Hirche, M. Buss, M. Rothbucher, K. Diepold, J. Kammerl, and E. Steinbach, “Beyond classical teleoperation: Assistance, cooperation, data reduction, and spatial audio,” in *International Conference on Robotics and Automation (ICRA)*, (Saint Paul, Minnesota, USA), pp. 3553–3554, IEEE, 2012.

[97] M. Rank, T. Schauss, A. Peer, S. Hirche, and R. Klatzky, “Masking Effects for Damping JND,” in *Haptics: Perception, Devices, Mobility, and Communication* (P. Isokoski and J. Springare, eds.), vol. 7283 of *Lecture Notes in Computer Science*, pp. 145–150, Springer Berlin / Heidelberg, 2012.

[98] E. M. Rank, *Dynamic Models of Human Perception and Action and Their Application in Telepresence.* PhD thesis, Technische Universität München, München, 2012.

[99] P. Apkarian, M. N. Dao, and D. Noll, “Parametric Robust Structured Control Design,” *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1857–1869, 2015.

[100] P. S. V. Nataray and K. Kotecha, “An algorithm for global optimization using the Taylor–Bernstein form as inclusion function,” *Journal of Global Optimization*, vol. 24, no. 4, pp. 417–436, 2002.