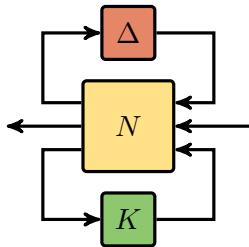




Uncertain systems: analysis and synthesis with application to flutter suppression control



Ph.D. Thesis by
BÁLINT PATARTICS

In Partial Fulfillment of the Requirements for the Degree of Doctor of
Philosophy

Supervisor:
Dr. Bálint Vanek

October 14, 2022

Soli Deo gloria

Declaration

Undersigned, Bálint Patartics, hereby state that this Ph.D. Thesis is my own work wherein I have used only the sources listed in the Bibliography. All parts taken from other works, either in a word for word citation or rewritten keeping the original contents, have been unambiguously marked by a reference to the source.

Nyilatkozat

Alulírott Patartics Bálint kijelentem, hogy ezt a doktori értekezést magam készítettem és abban csak a megadott forrásokat használtam fel. minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettettem, egyértelműen, a forrás megadásával megjelöltetem.

Budapest, 2022. 09. 05.

Patartics Bálint

The reviews of this Ph.D. Thesis and the record of defense will be available later in the Dean Office of the Faculty of Transportation Engineering and Vehicle Engineering of the Budapest University of Technology and Economics.

Az értekezésről készült bírálatok és a jegyzőkönyv a későbbiekben a Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésmérnöki és Járműmérnöki Karának Dékáni Hivatalában elérhetőek.

Abstract

Robustness has been a central idea of control design ever since the introduction of the Bode and Nyquist stability criteria. Beyond gain and phase margins, modern control theory increasingly makes use of model uncertainty to enable the direct handling of neglected dynamics and inaccuracies in physical parameters of the plant. The robust controller design task is to synthesize a controller with which the closed-loop is stable and meets the performance requirements for all possible plants in the set characterized by the uncertainty. Appropriate analysis tools are also needed to verify stability and performance for uncertain systems.

In the present thesis, a robust control design and a robustness analysis method are developed. The synthesis algorithm is capable of optimizing a controller with fixed structure in the presence of mixed (both dynamic and parametric) uncertainty. The proposed method alternates between an analysis step and a synthesis step. Samples of the parametric uncertainty are computed during the analysis steps thus yielding an array of uncertain systems containing only dynamic uncertainty. The controller is then synthesized on this array of uncertain models. This synthesis step itself involves an alternation between constructing a D-scale for each of the uncertain systems and tuning a single controller for the entire collection of scaled plants. The controller tuning is performed using existing structured control design techniques.

The goal of the presented analysis technique is to evaluate the behavior of the closed-loop by selecting a worst-case uncertainty sample. In the analysis of uncertain systems, we often search for a worst-case perturbation that drives the H_∞ gain of the system to the maximum over the set of allowable uncertainties. Employing the classical technique, an uncertainty sample is obtained that indeed maximizes the gain but only at the single frequency where that maximum occurs. In contrast, this thesis provides a method to calculate a worst-case perturbation that maximizes the gain of a system with mixed uncertainty at multiple frequencies simultaneously. This approach involves a nonlinear optimization that selects the worst-case value of the uncertain parameters and the application of the boundary Nevanlinna-Pick interpolation to calculate the dynamic uncertainty sample. Such a perturbation can be used to augment Monte Carlo simulations of uncertain systems, especially if the system has multiple resonance frequencies.

This thesis also considers the solution of a practical control problem: the flutter control design method for a flexible aircraft. The dynamics of this aircraft are described by both a high fidelity and a reduced-order control-oriented model. The latter is used to obtain the uncertain design model. The synthesis objective for flutter suppression is to achieve robust stabilization in the presence of mixed uncertainty with limited actuator effort. The proposed structured robust synthesis algorithm is applied for the control design. It yields a single, low order, linear time-invariant controller which increases the flutter speed by 15%. Additional robustness analyses, including multi-frequency worst-case uncertainty construction, and high fidelity simulations are provided to assess the controller performance.

Kivonat

A Bode- és a Nyquist-stabilitáskritérium megjelenése óta a robusztusság az szabályozótervezés központi fogalma. A fázis- és erősítéstartalékon túlmutatóan, a modern irányításelmélet egyre inkább alkalmazza a modellbizonytalanságot, ezzel lehetővé téve elhanyagolt rendszerdinamika és a fizikai paraméterek pontatlan ismeretének közvetlen figyelembe vételét. A robusztus szabályozótervezés célja olyan szabályozó előállítása, amely stabilizálja a lehetséges zárt rendszerek halmazát, amelyet a bizonytalanság határoz meg. Ehhez szükségünk van megfelelő, bizonytalan rendszerek stabilitás és performancia analízisére alkalmas, matematikai eszközökre.

A jelen disszertáció egy robusztus szabályozó tervezésre és egy robusztus analízisre alkalmas eljárást is bemutat. A szintézis módszer egy rögzített struktúrájú szabályozó optimális hangolására képes kevert (azaz dinamikus és parametrikus) bizonytalansággal terhelt rendszerek esetén. Az algoritmus egy szintézis és egy analízis lépés között alternál. Az analízis lépés során a bizonytalan paraméterek worst-case mintáit számítjuk ki, amely eredményeként egy sereg bizonytalan rendszert kapunk, amelyekben már csak dinamikus bizonytalanság van. A szabályozót ehhez a bizonytalan rendszer seregez tervezük. A szintézis lépés maga is iteratív. Két lépés váltja egymást: D-skálák számítása a bizonytalan rendszerseregről minden eleméhez és egyetlen strukturált szabályozó hangolása az összes D-skálával ellátott rendszerhez. A szabályozó optimális hangolása létező strukturált tervezési módszerekkel történik.

A disszertációban bemutatott analízis módszer célja egy zárt szabályozási kör viselkedésének vizsgálata egy worst-case bizonytalanság minta előállításával. Bizonytalan rendszerek analízise során gyakran keressük a bizonytalanság worst-case értékét, amely maximalizálja a rendszer H_∞ erősítését a megengedett bizonytalanság halmaz felett. Az erre szolgáló klasszikus módszer valóban maximalizálja a rendszer erősítését, azonban csak a maximumhoz tartozó egyetlen frekvencia pontban. Ezzel szemben az itt bemutatott módszerrel olyan worst-case bizonytalanság minta számítható, amely a kevert bizonytalansággal terhelt rendszer erősítését több frekvencia pontban egyidejűleg maximalizálja. Ehhez nemlineáris optimalizálás segítségével keressük meg a bizonytalan paraméterek worst-case értékét, a bizonytalanság minta dinamikus részét pedig a Nevanlinna-Pick interpoláció egy speciális változatával állítjuk elő. Az így nyert worst-case bizonytalanság alkalmazható egy bizonytalan rendszer Monte Carlo szimulációjában, különösen olyan rendszer esetén, amely több rezonanciafrekvenciával is rendelkezik.

A jelen disszertáció egy gyakorlati problémával is foglalkozik: egy flexibilis repülőgép flatter elnyomó szabályozásával. A repülőgép dinamikáját egy nagy megbízhatóságú és egy redukált modellel is leírjuk. Az utóbbiból nyerjük a szabályozótervezéshez használt, bizonytalanságot is tartalmazó modellt. A tervezés célja a robusztus stabilitás biztosítása kevert bizonytalansággal szemben, a beavatkozó szervek terhelésének minimalizálásával. Az eredményül kapott kis dimenziós, lineáris időinvariáns szabályozó a flatter sebességet 15%-kal növeli meg. A szabályozó teljesítményét a többfrekvenciás worst-case bizonytalanság konstrukciót is felhasználó robusztus analízis, valamint a nagy megbízhatóságú modellel végzett szimuláció mutatja be.

Contents

Declaration	i
Abstract	ii
Kivonat	iii
Contents	iv
Nomenclature	v
1 Introduction	1
1.1 Robust synthesis	1
1.2 Worst-case analysis	2
1.3 Flutter suppression control	3
2 Theoretical foundation	5
2.1 Nominal systems and H_∞ control synthesis	5
2.1.1 Linear time-invariant systems	5
2.1.2 H_∞ synthesis	6
2.2 Robust stability and μ -synthesis	9
2.2.1 Uncertain LTI systems	9
2.2.2 Structured singular value	11
2.2.3 μ -synthesis	14
2.3 Worst-case gain	17
2.3.1 Upper bound	19
2.3.2 Lower bound	20
2.3.3 Worst-case gain as control objective	23
3 Structured robust synthesis method to minimize worst-case gain	25
3.1 Related literature	25
3.2 Problem statement	26
3.3 Overview of the design algorithm	27
3.4 Parameter-dependent D-scales	29
3.5 Structured synthesis for the collection of scaled plants	31
3.6 Worst-case analysis of the closed-loop	32
3.7 Numerical examples	33

3.7.1	Benchmark example from [2]	33
3.7.2	Example requiring parameter dependent D-scales	34
3.7.3	Further numerical testing	37
Thesis 1		38
4 Multi-frequency worst-case uncertainty construction		40
4.1	Related literature	40
4.2	Problems statement	41
4.3	Boundary Nevanlinna-Pick interpolation	41
4.4	Interpolation of the dynamic uncertainty	45
4.5	Optimization of the parametric uncertainty	48
4.6	Construction of the worst-case mixed uncertainty	50
4.7	Basic Validation of the Method	51
4.8	Analysis of a hard-disk drive control system	54
4.8.1	The Control Problem and Motivation	55
4.8.2	Uncertain Model of the Closed-Loop	56
4.8.3	Worst-case Analysis of the Closed-Loop	57
Thesis 2		58
5 Flutter suppression control of a flexible aircraft		61
5.1	Related literature	61
5.2	Aircraft model	62
5.2.1	Nonlinear high fidelity model of the flexible aircraft	62
5.2.2	Bottom-up model reduction	65
5.3	Flutter control synthesis	68
5.3.1	Design model	68
5.3.2	Performance specification	71
5.3.3	Synthesis	73
5.4	Analysis based on worst-case uncertainty construction	74
5.5	Validation of the flutter controller using the non-linear high fidelity model	77
5.5.1	Baseline control architecture	78
5.5.2	Closed-loop stability and performance evaluation	80
5.5.3	Robustness analysis	81
5.5.4	Time domain simulations	84
5.5.5	Comparison to potential alternatives	87
Thesis 3		87
6 Conclusions		88
Acknowledgements		90
Bibliography		91
Publications of the author		91
References		92

Nomenclature

Acronyms

ASE	AeroServoElastic
BNP	Boundary Nevanlinna-Pick
FLEXOP	flutter free FLight Envelope eXpansion for ecOnomical Performance improvement (project)
FliPASED	Flight Phase adaptive Aero-Servo-Elastic aircraft Design methods (project)
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LFT	Linear Fractional Transformation
LMI	Linear Matrix Inequality
LTI	Linear Time Invariant
MATLAB	MATrix LABoratory (software)
MIMO	Multiple-Input Multiple-Output
OLFS	Open-Loop Flutter Speed
PID	Proportional-Integral-Derivative
RFS	Robust Flutter Speed
SDP	SemiDefinite Program
SISO	Single-Input Single-Output
T-FLEX	name of the aircraft in the FLEXOP and FliPASED projects

Sets

\mathbb{R}	Real numbers
\mathbb{C}	Complex numbers
Δ	Block structured complex matrix uncertainties
\triangle	Block structured unit norm bounded LTI uncertainties
Δ_p	Block structured unit norm bounded parametric uncertainties
$\Delta_{p,s}$	Samples of the block structured unit norm bounded parametric uncertainties in the structured robust synthesis algorithm

Δ_d	Block structured unit norm bounded dynamic uncertainties
\mathbb{Q}	Block structured, unit norm bounded matrix samples of the uncertainty block
\mathbb{Q}_d	Block structured, unit norm bounded matrix samples of the dynamic uncertainty block
\mathbf{D}	D-scales as complex matrices
\mathbb{D}	D-scales as LTI systems
\mathbb{D}_d	D-scales corresponding to the dynamic uncertainty block as LTI systems
$\mathbb{D}_{d,s}$	Samples of the D-scales corresponding to the dynamic uncertainty block as LTI systems in the structured robust synthesis

Other symbols

$<, \leq, >, \geq$	Scalar inequalities
$\prec, \preceq, \succ, \succeq$	Matrix inequalities, e.g., $A \preceq B$ means $A - B$ is negative semidefinite
u	Control input signal of an LTI system
y	Measured output signal of an LTI system
x	State of an LTI system
d	Generalized error signal of a generalized plant
e	Generalized error signal of a generalized plant
v	Output of the uncertainty block $\Delta(s)$
z	Input of the uncertainty block $\Delta(s)$
A, B, C, D	State-space matrices of an LTI system
ω	Frequency in rad/s
$G(s)$	Transfer function of the plant
$K(s)$	Transfer function of the controller
κ	Design parameter vector of a structured controller
$K(\kappa, s)$	Transfer function of the structured controller
$P(s)$	Transfer function of the generalized plant
$T(s)$	Transfer function of a closed-loop control system
$S(s)$	Sensitivity function of a closed-loop control system
$W(s)$	Performance weight in the H_∞ synthesis
$ \cdot $	Absolute value of a complex number
$\bar{\sigma}(\cdot)$	Largest singular value of a complex matrix
$\ \cdot\ _\infty$	H_∞ norm of an LTI system
$\mu_{\Delta}(\cdot)$	Structured singular value (or μ) of a complex matrix with respect to the uncertainty block structure Δ
$\tilde{\mu}_{\Delta}(\cdot)$	Skewed structured singular value (or skewed- μ) of a complex matrix with respect to the uncertainty block structure Δ
$\mathcal{F}_U(\cdot, \cdot)$	Upper LFT of matrices or LTI systems
$\mathcal{F}_L(\cdot, \cdot)$	Lower LFT of matrices or LTI systems

$J(\cdot)$	Objective function of an optimization
$\Delta(s)$	Transfer function of the mixed uncertainty block ($\Delta(s) \in \mathbb{A}$)
$G(\Delta, s)$	Transfer function of an uncertain LTI system
Δ_p	Instance of the parametric uncertainty block ($\Delta_p \in \mathbb{A}_p$)
$\Delta_d(s)$	Transfer function of the dynamic uncertainty block ($\Delta_d(s) \in \mathbb{A}_d$)
$\Delta_s(s)$	Uncertainty sample that maximizes the gain of the uncertain system at a single frequency point
$\Delta_m(s)$	Uncertainty sample that maximizes the gain of the uncertain system at multiple frequency points
\mathcal{Q}	Block structured, unit norm bounded matrix sample of the uncertainty block ($\mathcal{Q} \in \mathbb{Q}$)
\mathcal{Q}_d	Block structured, unit norm bounded matrix sample of the dynamic uncertainty block ($\mathcal{Q}_d \in \mathbb{Q}_d$)
$D(s)$	D-scale
$D_d(s)$	D-scale corresponding to the dynamic uncertainty block

1

Introduction

When designing a controller for real systems, control engineers almost always utilize a mathematical model of that system. The model, however, usually only captures the actual phenomenon to some degree of precision but never perfectly. Robustness in control design means the resilience of the closed-loop system against differences between the actual system and its model which was used to design the controller. In particular, the closed-loop system has to fulfill performance requirements and has to maintain stability even in the presence of bounded model uncertainty. The present thesis considers three aspects of this problem: robust controller synthesis, worst-case analysis, and the development of a real-life flutter suppression control system.

1.1 Robust synthesis

For linear time-invariant (LTI) systems, the current literature offers a number of design techniques that can explicitly take these requirements into account [29, 67, 32, 63]. One of the most widely used methods is the H_∞ optimal synthesis [16], which is actively researched and applied to many practical problems. A more advanced approach, capable of handling structured uncertainty, is the μ -synthesis by D-K iteration [41]. Both of these methods are also well-established parts of the Robust Control Toolbox of MATLAB, one of the most significant software used for control design.

While the H_∞ and μ -syntheses are staples in the research community, they are not as relevant in the industry. Though the number of application examples are growing, industrial problems are mostly solved with more traditional and better known techniques, such as proportional-integral-derivative (PID) control. The required simplicity of the control law and the ease of tuning makes these methods more appealing over the modern optimal synthesis approaches. Structured H_∞ synthesis is a possible bridge between theory and practice [7, 20]. This enables the optimal tuning of controllers with pre-determined structure (e.g. PID), hence it brings the advanced optimization-based framework closer to industrial applications.

A notable example where structured H_∞ control design was applied is the Airbus A350 aircraft [15]. The flexible structure of the Airbus A350 resulted in unexpected fish-tailing phenomenon which is a serious impairment of passenger comfort. This design flaw was only discovered during flight tests by which time the controller structure could not be changed without starting the lengthy certification process from the beginning. However, the performance of the aircraft could be improved by applying structured H_∞ synthesis to tune the parameters of the control law. The Rosetta mission is another case when this kind of intervention was possible [8]. The failure of one of the thrusters aboard

the Philae lander almost made it impossible for the spacecraft to fulfill its intended purpose. However, the redesign of the controller parameters could compensate for the error to a degree that made the mission still possible.

Combining optimization-based robust and structured control, the present thesis proposes a design algorithm that is capable of synthesizing a structured controller for systems with mixed (both dynamic and parametric) uncertainty. The chosen objective of the optimal synthesis is the minimization of the worst-case gain, i.e., the maximal closed-loop H_∞ norm over the set of allowable uncertainties. The algorithm utilizes an iteration in which controller synthesis is followed by analysis. This iteration continuously updates a set of "bad" samples for the parametric uncertainty. This yields a corresponding set of sampled design plants. The synthesis step itself is an iterative process, which resembles the traditional D-K iteration. The K-step is performed as a structured design on the collection of sampled plants. The D-step solves for the scalings associated with each sampled plant. In the analysis step, the worst-case gain of the closed-loop is calculated along with a worst-case uncertainty, from which the sample of parametric uncertainty is extracted and added to the set of "bad" samples. The subsequent synthesis step is repeated using this updated set of sampled plants.

Similarly to its most closely related counterpart in the literature [2], this method bypasses numerical difficulties by handling the two types of uncertainties differently: it finds critical samples of the uncertain parameters and constructs D-scales for the dynamic uncertainty. The most notable difference of this algorithm compared to [2] is that the D-scales are allowed to vary with the samples of the uncertain parameters, which is less conservative than the unified D-scale approach of [2]. The MATLAB implementation of this algorithm is available on-line [52].

1.2 Worst-case analysis

Validation of control laws is just as important as designing them, especially for safety critical systems, such as vehicles and reactors. A controller designed for these systems has to be extensively analyzed before deployment. Mathematical analysis methods can provide stability guarantees to some extent, and the behavior of closed-loops can be studied in software-in-the-loop and hardware-in-the-loop simulations. However, high order dynamics, nonlinearities, complex sensors systems, and other phenomena make it very difficult to completely and rigorously test the control system.

Robustness analysis is the methodology to determine whether the closed-loop maintains stability and performance for all possible variations of the plant. In time-domain simulations, the Monte Carlo method is widely used, in which many samples of the uncertain plant is generated randomly and probabilistic guarantees are established based on the results [37]. This approach was successfully applied to the VEGA launch vehicle [38], among others. On the other hand, the systems appearing in modern control problems are often too complex to find all the relevant special cases by engineering intuition or by a reasonable sample size in a Monte Carlo simulation. Guided systematic methods are required to ensure that important corner cases are included in the tests.

Therefore, Monte Carlo simulations are increasingly complemented by the calculation of worst-case uncertainty samples [19]. MATLAB has had built-in tools for this purpose for decades. The so-called worst-case gain, the maximum H_∞ gain over all possible

uncertainties, is often used to evaluate robust performance, and the corresponding uncertainty sample, that drives the gain to its maximum, is usually chosen for worst-case uncertainty. This approach maximizes the gain of the system as a single frequency making it most sensitive to disturbances at that specific frequency. Therefore, the use of this worst-case uncertainty sample may not show a significant performance degradation in time-domain simulations.

This is especially significant for systems that have multiple resonance frequencies, e.g., hard disk drives [25] and flexible aircraft [P69]. To address this insufficiency, a novel method for worst-case uncertainty construction is presented in this thesis. The proposed algorithm maximizes the gain of an uncertain system at several given frequencies simultaneously. It consists of performing a nonlinear optimization to find the worst-case value of the uncertain parameters and the complex matrix uncertainty samples computed by the worst-case gain lower bound power iteration at multiple frequencies. Next, an interpolation technique, based on the boundary Nevanlinna-Pick (BNP) method [10], is used to create a stable, norm-bounded LTI uncertainty that interpolates the collection of matrix samples. In addition to systems with multiple resonance frequencies, this solution is also useful for constructing worst-case samples when the spectrum of the disturbance is known. Maximizing the gain at specific frequencies makes it possible to focus on a frequency range in the analysis which is relevant for many engineering problems. The MATLAB code of this algorithm can be downloaded from [51].

1.3 Flutter suppression control

The present thesis also considers a real-life control problem: the flutter suppression control for a flexible aircraft called T-FLEX and illustrated in Figure 1.1. Aeroelastic flutter is an unstable oscillation, which is the result of an adverse interaction of the structural dynamics and the aerodynamics of aircraft. Since this phenomenon leads to structural failure, it must be avoided. Presently, this is achieved passively, by the increase of structural stiffness of the airframe. At the same time, the use of lighter and therefore more flexible materials for the construction of the wings is desirable for fuel efficiency.

In recent years, this motivated research aiming to develop active control solutions to suppress flutter. NASA built the X-56 experimental platform to investigate the control of flexible aircraft [62]. A similar flying-wing aircraft, called mAEWing1, is used in the PAAW project launched in 2014 in the United States [65]. This collaboration has already produced promising ideas, which are all put to flight test aboard an unmanned aircraft designed and constructed for this purpose [14]. Because of the special construction of mAEWing1, it exhibits slightly different flutter behavior (called body-freedom flutter) than conventional aircraft (bending-torsion coupled flutter). Joining to this effort, the FLEXOP project was launched in 2015 with the funding of the European Union [18]. This research topic was also carried over to the FliPASED project started in 2019 [17]. The flutter characteristics of the T-FLEX demonstrator aircraft (Figure 1.1) built for these projects resemble airliners more, therefore the application of the results of FLEXOP and FliPASED in commercial aviation is expected to be more straightforward than those of PAAW and X-56.

The flutter control design task embodies all the challenges mentioned thus far. The behavior of the system is complex which makes it impossible to capture all its intricacies in a model that is still usable for control synthesis. The system also has a large number



Figure 1.1: The flexible aircraft, called T-FLEX, built for flutter experimentation in the FLEXOP project.

of physical parameters for which accurate values are difficult to determine. Therefore, the representation of model uncertainty is an integral part of this problem. Besides the apparent need for uncertain modelling, results emerging from PAAW, FLEXOP, and FlIPASED also indicate that the robust control framework is suitable for flutter suppression control design. In [70], H_∞ synthesis is used successfully to increase the damping of the flutter modes. Following the basic idea of this approach, an induced L_2 norm optimal linear parameter varying controller is designed in [P47]. This further cements the applicability of robust control to flutter suppression.

The onboard systems of the aircraft have limited computational capacity. In order to obtain a sufficiently simple control law that can be implemented on the flight control computer, structured control synthesis has to be employed. Above a certain airspeed, the system is inherently unstable because of the undamped oscillations caused by flutter, which makes the control of the aircraft safety critical. Hence, the validation of any control law needs to be sufficiently extensive to ensure safe operation.

The proposed solution incorporates all of these aspects. An uncertain design model is constructed that takes the key variations of the dynamics into account. Then, the flutter suppression controller is designed using the structured synthesis algorithm. The closed-loop is subjected to time and frequency domain analysis. A multi-frequency worst-case uncertainty sample is constructed to establish bounds on the achieved performance. The obtained control law is also validated by disk margin analysis and time domain simulations using the high fidelity model of the aircraft. Based on these results, the controller is capable of extending the safe flight envelope by 15%.

The rest of the thesis is structured as follows. Chapter 2 reviews existing results of robust control theory that are going to be expanded upon in the following chapters. Chapter 3 describes the robust structured synthesis algorithm. The multi-frequency worst-case uncertainty construction is laid out in Chapter 4 and its application is also demonstrated by the analysis of a hard disk drive control system. The flutter control methodology along with the validation of the obtained flutter controller is elaborated in Chapter 5. Finally, concluding remarks are made in Chapter 6.

2

Theoretical foundation

This chapter reviews the elements of liner system analysis and control synthesis that is strongly relied on throughout this thesis. First, in Section 2.1, the basic notions of LTI systems and the H_∞ control synthesis method are introduced. Section 2.2 covers the modeling of uncertain systems, robust stability analysis using the structured singular value, and μ -synthesis. Finally, Section 2.3 defines the worst-case gain of uncertain systems and provides the computation method. The theoretical statements in this Chapter are based on [67].

2.1 Nominal systems and H_∞ control synthesis

2.1.1 Linear time-invariant systems

In the design and analysis of control systems, the plant, controller, actuators, sensors, and other components involved are usually modeled by a system of nonlinear differential equations. These equations are linearized to obtain LTI systems that are more tractable for control design and analysis. An LTI system is given by its state-space representation [27] or transfer function. The state-space representation is the system of ordinary differential equations

$$\begin{aligned}\dot{x} &= Ax + Bu, \\ y &= Cx + Du.\end{aligned}\tag{2.1}$$

Here, $u : \mathbb{R} \rightarrow \mathbb{R}^{n_u}$ is the input, $y : \mathbb{R} \rightarrow \mathbb{R}^{n_y}$ is the output, and $x : \mathbb{R} \rightarrow \mathbb{R}^{n_x}$ is the state of the system. The $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$, and $D \in \mathbb{R}^{n_y \times n_u}$ are real matrices. We refer to the system with only one input and output ($n_y = n_u = 1$) as a single-input single-output (SISO) system. Otherwise, the system is called multiple-input multiple-output (MIMO). In the Laplace domain, an LTI system is also represented by its transfer function

$$G(s) = C(sI - A)^{-1}B + D,\tag{2.2}$$

where $s \in \mathbb{C}$ is the complex variable of the Laplace transform, and I is the identity matrix of appropriate size. The poles of the transfer function are the roots of the equation $\det(sI - A) = 0$.

Whether the LTI system models the plant, controller or the closed-loop, stability is one of its key properties. The following theorem provides equivalent definitions of stability and also gives tools to assess it.

Theorem 2.1. The following statements are equivalent for an LTI system.

1. For all bounded inputs u , the output of the system y is also bounded.
2. For all minimal state-space representations[†] of the system and for arbitrary initial states, the state converges to 0 when $u = 0$ and $t \rightarrow \infty$.
3. For all minimal state-space representations, the eigenvalues of A are in the open left half of the complex plane. (For all eigenvalues λ of A , $\text{Re } \lambda < 0$.)
4. The poles of $G(s)$ are in the open left half of the complex plane. (For all poles p of $G(s)$, $\text{Re } p < 0$.)

Proof. The proof is found in standard textbooks, such as [67]. \diamond

[†]A state-space representation is called minimal if the pair (A, B) is controllable, and the pair (A, C) is observable. See Section 4.2 of [67] for details.

The mathematical articulations of the expected behavior of the closed-loop system are called performance measures. Performance measures, as well as other important properties of the closed-loop systems, are often defined in terms of gain. We use the H_∞ norm of $G(s)$ to define the gain of an LTI system. The H_∞ norm is the maximum over frequency of the largest singular value of $G(j\omega)$. If $G(s)$ is stable, then

$$\|G(s)\|_\infty = \sup_{\omega \in \mathbb{R} \cup \{\infty\}} \bar{\sigma}(G(j\omega)) \quad (2.3)$$

and $\|G(s)\|_\infty = \infty$ otherwise. The symbol $\bar{\sigma}(\cdot)$ denotes the largest singular value of its argument.

2.1.2 H_∞ synthesis

In general, the objective of a control system is to make the output y behave in a desired way by manipulating the plant input u . The controller is the algorithm for adjusting u based on the available information. To arrive at a good design for the controller, we need a priori information about the expected disturbances and reference inputs, and of the plant model. The design is the selection of a controller for which the stability of the closed-loop and acceptable values of the performance measures are guaranteed. In an optimal design strategy, the performance measures are incorporated into an objective function which is minimized to attain the optimal controller.

The H_∞ synthesis is an optimal control design method in which the performance requirements are given as an H_∞ gain minimization task. The block diagram in Figure 2.1 depicts the general control configuration used for the H_∞ control synthesis in any closed-loop control problem. Here, $K(s)$ is the controller, and $P(s)$ is called the generalized plant. The signal y represents measured signals of $P(s)$, and u is the control input. The performance inputs and outputs are d , the generalized disturbance, and e , the generalized error. This closed-loop system is described by

$$\begin{bmatrix} e \\ y \end{bmatrix} = P(s) \begin{bmatrix} d \\ u \end{bmatrix}, \quad u = K(s) y. \quad (2.4)$$

The generalized plant $P(s)$ is the interconnection of the actual plant and frequency dependent weighting functions that turn the performance requirements into an H_∞ gain minimization task.

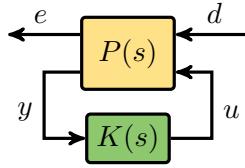


Figure 2.1: General control configuration for the H_∞ control design.

Example 2.1. Consider the generalized plant interconnection depicted in Figure 2.2. This design setup aims to shape the sensitivity function of the closed-loop, while limiting the control effort. The sensitivity function is defined as

$$S(s) := (I + G(s) K(s))^{-1}. \quad (2.5)$$

It is the transfer function from the reference signal to the tracking error and also form a disturbance on the plant output to the output itself. Hence, $\bar{\sigma}(S(j\omega))$ determines the tracking and disturbance rejection performance of the closed-loop. The weighting function $W_y(s)$ captures the desired shape of $S(s)$ since the optimization

$$\min_{K(s)} \|W_y(s) S(s)\|_\infty \quad (2.6)$$

yields $K(s)$ such that $\bar{\sigma}(S(j\omega))$ is optimally close to $\bar{\sigma}(W_y^{-1}(j\omega))$. However, this is achieved with arbitrary controller bandwidth and potentially very large control signals. Hence, the control effort, which is the energy output of the controller, is taken into account. The desired gain of the controller is expressed by $\bar{\sigma}(W_u^{-1}(j\omega))$.

With both control objectives included, the equations of the weighted interconnection in Figure 2.2 are

$$\begin{aligned} e_y &= W_y(s) (d - G(s) u), \\ e_u &= W_u(s) u, \\ y &= d - G(s) u, \\ u &= K(s) y. \end{aligned} \quad (2.7)$$

These equations are equivalent to

$$\begin{bmatrix} e_y \\ e_u \\ y \end{bmatrix} = P(s) \begin{bmatrix} d \\ u \\ u \end{bmatrix}, \quad (2.8)$$

$$u = K(s) y$$

with

$$P(s) = \begin{bmatrix} W_y(s) & -W_y(s) G(s) \\ 0 & W_u(s) \\ 1 & -G(s) \end{bmatrix}. \quad (2.9)$$

The lower linear fractional transformation (LFT) is used to express the transfer function of the closed-loop. It is defined as follows. Denote the set of complex matrices with r rows and c columns by $\mathbb{C}^{r \times c}$. Let $P \in \mathbb{C}^{r_P \times c_P}$ and $K \in \mathbb{C}^{r_K \times c_K}$ be given matrices with $r_K < c_P$ and $c_K < r_P$. Partition P , such that, the lower right block P_{22} is $c_K \times r_K$,

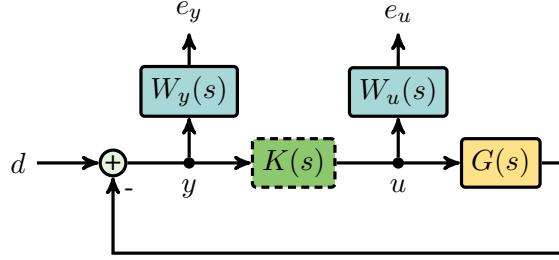


Figure 2.2: Generalized plant interconnection in Example 2.1.

and

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}. \quad (2.10)$$

Then, the lower linear fractional transformation is defined as

$$\mathcal{F}_L(P, K) := P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}. \quad (2.11)$$

The same definition of the lower LFT applies if P and K are LTI systems. Hence, the MIMO closed-loop interconnection in Figure 2.1 is $T(s) = \mathcal{F}_L(P(s), K(s))$.

The H_∞ optimal control problem is

$$\min_{K(s)} \|\mathcal{F}_L(P(s), K(s))\|_\infty, \quad (2.12)$$

where $P(s)$ is partitioned, such that, the input and output of $P_{22}(s)$ is u and y , respectively. This is formulated as a convex optimization task by either turning it into a semidefinite program (SDP) with linear matrix inequality (LMI) constraints or by solving two Riccati equations [72]. Both approaches are available in the MATLAB function `hinfsyn`. The resulting controller $K(s)$ is an unstructured LTI system with state dimension equal to that of $P(s)$. It has to be noted that while the method guarantees the stability of the closed-loop $\mathcal{F}_L(P(s), K(s))$, it does not explicitly enforce the stability of the controller.

Industrial applications often prefer or require controllers with specific structure, such as PID or controllers with fixed state dimension. This motivates the use of structured H_∞ synthesis [5]. In a structured control problem, the dynamics of the controller are fixed, up to a set of design parameters $\kappa \in \mathbb{R}^{n_\kappa}$. Such a controller is denoted as $K(\kappa, s)$. The design task is to find the optimal value of the parameters κ , such that, the H_∞ norm of the closed-loop is minimal, i.e.,

$$\min_{\kappa \in \mathbb{R}^{n_\kappa}} \|\mathcal{F}_L(P(s), K(\kappa, s))\|_\infty. \quad (2.13)$$

This problem is non-convex and non-smooth, therefore the global optimality of the solution is not guaranteed. This synthesis method is available in MATLAB in the `hinfsyn` function [7, 20]. The `systune` algorithm in MATLAB can also handle more general design objectives, such as constraints on the closed-loop dynamics, loop-shaping constraints at specified loop opening sites, and controller stability enforcement [6].

Example 2.2. A PID controller in a structured design problem is written as

$$K(\kappa, s) = \kappa_1 + \frac{\kappa_2}{s} + \frac{\kappa_3 s}{\kappa_4 s + 1}, \quad (2.14)$$

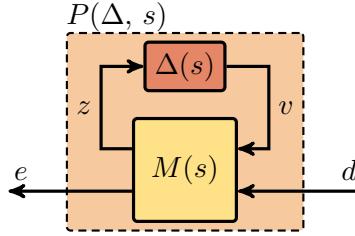


Figure 2.3: General uncertain system interconnection.

where $\kappa_1, \kappa_2, \kappa_3, \kappa_4 \in \mathbb{R}$ are the tunable design parameters. The parameter vector is $\kappa^T = [\kappa_1 \ \kappa_2 \ \kappa_3 \ \kappa_4]$.

Example 2.3. When the controller is required to be strictly proper, SISO and third order, then it is written as

$$K(\kappa, s) = \frac{\kappa_1 s^2 + \kappa_2 s + \kappa_3}{s^3 + \kappa_4 s^2 + \kappa_5 s + \kappa_6} \quad (2.15)$$

with $\kappa_1, \kappa_2, \kappa_3, \kappa_4, \kappa_5, \kappa_6 \in \mathbb{R}$ and $\kappa^T = [\kappa_1 \ \kappa_2 \ \kappa_3 \ \kappa_4 \ \kappa_5 \ \kappa_6]$.

2.2 Robust stability and μ -synthesis

In modern control system design approaches, we increasingly account for the fact that we can only acquire a limited amount of information about the physical system in question to construct its model. These gaps in our knowledge are articulated by the introduction of uncertainty. Uncertainty also arises when the non-linear system is approximated by an LTI model or when complex dynamics are simplified via a suitable model reduction technique for control design purposes. This section extends the notion of stability to uncertain systems, provides a tool for robust stability analysis, and introduces μ -synthesis.

2.2.1 Uncertain LTI systems

Consider the block diagram in Figure 2.3. This is an interconnection of a stable MIMO LTI system $M(s)$ and the MIMO uncertainty block $\Delta(s)$. The uncertain system $P(\Delta, s)$ is defined as the upper LFT of these components. Similarly to the lower LFT in Section 2.1.2, let $M \in \mathbb{C}^{r_M \times c_M}$ and $\Delta \in \mathbb{C}^{r_\Delta \times c_\Delta}$ be given matrices with $r_\Delta < c_M$ and $c_\Delta < r_M$. Partition M , such that, the upper left block is $c_\Delta \times r_\Delta$, and

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}. \quad (2.16)$$

The upper LFT of M and Δ is defined as

$$\mathcal{F}_U(M, \Delta) := M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12}. \quad (2.17)$$

Then, the uncertain system with input d and output e in Figure 2.3 is

$$P(\Delta, s) = \mathcal{F}_U(M(s), \Delta(s)). \quad (2.18)$$

The uncertainty block $\Delta(s)$ is also a stable MIMO LTI system but with additional properties. Specifically, we use two distinct types of uncertainties. When the values of

certain physical parameters, such as mass, inertia or aerodynamic coefficients, are not accurately known, parametric uncertainty is introduced into the system. To characterize $\Delta(s)$ when only parametric uncertainty is present, define the set of parametric uncertainties as

$$\mathbb{A}_p := \left\{ \begin{bmatrix} \delta_1 I_{r_1} & & 0 \\ & \ddots & \\ 0 & & \delta_{N_p} I_{r_{N_p}} \end{bmatrix} : \delta_i \in \mathbb{R}, |\delta_i| \leq 1, i = 1, \dots, N_p \right\}. \quad (2.19)$$

The uncertain parameters are denoted by δ_i and r_i describe the number of instances of δ_i in the system. In general, r_i depends on how many times the parameter δ_i occurs in the equations of the system.

The other type of uncertainty besides parametric is called dynamic. Dynamic uncertainty represents unmodelled behavior of the system, e.g., high frequency vibrations, dynamics neglected when applying model reduction, or general lack of knowledge arising from system identification for example. The set of structured and unit norm bounded dynamic uncertainties is defined as

$$\mathbb{A}_d := \left\{ \begin{bmatrix} \Delta_1(s) & & 0 \\ & \ddots & \\ 0 & & \Delta_{N_d}(s) \end{bmatrix} : \Delta_i(s) \text{ is LTI, } \|\Delta_i(s)\|_\infty \leq 1, \right. \\ \left. i = 1, \dots, N_d \right\}. \quad (2.20)$$

Notice, that $\Delta_d(s) \in \mathbb{A}_d$ implies that $\Delta_d(s)$ is stable and $\|\Delta_d(s)\|_\infty \leq 1$. The block $\Delta_i(s)$ can be MIMO and has dimensions $r_{d,i} \times c_{d,i}$. It is also true that, if $\Delta_p \in \mathbb{A}_p$, then $\bar{\sigma}(\Delta_p) \leq 1$. The mixed uncertainty set is

$$\mathbb{A} := \left\{ \begin{bmatrix} \Delta_p & 0 \\ 0 & \Delta_d(s) \end{bmatrix} : \Delta_p \in \mathbb{A}_p, \Delta_d(s) \in \mathbb{A}_d \right\}. \quad (2.21)$$

Again, $\Delta(s) \in \mathbb{A}$ means $\|\Delta(s)\|_\infty \leq 1$. To illustrate these concepts, the following example is provided.

Example 2.4. Consider the first order system with nominal pole at -1 and unit nominal gain at zero frequency. The pole is $\pm 20\%$ uncertain and $\pm 10\%$ uncertainty in the dynamics is introduced across all frequencies. This system is described by

$$P(\Delta, s) = \frac{1}{s + (1 + 0.2\delta_1)} (1 + 0.1\Delta_1(s)), \quad (2.22)$$

where $\Delta_1(s)$ is the dynamic uncertainty block with $\|\Delta_1(s)\|_\infty \leq 1$, and $\delta_1 \in \mathbb{R}$ is the uncertain parameter with $|\delta_1| \leq 1$. This makes the corresponding uncertainty sets

$$\begin{aligned} \mathbb{A}_p &= \{\delta_1 : \delta_1 \in \mathbb{R}, |\delta_1| \leq 1\}, \\ \mathbb{A}_d &= \{\Delta_1(s) : \Delta_1(s) \text{ is LTI, } \|\Delta_1(s)\|_\infty \leq 1\}, \\ \mathbb{A} &= \left\{ \begin{bmatrix} \delta_1 & 0 \\ 0 & \Delta_1(s) \end{bmatrix} : \delta_1 \in \mathbb{A}_p, \Delta_1(s) \in \mathbb{A}_d \right\}. \end{aligned} \quad (2.23)$$

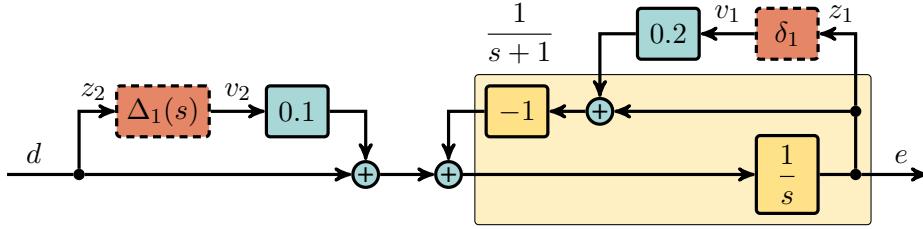


Figure 2.4: Block diagram of the uncertain system in Example 2.4.

The block diagram of $P(\Delta, s)$, depicted in Figure 2.4, is equivalent to the equations

$$\begin{aligned} z_1 &= e, \\ z_2 &= d, \\ e &= \frac{1}{s} [(-1)(0.2v_1 + e) + (0.1v_2 + d)]. \end{aligned} \quad (2.24)$$

From the last equation,

$$e = \frac{-0.2v_1 + 0.1v_2 + d}{s + 1}. \quad (2.25)$$

Therefore

$$M(s) = \begin{bmatrix} M_{11}(s) & M_{12}(s) \\ M_{21}(s) & M_{22}(s) \end{bmatrix}, \quad (2.26)$$

where

$$\begin{aligned} M_{11}(s) &= \begin{bmatrix} \frac{-0.2}{s+1} & \frac{0.1}{s+1} \\ 0 & 0 \end{bmatrix}, & M_{12}(s) &= \begin{bmatrix} \frac{1}{s+1} \\ 1 \end{bmatrix}, \\ M_{21}(s) &= \begin{bmatrix} \frac{-0.2}{s+1} & \frac{0.1}{s+1} \end{bmatrix}, \text{ and } M_{22}(s) &= \frac{1}{s+1}. \end{aligned} \quad (2.27)$$

Notice, that the nominal system is $P(0, s) = M_{22}(s)$.

Using this description of an uncertain system, for each $\Delta(s) \in \Delta$, we get an LTI system $P(\Delta, s)$. Since Δ is an infinite set, this yields an infinite set of LTI systems $\mathcal{F}_U(M(s), \Delta(s))$ in which the state dimension, stability, and gain of the members vary. In the following section, we define the stability and gain of uncertain systems and introduce the methods with which to assess these properties.

2.2.2 Structured singular value

The most important notion of this section is robust stability.

Definition 2.1 (robust stability). The uncertain system $P(\Delta, s)$ is called robustly stable if it is stable for all possible $\Delta(s) \in \Delta$.

Our first approach to assess robust stability is the small gain theorem (Theorem 9.1 in [73]) that provides a conservative necessary condition. Let Δ_{full} denote the full block uncertainty set, i.e., when $\Delta(s)$ is a single dynamic uncertainty block. The small gain theorem is a condition for robust stability when $\Delta = \Delta_{\text{full}}$.

Theorem 2.2 (small gain theorem). $P(\Delta, s) = \mathcal{F}_U(M(s), \Delta(s))$ is stable for all $\Delta(s) \in \Delta_{\text{full}}$ and for stable $M(s)$ if and only if $\bar{\sigma}(M_{11}(j\omega)) < 1$ for all ω .

Proof. See [73]. \diamond

Using this theorem, the following necessary condition is given.

Lemma 2.1. If $\bar{\sigma}(M_{11}(j\omega)) < 1$ for all ω , then the uncertain system $P(\Delta, s) = \mathcal{F}_U(M(s), \Delta(s))$ is robustly stable.

Proof. The statement of the theorem follows from Theorem 2.2 considering the fact that for appropriate sizes, $\Delta \subseteq \Delta_{\text{full}}$. \diamond

Lemma 2.1 is conservative, since Δ usually describes a nontrivial block diagonal structure. Hence, a more sophisticated tool is developed to obtain a necessary and sufficient condition. To take the structure of the uncertainty into account, the so called structured singular value, also known as μ , is introduced. Let Δ be a set of complex matrices with the block structure of Δ . Also, with some overload of notation, denote the frequency response of LTI systems, such as $M_{11}(s)$ and $M(s)$ by M_{11} and M without the (s) or $(j\omega)$ arguments.

Definition 2.2 (structured singular value, μ). The structured singular value of the complex matrix M_{11} with respect to the block structure Δ is

$$\mu_{\Delta}(M_{11}) = \frac{1}{\min_{\Delta \in \Delta} \{\bar{\sigma}(\Delta) : \det(I - M_{11}\Delta) = 0\}} \quad (2.28)$$

and $\mu_{\Delta}(M_{11}) = 0$ if there is no $\Delta \in \Delta$ that makes $\det(I - M_{11}\Delta) = 0$.

The interpretation of Definition 2.2 is that the structured singular value of $M_{11}(j\omega_0)$ at a fixed frequency ω_0 is the inverse of the largest singular value of $\Delta(j\omega_0)$ ($\Delta(s)$ has block structure of Δ) for which $P(\Delta, s)$ is marginally unstable with poles $\pm j\omega_0$. We assumed that $M(s)$ is stable which means that $P(0, s)$ is also stable. Because the poles of $P(\Delta, s)$ change continuously with increasing $\|\Delta(s)\|_{\infty}$, the system $P(\Delta, s)$ is robustly stable if there is no uncertainty in Δ that can place the poles of $P(\Delta, s)$ on the imaginary axis. This leads us to our main robust stability theorem.

Theorem 2.3. The uncertain system $P(\Delta, s) = \mathcal{F}_U(M(s), \Delta(s))$ is robustly stable if and only if $\mu_{\Delta}(M_{11}(j\omega)) < 1$ for all ω .

Proof. The statement of the theorem is clear from the definition of μ (Definition 2.2). \diamond

To illustrate μ , we calculate it for the system in Example 2.4.

Example 2.4 (continued from page 10). Is $P(\Delta, s)$ robustly stable? For this simple system, we can assess robust stability using basic algebra. $P(\Delta, s)$ has a pole in the origin if $1 + 0.2\delta_1 = 0$, i.e., $\delta_1 = -5$. Since $\Delta_1(s)$ is stable, this is the only pole that can cross over to the right half plane, therefore there is no destabilizing uncertainty if $\omega \neq 0$. Also, the stability of $P(\Delta, s)$ is clearly independent of $\Delta_1(s)$.

We also compute μ to verify this. Denote the complex block structure corresponding

to Δ by Δ . To calculate $\mu_{\Delta}(M_{11}(j\omega))$, consider

$$\begin{aligned} I - M_{11}(j\omega) \Delta &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{-0.2}{j\omega+1} \delta_1 & \frac{0.1}{j\omega+1} \Delta_1 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 + \frac{0.2\delta_1}{j\omega+1} & -\frac{0.1\Delta_1}{j\omega+1} \\ 0 & 1 \end{bmatrix}. \end{aligned} \quad (2.29)$$

The determinant is

$$\det(I - M_{11}(j\omega) \Delta) = 1 + \frac{0.2\delta_1}{j\omega+1} = \frac{j\omega+1+0.2\delta_1}{j\omega+1} = 0 \quad (2.30)$$

for $\delta_1 = -5 - j5\omega$. If $\omega = 0$, $\delta_1 = -5$ is the solution with arbitrary Δ_1 . If $\omega \neq 0$ however, there is no solution since $\delta_1 \in \mathbb{R}$. Therefore, the smallest $\Delta \in \Delta$ that makes the determinant zero at $\omega = 0$ is

$$\Delta = \begin{bmatrix} -5 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.31)$$

for which $\bar{\sigma}(\Delta) = 5$. This makes μ

$$\mu_{\Delta}(M_{11}(j\omega)) = \begin{cases} 0.2 & \text{if } \omega = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (2.32)$$

Clearly, $\mu_{\Delta}(M_{11}(j\omega)) < 1$ for all ω , therefore $P(\Delta, s)$ is robustly stable.

In general, the numerical computation of μ is known to be NP-hard for many cases [12]. Instead of a direct computation, lower bounds and upper bounds are calculated. This is reviewed next. Using these results, robust stability analysis amounts to calculating the upper and lower bounds over a sufficiently dense frequency grid.

Upper bound

To develop the upper bound of μ , define the set of D-scales as the invertible complex matrices that commute with the elements of Δ , i.e.,

$$\mathbf{D} := \{D : \exists D^{-1}, D\Delta = \Delta D \text{ for all } \Delta \in \Delta\}. \quad (2.33)$$

For the rest of this thesis, we assume that all blocks in Δ (and Δ) are square. This assumption is easily relaxed with additional notation. To obtain the D-scales, construct $D \in \mathbf{D}$, such that, it has the same block diagonal structure as Δ . When the diagonal block of Δ is a scalar matrix (constant multiple of the identity matrix), then the corresponding diagonal block in D is an unstructured (full) matrix and vice versa. Using the D-scales, the upper bound of μ is given next.

Lemma 2.2. The upper bound of μ is calculated as

$$\mu_{\Delta}(M_{11}) \leq \inf_{D \in \mathbf{D}} \bar{\sigma}(DM_{11}D^{-1}). \quad (2.34)$$

Proof. Because of Lemma 2.1, it is clear that $\mu_{\Delta}(M_{11}) \leq \bar{\sigma}(M_{11})$. Also, because $\Delta D = D\Delta$ holds for all $\Delta \in \Delta$ and $D \in \mathbf{D}$,

$$\begin{aligned} \det(I - DM_{11}D^{-1}\Delta) &= \det(DD^{-1} - DM_{11}\Delta D^{-1}) \\ &= \det(I - M_{11}\Delta). \end{aligned} \quad (2.35)$$

Hence,

$$\mu_{\Delta}(M_{11}) = \mu_{\Delta}(DM_{11}D^{-1}) \leq \bar{\sigma}(DM_{11}D^{-1}) \quad (2.36)$$

which implies the statement of the lemma. \diamond

The standard way to evaluate the right hand side of (2.34) is to turn it into an SDP using the following lemma. For symmetric matrices A and B , the notation $A \succ B$ means that the matrix $A - B$ is positive definite. The symbols \prec , \succeq , and \preceq is to be understood accordingly. Also, for a complex matrix A , A^* denotes the conjugate transpose.

Lemma 2.3 (μ upper bound). The upper bound of μ is expressed as $\mu_{\Delta}(M_{11}) \leq \sqrt{\hat{\beta}}$, where

$$\hat{\beta} = \min_{X \in \mathbf{D}} \beta$$

subject to:

$$M_{11}^* X M_{11} - \beta X \preceq 0, \quad (2.37)$$

$$X \succ 0,$$

$$\beta > 0.$$

Proof. The proof relies on the fact that for any complex matrix A , $\bar{\sigma}(A) \leq \gamma$ is equivalent to $A^* A \preceq \gamma^2 I$. Hence, $\bar{\sigma}(DM_{11}D^{-1}) \leq \gamma$ is equivalent to

$$D^{-*} M_{11} D^* D M_{11} D^{-1} - \gamma^2 I \preceq 0. \quad (2.38)$$

Perform the multiplication $D^* (\dots) D$ on both sides. Since D has full column rank, this does not change the definiteness. Introduce $X := D^* D$ and $\beta = \gamma^2$ to get the constraint

$$M_{11}^* X M_{11} - \beta X \preceq 0. \quad (2.39)$$

To ensure that $D = \sqrt{X}$ and $\gamma = \sqrt{\beta}$ exists, the constraints $X \succ 0$ and $\beta > 0$ are added. \diamond

Lower bound

The lower bound of the structured singular value is obtained by constructing a worst-case complex matrix uncertainty sample $\Delta_{wc} \in \Delta$ with minimal $\bar{\sigma}(\Delta_{wc})$ that makes $\det(I - M_{11}\Delta_{wc}) = 0$. This is achieved using a special power iteration detailed in [43]. Just like the upper bound, the lower bound is also calculated on a frequency grid. This means that for this calculation, the value of the uncertain parameters is allowed to vary from frequency to frequency. The smallest destabilizing uncertainty sample is obtained by fitting the complex matrix sample at the minimum of $\mu_{\Delta}(M_{11}(j\omega))$. This process can be construed as a simplified version of the skewed- μ lower bound computation which is described in Section 2.3.2. The skewed- μ lower bound is directly applied in the subsequent chapters of this thesis therefore it is presented in detail. Hence, the reader is referred to Section 2.3.2 for the actual calculations involved.

2.2.3 μ -synthesis

In robust control synthesis, our aim is to design a controller that suppresses external disturbances to the closed-loop and ensures stability and performance even in the presence

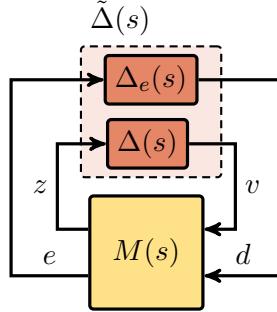


Figure 2.5: Introduction of the performance block $\Delta_e(s)$.

of model uncertainty. Since due to the uncertainty, we have an infinite set of plants instead of just a single LTI system, this means that stability, performance and disturbance rejection requirements have to be fulfilled for an infinite set of possible closed-loop systems. The most prominent method that address this problem is μ -synthesis by D-K iteration. This method is reviewed next.

Using the results of the previous section, we define a performance criterion based on robust stability. Consider the block diagram in Figure 2.5. Here, $\Delta_e(s)$ is a full LTI uncertainty block and is called the performance block. The corresponding block structure is

$$\Delta_e := \{ \Delta_e(s) : \Delta_e(s) \text{ is } n_d \times n_e \text{ LTI}, \Delta_e(s) \in \mathbb{RH}_\infty \}. \quad (2.40)$$

Notice, that the norm of $\Delta_e(s)$ is not restricted. We define the robust performance criterion as the robust stability of the system in Figure 2.5. This definition of robust performance is equivalent to

$$\max_{\omega \in \mathbb{R} \cup \{\infty\}} \max_{\substack{\Delta \in \Delta \\ \bar{\sigma}(\Delta) \leq 1}} \mu_{\Delta_e}(\mathcal{F}_U(M(j\omega), \Delta)) < 1. \quad (2.41)$$

The main loop theorem combines robust stability with robust performance and provides a single μ test to establish both. The theorem requires the definition of the augmented block structure in Figure 2.5:

$$\tilde{\Delta} := \left\{ \begin{bmatrix} \Delta_e(s) & 0 \\ 0 & \Delta(s) \end{bmatrix} : \Delta_e(s) \in \Delta_e, \Delta(s) \in \Delta \right\}. \quad (2.42)$$

Theorem 2.4 (main loop theorem).

$$\left. \begin{array}{l} \mu_{\Delta}(M_{11}) < 1 \\ \max_{\substack{\Delta \in \Delta \\ \bar{\sigma}(\Delta) \leq 1}} \mu_{\Delta_e}(\mathcal{F}_U(M, \Delta)) < 1 \end{array} \right\} \Leftrightarrow \mu_{\tilde{\Delta}}(M) < 1. \quad (2.43)$$

The statement of the theorem is easy to verify using the definition of μ and block determinant lemmas. \diamond

Theorem 2.4 states that the robust stability and robust performance of the closed-loop is equivalent to $\mu_{\tilde{\Delta}}(M(j\omega)) < 1$ for all ω . This gives rise to the μ -synthesis problem. The general control configuration with uncertainty is depicted in Figure 2.6. The μ -

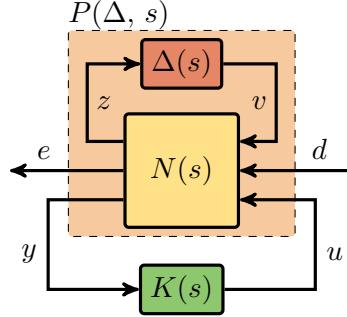


Figure 2.6: General control configuration with uncertainty.

synthesis problem is articulated as the optimization

$$\min_{K(s)} \max_{\omega \in \mathbb{R} \cup \{\infty\}} \mu_{\tilde{\Delta}}(\mathcal{F}_L(N(j\omega), K(j\omega))). \quad (2.44)$$

Since this is not directly solvable, $\mu_{\tilde{\Delta}}$ is replaced by its upper bound as described in Section 2.2.2. For the control synthesis, we extend the set of D-scales \mathbb{D} to LTI systems as follows. Define the set of LTI D-scales associated with the uncertainty set Δ as a set of stable, minimum phase, and invertible LTI systems which commute with the elements of Δ . Similarly to the matrix case, the D-scales are chosen differently for the two types of uncertainties.

Because the parametric uncertainty blocks are scalar matrices, they commute with all matrices of appropriate size. Hence, the D-scales corresponding to them are unstructured LTI systems, i.e.,

$$\mathbb{D}_p := \left\{ \begin{bmatrix} D_1(s) & & 0 \\ & \ddots & \\ 0 & & D_{N_p}(s) \end{bmatrix} : D_i(s) \text{ is } r_i \times r_i, \text{ stable, minimum phase, and invertible LTI, } i = 1, \dots, N_p \right\}. \quad (2.45)$$

The situation is reversed for the dynamic uncertainty blocks therefore the D-scales are scalar matrices. This makes

$$\mathbb{D}_d := \left\{ \begin{bmatrix} d_1(s) I & & 0 \\ & \ddots & \\ 0 & & d_{N_d}(s) I \end{bmatrix} : d_i(s) \text{ is stable, minimum phase, proper, SISO, LTI, } i = 1, \dots, N_d \right\}. \quad (2.46)$$

The set of D-scales for the entire uncertainty set is

$$\mathbb{D} := \left\{ \begin{bmatrix} D_p(s) & 0 \\ 0 & D_d(s) \end{bmatrix} : D_p(s) \in \mathbb{D}_p, D_d(s) \in \mathbb{D}_d \right\}. \quad (2.47)$$

Applying Lemma 2.2 with LTI D-scales, the μ -synthesis problem in (2.44) becomes

$$\min_{\substack{K(s) \\ D(s) \in \mathbb{D}}} \left\| D(s) \mathcal{F}_L(N(s), K(s)) D(s)^{-1} \right\|_\infty. \quad (2.48)$$

This is still a non-convex optimization problem, in general. The D-K iteration provides a workaround to this issue. The main idea of the D-K iteration is to only optimize either

$D(s)$ or $K(s)$ at a time, while holding the other fixed. Assume some $D_{k-1}(s) \in \mathbb{D}$ is available. Then, the control synthesis step (or K-step) of the iteration is

$$K_k(s) = \arg \min_{K(s)} \left\| D_{k-1}(s) \mathcal{F}_L(N(s), K(s)) D_{k-1}(s)^{-1} \right\|_\infty. \quad (2.49)$$

Since $D_{k-1}(s)$ is assumed to be a fixed LTI system, this is simply an H_∞ synthesis task described in Section 2.1.2. The initial value of the D-scale is usually chosen as $D_0(s) = I$.

Next, in the analysis step (or D-step), new D-scales are computed to provide a better upper bound for μ , i.e.,

$$D_k(s) = \arg \min_{D(s) \in \mathbb{D}} \left\| D(s) \mathcal{F}_L(N(s), K_k(s)) D(s)^{-1} \right\|_\infty. \quad (2.50)$$

This is solved by calculating D matrices frequency by frequency over a grid and the LTI system $D(s)$ is the result of weighted least-squares fitting. This method usually works well for dynamic uncertainty since fitting the D matrices amounts to fitting a scalar value for each uncertainty block. On the other hand, the fitting of a full matrix is required for the parametric uncertainty blocks. If the number of repeated parameters is high, this mostly results in an ill-conditioned problem which yields an inaccurate and high order D-scale.

By going back and forth between the K and D-steps, this iteration usually eventually converges to a solution. There is no theoretical proof of convergence, but this method works well in practice and was successfully applied to industrial problems. The method is available in MATLAB in the function `dksyn` or `musyn` in more recent versions. The latter also allows for structured D-K synthesis by replacing the K-step with structured control design. In other words, `musyn` has the option to replace `hinfsyn` by `hinfstruct`.

2.3 Worst-case gain

In Section 2.1.1, the gain of the nominal system is defined in terms of the H_∞ norm. However, with the addition of uncertainty, we have to consider an infinite set of LTI systems. We introduce the worst-case gain which is a natural extension of the notion of gain to uncertain systems. The worst-case gain of the uncertain system is the maximum H_∞ norm of $P(\Delta, s)$ in Figure 2.3, over the set of allowable uncertainties.

Definition 2.3 (worst-case gain). The worst-case gain of the uncertain system of the form $\mathcal{F}_U(M(s), \Delta(s))$ is

$$\hat{\Gamma} := \max_{\Delta(s) \in \Delta} \|\mathcal{F}_U(M(s), \Delta(s))\|_\infty \quad (2.51)$$

if $\mathcal{F}_U(M(s), \Delta(s))$ is robustly stable, and $\hat{\Gamma} = \infty$ otherwise.

The worst-case gain defined in (2.51) is equivalent to calculating the peak gain of the system frequency by frequency and finding the maximum, i.e.,

$$\hat{\Gamma} = \max_{\omega \in \mathbb{R} \cup \{\infty\}} \Gamma(\omega), \quad (2.52)$$

where $\Gamma(\omega)$ is the peak gain of $\mathcal{F}_U(M(s), \Delta(s))$ at ω , i.e.,

$$\Gamma(\omega) := \max_{\substack{\Delta \in \Delta \\ \bar{\sigma}(\Delta) \leq 1}} \bar{\sigma}(\mathcal{F}_U(M(j\omega), \Delta)). \quad (2.53)$$

Note that in general, there is no $\Delta_p \in \Delta_p$ for which $\Gamma(\omega)$ is the gain of $\mathcal{F}_U(M(s), \Delta(s))$ at all frequencies, since the value of Δ_p corresponding to $\Gamma(\omega)$ at some ω may vary with ω . For simplicity, let us drop the (ω) and $(j\omega)$ arguments of Γ and M for the rest of this section.

Next, we define the skewed structured singular value, or skewed- μ , which is the mathematical tool for calculating Γ . For this, we use the augmented uncertainty block structure defined in Section 2.2.3 and in Figure 2.5.

Definition 2.4 (skewed structured singular value, skewed- μ). The skewed structured singular value of the complex matrix M with respect to the block structure $\tilde{\Delta}$ is

$$\tilde{\mu}_{\tilde{\Delta}}(M) := \frac{1}{\min_{\substack{\tilde{\Delta} \in \tilde{\Delta} \\ \bar{\sigma}(\tilde{\Delta}) \leq 1}} \{\bar{\sigma}(\Delta_e) : \det(I - M\tilde{\Delta}) = 0\}} \quad (2.54)$$

and $\tilde{\mu}_{\tilde{\Delta}}(M) = 0$ if there is no $\tilde{\Delta} \in \tilde{\Delta}$ with $\bar{\sigma}(\tilde{\Delta}) \leq 1$ that makes $\det(I - M_{11}\tilde{\Delta}) = 0$.

The connection between Γ and skewed- μ is made by the following lemma.

Lemma 2.4. $\Gamma < \gamma$ is equivalent to $\tilde{\mu}_{\tilde{\Delta}}(M) < \gamma$.

Proof. First, we use the linear algebra fact that since Δ_e is a full uncertainty block, than for any matrix P , $\mu_{\Delta_e}(P) = \bar{\sigma}(P)$. With this, the definition of $\Gamma(\omega)$ in (2.53) is rewritten as

$$\Gamma = \max_{\substack{\Delta \in \Delta \\ \bar{\sigma}(\Delta) \leq 1}} \mu_{\Delta_e}(\mathcal{F}_U(M, \Delta)). \quad (2.55)$$

In the definition of the worst-case gain (Definition 2.3), it is assumed that the system is robustly stable, otherwise the worst-case gain is infinite. Hence it is assumed that $\mu_{\Delta}(M_{11}) < 1$. Because of the main loop theorem (Theorem 2.4), $\Gamma < 1$ and $\mu_{\Delta}(M_{11}) < 1$ is equivalent to $\mu_{\tilde{\Delta}}(M) < 1$.

Instead of $\Gamma < 1$, the more general bound $\Gamma < \gamma$ is established as follows. Because of elementary properties of μ , $\mu_{\Delta_e}(\mathcal{F}_U(M, \Delta)) < \gamma$ is equivalent to

$$\mu_{\Delta_e} \left(\frac{1}{\gamma} \mathcal{F}_U(M, \Delta) \right) < 1. \quad (2.56)$$

Define the scaling matrix

$$S_\gamma = \begin{bmatrix} I & 0 \\ 0 & \frac{1}{\gamma} I \end{bmatrix} \quad (2.57)$$

that scales the performance channels of M by γ and leaves the uncertainty channels unchanged. The scaled matrix $S_\gamma M$ is depicted in Figure 2.7 on the top left. Then, $\frac{1}{\gamma} \mathcal{F}_U(M, \Delta)$ can be written as $\mathcal{F}_U(S_\gamma M, \Delta)$. Therefore, assuming robust stability, $\Gamma < \gamma$ is equivalent to $\mu_{\tilde{\Delta}}(S_\gamma M) < 1$.

By Definition 2.2, $\mu_{\tilde{\Delta}}(S_\gamma M) < 1$ means that there is no $\tilde{\Delta} \in \tilde{\Delta}$ with $\bar{\sigma}(\tilde{\Delta}) \leq 1$, such that, $\det(I - S_\gamma M \tilde{\Delta}) = 0$. Because of the structure of $\tilde{\Delta}$, $\bar{\sigma}(\tilde{\Delta}) \leq 1$ is equivalent

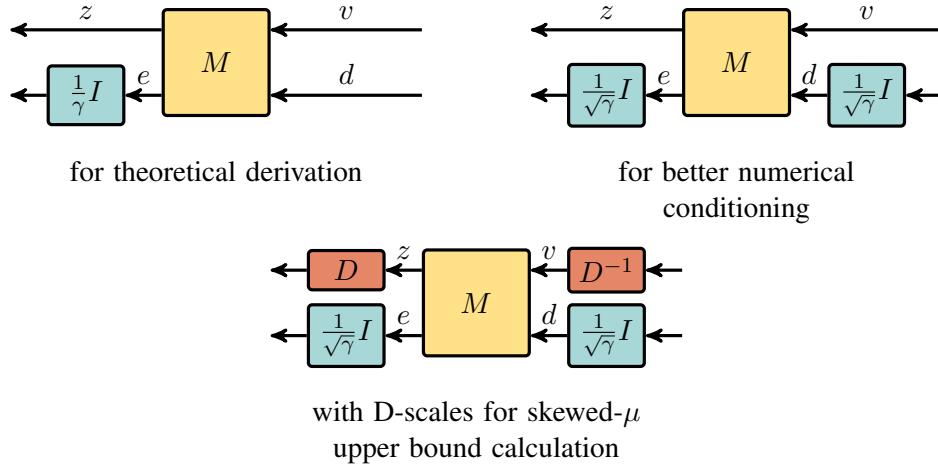


Figure 2.7: Scalings used on the performance and uncertainty channels when calculating the worst-case gain.

to $\bar{\sigma}(\Delta) \leq 1$ and $\bar{\sigma}(\Delta_e) \leq 1$. Therefore, according to Definition 2.4, $\mu_{\tilde{\Delta}}(S_\gamma M) < 1$ is equivalent to $\tilde{\mu}_{\tilde{\Delta}}(S_\gamma M) < 1$. It is easy to see, that this is in turn equivalent to $\tilde{\mu}_{\tilde{\Delta}}(M) < \gamma$ which proves the statement of the lemma. \diamond

The scaling matrix in the proof of Lemma 2.4 is also referred to as the skewing matrix. This is where skewed- μ gets its name. This definition of S_γ is only used for theoretical derivations. In practice, the e and d channels are both scaled by $1/\sqrt{\gamma}$ for better numerical conditioning. This is illustrated by the top right block diagram in Figure 2.7.

Because of Lemma 2.4, the computation of Γ amounts to a skewed- μ calculation. Similarly to μ in Section 2.2.2, the calculation of skewed- μ is also NP-hard for many cases [12]. Therefore, the direct computation is replaced by the lower and upper bound. This computation is performed in the MATLAB function `wcgain` [9]. Similar calculations are performed by the MATLAB-Simulink Systems Modeling, Analysis and Control (SMAC) toolbox [61].

2.3.1 Upper bound

Similarly to the μ upper bound in Section 2.2.2, the calculation of the skewed- μ upper bound is turned into an SDP [24]. The following lemma, taken from [42], provides the computation method making use of D-scales.

Lemma 2.5 (skewed- μ upper bound). The skewed- μ upper bound is expressed as

$$\tilde{\mu}_{\tilde{\Delta}}(M) \leq \sqrt{\hat{\beta}}, \quad (2.58)$$

where

$$\hat{\beta} = \min_{X \in \mathbf{D}} \beta$$

subject to:

$$\begin{aligned} M^* \begin{bmatrix} X & 0 \\ 0 & I \end{bmatrix} M - \begin{bmatrix} X & 0 \\ 0 & \beta I \end{bmatrix} &\preceq 0 \\ X > 0 \\ \beta > 0. \end{aligned} \tag{2.59}$$

Proof. The proof uses the $1/\sqrt{\gamma}$ scaling on the performance channels and D-scales on the uncertainty channels as depicted in Figure 2.7 on the bottom. The D-scales are obtained by the matrix square-root $D = \sqrt{X}$. For details, see [42] and references therein. \diamond

2.3.2 Lower bound

The lower bound of skewed- μ is obtained by the construction of worst-case uncertainty samples that achieve the lower bound at the given frequency. The lower bounds L_k at frequencies ω_k are obtained employing a skewed- μ power iteration [23]. To describe the method in more detail, we define the matrix sample of the uncertainty. The response of $\Delta_d(s) \in \mathbb{A}_d$ at any frequency is a block structured, complex matrix \mathcal{Q}_d . The set of block structured, unit norm bounded matrix samples is defined as

$$\mathbb{Q}_d := \left\{ \begin{bmatrix} Q_1 & & 0 \\ & \ddots & \\ 0 & & Q_{N_d} \end{bmatrix} : Q_i \in \mathbb{C}^{r_i \times c_i}, \bar{\sigma}(Q_i) = 1, \text{rank}(Q_i) = 1, \right. \\ \left. i = 1, \dots, N_d \right\}. \tag{2.60}$$

Note that $\mathcal{Q}_d \in \mathbb{Q}_d$ implies $\bar{\sigma}(\mathcal{Q}_d) = 1$. In addition, note that it is sufficient to restrict the complex matrices to be rank one as in the definition of \mathbb{Q}_d [40, 73]. The set of mixed uncertainty samples at a fixed frequency is

$$\mathbb{Q} := \left\{ \begin{bmatrix} \Delta_p & 0 \\ 0 & \mathcal{Q}_d \end{bmatrix} : \Delta_p \in \mathbb{A}_p, \mathcal{Q}_d \in \mathbb{Q}_d \right\}. \tag{2.61}$$

The power iteration at ω_k yields $\mathcal{Q}_k \in \mathbb{Q}$, such that,

$$\bar{\sigma}(\mathcal{F}_U(M(j\omega_k), \mathcal{Q}_k)) = L_k. \tag{2.62}$$

Suppose that the skewed- μ lower bound power iteration at ω_0 yields the lower bound L_0 and worst-case perturbation

$$\mathcal{Q}_0 = \begin{bmatrix} \Delta_{p,0} & 0 \\ 0 & \mathcal{Q}_{d,0} \end{bmatrix} \in \mathbb{Q}, \quad \text{where} \quad \mathcal{Q}_{d,0} = \begin{bmatrix} Q_1 & & 0 \\ & \ddots & \\ 0 & & Q_{N_d} \end{bmatrix} \in \mathbb{Q}_d. \tag{2.63}$$

The sample $\Delta_{p,0}$ can be substituted into $P(\Delta, s)$ directly but $\mathcal{Q}_{d,0}$ is a complex matrix at this point. Therefore, the goal is to find an LTI uncertainty which interpolates $\mathcal{Q}_{d,0}$ at the single frequency ω_0 . The uncertainty must be stable, unit norm bounded and have the correct block structure, i.e., we want to find $\Delta_{d,s}(s) \in \mathbb{A}_d$. The 's' subscript refers to the

fact that the interpolation is performed at a single frequency. The uncertainty

$$\Delta_s(s) = \begin{bmatrix} \Delta_{p,0} & 0 \\ 0 & \Delta_{d,s}(s) \end{bmatrix} \quad (2.64)$$

drives the gain of the uncertain system to the power iteration lower bound at ω_0 , in other words it satisfies $\bar{\sigma}(\mathcal{F}_U(M(j\omega_0), \Delta_s(j\omega_0))) = L_0$. The construction of $\Delta_{d,s}(s)$ is given in the proof of the small gain theorem in [73, Theorem 9.1] and is summarized in the following. The SISO case is addressed first.

Theorem 2.5. A finite frequency ω_0 and a complex number $\Delta_0 \in \mathbb{C}$ are given. There exists a stable SISO LTI system $\Delta(s)$ that interpolates the data $\Delta(j\omega_0) = \Delta_0$, and whose magnitude $|\Delta(j\omega)| \leq |\Delta_0| \forall \omega$.

Proof. First, consider the fact that a system of the form $T(s) = \frac{s-\beta}{s+\beta}$ with $\beta > 0$ is stable and has unit magnitude over all frequencies. Moreover, the phase of $T(s)$ goes from 180° to 0° with increasing frequency. Similarly, $-T(s)$ is stable with unit magnitude and has phase that goes from 360° to 180° . Thus, a transfer function of the form $\pm c \frac{s-\beta}{s+\beta}$ with $c > 0$ can achieve any desired phase and magnitude at any frequency.

If $\Delta_0 \in \mathbb{R}$, then select $\Delta(s) = \Delta_0$. If $\text{Im}(\Delta_0) \neq 0$, then $\Delta_0 = \pm ce^{j\phi}$ for some $c > 0$ and $\phi \in (0^\circ, 180^\circ)$. Note that since $\beta > 0$, the phase of $T(s)$ is

$$\begin{aligned} \arg T(j\omega) &= \arg \frac{\omega + j\beta}{\omega - j\beta} = \arg(\omega + j\beta) - \arg(\omega - j\beta) \\ &= \tan^{-1}\left(\frac{\beta}{\omega}\right) - \tan^{-1}\left(\frac{-\beta}{\omega}\right) = 2\tan^{-1}\left(\frac{\beta}{\omega}\right). \end{aligned} \quad (2.65)$$

To achieve the desired phase ϕ at ω_0 , select $\beta = \omega_0 \tan\left(\frac{\phi}{2}\right)$ so that $T(j\omega_0) = e^{j\phi}$.

Finally, define $\Delta(s) = \pm |\Delta_0| \frac{s-\beta}{s+\beta}$ with the appropriate sign. Then, $\Delta(s)$ is stable with pole $-\beta$ ($\beta > 0$), $\Delta(j\omega_0) = \Delta_0$, and $\|\Delta(s)\|_\infty = |\Delta_0|$. \diamond

Each block $Q_i \in \mathbb{C}^{r_i \times c_i}$ of $\mathcal{Q}_{d,0}$ is rank-one with unit maximum singular value, i.e., each block can be expressed as $Q_i = uv^*$ for some vectors $\|u\|_2 = \|v\|_2 = 1$. Let $u_l \in \mathbb{C}$ denote the l^{th} element of the vector u . It is possible to find a SISO transfer function $\hat{u}_l(s)$, such that, $\hat{u}_l(j\omega_0) = u_l$ and $\|\hat{u}_l(s)\|_\infty \leq |u_l|$. This SISO interpolation can be performed with a constant or first-order transfer function $\hat{u}_l(s)$ as described in the proof of Theorem 2.5. Each entry of u can be interpolated to obtain an $r_i \times 1$ stable system $\hat{u}(s)$ with $\|\hat{u}(s)\|_\infty \leq 1$. Similarly, each entry of v^* can be interpolated to obtain a $1 \times c_i$, stable system $\hat{v}(s)$ with $\|\hat{v}(s)\|_\infty \leq 1$. Finally, the block $\Delta_i(s) = \hat{u}(s) \hat{v}(s)$ is stable and satisfies $\Delta_i(j\omega_0) = Q_i$ and $\|\Delta_i\|_\infty \leq 1$. As a result,

$$\Delta_{d,s}(s) = \begin{bmatrix} \Delta_1(s) & 0 \\ \ddots & \Delta_{N_d}(s) \end{bmatrix} \in \Delta \quad (2.66)$$

interpolates $\mathcal{Q}_{d,0}$ at frequency ω_0 .

One entry of u can be normalized in each block of $Q_i = uv^*$. Specifically, $\|u\|_2 = 1$ implies u has at least one non-zero entry, say $u_l \neq 0$. Then $Q_i = \tilde{u}\tilde{v}^*$ where $\tilde{u} := u/u_l$ and $\tilde{v} := u_l v$. This normalizes the l^{th} entry, i.e., $\tilde{u}_l = 1$. Therefore, if $\mathcal{Q}_{d,0} \in \mathbb{C}^{r_Q \times c_Q}$, then in general $\Delta_{d,s}(s)$ has $r_Q + c_Q - N_d$ states. The number of states can be less if there

are real numbers in the rank-one decomposition of the blocks of $\mathcal{Q}_{d,0}$. The most notable situations when that happens are when $\omega_0 = 0$ or $\omega_0 = \infty$. In those cases, $M(j\omega_0)$ is a real matrix therefore in some cases, the power iteration results in a real $\mathcal{Q}_{d,0}$.

Typically, this method is used to interpolate the matrix $\mathcal{Q}_{d,0}$ that achieves the maximal lower bound at frequency ω_0 . Therefore, $\Delta_s(s)$ achieves the largest gain found by the power iteration over all frequencies in the grid. However, the gain of the system $\bar{\sigma}(\mathcal{F}_U(M(j\omega), \Delta_s(j\omega)))$ may not be large at other frequencies ($\omega \neq \omega_0$). The `wcgain` function in MATLAB computes the worst-case uncertainty using this method. In the implementation, a band-pass filter is also added to the uncertainty. This band-pass filter is omitted from the discussions in this thesis. A simple example is presented next to demonstrate this.

Example 2.5. Consider two band-pass filters with peaks at 1 rad/s and 100 rad/s, respectively:

$$F_1(s) = \frac{2s}{(s+1)^2}, \text{ and } F_2(s) = \frac{200s}{(s+100)^2}. \quad (2.67)$$

Let $\delta_1 \in \mathbb{R}$ with $|\delta_1| \leq 1$ and $\Delta_1(s)$ with $\|\Delta_1(s)\|_\infty \leq 1$. The uncertain system is given as

$$P(\Delta, s) = [2(0.2 + \delta_1)F_1(s) + (0.9 - \delta_1)F_2(s)](1 + W_d(s)\Delta_1(s)) \quad (2.68)$$

where

$$W_d(s) = 0.2 \frac{s+7}{s+14}. \quad (2.69)$$

That is, $P(\Delta, s)$ is the weighted sum of $F_1(s)$ and $F_2(s)$ with 10% dynamic uncertainty on low frequencies and 20% on high frequencies. The uncertainty sets are

$$\begin{aligned} \mathbb{A}_p &= \{\delta_1 I_2 : \delta_1 \in \mathbb{R}, |\delta_1| \leq 1\}, \\ \mathbb{A}_d &= \{\Delta_1(s) : \Delta_1(s) \text{ is LTI}, \|\Delta_1(s)\|_\infty \leq 1\}, \\ \mathbb{A} &= \left\{ \begin{bmatrix} \mathbb{A}_p & 0 \\ 0 & \Delta_d(s) \end{bmatrix} : \Delta_p \in \mathbb{A}_p, \Delta_d(s) \in \mathbb{A}_d \right\}. \end{aligned} \quad (2.70)$$

Due to the construction of $P(\Delta, s)$, it is sensitive to excitation at 1 rad/s and 100 rad/s. This is illustrated in Figure 2.8 in which the gain of the nominal system $P(0, s)$ and the worst-case gain lower bound $L(\omega)$ are depicted. The maximum of $L(\omega)$ is at $\omega_0 = 1$ rad/s. The skewed- μ power iteration yields

$$\mathcal{Q}_0 = \begin{bmatrix} I_2 & 0 \\ 0 & 0.998 - 0.071j \end{bmatrix}. \quad (2.71)$$

The dynamic uncertainty constructed with the classical method in Theorem 2.5 interpolates $Q_1 = 0.998 - 0.071j$ at $\omega_0 = 1$ rad/s with a stable norm bounded LTI system. The result is

$$\Delta_{1,s}(s) = -\frac{s - 28.32}{s + 28.32}. \quad (2.72)$$

Thus,

$$\Delta_s(s) = \begin{bmatrix} I_2 & 0 \\ 0 & \Delta_{1,s}(s) \end{bmatrix}. \quad (2.73)$$

As depicted in Figure 2.8, this uncertainty sample drives the magnitude of $P(\Delta_s, s)$ to

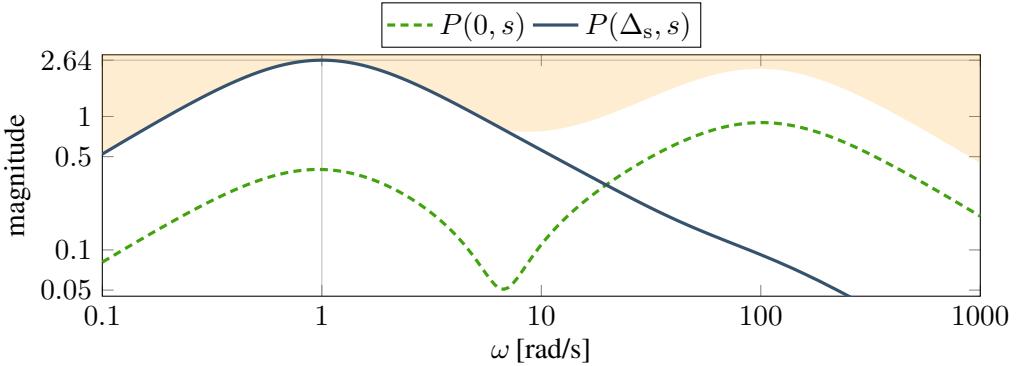


Figure 2.8: The gain of the uncertain system $P(\Delta_s, s)$, in Example 2.5, for different values of the uncertainty. The boundary of the shaded area is $L(\omega)$.

its maximum at ω_0 . However, the gain of $P(\Delta_s, s)$ at the second peak is less than the gain of the nominal system $P(0, s)$.

2.3.3 Worst-case gain as control objective

The worst-case gain minimization is not an often considered objective for control design in the literature. This is mainly the case because of the difficulty it poses for computation. For optimal robust control design in current engineering practice, the H_∞ and structured H_∞ synthesis in Section 2.1.2 are the most applied techniques. Using these methods, the uncertainty is accounted for implicitly, or it is represented as a full block dynamic uncertainty. To summarize and compare the different robust control design methods, the three performance measures mentioned thus far are illustrated in Figure 2.9. The figure shows the maximum H_∞ norm of an uncertain system as a function of the level of uncertainty (norm of $\Delta(s)$). The nominal gain of the system is at $\|\Delta(s)\|_\infty = 0$, i.e., it is $\|P(0, s)\|_\infty$. The minimization of the nominal gain is the objective of both the structured and unstructured nominal H_∞ synthesis.

To a lesser extent, more advanced problems with complex uncertainty structures are considered in the literature, see Chapter 1 for examples. For control design in these applications, the μ -synthesis by D-K iteration is used almost exclusively. This prevalence of μ -synthesis is due to it being a well-established method readily available in MATLAB. However, by minimizing μ , μ -synthesis optimizes the so called robust performance which has an important drawback compared to worst-case gain. As depicted in Figure 2.9, decreasing the robust performance requires the extension of the stability region which, on its own, is unnecessary when $\|\Delta(s)\|_\infty > 1$.

The worst-case gain is the more natural extension of the H_∞ metric to uncertain systems, since it is the maximum of the H_∞ norm over the allowable uncertainty set, and optimizing it does not necessitate the stabilization of the system beyond $\|\Delta(s)\|_\infty \leq 1$. Even if worst-case gain could be a more useful design objective in engineering problems, it is rarely used primarily because of the lack of synthesis tools. A notable recently

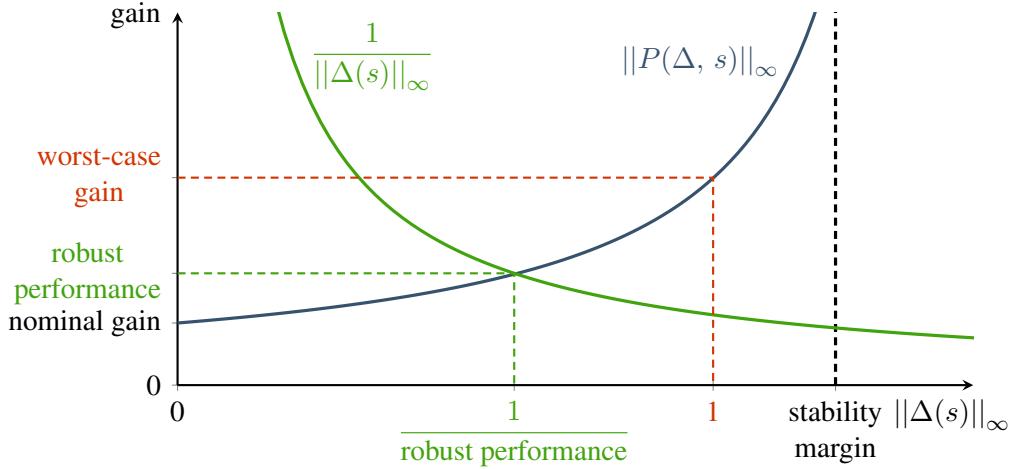


Figure 2.9: Comparison of robust control design performance measures.

published method for worst-case gain optimal control design is the hybrid relaxation approach in [2, Algorithm 2]. Inspired by this algorithm, Chapter 3 provides a novel method for worst-case gain minimization with structured control synthesis.

3

Structured robust synthesis method to minimize worst-case gain

This chapter presents a method for worst-case gain minimization using structured robust control design for systems with a mixture of parametric and dynamic uncertainty. The proposed method alternates between an analysis step and a synthesis step. Samples of the parametric uncertainty are computed during the analysis steps thus yielding an array of uncertain systems containing only dynamic uncertainty. The controller is then synthesized on this array of uncertain models. This synthesis step itself involves an alternation between constructing a D-scale for each of the uncertain systems and tuning a single controller for the entire collection of scaled plants. The controller tuning is performed using structured control design techniques. After a literature review in Section 3.1, the mathematical formulation of the design problem is given in Section 3.2. Then, Section 3.3 provides an overview of the algorithm. The construction of the D-scales, and the synthesis and analysis steps are elaborated in Sections 3.4, 3.5, and 3.6 respectively. Finally numerical examples are provided in Section 3.7.

3.1 Related literature

The best known and most widely used control design method against mixed uncertainty is the μ -synthesis by D-K iteration detailed in Section 2.2.3. The major difference between the solution in this Chapter and the traditional D-K iteration lies in the choice of the objective function, as explained in Section 2.3.3 and illustrated in Figure 2.9. Another key distinction is that during the D-K iteration, D-scales are calculated to treat the parametric uncertainty, as well as the dynamic uncertainty. If the number of repetitions for an uncertain parameter is high, then the fitting of the corresponding D-scale can get numerically intractable causing the iteration to terminate prematurely. Physical systems in engineering problems are often described by equations in which a certain parameter (mass, inertia, position of the center of gravity, etc.) appears a large number of times, hence this is often a concern.

The algorithm proposed in the present chapter resembles the hybrid relaxation approach in [2, Algorithm 2]. This is a structured synthesis algorithm that improves numerical conditioning by collecting worst-case samples of the uncertain parameters thus eliminating the need for the construction of a D-scale for parametric uncertainty. A single structured controller and structured D-scale (for the dynamic uncertainty only) is

designed simultaneously for all the samples to minimize the worst-case gain. The subsequent analysis step calculates the worst-case gain along with the worst-case uncertainty sample which is then added to the collection.

The novelty of the method in this thesis lies in the control synthesis step. In particular, the hybrid relaxation has a synthesis step that involves parameterizing a single dynamic D-scale for all uncertainty samples. It then jointly optimizes over the tunable parameters in both the scaling and controller using structured synthesis techniques. This approach is effective for many problems but the restriction to a single dynamic scaling might be conservative on certain examples (e.g., see Section 3.7.2). As remarked in [2], it is possible to extend the hybrid relaxation to parameterize one scaling for each parametric uncertainty sample at the expense of a larger number of optimization variables.

In contrast, the approach given in this thesis utilizes dynamic scalings that depend on the parametric uncertainty sample. The use of parameter-dependent scalings reduces conservatism at the expense of requiring a coordinate-wise D-K iteration for the synthesis, similarly to the traditional μ -synthesis. This coordinate-wise iteration alternates between optimizing the D-scaling while holding the controller fixed and vice versa. This tends to be less effective than the joint scaling and controller optimization used for the synthesis in the hybrid relaxation approach. If the D-scales depend strongly on the value of the uncertain parameters however, the individual D-scale construction has the advantage. Another distinction is that a branch-and-bound implementation for the worst-case gain analysis is used in the proposed algorithm, which provides tighter analysis bounds.

3.2 Problem statement

The robust control synthesis problem is formulated using the feedback interconnection shown in Figure 2.6. For simplicity, the dependence of transfer functions on the Laplace variable s is omitted in this chapter. The uncertain plant $P(\Delta) = \mathcal{F}_U(N, \Delta)$ consists of an LTI system N and structured LTI uncertainty Δ . The objective is to design an LTI controller with fixed structure, $K(\kappa)$, where $\kappa \in \mathbb{R}^{n_\kappa}$ is a vector of tunable design parameters. The goal of the synthesis is to minimize the worst-case gain of the closed-loop in Figure 2.6 by tuning the design parameters κ . Hence, the objective function is

$$J(\kappa) := \max_{\Delta \in \Delta} \|\mathcal{F}_L(P(\Delta), K(\kappa))\|_\infty. \quad (3.1)$$

This definition assumes that $\mathcal{F}_L(P(\Delta), K(\kappa))$ is robustly stable, i.e., stable for all $\Delta \in \Delta$. If the closed-loop is not robustly stable, then $J(\kappa)$ is defined to be $+\infty$. The synthesis objective is then to find the optimal controller parameters

$$\kappa^* = \arg \min_{\kappa \in \mathbb{R}^{n_\kappa}} J(\kappa). \quad (3.2)$$

(See Chapter 2 for a more detailed explanation of the mentioned concepts.) According to our engineering experience, optimizing the worst-case gain is a better performance objective than the usual robust performance detailed in Section 2.2.3 and also in [41]. If robust stability is reached, the design objective is to improve performance without extending the stability margin any further. Structured robust synthesis is, in general, a non-convex optimization problem. The algorithm described in the chapter (in Sections 3.3-3.6) is an iterative method to compute sub-optimal controller parameters. This solution is inspired by the hybrid relaxation approach in [2].

3.3 Overview of the design algorithm

This section introduces the proposed algorithm for computing a structured controller to minimize the closed-loop worst-case gain. An overview summary of the proposed approach is given in Algorithm 1. In Step 1, the algorithm expects that a controller $K(\kappa_{\text{init}})$ is available for initialization. It is further assumed that this controller robustly stabilizes the system for the dynamic uncertainty (Δ_d) and the nominal value of the parametric uncertainty ($\Delta_p = 0$). In some problems, the open loop is robustly stable and hence $K(\kappa_{\text{init}}) = 0$ is sufficient for initialization. Otherwise an initial robustly stabilizing controller can be computed with a related stability margin maximization method. Algorithm 1 is iterative and utilizes the sample set of the parametric uncertainty, $\Delta_{p,s}$, and the D-scales, $\mathbb{D}_{d,s}$. The set $\Delta_{p,s}$ is initialized with the nominal value of the parametric uncertainty, i.e., $\Delta_p = 0$ (Step 2). These sets are updated throughout the iteration, as described further below.

The first key step of Algorithm 1 is to synthesize a structured robust controller using an iteration (Steps 5-8) that is similar to the unstructured, full-order D-K iteration in Section 2.2.3. This alternates between a D-step (Step 6) which computes a set of scalings $\mathbb{D}_{d,s}$ and a K-step (Step 7) which computes a structured robust controller $K(\kappa)$. This iteration terminates when $\bar{\gamma}_d$ does not decrease significantly compared to the previous iteration, or when the maximum number of iterations is reached. Our specific implementation terminates the D-K inner loop if $\bar{\gamma}_d$ fails to decrease by more than 1%, or the maximum number of 15 iterations is reached. These values provide a reasonable compromise between precision and computation time, however, other choices are conceivable. (See Section 3.7 for a discussion of computation time.)

The D-step (Step 6) first constructs a set of closed-loops with the structured controller $K(\kappa)$ and each sample of the parametric uncertainty $\Delta_p \in \Delta_{p,s}$ obtained in Step 9. Each closed-loop sample still has the remaining dynamic uncertainty Δ_d . Worst-case gain analyses are performed to compute a scaling $D_d \in \mathbb{D}_d$ for each closed-loop with corresponding sample $\Delta_p \in \Delta_{p,s}$. Thus $\mathbb{D}_{d,s}$ has the same number of elements as $\Delta_{p,s}$. Details on this step are given in Section 3.4.

The K-step (Step 7) forms a collection of scaled plants from the samples of parametric uncertainty and D-scales. Specifically, a scaled plant is constructed for each sample $(\Delta_p, D_d) \in \Delta_{p,s} \times \mathbb{D}_{d,s}$ as shown in Figure 3.1. Then systune is used to compute a structured controller $K(\kappa)$ to minimize the worst-case gain on this collection of sampled, scaled systems. Details on this step are provided in Section 3.5.

Finally, an analysis step (Step 9) is performed on the resulting closed-loop which is $\mathcal{F}_L(P(\Delta), K(\kappa))$. In the analysis, the entire uncertainty Δ is considered using a version of wcgain with additional branch-and-bounding on the parametric uncertainty. (With branch-and-bounding, the worst-case gain is computed iteratively to bring the upper and lower bounds as close to each other as possible.) This generates lower and upper bounds on the worst-case gain. It also computes a worst-case uncertainty that achieves the lower bound. The parametric block of this worst-case uncertainty is added to the sample set $\Delta_{p,s}$ (Step 10). Details on the analysis are given in Section 3.6.

The outer-loop of the algorithm terminates (Step 3) if the worst-case synthesis gain is within the interval of the analysis bounds, i.e., $\gamma_s \in [\underline{\gamma}_a, \bar{\gamma}_a]$, and $\bar{\gamma}_a$ fails to decrease significantly (more than 1%) over the last iteration. The loop also terminates if the max-

Algorithm 1 Structured robust synthesis.

1: **Given:** Initial controller $K(\kappa_{\text{init}})$ which is robustly stabilizing for Δ_d and $\Delta_p = 0$.
 2: **Initialize:** $\Delta_{p,s} := \{0\}$ and $\kappa := \kappa_{\text{init}}$.
 3: **while** termination criteria are not met **do**
 4: **Synthesis Iteration**
 5: **while** D-K termination criteria are not met **do**
 6: **Parameter-Dependent D-Scales (Section 3.4):** Form a set of closed-loops T with $K(\kappa)$ and each sample of parametric uncertainty $\Delta_p \in \Delta_{p,s}$ (Figure 3.1). For each closed-loop, solve for the scaling $D_d \in \Delta_d$ to minimize the worst-case gain upper bound over the dynamic uncertainty Δ_d .
 Output: Set of dynamic scalings $\mathbb{D}_{d,s}$ and largest worst-case gain upper bound $\bar{\gamma}_d$.
 7: **Controller Synthesis (Section 3.5):** Form scaled plants from each sample of parametric uncertainty and scaling $(\Delta_p, D_d) \in \Delta_{p,s} \times \mathbb{D}_{d,s}$ (Figure 3.1). Use `systune` to solve for a structured controller to minimize the worst-case gain on this collection of sampled, scaled systems.
 Output: Controller parameters κ and worst-case synthesis gain γ_s .
 8: **end while**
 9: **Analysis (Section 3.6):** Use a modified version of `wcgain` to compute upper and lower bounds on the worst-case gain of the closed-loop with structured controller $K(\kappa)$ over the set of uncertainty Δ .
 Output: Bounds $(\underline{\gamma}_a, \bar{\gamma}_a)$ on worst-case gain, and worst-case uncertainty

$$\Delta^* = \begin{bmatrix} \Delta_p^* & 0 \\ 0 & \Delta_d^* \end{bmatrix}.$$

 10: **Sample Update:** Set $\Delta_{p,s} := \Delta_{p,s} \cup \{\Delta_p^*\}.$
 11: **end while**

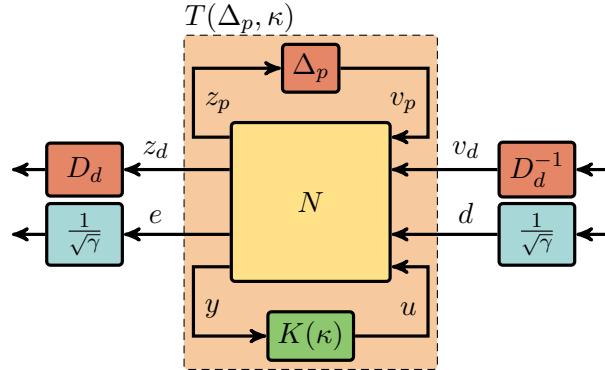


Figure 3.1: Scaled plant used in the D-K iteration in Algorithm 1.

imum number of iteration counts (which is chosen as 30) is reached.

The outline of Algorithm 1 shows resemblance to the hybrid relaxation approach in [2]. The key difference between the two approaches is the control synthesis in Steps 5-8. Specifically, in Algorithm 1 a separate $D_d \in \mathbb{D}_{d,s}$ corresponds to the individual samples in $\Delta_{p,s}$, instead of a common D_d . This solution reduces conservatism, and decreases the number of decision variables for the structured synthesis. This comes at the expense of having to use an iterative process for the controller synthesis, however. Another distinction between the two approaches is that we use branch-and-bounding for the worst-case gain computation, which results in tighter bounds.

The proposed algorithm is designed to handle mixed uncertainty. If the system only contains dynamic uncertainty ($\Delta = \Delta_d$) then only Steps 5-8 are performed. On the other hand, if the uncertainty is purely parametric ($\Delta = \Delta_p$) then the samples are LTI systems without uncertainty. In this case, Steps 5-8 are replaced by a single controller synthesis step, where we find a controller that simultaneously stabilizes all the samples and minimizes performance at the same time. This is done using `hinfstruct`.

3.4 Parameter-dependent D-scales

Since a different D-scale is constructed for each parametric uncertainty sample, this section only discusses the calculations involving a fixed value of $\Delta_p \in \Delta_p$. Section 3.5 treats the collection of samples in $\Delta_{p,s}$. Let Δ_p and a controller $K(\kappa)$ be given. The corresponding closed-loop $T(\Delta_p, \kappa)$, shown in Figure 3.1, is defined as

$$T(\Delta_p, \kappa) := \mathcal{F}_L(\mathcal{F}_U(N, \Delta_p), K(\kappa)). \quad (3.3)$$

The worst-case gain for the closed-loop over the remaining dynamic uncertainty is given by

$$\max_{\Delta_d \in \Delta_d} \|\mathcal{F}_U(T(\Delta_p, \kappa), \Delta_d)\|_\infty. \quad (3.4)$$

It follows from standard μ analysis results [72, 40] that an upper bound on this worst-case gain over Δ_d is the result of the optimization

$$\begin{aligned} & \min_{\substack{D_d \in \mathbb{D}_d \\ \gamma > 0}} \gamma \\ & \text{subject to:} \end{aligned} \quad (3.5)$$

$$\left\| \begin{bmatrix} D_d & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{bmatrix} T(\Delta_p, \kappa) \begin{bmatrix} D_d^{-1} & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{bmatrix} \right\|_\infty \leq 1.$$

Figure 3.1 shows the sampled closed-loop with D-scales and performance gain scaling. For the remainder of the derivation, the notational dependence of T on (Δ_p, κ) is dropped for simplicity. Form the partitioning $T =: \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$ where T_2 has the row dimension of the generalized error signal e . Recall that, for any complex matrix L , it follows that $\bar{\sigma}(L) \leq 1$ if and only if $I - L^* L \succeq 0$. This can be used to express the constraint in (3.5) as a (frequency-dependent) LMI. By the Schur complement lemma and the definition of the H_∞ norm, this constraint is equivalent to the frequency domain condition

$$\begin{bmatrix} \gamma I & T_2(j\omega) \\ T_2(j\omega)^* & \begin{bmatrix} X(j\omega) & 0 \\ 0 & \gamma I \end{bmatrix} - T(j\omega)^* \begin{bmatrix} X(j\omega) & 0 \\ 0 & 0 \end{bmatrix} T(j\omega) \end{bmatrix} \succeq 0, \quad \forall \omega, \quad (3.6)$$

where $X(j\omega) := D_d(j\omega)^* D_d(j\omega)$.

Every $D_d \in \mathbb{D}_d$ has the structure

$$D_d = \begin{bmatrix} d_1 I, & 0 \\ & \ddots \\ 0 & d_{N_d} I \end{bmatrix}, \quad (3.7)$$

with each d_i as a stable, minimum-phase, invertible SISO system. Thus, $X(j\omega)$ has the structure

$$X(j\omega) = \begin{bmatrix} x_1(j\omega) I & 0 \\ & \ddots \\ 0 & x_{N_d}(j\omega) I \end{bmatrix}, \quad (3.8)$$

where $x_i(j\omega) = d_i(j\omega)^* d_i(j\omega) \geq 0$ for all ω . Next, we write x_i in terms of basis functions and coefficients which results in a convex optimization problem for calculating the D-scales. Express each x_i as $x_i = \psi^* \Lambda_i \psi$ where ψ is a (fixed) column vector of stable, minimum phase basis functions and $\Lambda_i = \Lambda_i^*$ are decision variables to be optimized.

For the subsequent synthesis step (in Section 3.5), it is advantageous to find a D-scale, that does not only minimize the peak of γ , but minimizes it over a broad frequency range. To illustrate this, consider Figure 3.2. If we solve (3.5) frequency by frequency while allowing the D-scales to vary arbitrarily between each point, then we obtain the theoretical worst-case gain lower bound. This is the dashed blue curve in Figure 3.2, labeled "theoretical". If we search for a realizable $D_d(s)$ by enforcing (3.6) only at the frequency of the peak, then the $\gamma(\omega)$ we get is, in general, only accurate at the peak and can be far from the theoretical value at other frequencies. See the red dash-dotted line in Figure 3.2 with label "single-frequency optimum". Using such a D-scale leaves little room for the control synthesis to improve the performance further. Therefore, it is better to enforce (3.6) at several frequencies at the same time. In practice, this calculation

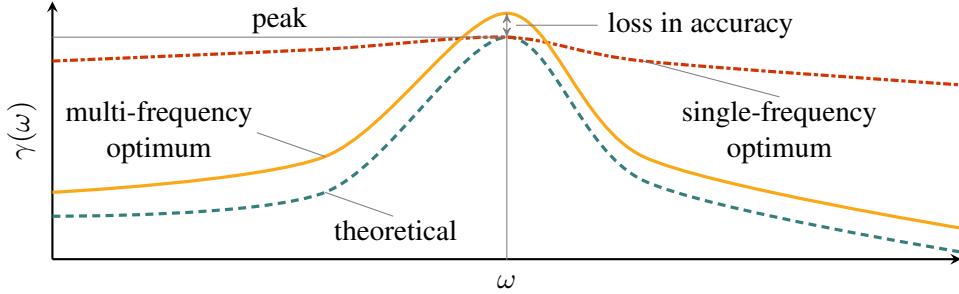


Figure 3.2: Gain of the scaled system with D -scales obtained using different methods.

usually leads to some inaccuracy at the peak but it provides lower γ values at the rest of the frequencies, as illustrated by the continuous yellow curve in Figure 3.2 called "multi-frequency optimum". To account for this, the constraint in (3.6) is enforced on a frequency grid $\{\omega_k\}_{k=1}^{N_\omega}$. The cost function to be minimized is turned into

$$\hat{\gamma} + \sum_{k=1}^{N_\omega} \gamma_k, \quad (3.9)$$

where γ_k is the gain at ω_k and $\hat{\gamma}$ is the peak gain over all frequencies, i.e.,

$$\hat{\gamma} \geq \gamma_k, \quad k = 1, \dots, N_\omega \quad (3.10)$$

is added to the constraints.

To ensure that $x_i(j\omega) \geq 0$ for all ω , additional constraints from the KYP (Kalman-Yakubovich-Popov) lemma are added [58]. This guarantees that the solution X obtained from the optimization can be spectral factorized to get a stable, minimum phase scaling D_d . Let $\psi(j\omega) = C_\psi (j\omega I - A_\psi)^{-1} B_\psi + D_\psi$, then $\psi(j\omega)^* \Lambda_i \psi(j\omega) \geq 0$ for all ω is expressed as

$$\begin{bmatrix} (j\omega I - A_\psi)^{-1} B_\psi \\ I \end{bmatrix}^* \begin{bmatrix} C_\psi^* \\ D_\psi^* \end{bmatrix} \Lambda_i \begin{bmatrix} C_\psi & D_\psi \end{bmatrix} \begin{bmatrix} (j\omega I - A_\psi)^{-1} B_\psi \\ I \end{bmatrix} \geq 0 \quad (3.11)$$

for all ω . By the KYP lemma, this condition is equivalent to the existence of $Q_i = Q_i^*$, such that,

$$\begin{bmatrix} A_\psi^* Q_i + Q_i A_\psi & Q_i B_\psi \\ B_\psi^* Q_i & 0 \end{bmatrix} - \begin{bmatrix} C_\psi^* \\ D_\psi^* \end{bmatrix} \Lambda_i \begin{bmatrix} C_\psi & D_\psi \end{bmatrix} \preceq 0. \quad (3.12)$$

The conclusion of the D -scale construction is that the optimization in (3.5) is reformulated as the minimization of the cost function in (3.9) with constraints (3.6) for each ω_k and γ_k , (3.10), and (3.12). This is a convex SDP in the variables $\hat{\gamma}$, $\{\gamma_k\}_{k=1}^{N_\omega}$, $\{\Lambda_i\}_{i=1}^{N_d}$, and $\{Q_i\}_{i=1}^{N_d}$. Finally, this optimization is solved to compute a dynamic scaling D_d for each uncertainty sample $\Delta_p \in \Delta_{p,s}$.

3.5 Structured synthesis for the collection of scaled plants

The output of the D -step is a set of dynamic scalings $\mathbb{D}_{d,s}$ each computed for a corresponding element of the parameter uncertainty set $\Delta_{p,s}$. The synthesis step optimizes

the controller $K(\kappa)$ in Figure 3.1 to minimize the worst-case gain over the set of scaled, closed-loop samples. This is formulated as the optimization:

$$\min_{\substack{\kappa \in \mathbb{R}^{n_\kappa} \\ \gamma > 0}} \gamma$$

subject to:

$$\begin{aligned} & \left\| \begin{bmatrix} D_d & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{bmatrix} T(\Delta_p, \kappa) \begin{bmatrix} D_d^{-1} & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{bmatrix} \right\|_\infty \leq 1 \\ & \forall (\Delta_p, D_d) \in \Delta_{p,s} \times \mathbb{D}_{d,s}. \end{aligned} \tag{3.13}$$

Here, the H_∞ norm constraint is added for each parametric uncertainty sample and corresponding D-scale. This control synthesis is directly addressable using the `systune` command in MATLAB [4].

If $K(\kappa_{\text{init}})$ is not robustly stabilizing for the dynamic uncertainty, as assumed in Algorithm 1, the D-scales are set to identity in Step 6. Under these conditions, it is not guaranteed that the optimization in (3.13) yields a solution. If it does not, a different algorithm is invoked, which maximizes the stability margin of the system, thus provides a robustly stabilizing controller if possible. This algorithm is not detailed in this thesis to keep the focus on worst-case gain minimization problem. Also, for the problems we encountered, (3.13) always provided a stabilizing controller which suggests that it does so for most systems. (See Section 3.7.3 for a list of examples the algorithm was tested on.)

3.6 Worst-case analysis of the closed-loop

The analysis step computes upper and lower bounds on the worst-case gain of the closed-loop given in (2.51) for a fixed controller $K(\kappa)$. This is performed using the `wcgain` function in MATLAB but modified to incorporate branch-and-bound on the parametric uncertainty. The lower bound $\underline{\gamma}_a$ is computed using a combination of the skewed- μ power iteration for the dynamic uncertainty blocks and a Hamiltonian-based coordinate-wise iteration on the parametric uncertainty. This returns a worst-case uncertainty Δ^* that achieves the lower bound. The upper bound $\bar{\gamma}_a$ is computed using a frequency-gridded SDP with D-scales for the dynamic uncertainty and (D, G)-scales for the parametric uncertainty.

Branch-and-bounding of the real uncertainty is used to reduce the gap between the upper and lower bounds. Specifically, the real parameter uncertainty set Δ_p is described by a normalized hypercube. This hypercube is split and the upper/lower bounds are computed on each sub-cube. The splitting of sub-cubes continues until the gap between the overall upper and lower bounds gets below some relative error or a maximum number of cube splits is reached. Details on this analysis step are given in [42].

As opposed to the hybrid relaxation approach in [2], the case when the worst-case gain upper bound $\bar{\gamma}_a$ is infinite is treated differently. Instead of the uncertainty sample that corresponds the worst-case gain lower bound, the destabilizing uncertainty sample is chosen. This is obtained by evaluating the upper bound of the stability margin of the closed-loop, which is less than one in this case (see Figure 2.9). The computation is performed using the `robstab` function in MATLAB [26].

3.7 Numerical examples

In this section, two numerical examples are presented to demonstrate the applicability of Algorithm 1. The first example is one of the benchmarks (Example 12) used in [2]. The second is constructed to highlight the benefits of using different dynamic scalings $D_d \in \mathbb{D}_d$ for each uncertainty sample $\Delta_p \in \mathbb{A}_{p,s}$. Also, further numerical results are presented that were obtained by testing the algorithm on 31 example systems. For the application of this method on a real-life flutter control problem, see Chapter 5.

3.7.1 Benchmark example from [2]

The system N is ninth order with five inputs and outputs including three uncertainty, one performance, and one control channels. It has all but one pole in the left half of the complex plane. The uncertainty is

$$\Delta = \begin{bmatrix} \Delta_d & 0 \\ 0 & \delta_1 \\ 0 & \delta_2 \end{bmatrix}, \quad (3.14)$$

where Δ_d is SISO. The controller is parameterized as

$$K(\kappa) = \left[\begin{array}{ccc|c} \kappa_1 & \kappa_2 & \kappa_3 & \kappa_6 \\ \kappa_4 & 0 & 0 & \kappa_7 \\ 0 & \kappa_5 & 0 & \kappa_8 \\ \hline \kappa_9 & \kappa_{10} & \kappa_{11} & \kappa_{12} \end{array} \right]. \quad (3.15)$$

The initial controller parameter values are $\kappa_1 = -103.26$, $\kappa_2 = -20.41$, $\kappa_3 = -1.98$, $\kappa_4 = 16$, $\kappa_5 = \kappa_6 = \kappa_9 = \kappa_{10} = \kappa_{11} = 1$, and $\kappa_7 = \kappa_8 = \kappa_{12} = 0$. This parametrization is one of the built-in options (called 'companion') of the `tunableSS` function in MATLAB.

The basis functions for the D-scale construction (see Section 3.4) are

$$\psi_1(s) = 1, \quad \psi_i(s) = \frac{p_i}{s + p_i}, \quad i = 2, \dots, 9. \quad (3.16)$$

Hence, the D-scales are eighth order. The poles p_i are placed logarithmically inside the frequency interval of interest, which is determined using the `wcgain` function.

The results of each iteration are presented in Table 3.1. In the first iteration, the controller designed for the nominal value of the parametric uncertainty ($\delta_1 = \delta_2 = 0$) achieves the synthesis gain 0.1727. This controller does not provide robust stability over the entire Δ , hence the worst-case gain analysis yields $\underline{\gamma}_a = \bar{\gamma}_a = \infty$, and worst-case uncertainty with $\delta_1^* = 0.6382$ and $\delta_2^* = 0.6280$. This parametric uncertainty is added to the sample set $\mathbb{A}_{p,s}$. The second iteration proceeds similarly. The third iteration generates a controller with a synthesis gain $\gamma_s = 0.4381$. This controller is the first to achieve robust stability over Δ with corresponding (finite) worst-case gain upper bound $\bar{\gamma}_a = 3.6743$.

The synthesis and analysis gains converge in iteration 5, where the design process effectively ends (the new worst-case uncertainty sample is $\delta_1^* = \delta_2^* = 0$, which is already in $\mathbb{A}_{p,s}$). One additional iteration is performed to check whether $\bar{\gamma}_a$ can be decreased any

Iter.	γ_s	$\underline{\gamma}_a$	$\bar{\gamma}_a$	δ_1^*	δ_2^*
1.	0.1727	∞	∞	0.6382	0.6280
2.	0.3481	∞	∞	0.9218	0.9103
3.	0.4381	3.6387	3.6743	1	0.7129
4.	0.4653	0.5597	0.5609	-1	-1
5.	0.5023	0.5023	0.5033	0	0
6.	0.5023	0.5023	0.5033	\emptyset	\emptyset

Table 3.1: Iteration results of Algorithm 1 for the example of Section 3.7.1. (The \emptyset symbol in the last row means that no samples are collected because the iteration terminated.)

further. The final controller is

$$K(\kappa^*) = \left[\begin{array}{ccc|c} -15.874 & -6.7578 & 8.2503 & -28.2936 \\ 22.3798 & 0 & 0 & 36.0798 \\ 0 & -4.3234 & 0 & 20.2262 \\ \hline 2.7465 & 4.3159 & -3.9798 & -0.2511 \end{array} \right]. \quad (3.17)$$

Both our method and the algorithm in [2] ends with the gain 0.503.[†]

3.7.2 Example requiring parameter dependent D-scales

Consider the generalized design interconnection shown in Figure 3.3. The plant is given by $G(s) = \frac{1}{s+1}$. The additive uncertainty is

$$\tilde{\Delta} = \frac{\sqrt{0.05}}{1 + 0.98\delta_p} \cdot \Delta_d \cdot \frac{\sqrt{0.05}}{1 - 0.98\delta_p}, \quad (3.18)$$

where $|\delta_p| \leq 1$ and $\|\Delta_d\|_\infty \leq 1$. Therefore,

$$\Delta = \begin{bmatrix} \Delta_d & 0 \\ 0 & \delta_p I_2 \end{bmatrix}. \quad (3.19)$$

The worst-case gain upper bound of the uncertain plant $G + \tilde{\Delta}$ over Δ_d is depicted in Figure 3.4. Rewriting the plant as $G + \tilde{\Delta} = \mathcal{F}_U(T(\delta_p), \Delta_d)$, we get

$$T(\delta_p) = \begin{bmatrix} 0 & \frac{\sqrt{0.05}}{1-0.98\delta_p} \\ \frac{\sqrt{0.05}}{1+0.98\delta_p} & G \end{bmatrix}. \quad (3.20)$$

Because of the way the parametric uncertainty enters into $T(\delta_p)$, the D-scales corresponding to the worst-case gain upper bound depend on δ_p and the sign of δ_p as well. To demonstrate this, compute the D-scales on a frequency grid, i.e., for a fixed frequency

[†]The examples in [2] are normalized, therefore all gains are close to one in the paper. The authors shared the unscaled results with us via personal communication.

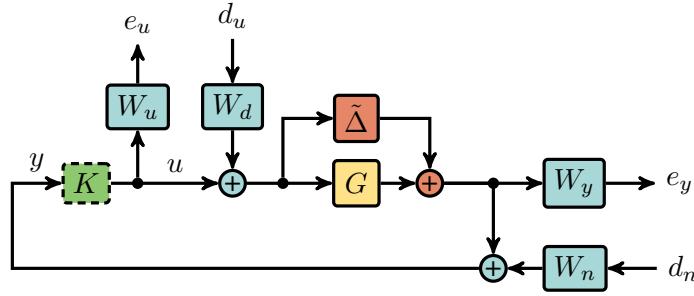


Figure 3.3: Generalized plant interconnection for the example of Section 3.7.2

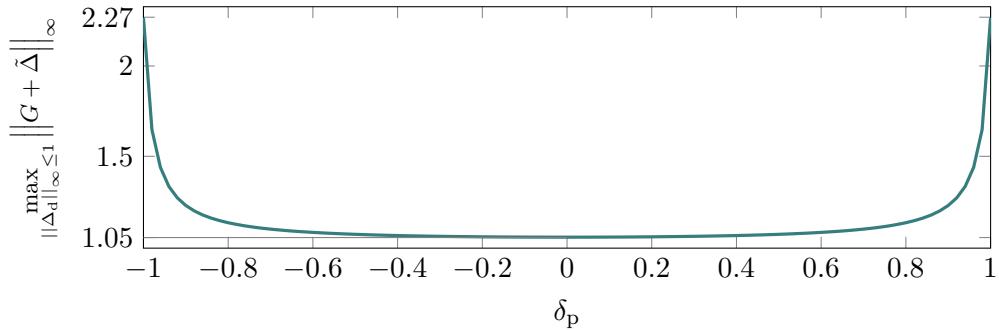


Figure 3.4: Worst case gain of $G + \tilde{\Delta}$ versus the parametric uncertainty in the example of Section 3.7.2.

and a fixed δ_p compute

$$\begin{aligned} & \max_{D_d \in \mathbb{C}} \gamma \\ & \text{subject to:} \\ & \bar{\sigma} \left(\begin{bmatrix} D_d & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} \end{bmatrix} T(\delta_p) \begin{bmatrix} D_d^{-1} & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} \end{bmatrix} \right) \leq 1. \end{aligned} \tag{3.21}$$

For a few values of δ_p , the resulting D-scales are shown in Figure 3.5. Notice, that both the gain in Figure 3.4, and the D-scales in Figure 3.5 vary with δ_p significantly. Also, the D-scales are different even if the corresponding gains are the same (e.g., $\delta_p = -1, +1$). Therefore, the analysis of multiple samples with a single D-scale yields inaccurate worst-case gain upper bound.

The weighting functions in Figure 3.3 serve to separate, in frequency, the disturbance rejection and noise cancellation requirements. These are given by

$$\begin{aligned} W_y(s) &= W_d(s) = \frac{1}{0.1s + 1}, \\ W_u(s) &= \frac{10s + 701.76}{s + 7088.46}, \quad \text{and} \\ W_n(s) &= \frac{s + 70.18}{s + 70888.1}. \end{aligned} \tag{3.22}$$

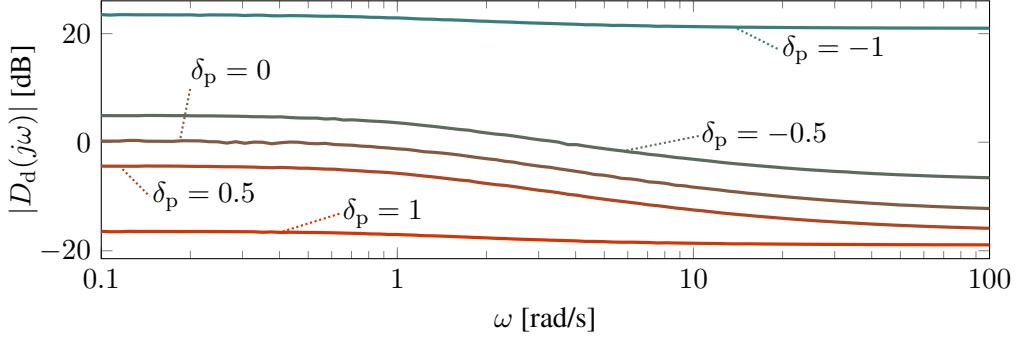


Figure 3.5: *D-scales corresponding to the worst-case gain upper bound of $G + \bar{\Delta}$ over Δ_d for a few values of the parametric uncertainty in the example of Section 3.7.2.*

Iter.	γ_s	$\underline{\gamma}_a$	$\bar{\gamma}_a$	δ_p^*
1.	0.1132	∞	∞	0.7127
2.	0.1453	∞	∞	-0.8390
3.	0.2095	∞	∞	0.9128
4.	0.4999	∞	∞	-0.9765
5.	0.8136	4.6441	4.6515	-1.0000
6.	1.5755	1.5755	1.5793	-1
7.	1.5755	1.5755	1.5793	\emptyset

Table 3.2: Algorithm 1 iteration results for the example of Section 3.7.2.

The controller is parameterized by a first order system

$$K(\kappa) = \left[\begin{array}{c|c} \kappa_1 & \kappa_2 \\ \hline \kappa_3 & \kappa_4 \end{array} \right], \quad (3.23)$$

with $K(\kappa_{\text{init}}) = 0$. Similarly to Section 3.7.1, the tunableSS function is used to get this parametrization. The five D-scale basis functions are chosen as in (3.16) using four poles. This choice yields fourth order D-scales.

The design gains and worst-case uncertainty samples during the iteration are listed in Table 3.2. The controller designed in the first iteration with the nominal sample $\delta_p = 0$ achieves the synthesis gain 0.1132 over the dynamic uncertainty. However, this controller does not provide closed-loop robust stability over the entire Δ . This analysis results in the new sample $\delta_p^* = 0.7127$. Consequently, the sample set is expanded to $\Delta_{p,s} = \{0 \cdot I_2, 0.7127 \cdot I_2\}$. The controller becomes robustly stabilizing during iteration 5. The algorithm terminates because γ_s is in the interval of the analysis bounds and due to the lack of improvement in the analysis gain. The resulting controller is

$$K(\kappa^*) = \left[\begin{array}{c|c} -2278.9495 & 100.3528 \\ \hline -2.7772 & -0.0832 \end{array} \right]. \quad (3.24)$$

This controller achieves a worst-case gain of 1.5793.

To demonstrate the necessity of D-scales that vary with the parametric uncertainty samples, we analyze $K(\kappa^*)$ using a single D-scale and two samples. Choose the sample set as $\Delta_{p,s} = \{0 \cdot I_2, I_2\}$, and compute the worst-case gain upper bound obtained with a single, fourth-order D-scale on these two samples. Formally this amounts to solving the optimization

$$\begin{aligned} & \min_{\substack{\gamma > 0 \\ D_d \in \mathbb{D}_d}} \gamma \\ & \text{subject to:} \end{aligned} \tag{3.25}$$

$$\left\| \begin{bmatrix} D_d & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{bmatrix} T(\Delta_p, \kappa^*) \begin{bmatrix} D_d^{-1} & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{bmatrix} \right\|_\infty \leq 1 \quad \forall \Delta_p \in \Delta_{p,s}$$

for a single D-scale. A suboptimal solution to this computation is obtained using systune. The resulting D-scale is

$$D_d = \frac{0.14s^4 + 15.40s^3 + 155.31s^2 + 157.04s + 13.51}{s^4 + 111.1s^3 + 1121.1s^2 + 1110.91s + 99.97}, \tag{3.26}$$

which yields a gain upper bound of 2.3573. This value exceeds the worst-case gain over the entire Δ . Changing the sample set to $\Delta_{p,s} = \{-I_2, I_2\}$ increases the gain upper bound to 118.2761. These results demonstrate the conservativeness of using a single D-scale for the control design on this example.

3.7.3 Further numerical testing

Further test results are presented to give the reader an overview of the expected performance of the algorithm. A collection of 31 examples are used to test the method. All of the examples are composed of an uncertain generalized plant $P(\Delta) = \mathcal{F}_U(N, \Delta)$ with corresponding uncertainty set Δ , and a controller with fixed state order $K(\kappa)$. The test results are given in Table 3.3. The columns of the table are:

- #: ordinal number of the example,
- $n_{x,N}$: state order of N ,
- $n_{x,K}$: fixed state order of the controller $K(\kappa)$,
- Δ_d : structure of the dynamic uncertainty block (\emptyset means no dynamic uncertainty),
- Δ_p : structure of the parametric uncertainty block (\emptyset means no parametric uncertainty),
- time [sec]: required time for the algorithm to complete (" - " means that the synthesis failed to stabilize the closed-loop), and
- gain: final worst-case gain of the closed-loop (" - " means that the synthesis failed to stabilize the closed-loop).

The first thirty examples are taken from [2], Example 12 and 31 are the systems detailed in Sections 3.7.1 and 3.7.2, respectively. The computations are performed on a computer that runs Ubuntu 16.04 LTS and features an eight-core 2.1 GHz Intel Xenon CPU with 20 GB RAM. The algorithm is run on MATLAB R2016b making use of the Parallel Computing Toolbox.

These results show that the method is applicable to a wide range of problems. It succeeds for 28 of the 31 test cases that include problems with both dynamic and parametric uncertainty. Running the algorithm can take from a couple of seconds (e.g., Examples 6

and 11) up to about an hour (Example 25) on our setup. Both the resulting gain and the synthesis time depends more on the numerical properties of the system than on the complexity of the problem. It is our observation that the final objective value is greatly influenced by the number of controller parameters. Choosing a parametrization with minimal number of parameters is advantageous even if it is numerically ill-conditioned, such as the 'companion' form in `tunableSS`.

The '-' symbols in the last two columns in Table 3.3 indicate that the design process failed to produce a robustly stabilizing controller. This can occur when the maximum number of iterations is reached and the controller is still unable to stabilize the closed-loop against the entire uncertainty domain. It is also possible that the original problem data is ill conditioned and the numerical errors accumulate to such an extent, that Algorithm 1 cannot continue after a certain step. This results in a failure if the controller from the previous step is not robustly stabilizing.

Thesis 1: I propose a method for structured robust control design for systems with a mixture of parametric and dynamic uncertainty. The algorithm minimizes the worst-case gain of the uncertain system by alternating between an analysis step and a synthesis step. Samples of the parametric uncertainty are computed during the analysis steps thus yielding an array of uncertain systems containing only dynamic uncertainty. The controller is then synthesized on this array of uncertain models. This synthesis step itself involves an alternation between constructing a D-scale for each of the uncertain systems and tuning a single controller for the entire collection of scaled plants. The controller tuning is performed using structured control design techniques.

Related publications: [P54, P46]

#	$n_{x,N}$	$n_{x,K}$	Δ_d	Δ_p	time [sec]	gain
1.	9	2	[1 × 1]	[∅]	73	1.02
2.	7	1	[3 × 3]	[∅]	42	1.78
3.	8	4	[3 × 3]	[∅]	41	3.46
4.	12	3	[8 × 8]	[∅]	35	2.44
5.	22	5	[1 × 1, 1 × 1]	[∅]	290	1.34
6.	3	1	[1 × 1]	[∅]	15	1.09
7.	26	4	[2 × 2]	[∅]	835	1.40
8.	3	2	[∅]	[1]	12	1.00
9.	23	2	[∅]	[2, 18]	447	3.77
10.	10	2	[∅]	[20]	152	0.78
11.	5	1	[∅]	[21]	5	45.55
12.	9	3	[1 × 1]	[1, 1]	-	-
13.	7	1	[1 × 1]	[2, 2]	35	1.00
14.	8	2	[1 × 1]	[3, 1]	406	1.00
15.	8	3	[1 × 1]	[1]	386	74.54
16.	14	4	[2 × 2]	[1, 1, 1, 1, 1, 1]	1147	1.60
17.	9	4	[1 × 1]	[1, 1, 1, 3, 1]	141	10.80
18.	6	3	[1 × 1]	[1, 2, 2]	299	5.78
19.	6	1	[1 × 1]	[1, 3, 3, 3, 1, 3]	-	-
20.	11	2	[1 × 1]	[1, 1, 3, 3, 3, 3, 3, 1, 1, 1]	-	-
21.	8	1	[4 × 4]	[1, 1, 1, 1]	44	2.13
22.	19	3	[1 × 1]	[1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1]	2335	0.07
23.	8	4	[3 × 3]	[1, 6, 1]	268	2.56
24.	7	4	[3 × 3]	[1]	92	7.15
25.	24	3	[1 × 1]	[1, 1, 1, 6, 6, 6]	3806	21.67
26.	8	1	[1 × 1, 1 × 1, 1 × 1, 1 × 1]	[1, 1, 1, 1]	59	4.17
27.	7	3	[1 × 1]	[1]	192	1.41
28.	7	4	[1 × 1]	[1, 5]	1202	1.01
29.	4	2	[1 × 1]	[1]	135	1.25
30.	8	5	[1 × 1]	[1]	583	1.36
31.	5	1	[1 × 1]	[2]	299	1.58

Table 3.3: Numerical results with the example system used for testing Algorithm 1.

4

Multi-frequency worst-case uncertainty construction

In most control design problems, it is useful to have an appropriate analysis method to verify the synthesis results. This chapter presents an algorithm that constructs a worst-case uncertainty sample that maximizes the gain of an uncertain system at multiple frequency points simultaneously. Such a sample can be used to complement Monte Carlo simulations when observing the behavior of the system in the time domain. The Nevanlinna-Pick interpolation is the basis of the approach with several modifications that make it applicable to the worst-case perturbation construction problem. The dynamic uncertainty sample is directly obtainable by employing this interpolation technique. The worst-case value of the parametric uncertainty is selected by the solution of a nonlinear optimization. For the calculation of the mixed uncertainty sample, these two results are combined. The algorithm is tested on a number of examples to assess its applicability on a wide variety of problems. The analysis of a real-life hard disk drive control system is also presented in which the closed-loop is susceptible to excitations at multiple distinct frequencies. The chapter is structured as follows. The related literature is described in Section 4.1, then the mathematical problem statement is given in Section 4.2. Section 4.3 describes the boundary Nevanlinna-Pick interpolation. The worst-case uncertainty construction is elaborated for the purely dynamic, purely parametric, and the mixed uncertainty cases in Sections 4.4, 4.5, and 4.6, respectively. The results for the example systems is presented in Section 4.7, and for the hard disk drive control system in Section 4.8.

4.1 Related literature

In current engineering practice, worst-case uncertainty samples are usually obtained by calculating the worst-case gain lower bound using the skewed- μ power iteration as described in Section 2.3. This yields a sample that maximizes the gain of the uncertain system at the frequency where the peak occurs. The method presented in this chapter can provide the same sample as a special case. However, this solution allows for more flexibility, as multiple frequencies can be incorporated into the sample search.

The most closely related work to this uncertainty calculation approach is [30] which considers sample construction for randomly sampling an uncertainty set. The Nevanlinna-Pick theorem is employed to sample unit H_∞ norm systems. The approach involves randomly chosen frequencies and data that are interpolated while constraining the resulting interpolant to be stable and norm bounded by one. The construction in [30] is similar to

the one presented in this thesis. The main difference is that random uncertainty samples are created in [30] while the present method aims to maximize gain at multiple frequencies.

4.2 Problems statement

Assume the following are given: a robustly stable[†] uncertain system of the form

$$P(\Delta, s) = \mathcal{F}_U(M(s), \Delta(s)) \quad (4.1)$$

with $\Delta(s) \in \Delta$, a collection of frequencies $\{\omega_k\}_{k=1}^{N_\omega}$ and worst-case gain lower bounds $\{L_k\}_{k=1}^{N_\omega}$. Define the objective function $J : \Delta \rightarrow \mathbb{R}$, such that, it is the sum of the gains at the given frequencies, i.e.,

$$J(\Delta(s)) := \sum_{k=1}^{N_\omega} \bar{\sigma}(\mathcal{F}_U(M(j\omega_k), \Delta(j\omega_k))) . \quad (4.2)$$

The worst-case construction problem is to find $\Delta_m(s) \in \Delta$ for which $J(\Delta_m(s))$ is maximal. The 'm' subscript stands for multiple peak.

The objective function J has a theoretical upper bound: the sum of the worst-case gain lower bounds at the given frequencies. More precisely, for any $\Delta(s) \in \Delta$, $J(\Delta(s)) \leq J_U$, where

$$J_U := \sum_{k=1}^{N_\omega} L_k . \quad (4.3)$$

If the system only has dynamic uncertainty ($\Delta = \Delta_d$) then it is possible to find a $\Delta_m(s)$, such that, $J(\Delta_m(s)) = J_U$. This follows from the boundary Nevanlinna-Pick (BNP) results in Section 4.4. Specifically, a single $\Delta_m(s) \in \Delta_d$ can be constructed to interpolate the complex matrices found by the power iteration at the given collection of frequencies. However, in the mixed case, it is generally not possible to find a $\Delta_m(s)$, such that, $J(\Delta_m(s)) = J_U$. The parametric uncertainty Δ_p couples the frequencies together and different values of the same parameter may be required to achieve the lower bound at each frequency.

To address these issues, we propose a search over the parametric uncertainty and an interpolation of the dynamic uncertainty. In this approach, a nonlinear optimization is performed that yields $\Delta_{p,m} \in \Delta_p$ and the samples of the complex uncertainty $\{\mathcal{Q}_{d,k}\}_{k=1}^{N_\omega}$ at frequencies $\{\omega_k\}_{k=1}^{N_\omega}$. The interpolant $\Delta_{d,m}(s) \in \Delta_d$ is obtained by the application of the BNP interpolation method.

4.3 Boundary Nevanlinna-Pick interpolation

The BNP interpolation method is detailed next. The algorithm is given in Example 21.3.1 and Corollary 21.4.2 in [10] and is repeated here for ease of reference. The $(\cdot)'$ superscript

[†]We remark that if the system is not robustly stable, then we can use the μ lower bound power iteration to obtain a destabilizing complex matrix sample of the uncertainty. The state-space realization of this sample is then the result of the single frequency interpolation in Section 2.3.2. This calculation is performed by the MATLAB function `robstab`.

denotes derivative with respect to the independent variable.

Theorem 4.1. Let $\{\vartheta_k\}_{k=1}^{N_\vartheta}$ be given, distinct numbers on the imaginary axis. In addition, let $\{u_k\}_{k=1}^{N_\vartheta} \subset \mathbb{C}^q$ and $\{v_k\}_{k=1}^{N_\vartheta} \subset \mathbb{C}^q$ be given unit-length vectors. The following statements hold for any such interpolation data.

1. There exist non-negative real numbers $\{\varrho_k\}_{k=1}^{N_\vartheta}$, such that, the boundary Pick matrix $H(\varrho) \in \mathbb{C}^{N_\vartheta \times N_\vartheta}$ defined by

$$H_{ik}(\varrho) := \begin{cases} \frac{v_i^* v_k - u_i^* u_k}{\vartheta_i^* + \vartheta_k}, & i \neq k \\ \varrho_k, & i = k, \end{cases} \quad (4.4)$$

satisfies $H(\varrho) \succ 0$.

2. Assume $H(\varrho) \succ 0$ and define

$$C_{0-} := [v_1 \ \dots \ v_{N_\vartheta}] \in \mathbb{C}^{q \times N_\vartheta}, \quad (4.5)$$

$$C_{0+} := [u_1 \ \dots \ u_{N_\vartheta}] \in \mathbb{C}^{q \times N_\vartheta}, \quad (4.6)$$

$$A_0 := \begin{bmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_{N_\vartheta} \end{bmatrix} \in \mathbb{C}^{N_\vartheta \times N_\vartheta}, \quad (4.7)$$

$$\Theta(s) := I + \begin{bmatrix} C_{0+} \\ C_{0-} \end{bmatrix} (sI - A_0)^{-1} H(\varrho)^{-1} \begin{bmatrix} -C_{0+} \\ C_{0-} \end{bmatrix}^*. \quad (4.8)$$

Let $G(s)$ be any $q \times q$ rational function analytic on the closed right-half plane with $\|G(s)\|_\infty \leq 1$. Define

$$\Delta_d(s) := (\Theta_{11}(s) G(s) + \Theta_{12}(s)) (\Theta_{21}(s) G(s) + \Theta_{22}(s))^{-1}, \quad (4.9)$$

where $\Theta(s) := \begin{bmatrix} \Theta_{11}(s) & \Theta_{12}(s) \\ \Theta_{21}(s) & \Theta_{22}(s) \end{bmatrix}$ is partitioned into $q \times q$ blocks. Then $\Delta_d(s)$ is analytic on the closed right-half plane and $\|\Delta_d(s)\|_\infty \leq 1$. Moreover, $\Delta_d(\vartheta_k) v_k = u_k$, $u_k^* \Delta_d(\vartheta_k) = v_k^*$, and $-u_k^* \Delta'_d(\vartheta_k) v_k = \varrho_k$ for $k = 1, \dots, N_\vartheta$.

Proof. See [10]. ◊

In the rest of this section, several modifications are added to the basic BNP interpolation result in Theorem 4.1 that make it applicable to worst-case dynamic uncertainty construction. First, the theorem assumes that the interpolation vectors are both of the same dimension. Zero-padding is used if the dimensions are not equal. For example, assume $\{u_k\}_{k=1}^{N_\vartheta} \subset \mathbb{C}^r$ and $\{v_k\}_{k=1}^{N_\vartheta} \subset \mathbb{C}^c$, with $r > c$. In this case, the vectors $\{v_k\}_{k=1}^{N_\vartheta}$ are augmented with $r - c$ zeros so that $\tilde{v}_k^* = [v_k^* \ 0]$. The resulting interpolant is $\tilde{\Delta}_d(s) = [\Delta_d(s) \ \Delta_0(s)]$ where $\tilde{\Delta}_d(s)$ satisfies the interpolation conditions with $\{u_k\}_{k=1}^{N_\vartheta}$ and $\{\tilde{v}_k\}_{k=1}^{N_\vartheta}$. Then $\Delta_d(s)$ satisfies the interpolation conditions with $\{u_k\}_{k=1}^{N_\vartheta}$ and $\{v_k\}_{k=1}^{N_\vartheta}$. Similarly, $\{u_k\}_{k=1}^{N_\vartheta}$ are augmented with zeros when $r < c$.

Next, the interpolant provided by Theorem 4.1 is not necessarily a real system, i.e., its coefficients in transfer function or state-space form can be complex. A system with real coefficients is obtained by interpolating the given data as well as its complex conjugate. Specifically, assume the given interpolation data is $\{u_k\}_{k=1}^{N_\omega}$, $\{v_k\}_{k=1}^{N_\omega}$, and a collection of non-negative frequencies $\{\omega_k\}_{k=1}^{N_\omega}$. The interpolant in Theorem 4.1 is constructed with

the data

$$\begin{aligned}\vartheta_k : & \quad -j\omega_1 \quad \dots \quad -j\omega_{N_\omega} \quad j\omega_1 \quad \dots \quad j\omega_{N_\omega}, \\ u_k : & \quad \bar{u}_1 \quad \dots \quad \bar{u}_{N_\omega} \quad u_1 \quad \dots \quad u_{N_\omega}, \\ v_k : & \quad \bar{v}_1 \quad \dots \quad \bar{v}_{N_\omega} \quad v_1 \quad \dots \quad v_{N_\omega}.\end{aligned}\tag{4.10}$$

If zero is among the frequency points then it is not duplicated. This makes $N_\vartheta = 2N_\omega$ or $N_\vartheta = 2N_\omega - 1$ if the zero frequency is included. Interpolating such conjugate data yields an interpolant $\Delta_d(s)$ with real coefficients. The following lemma proves this statement.

Lemma 4.1. The interpolation data $\{\omega_k\}_{k=1}^{N_\omega}$, $\{u_k\}_{k=1}^{N_\omega}$, and $\{v_k\}_{k=1}^{N_\omega}$ are given. If Theorem 4.1 is applied with the input in (4.10), then there exists a realization of the interpolant $\Delta_d(s)$ with real coefficients.

Proof. We provide a state-space transformation that yields $\Theta(s)$ with real coefficients. This implies the statement of the lemma. Let $\Theta(s) = C_0(sI - A_0)^{-1}B_0 + D_0$ where according to Theorem 4.1,

$$\begin{aligned}A_0 &= \text{diag } \{-j\omega_1, \dots, -j\omega_{N_\omega}, j\omega_1, \dots, j\omega_{N_\omega}\}, \\ B_0 &= H(\rho)^{-1} \begin{bmatrix} -C_{0+} \\ C_{0-} \end{bmatrix}^*, \\ C_0 &= \begin{bmatrix} C_{0+} \\ C_{0-} \end{bmatrix}, \\ D_0 &= I.\end{aligned}\tag{4.11}$$

Applying the state-space transformation

$$T = \frac{1}{2} \begin{bmatrix} I & I \\ jI & -jI \end{bmatrix}, \quad T^{-1} = \begin{bmatrix} I & -jI \\ I & jI \end{bmatrix}\tag{4.12}$$

yields $\Theta(s) = C_{\text{real}}(sI - A_{\text{real}})^{-1}B_{\text{real}} + D_{\text{real}}$ where A_{real} , B_{real} , C_{real} , and $D_{\text{real}} = D_0 = I$ are real valued matrices.

Because of the construction of A_0 , it can be written as

$$A_0 = \begin{bmatrix} -j\Omega & 0 \\ 0 & j\Omega \end{bmatrix}, \quad \text{where } \Omega = \begin{bmatrix} \omega_1 & & 0 \\ & \ddots & \\ 0 & & \omega_{N_\omega} \end{bmatrix}.\tag{4.13}$$

The transformation results in

$$A_{\text{real}} = TA_0T^{-1} = \begin{bmatrix} 0 & -\Omega \\ \Omega & 0 \end{bmatrix}.\tag{4.14}$$

Define

$$F_1 = \begin{bmatrix} u_1 & \dots & u_{N_\omega} \\ v_1 & \dots & v_{N_\omega} \end{bmatrix}.\tag{4.15}$$

Then, $C_0 = [\bar{F}_1 \ F_1]$ which makes

$$C_{\text{real}} = C_0T^{-1} = 2[\text{Re}(F_1) \ \text{Im}(F_1)].\tag{4.16}$$

To prove that $B_{\text{real}} = TB_0$ is a real valued matrix, we establish that because of the

definition of $H(\rho)$ in (4.4), it can be written as

$$H(\rho) = \begin{bmatrix} \bar{H}_0 & S \\ \bar{S} & H_0 \end{bmatrix}, \quad (4.17)$$

where $\bar{S} = S^*$, and $H_0 = H_0^*$. (H_0 is the boundary Pick matrix corresponding to the interpolation data without the conjugation.) Using both expansions of the block matrix inverse lemma, we obtain that the inverse is of the form

$$H(\rho)^{-1} = \begin{bmatrix} \bar{X} & Y \\ \bar{Y} & X \end{bmatrix}, \quad (4.18)$$

where $\bar{Y} = Y^*$ and $X^* = X$. Defining

$$F_2 = \begin{bmatrix} -u_1 & \dots & -u_{N_\omega} \\ v_1 & \dots & v_{N_\omega} \end{bmatrix}, \quad (4.19)$$

we can write

$$\begin{bmatrix} -C_{0+} \\ C_{0-} \end{bmatrix} = [\bar{F}_2 \ F_2]. \quad (4.20)$$

Also,

$$\begin{aligned} B_0^* &= \begin{bmatrix} -C_{0+} \\ C_{0-} \end{bmatrix} H(\rho)^{-*} = [\bar{F}_2 \ F_2] H(\rho)^{-1} \\ &= [\bar{F}_2 \bar{X} + F_2 \bar{Y} \ F_2 X + \bar{F}_2 Y] \end{aligned} \quad (4.21)$$

which has the structure $[\bar{F}_3 \ F_3]$ with $F_3 = F_2 X + \bar{F}_2 Y$. Because $T^* = \frac{1}{2}T^{-1}$, $B_{\text{real}} = \frac{1}{2}(B_0^* T^{-1})^*$ and since $B_0^* T^{-1}$ has the same structure as C_{real} in (4.16),

$$B_{\text{real}} = \begin{bmatrix} \text{Re}(F_3^*) \\ \text{Im}(F_3^*) \end{bmatrix}. \quad (4.22)$$

If zero is among the frequency points, delete the first row and column of A_{real} and D_{real} , the first row of B_{real} , and the first column of C_{real} . With that, the elements corresponding to the data duplicated by the conjugation are removed. \diamond

We also remark that a minimal realization of the interpolant is always obtainable by Kalman decomposition. Since this transformation does not affect the frequency response of the interpolant, the interpolation criteria (given in the frequency domain) are still met by the minimal interpolant.

Finally, Theorem 4.1 provides a set of interpolants depending on the choices for the non-negative numbers $\{\varrho_k\}_{k=1}^{N_\vartheta}$ and rational function $G(s)$. An all-pass $\Delta_d(s)$ is obtained if $G(s)$ is chosen to be all-pass [16]. For simplicity, we always pick $G(s) = I$. In addition, the interpolant satisfies $-u_k^* \Delta'_d(\vartheta_k) v_k = \varrho_k$ and hence smaller values of $\{\varrho_k\}_{k=1}^{N_\vartheta}$ are related to smaller derivatives of the interpolant. Specifically, if $\Delta_d(s)$ is SISO, smaller ϱ_k values mean that the phase of $\Delta_d(s)$ varies more gradually with frequency (see Example 4.1). Because of the construction of the data in (4.10), there are only N_ω independent variables in $\{\varrho_k\}_{k=1}^{N_\vartheta}$ since for negative frequencies,

$$\begin{aligned} -\bar{u}_k^* \Delta'_d(-j\omega_k) \bar{v}_k &= -\bar{u}_k^* \overline{\Delta'_d(j\omega_k)} \bar{v}_k = -\overline{u_k^* \Delta'_d(j\omega_k) v_k} \\ &= -u_k^* \Delta'_d(j\omega_k) v_k = \varrho_k. \end{aligned} \quad (4.23)$$

The following optimization is formulated to minimize ϱ_k . Let

$$R := \begin{bmatrix} \varrho_1 & & 0 \\ & \ddots & \\ 0 & & \varrho_{N_\omega} \end{bmatrix} \quad (4.24)$$

so that $\begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix}$ is the main diagonal of the boundary Pick matrix $H(\varrho)$ defined in (4.4). Define

$$H_0 := H(\varrho) - \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \quad (4.25)$$

and solve the optimization

$$\min_{R, \hat{\varrho}} \hat{\varrho} + \text{trace}(R)$$

subject to:

$$\hat{\varrho}I \geq R \geq 0, \quad (4.26)$$

$$\kappa I \succeq H_0 + \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \succeq \frac{1}{\kappa} I.$$

to obtain $\{\varrho_k\}_{k=1}^{N_\omega}$. The objective function $\hat{\varrho} + \text{trace}(R)$ combines a bound on the peak value and the sum of $\{\varrho_k\}_{k=1}^{N_\omega}$. The two terms are weighted equally in this formulation but alternative weightings could be used. The optimization also includes an upper bound constraint on the condition number κ of the boundary Pick matrix $H(\varrho)$. This constraint improves the conditioning of the matrix inversion $H(\varrho)^{-1}$ that appears in (4.8). The condition number bound is selected as $\kappa = 10^4$. The following example illustrates the effect of this optimization on the phase of the interpolant.

Example 4.1 (the effect of the derivatives). Interpolate the complex numbers $e^{-j\frac{\pi}{4}}$ and $e^{-j\frac{5\pi}{2}}$ at the frequencies 1 rad/s and 100 rad/s, respectively. When using the optimization in (4.26), the derivatives become 0.72 and 0.02, respectively. Perform the interpolation two more times but instead of minimizing the derivatives, set them both to 1, then 10. Because of the BNP construction, all three transfer function satisfy the interpolation criteria and their magnitude is one at all frequencies. The only difference between them is their phase, except for 1 rad/s and 100 rad/s where it is also identical. The phase of these three transfer functions are depicted in Figure 4.1. The figure clearly illustrates the effect of the derivatives on the interpolant.

4.4 Interpolation of the dynamic uncertainty

Using the method detailed in Section 4.3, the worst-case dynamic uncertainty construction is described next. The following are given: a robustly stable uncertain system $\mathcal{F}_U(M(s), \Delta(s))$ along with a block structure $\Delta(s) \in \mathbb{A}_d$, distinct frequencies $\{\omega_k\}_{k=1}^{N_\omega}$, and corresponding worst-case gain lower bounds $\{L_k\}_{k=1}^{N_\omega}$ at those frequencies. We want to find a $\Delta_{d,m}(s) \in \mathbb{A}_d$, such that,

$$\bar{\sigma}(\mathcal{F}_U(M(j\omega_k), \Delta_{d,m}(j\omega_k))) = L_k \quad (4.27)$$

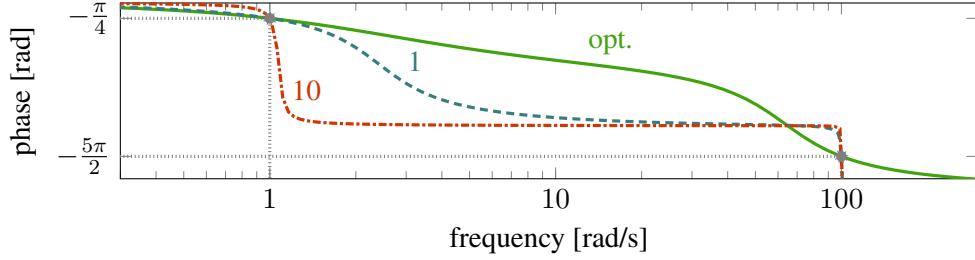


Figure 4.1: The effect of the values of the derivatives when using the BNP interpolation.

for $k = 1, \dots, N_\omega$. To achieve this, we first compute the complex uncertainty samples at $\{\omega_k\}_{k=1}^{N_\omega}$ and use the BNP interpolation for each diagonal block of Δ_d to obtain the interpolant $\Delta_{d,m}(s)$. The method is summarized in Algorithm 2.

The worst-case uncertainty samples $\{\mathcal{Q}_{d,k}\}_{k=1}^{N_\omega} \subset \mathbb{Q}_d$ are complex matrices computed on the frequency grid $\{\omega_k\}_{k=1}^{N_\omega}$ using the existing worst-case gain lower bound power iteration in Section 2.3.2. The uncertainty $\Delta_{d,m}(s) \in \Delta_d$ is the result of interpolation between these matrices. The uncertainty samples have block diagonal structure, i.e.,

$$\mathcal{Q}_{d,k} = \begin{bmatrix} Q_{1,k} & & 0 \\ & \ddots & \\ 0 & & Q_{N_b,k} \end{bmatrix} \in \mathbb{Q}_d \quad (4.28)$$

has block-diagonal structure. Thus, the interpolation procedure in Theorem 4.1 and Section 4.3 is repeated for each block separately. For the block Q_k of the sample $\mathcal{Q}_{d,k}$, compute rank-one decomposition at each frequency so that $Q_k = u_k v_k^*$, $k = 1, \dots, N_\omega$. If Q_k is not square, add zeros at the end of u_k or v_k , whichever has fewer elements, so that they have the same size. Form the input of the BNP procedure as in (4.10). Apply the BNP interpolation as described in Sections 4.3. This procedure interpolates through the rank-one blocks of the uncertainty sample. Every block of the resulting $\Delta_d(s)$ will interpolate the $\{u_k\}_{k=1}^{N_\vartheta}$ and $\{v_k\}_{k=1}^{N_\vartheta}$ corresponding to that block but they can, in general, be full rank matrices at the interpolation frequencies.

In Theorem 4.1, $\Delta_d(s)$ interpolates the vectors u_k and v_k in the sense that $\Delta_d(j\omega_k) v_k = u_k$ for $k = 1, \dots, N_\omega$. At ω_k , the lower bound power iteration yields $\mathcal{Q}_{d,k}$ for which $\bar{\sigma}(\mathcal{F}_U(M(j\omega_k), \mathcal{Q}_{d,k})) = L_k$. That means that there exist unit vectors $u_{p,k}$ and $v_{p,k}$, such that, $\mathcal{F}_U(M(j\omega_k), \mathcal{Q}_{d,k}) u_{p,k} = L_k v_{p,k}$. (The 'p' subscript refers to the word performance, since in skewed- μ analysis, e and d are usually called performance channels and the fictitious uncertainty block associated with them is called the performance block.) The interpolation means that $\Delta_{d,m}(j\omega_k) v_k = u_k$. This is sufficient since the uncertainty satisfies the equations

$$\begin{aligned} M(j\omega_k) \begin{bmatrix} u_k \\ u_{p,k} \end{bmatrix} &= \begin{bmatrix} v_k \\ L_k v_{p,k} \end{bmatrix}, \\ \Delta_{d,m}(j\omega_k) v_k &= u_k \end{aligned} \quad (4.29)$$

which implies (4.27). This applies when $\Delta(s)$ is a full block uncertainty. The generalization to the case when $\Delta(s)$ has a block structure is straightforward but is omitted for

Algorithm 2 Worst-case dynamic uncertainty construction.

Given: Uncertain system $\mathcal{F}_U(M(s), \Delta_d(s))$ robustly stable for all $\Delta_d(s) \in \Delta_d$ and distinct frequencies $\{\omega_k\}_{k=1}^{N_\omega}$.

Compute worst-case uncertainty samples: Use skewed- μ lower bound power iteration (wcgain) to obtain $\{\mathcal{Q}_{d,k}\}_{k=1}^{N_\omega} \subset \mathbb{Q}_d$.

for each diagonal block $i = 1, \dots, N_d$ of the structured uncertainty
do

Rank-one decomposition: Decompose the i^{th} block of each $\mathcal{Q}_{d,k}$ as $u_k v_k^*$. If the block is non-square then augment u_k or v_k with zeros.

Interpolation: Interpolate the conjugate data

$$\begin{aligned} -j\omega_{N_\omega}, & \quad \dots, \quad -j\omega_1, \quad j\omega_1, \quad \dots, \quad j\omega_{N_\omega}, \\ \bar{u}_1, & \quad \dots, \quad \bar{u}_{N_\omega}, \quad u_1, \quad \dots, \quad u_{N_\omega}, \\ \bar{v}_1, & \quad \dots, \quad \bar{v}_{N_\omega}, \quad v_1, \quad \dots, \quad v_{N_\omega}. \end{aligned}$$

Do not duplicate 0 if it is among the frequency points. Solve the optimization in (4.26) to obtain $\{\varrho_k\}_{k=1}^{N_\omega}$ and then construct $\Delta_{d,i}(s)$ using BNP interpolation. Remove appropriate rows or columns if the i^{th} block is non-square.

Concatenation: Append the i^{th} block:

$$\Delta_d(s) := \begin{bmatrix} \Delta_d(s) & 0 \\ 0 & \Delta_{d,i}(s) \end{bmatrix}$$

end for

notational simplicity.

Note that because of the definition of A_0 in (4.7) and our choice of $G(s) = I$, the state order of the interpolant in Theorem 4.1 is N_ϑ . Because the interpolation is repeated for every block, the number of states in $\Delta_{d,m}(s)$ is $N_\vartheta N_d = 2N_\omega N_d$ (or $(2N_\omega - 1) N_d$). Thus, if $N_\omega = 1$ this method generally provides a lower dimensional uncertainty sample than the classical solution in Section 2.3.2 which is also the current interpolation used in `wcgain` and `robstab` in MATLAB.

4.5 Optimization of the parametric uncertainty

Before we study the construction of the mixed uncertainty sample, let us consider the case when there is only parametric uncertainty in the system, i.e., $\Delta = \Delta_p$. In this case, the objective function in (4.2) simplifies to

$$J(\Delta_p) = \sum_{k=1}^{N_\omega} \bar{\sigma}(\mathcal{F}_U(M(j\omega_k), \Delta_p)) \quad (4.30)$$

and the goal is to find

$$\Delta_{p,m} = \arg \max_{\Delta_p \in \Delta_p} J(\Delta_p). \quad (4.31)$$

Since the parametric uncertainty couples the frequencies together, it is not possible to maximize J in the given frequency points independently. Therefore, there is no guarantee that a $\Delta_p \in \Delta_p$ exists, such that,

$$J(\Delta_p) = J_U = \sum_{k=1}^{N_\omega} L_k. \quad (4.32)$$

Even if such a Δ_p does exist, J is a nonlinear and non-convex function therefore there is no guarantee that it can be found.

To perform the optimization in (4.36), a multi-dimensional version of the interval search is performed combined with a gradient ascent algorithm. We take advantage of the fact that the independent variables of Δ_p form an N_p -dimensional hypercube. First, J is evaluated at the center and at the corners of the hypercube. Then, the hypercube is split into 2^{N_p} smaller cubes that all have the center and one of the corners as their respective corners. Finally, the small hypercube is selected that contains the two points with the highest objective value. This process is repeated until convergence or until the maximum number of objective function evaluations is reached. A gradient ascent search implemented in the MATLAB function `fmincon` with the 'interior-point' solver is then run starting from the resulting point.

An alternative to this cube splitting search could be to simply run a nonlinear gradient ascent algorithm starting from the center of the hypercube. We use 37 example systems to test our method and compare it to potential alternatives (See Section 4.7 for more information). Based on our test results with the 30 examples that have uncertain parameters, the cube splitting algorithm combined with a gradient ascent performs better than a simple nonlinear optimization. The advantage of our solution arises from the fact that the $\Delta_{p,m}$ is often close to one of the corners (or to the border) of Δ_p . On the other hand, the number of corner points of each hypercube grows exponentially with N_p . Due to

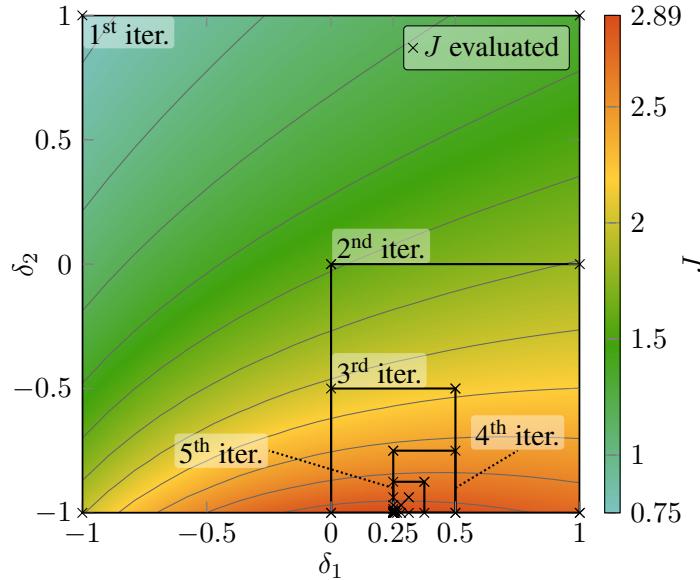


Figure 4.2: The value of the objective function and the hypercubes of the consecutive iterations during the optimization in Example 4.2.

constraints on computation time, this limits the accuracy of the result for problems with a high number of uncertain parameters. The following example illustrates this solution.

Example 4.2. Consider the second order system with resonance frequency ω_r and damping ξ :

$$P(\Delta, s) = \frac{\omega_r^2}{s^2 + 2\xi\omega_r s + \omega_r^2}. \quad (4.33)$$

The parameters of this system are uncertain, such that, $\omega_r = 1 + 0.3\delta_1$ and $\xi = \frac{1}{\sqrt{2}}(1 + 0.5\delta_2)$, where $\delta_1, \delta_2 \in \mathbb{R}$, $|\delta_1| \leq 1$, and $|\delta_2| \leq 1$. The uncertainty set corresponding to $P(\Delta, s)$ is

$$\Delta = \Delta_p = \left\{ \begin{bmatrix} \delta_1 I_2 & 0 \\ 0 & \delta_2 \end{bmatrix} : \delta_1, \delta_2 \in \mathbb{R}, |\delta_1| \leq 1, |\delta_2| \leq 1 \right\}. \quad (4.34)$$

Let us maximize the gain of $P(\Delta, s)$ at $\omega_1 = 0.75$ rad/s and $\omega_2 = 1$ rad/s simultaneously. Figure 4.2 shows a contour plot of the objective function. The figure also shows the points where the objective function was evaluated (black-x) during the hypercube refinement. The optimization yields

$$\Delta_{p,m} = \begin{bmatrix} 0.25I_2 & 0 \\ 0 & -1 \end{bmatrix} \quad (4.35)$$

that is the maximum on Δ_p with $J(\Delta_{p,m}) = 2.89$. The upper bound of the objective function is $J_U = 3.02$ which is clearly unattainable for this system.

Further possible replacements for the cube splitting algorithm include simulated annealing, pattern search, and the Nelder-Mead simplex method. These optimization techniques scale more preferably with the number of uncertain parameters (N_p) and could

potentially exploit the fact that the optimum is often on the boundary of the hypercube. Their inclusion will be the subject of future investigation.

4.6 Construction of the worst-case mixed uncertainty

If the system has dynamic uncertainty only, the interpolation method in Section 4.4 yields an uncertainty sample that ensures that the worst-case lower bound is reached at the given frequencies. As demonstrated in Section 4.5, this is not necessarily possible for the mixed uncertainty case. In the following a method is presented that combines the results of Sections 4.4 and 4.5. To maximize the objective function J in (4.2), consider the function $\tilde{J} : \Delta_p \rightarrow \mathbb{R}$, such that,

$$\tilde{J}(\Delta_p) = \max_{\mathcal{Q}_{d,k} \in \mathbb{Q}_d} \sum_{k=1}^{N_\omega} \bar{\sigma}\left(\mathcal{F}_U\left(M(j\omega_k), \begin{bmatrix} \Delta_p & 0 \\ 0 & \mathcal{Q}_{d,k} \end{bmatrix}\right)\right). \quad (4.36)$$

To evaluate \tilde{J} , substitute Δ_p into $P(\Delta, s)$ and perform the lower bound power iteration on Δ_d at the given frequencies $\{\omega_k\}_{k=1}^{N_\omega}$. This yields the complex uncertainty samples $\{\mathcal{Q}_{d,k}\}_{k=1}^{N_\omega}$ that maximize the largest singular values at the given frequencies. The interpolation method in Section 4.4 ensures that if $\Delta_{d,m}(s)$ interpolates $\{\mathcal{Q}_{d,k}\}_{k=1}^{N_\omega}$, then

$$\tilde{J}(\Delta_{p,m}) = J\left(\begin{bmatrix} \Delta_{p,m} & 0 \\ 0 & \Delta_{d,m}(s) \end{bmatrix}\right). \quad (4.37)$$

Therefore,

$$\Delta_{p,m} = \arg \max_{\Delta_p \in \Delta_p} \tilde{J}(\Delta_p). \quad (4.38)$$

To find the maximum of \tilde{J} , the optimization in Section 4.5 is employed. Aside from $\Delta_{p,m}$, this optimization also yields the samples of the complex uncertainty $\{\mathcal{Q}_{d,k}\}_{k=1}^{N_\omega}$. These samples are interpolated using Algorithm 2 to obtain $\Delta_{d,m}(s)$. The worst-case uncertainty is

$$\Delta_m(s) = \begin{bmatrix} \Delta_{p,m} & 0 \\ 0 & \Delta_{d,m}(s) \end{bmatrix}. \quad (4.39)$$

This method is summarized in Algorithm 3. Finally, the demonstrative example in Section 2.3.2 is continued.

Example 4.3 (Example 2.5 on page 22 continued). The worst-case gain lower bound in Figure 2.8 has two peaks at $\omega_1 = 1 \text{ rad/s}$ and $\omega_2 = 100 \text{ rad/s}$. Algorithm 3 is used to find the worst-case uncertainty $\Delta_m(s)$ that maximizes the gain of $P(\Delta, s)$ at ω_1 and ω_2 simultaneously. (Recall that $\Delta_s(s)$ only maximizes the gain at ω_1 in Example 2.5.) The value of the objective function $J(\delta_1 I_2)$ as a function of the uncertain parameter δ_1 is depicted in Figure 4.3. The maximum of $\tilde{J}(\delta_1 I_2)$ occurs at $\delta_1 = -1$ which makes $\Delta_{p,m} = -I_2$. This is different from $\Delta_{p,s} = I_2$ in Example 2.5. The skewed- μ power iteration yields $\mathcal{Q}_{d,1} = 0.998 - 0.07j$ at ω_1 and $\mathcal{Q}_{d,2} = 0.998 - 0.069j$ at ω_2 . Applying the boundary Nevanlinna-Pick interpolation in Section 4.4, we get

$$\Delta_{d,m}(s) = \frac{(s - 1.78 \cdot 10^4)(s - 1406)(s^2 - 4.15s + 121.2)}{(s + 1.78 \cdot 10^4)(s + 1406)(s^2 + 4.15s + 121.2)}. \quad (4.40)$$

Algorithm 3 Worst-case mixed uncertainty construction.

Given: Uncertain system $\mathcal{F}_U(M(s), \Delta(s))$ robustly stable for all $\Delta(s) \in \Delta$, and distinct frequencies $\{\omega_k\}_{k=1}^{N_\omega}$.

Compute worst-case uncertainty samples: Use nonlinear optimization to find $\Delta_{p,m} \in \Delta_p$ and $\{\mathcal{Q}_{d,k}\}_{k=1}^{N_\omega} \subset \mathbb{Q}_d$ for which $\tilde{J}(\Delta_{p,m})$ is maximal.

Interpolate the samples of the complex uncertainty: Use Algorithm 2 to obtain $\Delta_{d,m}(s) \in \Delta_d$ that interpolates $\{\mathcal{Q}_{d,k}\}_{k=1}^{N_\omega}$.

Concatenate the uncertainty:

$$\Delta_m(s) := \begin{bmatrix} \Delta_{p,m} & 0 \\ 0 & \Delta_{d,m}(s) \end{bmatrix}.$$

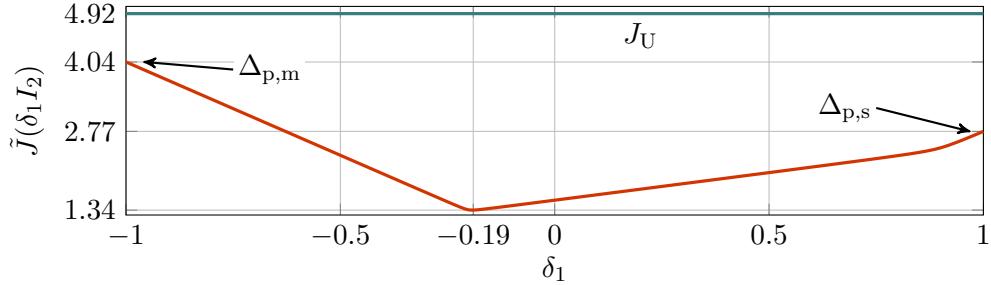


Figure 4.3: The value of the objective function as a function of the uncertain parameter in Example 4.3.

As shown in Figure 4.4, the gain of $P(\Delta_m, s)$ is large at both ω_1 and ω_2 . The worst-case gain lower bounds are $L_1 = 2.64$, and $L_2 = 2.28$ which makes the upper bound of the objective function $J_U = 4.92$. As depicted in Figure 4.3, the objective function does not reach this theoretical upper bound for any $\Delta_p \in \Delta_p$. The gain of $P(\Delta_m, s)$ attains L_2 at ω_2 but only at the cost of some drop at ω_1 .

4.7 Basic Validation of the Method

We use 37 example systems to test our method and compare it to potential alternatives. The first 31 are the closed-loop systems obtained while testing the synthesis algorithm (see Section 3.7). The rest are composed of randomly generated systems and physical systems we encountered in our research. The detailed results of these tests are listed in Table 4.1. The columns of the table are to be interpreted as:

- #: ordinal number of the example,
- n_x : state order of M ,
- N_ω : number of frequency points at which the gain is maximized,
- Δ_d : structure of the dynamic uncertainty block (\emptyset means no dynamic uncertainty),

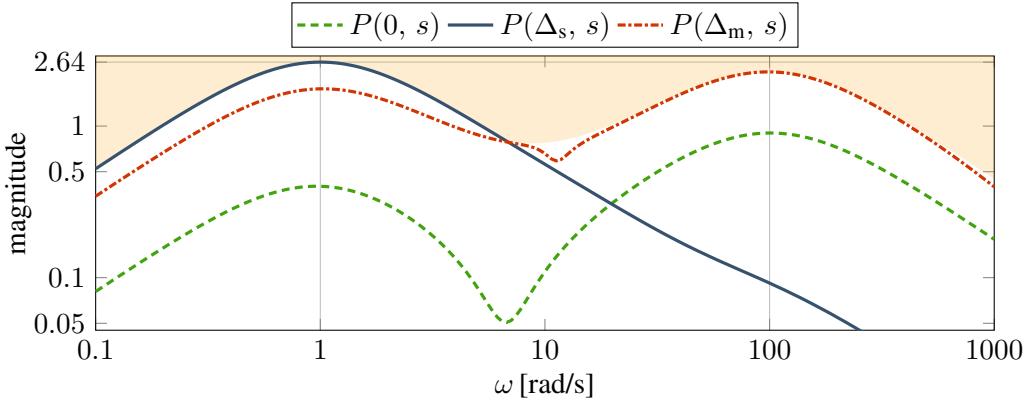


Figure 4.4: The gain of the uncertain system $P(\Delta, s)$ in Example 4.3 for different values of the uncertainty. The boundary of the shaded area is $L(\omega)$.

- Δ_p : structure of the parametric uncertainty block (\emptyset means no parametric uncertainty),
- time [sec]: required time for the algorithm to complete ("-" means that the algorithm did not yield any result),
- J/J_U : proportion of the achieved objective value and the theoretical upper bound for
 - c. s. [%]: the cube splitting search, and
 - nl. o. [%]: the nonlinear optimization.

The computations are performed on a computer that runs Ubuntu 18.04 LTS and features an four-core 2.6 GHz Intel® Core™ i5 processor with 8 GB memory. The algorithm is run on MATLAB R2016b. For our examples, the computation takes no more than two minutes, and approximately 20 seconds on average.

Various components of the algorithm were chosen based on the test results with these examples, e.g., the weighting in (4.26). Also, the cube splitting search described in Section 4.5 is compared to a gradient descent optimization in Table 4.1. The highlighted rows indicate that out of the 37 examples, the cube splitting algorithm yields better results in 7 cases and worse results in a single case. For the other examples, the performance of the two methods is identical.

To further investigate the method, we constructed an example which is designed to make the cube splitting algorithm find a local peak instead of the global maximum.

Example 4.4. Consider the fourth order Taylor polynomial of the function $\frac{\sin x}{x}$,

$$p(x) = 1 - \frac{x^2}{6} + \frac{x^4}{120}. \quad (4.41)$$

Shift and scale $p(x)$ to $p(2.9[x - 0.5])$ so that its maximum is moved to 0.5 and its peak is steeper. Construct the uncertain system with two uncertain parameters δ_1 and δ_2 of the form

$$G(\Delta, s) = \frac{1}{s+1} p(2.9[\delta_1 - 0.5]) p(2.9[\delta_2 - 0.5]). \quad (4.42)$$

#	n_x	N_ω	Δ_d	Δ_p	time [sec]	J/J_U c. s. [%]	nl. o. [%]
1.	11	7	[1×1]	[∅]	2	100.0	100.0
2.	8	5	[3×3]	[∅]	2	100.0	100.0
3.	12	7	[3×3]	[∅]	1	100.0	100.0
4.	15	4	[8×8]	[∅]	1	100.0	100.0
5.	27	7	[1×1, 1×1]	[∅]	2	100.0	100.0
6.	4	5	[1×1]	[∅]	1	100.0	100.0
7.	30	9	[2×2]	[∅]	2	100.0	100.0
8.	5	6	[∅]	[1]	1	83.4	83.4
9.	25	9	[∅]	[2, 18]	32	77.8	47.8
10.	12	6	[∅]	[20]	5	80.7	80.4
11.	6	4	[∅]	[21]	6	100.0	100.0
12.	12	7	[1×1]	[1, 1]	15	93.7	93.7
13.	8	6	[1×1]	[2, 2]	9	97.1	97.1
14.	10	4	[1×1]	[3, 1]	40	100.0	100.0
15.	11	7	[1×1]	[1]	27	100.0	69.7
16.	18	8	[2×2]	[1, 1, 1, 1, 1, 1]	41	67.9	44.3
17.	13	6	[1×1]	[1, 1, 1, 3, 1]	60	83.8	83.9
18.	9	6	[1×1]	[1, 2, 2]	14	68.9	68.9
19.	7	5	[1×1]	[1, 3, 3, 3, 1, 3]	16	100.0	100.0
20.	11	1	[∅]	[1, 1, 3, 3, 3, 3, 3, 1, 1, 1]	-	-	-
21.	9	6	[4×4]	[1, 1, 1, 1]	30	100.0	99.9
22.	22	5	[1×1]	[1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1]	202	94.2	94.2
23.	8	1	[3×3]	[1, 6, 1]	-	-	-
24.	11	5	[3×3]	[1]	9	100.0	93.2
25.	24	4	[1×1]	[1, 1, 1, 6, 6, 6]	34	100.0	100.0
26.	9	4	[1×1, 1×1, 1×1, 1×1, 1×1]	[1, 1, 1, 1]	16	99.8	99.8
27.	10	7	[1×1]	[1]	12	79.0	79.0
28.	11	8	[1×1]	[1, 5]	11	100.0	100.0
29.	6	7	[1×1]	[1]	10	71.1	71.1
30.	13	8	[1×1]	[1]	14	81.4	81.4
31.	6	7	[1×1]	[2]	40	100.0	3.9
32.	15	2	[1×1, 4×2, 2×2, 3×5]	[∅]	2	100.0	100.0
33.	15	8	[1×1, 4×2, 2×2, 3×5]	[∅]	3	100.0	100.0
34.	15	8	[1×1, 4×2, 2×2, 3×5]	[∅]	3	100.0	100.0
35.	20	2	[∅]	[10, 20]	24	100.0	100.0
36.	57	3	[2×2]	[46, 34, 7]	49	99.9	99.9
37.	4	2	[1×1]	[1]	6	52.5	52.5

Table 4.1: Numerical results with the example systems used for testing Algorithm 3. The highlighted rows indicate where the cube splitting algorithm performs better (green) or worse (orange) than the nonlinear optimization, and also the cases for which the algorithms fails to yield results (red). (c. s.: cube splitting, nl. o.: nonlinear optimization)

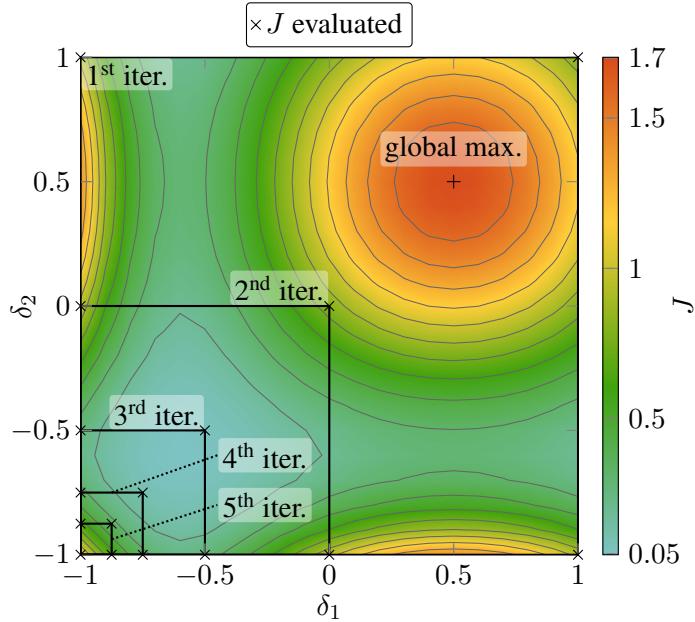


Figure 4.5: Contour plot of the objective function and the hypercubes of the iterations in Example 4.4.

Maximize the gain of this system at $\omega_1 = 0.1 \text{ rad/s}$ and $\omega_2 = 1 \text{ rad/s}$, simultaneously. The contour plot of the resulting objective function is depicted in Figure 4.5. The objective function that has four local maxima in the following points:

J	1.70	1.68	1.68	1.17
δ_1	0.5	-1	0.5	-1
δ_2	0.5	0.5	-1	-1

Notice that the first three maxima are away from the corners and the function rises steeply around them, therefore the cube splitting algorithm misses them and converges to the point $(-1, -1)$. The hypercubes of the consecutive iterations are also depicted in Figure 4.5. In the second phase of the optimization, a gradient ascent algorithm is run from the point the cube splitting converged to. The gradient ascent successfully finds the global maximum hence the worst case uncertainty really corresponds to the maximum of J . Although a gradient ascent starting from the origin would find the global maximum in fewer steps for this example, our method also ends up with the correct solution.

4.8 Analysis of a hard-disk drive control system

In this section, a hard disk drive control system is analyzed. A hard disk drive has several high frequency vibrational modes therefore it is sensitive to disturbance at certain resonance frequencies [25]. The effect of this phenomenon to the closed-loop performance is investigated. Most hard disk drives currently in production have dual-stage actuators. For simplicity, we will focus on the analysis of a single stage disk drive with a voice coil motor as an actuator.

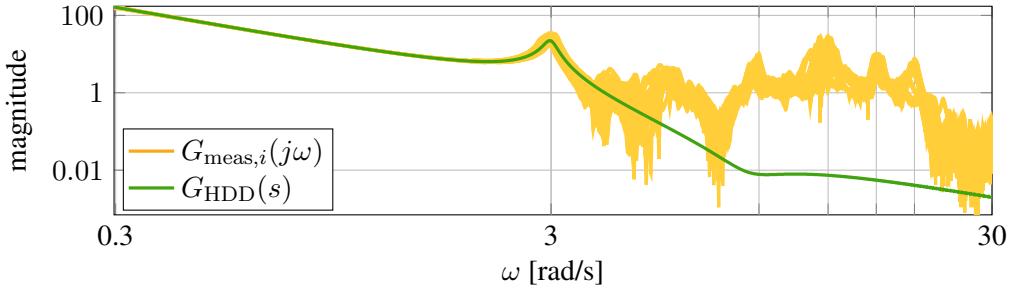


Figure 4.6: Measured frequency responses of the hard disk drive and the gain of the nominal (control oriented) model. ($i = 1, \dots, 50$)

4.8.1 The Control Problem and Motivation

The actuating signal of the hard disk drive is the input voltage of the voice coil motor actuator. The measured output is the position error signal of the head along the circular tracks of the drive. The system is modeled using the measured frequency response of a collection of 50 hard disk drives. The data was provided by Seagate with the frequency and gain scaled for proprietary reasons. (For simplicity, we will refer to every quantity as if they were not scaled in the rest of this section.) The magnitude of the measured responses $G_{\text{meas},i}(j\omega)$ where $i = 1, \dots, 50$ are depicted in Figure 4.6. A control oriented model is calculated by fitting the average of the data with a focus on the low frequency range which results in

$$G_{\text{HDD}}(s) = \frac{1.92(s^2 - 2.38s + 71.51)}{s^2(s^2 + 0.2s + 8.96)}. \quad (4.43)$$

The magnitude of $G_{\text{HDD}}(s)$ is also depicted in Figure 4.6. The fitting only aims to capture the dynamics at low frequencies including the first peak, because only this domain is significant from the point of view of the control design. Based on the measured data in Figure 4.6, high frequency dynamics are highly uncertain and the aim of the control design is to ensure that they are not excited.

A lead controller is designed of the form

$$K(s) = \frac{1}{|G_{\text{HDD}}(j\omega_c)|} \cdot \frac{\beta s + \omega_c}{s + \beta\omega_c} \quad (4.44)$$

with $\omega_c = 0.45$ rad/s, $\beta = 2$, and $|G_{\text{HDD}}(j\omega_c)| = 77.08$. The controller puts the cutoff frequency of the open-loop at ω_c . This is sufficiently low to balance tracking performance with the excitation of uncertain high frequency dynamics. The controller also provides 35° phase margin.

Figure 4.6 highlights frequencies where the measured data $G_{\text{meas},i}(j\omega)$, $i = 1, \dots, 50$ display significant peaks. These modes are the result of the complicated elastic behavior of the actuator. The goal of the worst case analysis is to investigate the effect of the sensitivity of the closed-loop system to these frequencies. For further details about the hard disk drive modeling and control, see [25, 3, 1].

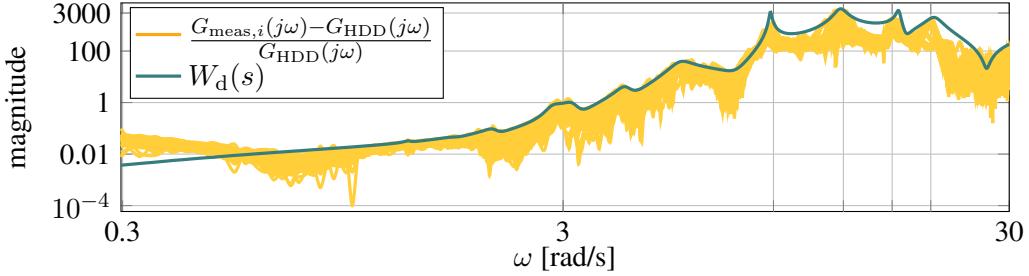


Figure 4.7: Weight of the dynamic uncertainty in the hard disk drive model. ($i = 1, \dots, 50$)

4.8.2 Uncertain Model of the Closed-Loop

Two factors are considered when constructing the uncertain model of the hard disk drive: perturbation in the rotational inertia of the actuator, and high frequency dynamics neglected by the design model $G_{\text{HDD}}(s)$. At low frequencies, the hard disk drive is essentially described by the rigid body rotational dynamics $\frac{1}{\theta s^2}$ where θ is the rotational inertia. Parametric uncertainty is introduced to model 10% variation in the denominator coefficient θ . Taking these into account, the uncertain system is of the form

$$G(\Delta, s) = \frac{1}{1 + 0.1\delta_\theta} G_{\text{HDD}}(s) [1 + W_d(s) \Delta_d(s)]. \quad (4.45)$$

Here, $\delta_\theta \in \mathbb{R}$ with $|\delta_\theta| \leq 1$, and $\|\Delta_d(s)\|_\infty \leq 1$. The weight of the dynamic uncertainty $W_d(s)$ is a 25th order system obtained by fitting the upper bound of the relative error terms

$$\frac{G_{\text{meas},i}(j\omega) - G_{\text{HDD}}(j\omega)}{G_{\text{HDD}}(j\omega)} \quad \text{for } i = 1, \dots, 50. \quad (4.46)$$

When fitting, the peaks in the frequency response in Figure 4.6 were weighted so that these peaks are properly represented in the uncertain model. The low frequency (below 0.5 rad/s) error is ignored, since here the nominal and the uncertain system are both double integrators, and since the variation in the rotational inertia θ accounts for these deviations. The relative error terms along with the resulting uncertainty weight are illustrated in Figure 4.7. The gain of the samples of $G(\Delta, s)$ is depicted in Figure 4.8. Compared to Figure 4.6, the high frequency behavior of the measured data is captured by $G(\Delta, s)$ with sufficient accuracy.

We evaluate the performance of the closed-loop by analyzing the sensitivity function of the uncertain system illustrated in Figure 4.9. The sensitivity function is the transfer function from the output disturbance d to e , i.e.,

$$S(\Delta, s) = \frac{1}{1 + G(\Delta, s) K(s)}. \quad (4.47)$$

The uncertainty sets associated with $S(\Delta, s)$ are

$$\begin{aligned} \mathbb{A}_p &= \{\delta_\theta : \delta_\theta \in \mathbb{R}, |\delta_\theta| \leq 1\}, \\ \mathbb{A}_d &= \{\Delta_d(s) : \Delta_d(s) \text{ is LTI}, \|\Delta_d(s)\|_\infty \leq 1\}. \end{aligned} \quad (4.48)$$

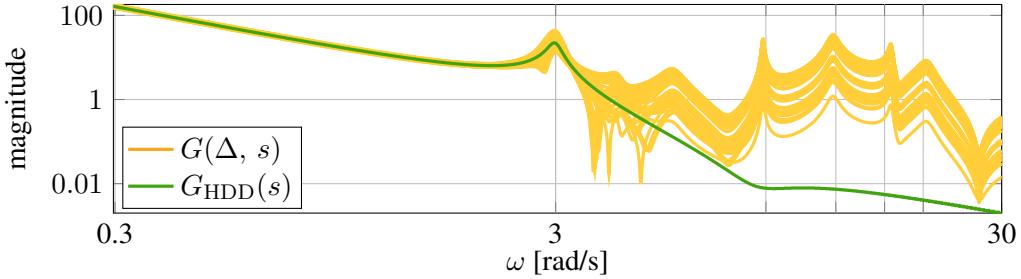


Figure 4.8: Frequency response of the uncertain hard disk drive model (random samples and the nominal system).

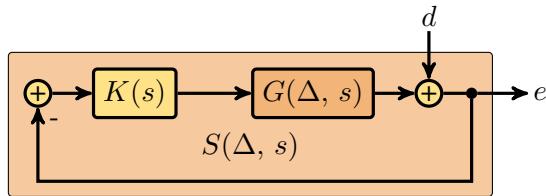


Figure 4.9: Closed-loop with the uncertain hard disk drive.

4.8.3 Worst-case Analysis of the Closed-Loop

The worst-case gain lower bound of the sensitivity function $S(\Delta, s)$ is depicted in Figure 4.10. The maximum of $L(\omega)$ is at 12.55 rad/s. There are local peaks at $\omega_1 = 0.59$ rad/s and, because of the elastic modes, also at $\omega_2 = 2.91$ rad/s, $\omega_3 = 8.74$ rad/s, $\omega_4 = 12.55$ rad/s, $\omega_5 = 16.92$ rad/s, and $\omega_6 = 20.28$ rad/s.

The uncertainty $\Delta_s(s)$ is calculated with the classical worst-case construction in Section 2.3.2. $\Delta_s(s)$ is second order and maximizes the gain of $S(\Delta, s)$ at ω_4 only. The gain of the resulting $S(\Delta_s, s)$ is depicted in Figure 4.10. Using Algorithm 3, $\Delta_m(s)$ is obtained that maximizes the gain at all of $\{\omega_k\}_{k=1}^6$. Since there is one dynamic uncertainty block and $N_\omega = 6$, $\Delta_m(s)$ has $2N_\omega = 12$ states. The gain of $S(\Delta_m, s)$ in Figure 4.10 is indeed maximal at all the chosen frequencies. This is especially significant at $\omega_2, \omega_3, \omega_5$, and ω_6 since the gain of $S(\Delta_s, s)$ is less than the nominal at these frequencies.

After running the optimization in Algorithm 3, the value of the objective function is $J(\Delta_m(s)) = 29.10$ which is almost identical to the upper bound J_U . For a point of comparison, $J(0) = 7.11$, and $J(\Delta_s(s)) = 20.95$. In this case, to achieve high gain at several frequencies, very little of the gain at ω_4 is sacrificed. Namely, $\|S(\Delta_s, s)\|_\infty = 15.7521$ while $\|S(\Delta_m, s)\|_\infty = 15.7499$.

The gain of $S(\Delta_m, s)$ at frequencies other than $\{\omega_k\}_{k=1}^6$ is not necessarily high. There are two reasons for this. The first is that the derivative of $\Delta_m(s)$ cannot be arbitrarily small at $\{\omega_k\}_{k=1}^6$ due to the boundary Nevanlinna-Pick interpolation (see Section 4.3). This causes the gain of $S(\Delta_m, s)$ to vary at frequencies close by. The other reason is that $S(\Delta, s)$ is a sensitivity function which puts a fundamental limitation on the shape of $|S(\Delta_m, j\omega)|$. The poles of the open-loop $K(s) G(\Delta, s)$ are in the closed left half of the complex plane and its relative degree is two for all $\Delta(s) \in \Delta$. Also, $S(\Delta, s)$ is robustly

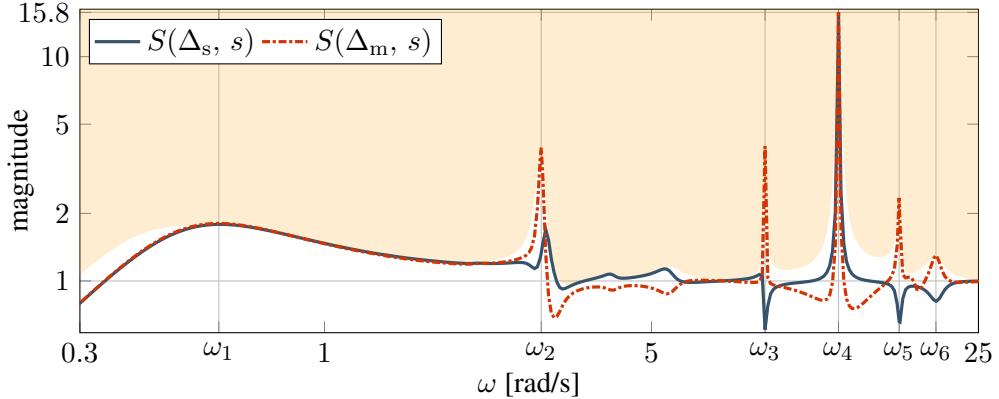


Figure 4.10: Magnitude of the sensitivity function of the hard disk drive control loop for the two samples of the uncertainty. The boundary of the shaded area is $L(\omega)$.

stable for all $\Delta(s) \in \Delta$. Hence, it follows from the Bode integral theorem [67, Section 5.2.3] that the sensitivity satisfies

$$\int_0^\infty \ln |S(\Delta, j\omega)| d\omega = 0 \quad (4.49)$$

for all $\Delta(s) \in \Delta$. This means that if the gain of $S(\Delta, s)$ increases for some frequencies, it must decrease for others. This effect is observable in Figure 4.10.

Time domain simulations are also conducted to demonstrate the effect of the gain maximization in the time response of the uncertain system. A unit step input disturbance is used with small sinusoidal signals added on, i.e.,

$$d(t) = 1 + 0.01 \sum_{k=1}^6 \sin\left(\omega_k t + \frac{k\pi}{3}\right) \quad (4.50)$$

for $t \geq 0$ and $d(t) = 0$ for $t < 0$. This input demonstrates the step response of the closed-loop and, to a smaller degree, also excites the dynamics at all the resonance frequencies. The simulation is run for 600 seconds. The output signal e is depicted in Figure 4.11. On the top diagram, the initial transient is illustrated. The bottom diagram shows the sinusoidal steady state at end of the simulation. The amplitude of both $S(\Delta_s, s)$ and $S(\Delta_m, s)$ is much larger than the nominal response. Figure 4.11 also demonstrates that $\Delta_m(s)$ produces higher amplitudes than $\Delta_s(s)$.

The steady state responses (after 300 s) are further compared by evaluating the maximum and the standard deviation of the output signal e . The comparison is done for the nominal system $S(0, s)$, for the system with the two uncertainty samples $S(\Delta_s, s)$ and $S(\Delta_m, s)$, and for 100 random samples of the uncertainty obtained using the `usample` function in MATLAB. The results are presented in Table 4.2. Out of the random samples only one is presented in Table 4.2 for which both the maximum and the standard deviation is coincidentally the largest. Table 4.2 shows that the multi-frequency uncertainty $\Delta_m(s)$ leads to significantly degraded performance as compared to the random samples and single-frequency worst-case uncertainty.

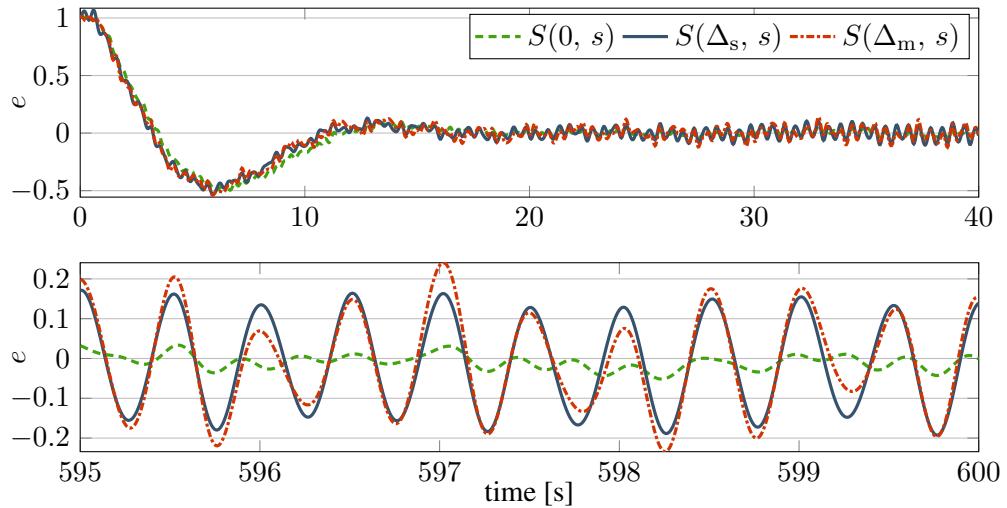


Figure 4.11: Transient and sinusoidal steady state response of the hard disk drive control loop for different samples of the uncertainty.

system	max.	std. dev.
$S(0, s)$	0.0656	0.0211
worst over 100 random samples	0.1023	0.0425
$S(\Delta_s, s)$	0.2050	0.1127
$S(\Delta_m, s)$	0.2757	0.1200

Table 4.2: Maximum and standard deviation of the hard disk drive control loop responses in sinusoidal steady state.

Thesis 2: I propose a method to calculate a worst-case uncertainty sample that maximizes the gain of an uncertain system with mixed uncertainty at multiple frequency points simultaneously. By contrast, employing the classical analysis technique, an uncertainty sample is obtained that maximizes the gain only at the single frequency where that maximum occurs. The presented approach involves a nonlinear optimization that selects the worst-case value of the uncertain parameters and the application of the boundary Nevanlinna-Pick interpolation to calculate the dynamic uncertainty sample. Such a perturbation can be used to augment Monte Carlo simulations of uncertain systems, especially if the system has multiple resonance frequencies.

Related publications: [P53, P50]

5

Flutter suppression control of a flexible aircraft

The aeroelastic flutter is an undamped oscillation of a flexible structure placed into an airflow. With increasing airspeed, the structural damping becomes insufficient to compensate the intensifying vibration caused by the aerodynamic forces. Beyond a certain airspeed value, which is called flutter speed, these vibrations become undamped with the threat of imminent catastrophic structural failure. Hence, this phenomenon has to be strictly avoided. Conventional aircraft are designed in a way that flutter does not occur under normal operating conditions. This is usually achieved through the use of stiffening materials and thus at the expense of additional structural mass. However, the future trends in aircraft design are oriented to build more economical vehicles, i.e., to increase fuel efficiency and decrease the operating costs. To achieve these goals, increasing the aspect ratio, decreasing the structural mass, and using more flexible components are possible ways to go. On the other hand, in more flexible aircraft, the flutter effect can occur during normal operation, hence an active flutter suppression control has to be developed to enlarge the flutter-free operating domain. One possible approach to flutter control is the use of traditional control surfaces, e.g., ailerons, to influence the airflow around the wing to limit aerodynamic loads.

The demonstrator of the FLEXOP project, called T-FLEX and shown in Figure 1.1 (on page 4), was built specifically for flutter control experimentation [18]. This chapter is dedicated to the flutter suppression control design for this platform with the application of the robust synthesis algorithm described in Chapter 3. After a brief literature review in Section 5.1, the high fidelity model of the aircraft used in the analysis of the closed-loop is given in Section 5.2, along with the construction of the reduced-order, control-oriented model. Section 5.3 presents the uncertain design model, the formulation of the control design task, and the synthesis results. Worst-case analysis of the closed-loop is conducted in Section 5.4. Finally, the validation of the flutter controller using the high fidelity model is detailed in Section 5.5.

5.1 Related literature

The robust control framework was successfully applied to flutter suppression in multiple studies. In [70], an H_∞ controller is designed to increase the damping of flutter-related modes. This technique was extended to the linear parameter varying case in [P47]. To maximize control effectiveness for the flutter modes and minimize the excitation of the remaining modes, optimal blending of the inputs and outputs is utilized in [56]. None

of these methods take model uncertainty explicitly into account, instead, they rely on the inherent robustness of the H_∞ and induced L_2 controllers. Aside from the different synthesis method, and the uncertain modeling of the aircraft, another distinguishing factor of the work presented in this chapter is the rigorous analysis of applicability. The robustness of the flutter control loop is evaluated using the high fidelity model of the aircraft and by computing loop margins. Time domain simulations are also conducted to investigate the extension of the safe flight envelope.

5.2 Aircraft model

The single-engine T-FLEX aircraft in Figure 1.1 features a wing span of 7 m, aspect ratio of 20, and takeoff weight between 55 and 65 kg. A set of wings were built for it that exhibit flutter at the relatively low airspeed of around 52 m/s. This section presents the most important aspects of the physical model of the T-FLEX aircraft. These results are taken from [68, 34]. The steps and products of the modelling process are summarized in Figure 5.1.

5.2.1 Nonlinear high fidelity model of the flexible aircraft

The construction of the aeroservoelastic (ASE) model of the T-FLEX aircraft is based on a subsystem approach which involves the integration of aerodynamics, structural dynamics, and flight dynamics [35, 28] as illustrated in Figure 5.2. The components in Figure 5.2 are developed separately and then combined to form the ASE model. The structural dynamics are modeled by high fidelity finite element method [34] and are then condensed with Guyan reduction [21]. The unsteady aerodynamics model of the aircraft is based on the vortex-lattice method (VLM) and the doublet-lattice method (DLM) and it is offset by results from computational fluid dynamics (CFD) simulations. The nonlinear equations of motions are derived based on a mean axes reference frame [64]. The mean axes approach describes the dynamics of the flexible body by a set of equations which decouple the rigid body modes from the vibrational modes. The mean axes coordinates ensure that the coupling is restricted to external forcing terms only [64].

Each wing of the T-FLEX aircraft is equipped with four control surfaces [60] with the outermost pair dedicated to flutter suppression as illustrated in Figure 5.3. To actuate these, a custom made direct drive actuator is developed with bandwidth greater than the flutter frequencies. Based on system identification, the direct drive actuator has 0.1 ms delay and transfer function

$$G_{\text{act}}(s) = \frac{0.78s + 2.74 \cdot 10^5}{s^2 + 564.51s + 2.74 \cdot 10^5}. \quad (5.1)$$

In addition to the GPS and air data probe, the aircraft features inertial measurement units (IMUs) at the center of gravity and in the wings as shown in Figure 5.3. The IMUs provide acceleration and angular rate measurements along all three body axes.

The nonlinear ASE model of the T-FLEX aircraft consists of 12 rigid body states, 100 states corresponding to flexible modes, and 1040 aerodynamic lag states in addition to the actuator dynamics. This model is considered as the high fidelity model. Based on this model, the aircraft has two unstable aeroelastic modes. The symmetric flutter mode becomes unstable at 52 m/s and the corresponding resonance frequency (flutter

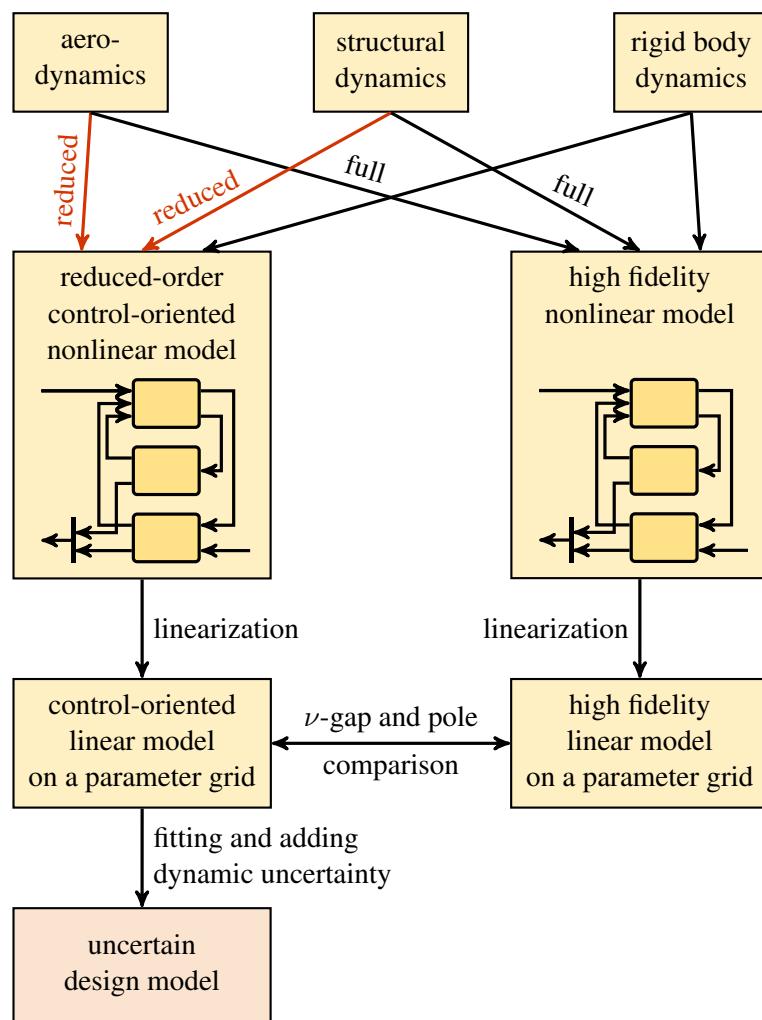


Figure 5.1: Process of the model construction for the flutter control design.

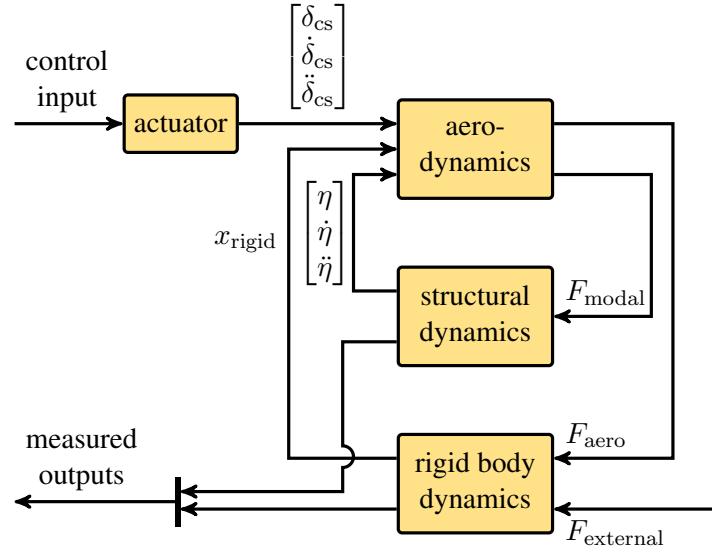


Figure 5.2: ASE subsystem interconnection. (F_{aero} represents the aerodynamic forces acting on the rigid body dynamics, F_{external} represents external (propulsion and gravitational) forces; the rest of the variables are defined in the remainder of the section.)

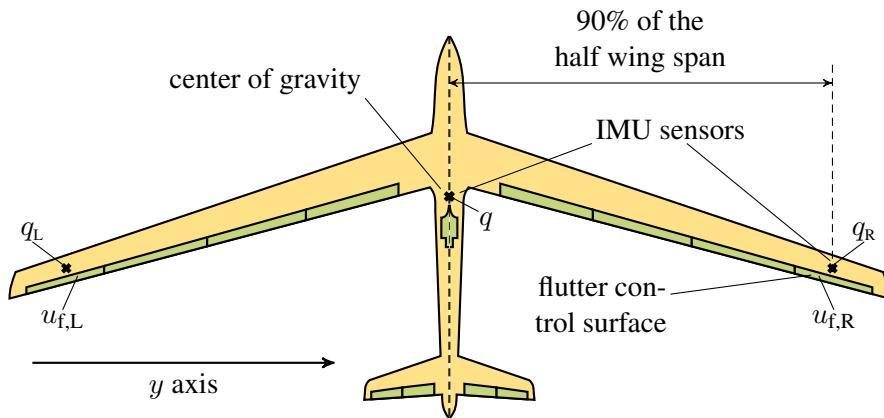


Figure 5.3: Positions of the sensors and control surfaces used for the flutter suppression control. The control inputs and measurements are marked at the corresponding control surface or sensor. The signals q_L , q_R , and q denote angular rates along the y axis, $u_{f,L}$ and $u_{f,R}$ are deflection commands for the actuators.

frequency) is 50.2 rad/s. The asymmetric mode becomes unstable at 55 m/s with flutter frequency 45.8 rad/s.

5.2.2 Bottom-up model reduction

The high fidelity model in 5.2.1 has over one thousand states. Control design for such a high dimensional model is not practical therefore an appropriately low order and numerically tractable control-oriented model is required [70]. Because the size of this model poses considerable difficulty to the linear parameter varying reduction techniques [31, 59], the "bottom-up" modeling approach in [68, 34] is applied instead. The key idea is to reduce the components of the system in Figure 5.2 separately. Since the structural and aerodynamics subsystems have simpler structure than the combined ASE model, it is possible to simplify them using more conventional reduction techniques. This approach leads to a sufficiently low order ASE control-oriented model.

Before reduction, the inputs and output of the system are selected that are involved in the control design. The outermost control surface pair is used for flutter suppression as illustrated in Figure 5.3. The actuating signal is the control surface deflection command received by the actuator denoted as $u_{f,L}$ and $u_{f,R}$ for the left and right wing respectively. The 'f' index refers to the word 'flutter'. The sensors used for control design are three IMUs: one in the center of gravity and two at the 90% of the half wing span on each wing. Only the angular rate measurements along the y axis of the three IMUs are used (see Figure 5.3). The signals of the IMUs on the left and right wings are denoted by q_L and q_R respectively while q denotes the pitch rate in the center of gravity. Only these inputs and outputs are considered in the reduction.

The frequency range of interest where the control-oriented model is expected to provide a good approximation of the high fidelity model is $[0, 100]$ rad/s. The 100 rad/s is roughly twice the flutter frequencies 50.2 rad/s and 45.8 rad/s which ensures the accurate representation of the flutter behavior. It is possible to retain the accuracy of the control-oriented model over a wider frequency range at the expense of additional dynamics.

The objective of the reduction is to decrease the number of states of the subsystems in Figure 5.2 while maintaining an acceptably low ν -gap between the high fidelity and the control-oriented model in the frequency range of interest ($[0, 100]$ rad/s). The ν -gap metric $\delta_\nu(\cdot, \cdot)$ is used since it takes into account the feedback control objective [71]. It assumes values between zero (for identical systems) and one. If a feedback controller stabilizes the system $P_1(s)$ with stability margin ε , then it also stabilizes $P_2(s)$ if $\delta_\nu(P_1(s), P_2(s)) < \varepsilon$. A plant at a distance greater than ε from $P_1(s)$, on the other hand, will in general not be stabilized by the same controller [71]. The ν -gap between $P_1(s)$ and $P_2(s)$ at the fixed frequency ω is

$$\delta_\nu(P_1(j\omega), P_2(j\omega)) = \bar{\sigma} \left[(I + P_2(j\omega) P_2^*(j\omega))^{-1/2} (P_1(j\omega) - P_2(j\omega)) \cdot (I + P_1^*(j\omega) P_1(j\omega))^{-1/2} \right], \quad (5.2)$$

where $\bar{\sigma}[\cdot]$ denotes largest singular value.

The aerodynamics of the model in Figure 5.2 are composed of static mappings and dynamic components that are called lag terms. The latter are more significant from the

point of view of the model reduction. The lag terms assume the state-space form

$$\dot{x}_{\text{lag}} = \frac{2V}{\bar{c}} A_{\text{lag}} x_{\text{lag}} + B_{\text{lag}} \begin{bmatrix} \dot{x}_{\text{rigid}} \\ \dot{\eta} \\ \dot{\delta}_{\text{cs}} \end{bmatrix}, \quad (5.3)$$

$$y_{\text{lag}} = C_{\text{lag}} x_{\text{lag}},$$

where V is the airspeed, \bar{c} is the reference chord, x_{rigid} is the rigid body state, η is the state of the structural dynamics, and δ_{cs} is the control surface deflection. Additional input-output mappings are used to map the input and output of the system in (5.3) to the input and output of the aerodynamics block in Figure 5.2.

Balancing transformation is applied to the state-space matrices A_{lag} , B_{lag} and C_{lag} of (5.3). The order reduction is achieved by residualizing the states with the smallest Hankel singular values. Keeping two lag states results in acceptable accuracy. Coefficient $C_{Q_3\eta_1}$ for the modal generalized force (see Chapter 7 of [64] for more details) affecting the symmetric flutter mode and coefficient C_{l_p} affecting the asymmetric flutter mode of the control-oriented model were scaled by 0.65 and 1.6 respectively. By this heuristic modification, the resulting control-oriented model matches the flutter speeds and frequencies of the high fidelity model better. The effect of this modification to the rest of the dynamics is negligible.

The structural dynamics of the aircraft are of the form

$$\mathcal{M}\ddot{\eta} + \mathcal{C}\dot{\eta} + \mathcal{K}\eta = F_{\text{modal}}, \quad (5.4)$$

where \mathcal{M} , \mathcal{C} and \mathcal{K} are the modal mass, damping and stiffness matrices respectively and F_{modal} is the external excitation in modal coordinates. The structural dynamics model is an LTI system therefore state truncation is applied. Along with the first six structural modes, modes 19, 20, 21 are retained since their removal results in a large increase in the ν -gap between the high fidelity and the control-oriented model. This way, the 100th order structural dynamics model is reduced to 18 states. It is assumed that the structural dynamics model has parametric uncertainty. Specifically, the first six modes of the control-oriented model have $\pm 1\%$ uncertainty in the natural frequency and $\pm 10\%$ in their damping.

The resulting bottom-up control-oriented model in Figure 5.1 has 56 states that consists of 12 rigid body states, 18 structural dynamics states, 2 aerodynamics lag states, and 24 actuator dynamics states. The design model is obtained by trimming and linearizing this model for straight and level flight at 36 equidistant points of the airspeed in the interval [30, 65] m/s and for each combination of the perturbed parameters in the structural dynamics. The ν -gap between the high fidelity and the control-oriented model for different airspeed values and for the nominal structural dynamics is depicted in Figure 5.4 (see also Figure 5.1). The ν -gap is calculated for the inputs and outputs used for flutter control (see Figure 5.3). The pole migration of the high fidelity and the control-oriented model is compared in Figure 5.5. The full order model predicts flutter at 52 and 55 m/s at frequencies of 50.2 and 45.8 rad/s respectively. In comparison, flutter occurs in the control-oriented model at 52 and 56.5 m/s at 50.3 and 46 rad/s respectively. The flutter speed and frequency accuracy of the control-oriented model is deemed sufficient for control design.

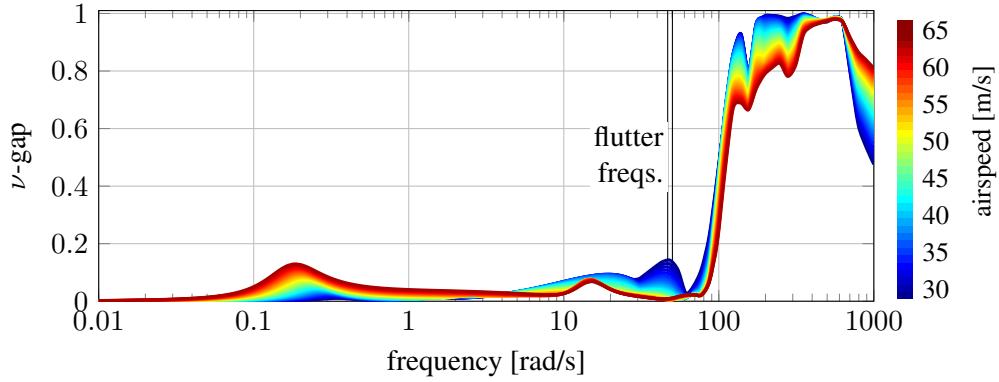


Figure 5.4: The ν -gap values as a function of frequency between the high fidelity and the control-oriented flexible aircraft model for the inputs and outputs used for the flutter suppression control design.

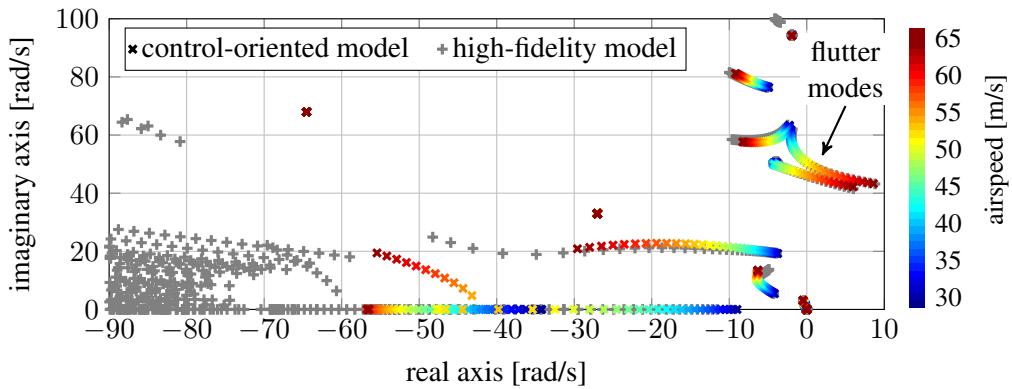


Figure 5.5: Pole migration of the control-oriented and high fidelity flexible aircraft model.

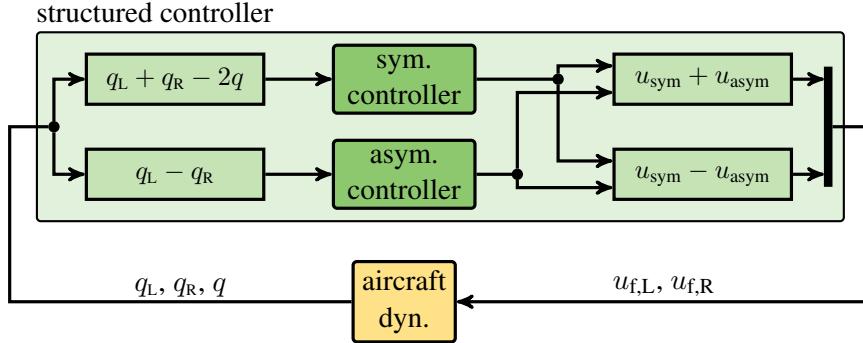


Figure 5.6: Structure of the closed flutter control loop using two SISO controllers to stabilize the symmetric and asymmetric flutter modes separately.

5.3 Flutter control synthesis

In this Section, the flutter suppression control design for the flexible aircraft in Section 5.2.1 is discussed. First, two uncertain SISO models are obtained from the reduced order control-oriented model detailed in Section 5.2.2. Then, the performance specification used for the optimal control design is described. Finally the results of the synthesis method in Chapter 3 are given.

5.3.1 Design model

As the last step of the modelling process in Figure 5.1, the uncertain design model is constructed next. The actuating signal and measured output used for flutter control, as described in Section 5.2.2 (and in Figure 5.3) are respectively

$$u_a = \begin{bmatrix} u_{f,L} \\ u_{f,R} \end{bmatrix} \quad \text{and} \quad y_m = \begin{bmatrix} q_L \\ q_R \\ q \end{bmatrix}. \quad (5.5)$$

Combining the elements of y_m and u_m in a specific way allows us to separate the symmetric and asymmetric flutter modes of the aircraft [57]. This makes it possible to design controllers that stabilize these two modes separately. To this end, define input and output blend matrices

$$b_u = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad b_y = \begin{bmatrix} -2 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}, \quad (5.6)$$

and the input and output combinations $u = b_u u_a$ and $y = b_y y_m$. The controller structure corresponding to these transformations is illustrated in Figure 5.6. The blended aircraft model is separated into two SISO systems: $\tilde{G}_{\text{sym}}(s)$ and $\tilde{G}_{\text{asym}}(s)$. The input and output of $\tilde{G}_{\text{sym}}(s)$ are $u_{f,L} + u_{f,R}$ and $q_L + q_R - 2q$ respectively. The states consist of w (vertical velocity), q (pitch rate), the modal coordinates that correspond to the symmetric wing deformations, the lag states, and the actuator states. For $\tilde{G}_{\text{asym}}(s)$, the input and output are $u_{f,L} - u_{f,R}$ and $q_L - q_R$ respectively. The states are v (horizontal velocity), p (roll rate), r (yaw rate), the modal coordinates corresponding to the asymmetric wing deformations, and lag and actuator states.

Both parametric and dynamic uncertainty are introduced using $\tilde{G}_{\text{sym}}(s)$ and $\tilde{G}_{\text{asym}}(s)$. As the result of the model reduction technique in Section 5.2.2, the state-space matrices of these systems are given on a parameter grid. The grid consists of 36 equidistant points of the airspeed between 30 m/s and 65 m/s, 5 points of the natural frequency in the structural dynamics between $\pm 1\%$ of the nominal value, and 5 points of the damping in the structural dynamics between $\pm 10\%$ of the nominal value. These two arrays of LTI systems are used to introduce parametric uncertainty in the system. The dependence of the dynamics on the airspeed is expressed via the uncertain parameter δ_V . The uncertainty of the natural frequency and the damping in the structural dynamics correspond to uncertain parameters δ_{ω_0} and δ_ξ respectively. The parameters are normalized uncertainties, i.e., $|\delta_V| \leq 1$, $|\delta_{\omega_0}| \leq 1$, $|\delta_\xi| \leq 1$. Least squares fitting is performed to obtain the uncertain state-space matrices of the form

$$\begin{aligned} A_\delta &= A_0 + A_1\delta_V + A_2\delta_V^2 + A_3\delta_{\omega_0} + A_4\delta_{\omega_0}^2 + A_5\delta_\xi, \\ B_\delta &= B_0 + B_1\delta_V + B_2\delta_V^2, \\ C_\delta &= C_0 + C_1\delta_V + C_2\delta_V^2, \\ D_\delta &= 0, \end{aligned} \tag{5.7}$$

where $\{A_i\}_{i=0}^4$, $\{B_i\}_{i=0}^2$, and $\{C_i\}_{i=0}^2$ are constant matrices. The elements of A_δ , B_δ , and C_δ are assumed to be a second order polynomial in δ_V based on Chapter 5 in [11]. Only A_δ depends on δ_{ω_0} and δ_ξ since the perturbation in the damping and natural frequency influences the position of the poles. Also, this form of A_δ , B_δ , and C_δ provides low error when compared to \tilde{G}_{asym} and \tilde{G}_{sym} in the parameter grid points. After the fitting, the Morton method, available in the LFR-toolbox for MATLAB, is used to minimize the size of the resulting uncertainty blocks without changing the dynamics of the uncertain systems [36, 22].

Dynamic uncertainty is added to account for the model reduction in Section 5.2.2. As illustrated in Figure 5.7, input multiplicative uncertainty structure is chosen, i.e., both uncertain SISO plants have the form

$$G(\Delta, s) = C_\delta(sI - A_\delta)^{-1} B_\delta (1 + W_d(s) \Delta_d(s)). \tag{5.8}$$

Here, $\Delta_d(s)$ is the stable SISO uncertainty block for which $\|\Delta_d(s)\|_\infty \leq 1$, and $W_d(s)$ is the weight of the uncertainty. The subscripts 'sym' and 'asym' are omitted since $G(\Delta, s)$ refers to both systems. The weight is obtained by comparing the singular values of the system $C_\delta(sI - A_\delta)^{-1} B_\delta$ to the high fidelity model similarly to the process used for the hard disk drive modeling in Section 4.8.2. This results in

$$\begin{aligned} W_{d,\text{sym}}(s) &= \frac{33.31(s + 111.7)(s^2 + 49.17s + 2195)}{(s + 409.5)(s^2 + 97.17s + 198900)}, \\ W_{d,\text{asym}}(s) &= \frac{55.703(s + 100)(s + 40)^2}{(s + 400)(s^2 + 350s + 25000)}. \end{aligned} \tag{5.9}$$

The Bode magnitude plot of both of the weights are depicted in Figure 5.8. These weighting functions add moderate uncertainty at low frequencies and significant uncertainty at high frequencies. The level of uncertainty starts to rise considerably at the flutter frequencies and it is above 1 (100%) after 100 rad/s. This is in accordance with the ν -gap results of the model reduction in Section 5.2.2 (see Figure 5.4).

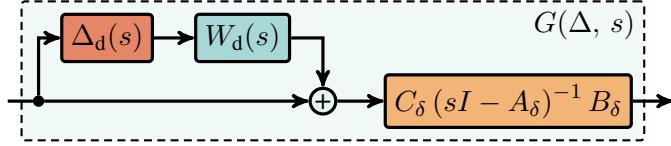


Figure 5.7: Uncertain flexible aircraft model interconnection with the dynamic uncertainty $\Delta_d(s)$ and A_δ , B_δ , C_δ matrices depending on the parametric uncertainty.

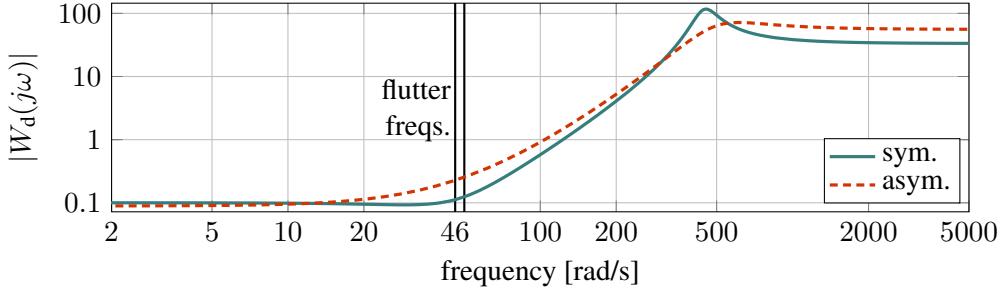


Figure 5.8: Bode magnitude plot of the dynamic uncertainty weights for the uncertain flexible aircraft model.

The resulting uncertain systems are written in LFT form as

$$G(\Delta, s) = \mathcal{F}_U(N(s), \Delta(s)), \quad (5.10)$$

where $\Delta(s)$ is the structured uncertainty block containing the three uncertain parameters δ_V , δ_{ω_0} , and δ_ξ and the dynamic uncertainty block $\Delta_d(s)$. Therefore, for $G_{\text{sym}}(\Delta, s)$ and $G_{\text{asym}}(\Delta, s)$ respectively, the uncertainty sets Δ_{sym} and Δ_{asym} are defined as

$$\begin{aligned} \Delta_{p,\text{sym}} &= \left\{ \begin{bmatrix} \delta_V I_{19} & & 0 \\ & \delta_{\omega_0} I_{15} & \\ 0 & & \delta_\xi I_4 \end{bmatrix} \right\}, \\ \Delta_{p,\text{asym}} &= \left\{ \begin{bmatrix} \delta_V I_{19} & & 0 \\ & \delta_{\omega_0} I_{10} & \\ 0 & & \delta_\xi I_2 \end{bmatrix} \right\}, \\ \Delta_{d,\text{sym}} &= \{\Delta_{d,\text{sym}}\}, \\ \Delta_{d,\text{asym}} &= \{\Delta_{d,\text{asym}}\}. \end{aligned} \quad (5.11)$$

As indicated by the indices of the identity matrices in $\Delta_{p,\text{sym}}$ and $\Delta_{p,\text{asym}}$, the uncertain parameters appear a large number of times in the systems. The number of repetitions are especially high for δ_V and δ_{ω_0} which makes the problem numerically intractable for the classical μ -synthesis by D-K iteration (see Section 2.2.3). However, this is not an issue for the control synthesis method in Chapter 3 since it samples the uncertain parameters instead of trying to construct D-scales for parametric uncertainty blocks.

The poles of $G_{\text{sym}}(\Delta, s)$ and $G_{\text{asym}}(\Delta, s)$ migrate with the change of the uncertain parameters. The domain of migration of the flutter modes is illustrated in Figure 5.9. The introduction of δ_{ω_0} and δ_ξ in $G_{\text{sym}}(\Delta, s)$ and $G_{\text{asym}}(\Delta, s)$ captures the uncertainty in frequency and damping ratio in addition to the variation with airspeed. Note that the nom-

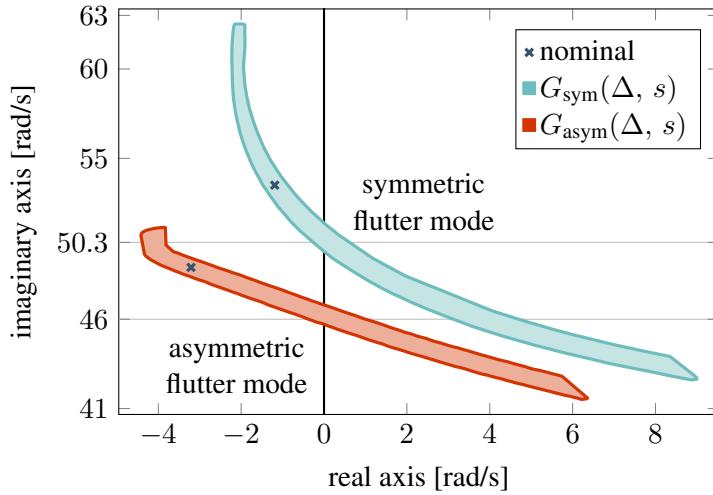


Figure 5.9: Domain of the migration of the flutter mode poles due to the parametric uncertainty in the uncertain flexible aircraft model.

inal systems, marked by 'x' in Figure 5.9, are stable, since they correspond to 47.5 m/s airspeed which is below the flutter speed of 52 m/s. The gain of the two systems change due to both types of uncertainty. In Figure 5.10, the Bode magnitude plot of the nominal systems are depicted along with random samples. In the frequency range below the flutter frequencies, only a moderate variation is observable while for frequencies above 100 rad/s, the gain of the systems is highly uncertain. Since the nominal systems are stable, the gain at the flutter frequencies are finite for both $G_{\text{sym}}(0, s)$ and $G_{\text{asym}}(0, s)$. For some values of the uncertainty, the gain of these systems is significantly higher or even unbounded.

5.3.2 Performance specification

The purpose of the flutter controller is to robustly stabilize the undamped vibrations of the wings of the aircraft. There are further considerations to be taken into account. The effect of the flutter controller on the rigid body behavior of the aircraft ought to be minimal so that the flutter controller does not interfere with the baseline controller governing the rigid body motion of the aircraft. The controller is implemented on an embedded computer whose sampling frequency is 200 Hz. Beyond this constraint, the bandwidth of the flutter controller must not breach the limitation posed by the actuator. Also, the autopilot hardware, the sensors, the actuator, and the communications channels within the flight control computer introduce delay into the closed-loop system. This is modeled by a 15 ms output delay in the plant. Therefore, another requirement is that the closed-loop must remain stable even in the presence of 15 ms delay.

The generalized plant interconnection in Figure 5.11 incorporates all the requirements above for both $G_{\text{sym}}(\Delta, s)$ and $G_{\text{asym}}(\Delta, s)$. The 15 ms delay is represented by $\tau(s)$ that is the 4th order Padé approximation of $e^{-0.015s}$. The controller is augmented with

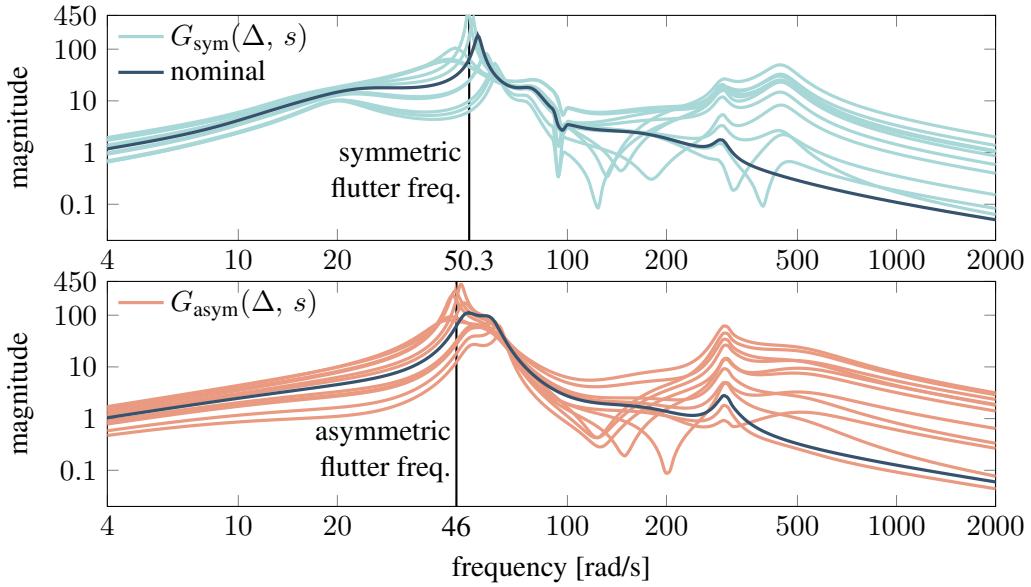


Figure 5.10: Bode magnitude plot of the nominal value and random samples of the uncertain systems $G_{\text{sym}}(\Delta, s)$ and $G_{\text{asym}}(\Delta, s)$.

the filter

$$F(s) = \frac{1.6 \cdot 10^5}{s^2 + 560s + 1.6 \cdot 10^5} \quad (5.12)$$

to ensure appropriate bandwidth. This way, the sampling constraint is met and the excitation of high frequency dynamics is avoided. The Bode magnitude plot of $F(s)$ along with the performance constraints are depicted in Figure 5.12.

The task of robust stabilization is expressed as sensitivity minimization. The sensitivity function of both closed-loops is

$$S(\Delta, s) = \frac{1}{1 + \tau(s) G(\Delta, s) K(s) F(s)}. \quad (5.13)$$

For any stable SISO loop, the minimal distance between the open loop Nyquist curve and

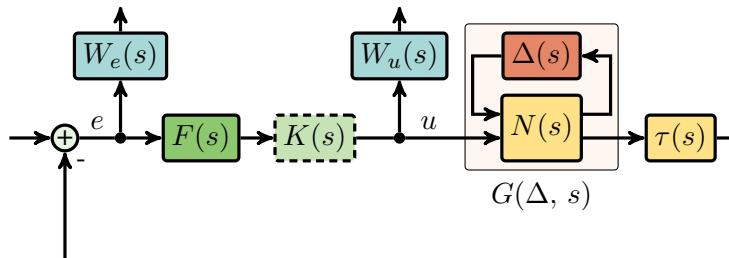


Figure 5.11: Generalized plant interconnection for flutter suppression control via sensitivity minimization.

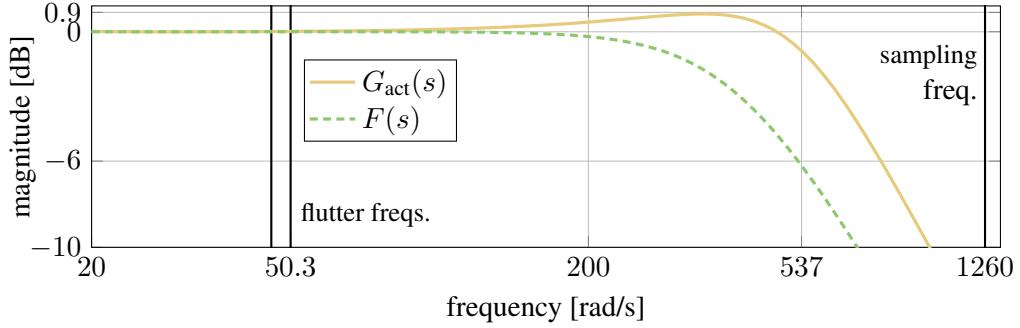


Figure 5.12: Bode magnitude plot of $F(s)$ and the performance constraints. The actuator dynamics $G_{\text{act}}(s)$ are given in (5.1).

the -1 point is the inverse of the peak of $|S(j\omega)|$ [67], i.e., it is

$$\frac{1}{\max_{\omega} |S(j\omega)|} = \frac{1}{\|S(s)\|_{\infty}}. \quad (5.14)$$

According to [67], the expressions

$$\frac{\|S(s)\|_{\infty}}{\|S(s)\|_{\infty} - 1} \quad \text{and} \quad 2 \sin^{-1} \left(\frac{1}{2 \|S(s)\|_{\infty}} \right) \quad (5.15)$$

provide a lower bound of the gain and phase margin, respectively. Therefore, we want to achieve $\|S(\Delta, s)\|_{\infty} \leq 2$ for all $\Delta(s) \in \Delta$ since that ensures at least 6 dB gain margin and close to 30° phase margin. Hence, the weight of the tracking error (sensitivity function) is chosen as $W_e(s) = 1/2$ with which the objective is to achieve $\|W_e(s) S(\Delta, s)\|_{\infty} \leq 1$ for all $\Delta(s) \in \Delta$. To limit actuator effort, the weight of the control input is $W_u(s) = 1/10^\circ = 5.78$.

5.3.3 Synthesis

The flutter suppression controller is designed with Algorithm 1 in Chapter 3. Following the modeling concept in Figure 5.6, the structure of the complete MIMO controller is depicted in Figure 5.13. The SISO controllers are parametrized as general second and fourth order transfer functions, i.e.,

$$K_{\text{asym}}(s) = \frac{\kappa_1 s^2 + \kappa_2 s + \kappa_3}{s^2 + \kappa_4 s + \kappa_5}, \quad (5.16)$$

$$K_{\text{sym}}(s) = \frac{\kappa_6 s^4 + \kappa_7 s^3 + \kappa_8 s^2 + \kappa_9 s + \kappa_{10}}{s^4 + \kappa_{11} s^3 + \kappa_{12} s^2 + \kappa_{13} s + \kappa_{14}}, \quad (5.17)$$

where $\{\kappa_k\}_{k=1}^{14}$ are tunable design parameters. The proposed state order of $K_{\text{asym}}(s)$ and $K_{\text{sym}}(s)$ is the result of design experiments. As depicted in Figure 5.13, the flutter controller is $u_a = K_f(s) y_m$, where

$$K_f(s) = b_u \begin{bmatrix} K_{\text{asym}}(s) F(s) & 0 \\ 0 & K_{\text{sym}}(s) F(s) \end{bmatrix} b_y. \quad (5.18)$$

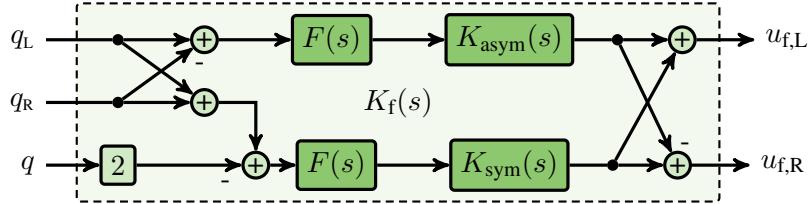


Figure 5.13: Structure of the flutter suppression controller.

Therefore, the final state order of $K_f(s)$ is 10.

For the synthesis of the D-scales for $K_{\text{asym}}(s)$, a frequency grid of 30 points and 5 basis functions were used. For $K_{\text{sym}}(s)$, the number of frequency points and basis functions are 90 and 9 respectively. These values represent a trade-off between accuracy and numerical conditioning. Increasing the number of basis functions allows for richer dynamics in the D-scales but it also increases the number of decision variables in the optimization in Section 3.4. Similarly, a denser frequency grid makes the worst-case gain upper bound more accurate, but the numerical conditioning of the calculations involved also deteriorate. The values given above are the result of experiments with the synthesis algorithm.

The two SISO loops are tuned separately. As the result of the synthesis, the tunable components of $K_f(s)$ are

$$\begin{aligned} K_{\text{asym}}(s) &= \frac{0.0106s^2 + 0.5476s + 53.6039}{s^2 + 40.4171s + 904.595}, \\ K_{\text{sym}}(s) &= \frac{0.0126s^4 + 3.7765s^3 + 1479.85s^2 + 48443.9s + 444523}{s^4 + 114.354s^3 + 59809.2s^2 + 538987s + 1.4737 \cdot 10^6}. \end{aligned} \quad (5.19)$$

It takes 10 iterations (i.e., 10 samples of $\Delta_{p,\text{asym}}$) to obtain $K_{\text{asym}}(s)$ and 9 iterations (9 samples of $\Delta_{p,\text{sym}}$) are required for $K_{\text{sym}}(s)$ [†]. The singular values of the flutter controller are depicted in Figure 5.14. According to the figure, the controller has enough control authority at the flutter frequencies and it is sufficiently rolled off at the sampling frequency. Next, the performance of $K_f(s)$ is analyzed in closed-loop with the uncertain design model and also with the high fidelity model linearized at certain speed values.

5.4 Analysis based on worst-case uncertainty construction

The objective of the controller designed in Section 5.3.3 is to robustly stabilize both flutter modes, i.e., increase damping in the closed-loop. The main point of the performance evaluation is to establish whether dangerous vibrations can occur during flight. Therefore, we conduct a preliminary analysis in which we look for resonance frequencies in the closed-loop and study their effect in time domain. This is followed by the more rigorous investigation in Section 5.5 that aims to establish whether the controller is viable for flight testing.

The closed-loop interconnection used for the worst-case analysis is depicted in Fig-

[†]The computation takes approximately three hours on a computer that runs Ubuntu 16.04 LTS and features an eight-core 2.1 GHz Intel Xenon CPU with 20 GB RAM. The algorithm is run on MATLAB R2016b making use of the Parallel Computing Toolbox.

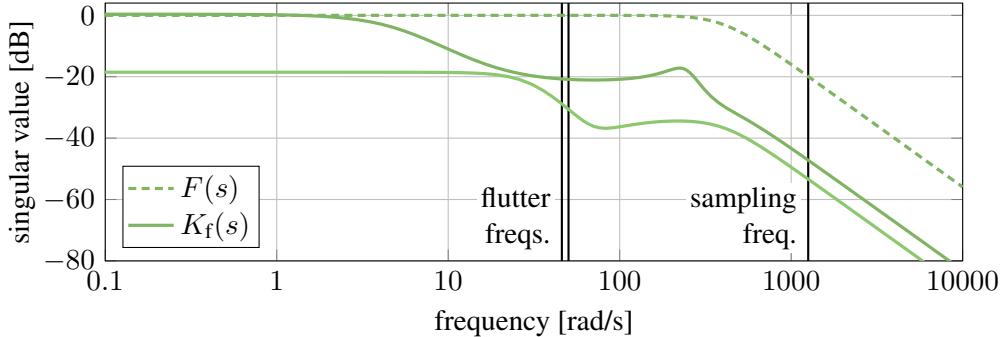


Figure 5.14: Singular values of the flutter controller along with the frequency domain constraints.

ure 5.15. $G(\Delta, s)$ is the uncertain MIMO aircraft model, with input u_a and output y_m , discussed in Section 5.3.1. Three signals are included in Figure 5.15 for analysis purposes: the input and output disturbances d_u and d_y , and the relative accelerations \hat{a}_z which is the difference between the vertical accelerations measured at the IMUs close to the wing tips and at the center of gravity. The diagonal entries of the controller $K_f(s)$ are designed with fixed bandwidth to minimize the diagonal entries of the MIMO sensitivity function from d_y to y in Figure 5.15, i.e.,

$$S(\Delta, s) = (I + b_y G(\Delta, s) b_u K_f(s))^{-1}. \quad (5.20)$$

The uncertainty sets of the MIMO $S(\Delta, s)$ are

$$\begin{aligned} \mathbb{A}_p &= \left\{ \begin{bmatrix} \delta_V I_{42} & 0 \\ 0 & \delta_{\omega_0} I_{32} & \delta_\xi I_6 \end{bmatrix} : \delta_V, \delta_{\omega_0}, \delta_\xi \in \mathbb{R}, |\delta_V|, |\delta_{\omega_0}|, |\delta_\xi| \leq 1 \right\}, \\ \mathbb{A}_d &= \{\Delta_d(s) : \Delta_d(s) \text{ is } 2 \times 2 \text{ LTI}, \|\Delta_d(s)\|_\infty \leq 1\}. \end{aligned} \quad (5.21)$$

The worst-case gain lower bound of $S(\Delta, s)$ is depicted in Figure 5.16. The gain is high around the flutter frequencies (50 rad/s) but the controller clearly provides damping. Two peaks are observable in the gain, $L_1 = 4.228$ at $\omega_1 = 26.59$ rad/s, and a slightly higher $L_2 = 4.229$ at $\omega_2 = 61.09$ rad/s. The classical worst-case uncertainty $\Delta_s(s)$ maximizes the gain of the system at ω_2 only. Based on Figure 5.16, the gain of $S(\Delta_s, s)$ indeed reaches L_2 at ω_2 but it is very close to the nominal gain at the rest of the frequencies. In contrast, Algorithm 3 finds an uncertainty sample that drives the gain of $S(\Delta_m, s)$ to L_1 at ω_1 and also produces high gain at ω_2 as illustrated in Figure 5.16[†].

Similarly to the hard disk drive analysis in Section 4.8, the Bode sensitivity integral provides insight for this result. For our MIMO control system with an unstable open-loop, the integral is of the form

$$\int_0^\infty \ln |\det S(\Delta, j\omega)| d\omega = \pi \sum_{\text{RHP}} \operatorname{Re}(p_k), \quad (5.22)$$

[†]The computation of this uncertainty sample takes 61 seconds. For details about our setup, see Section 4.7.

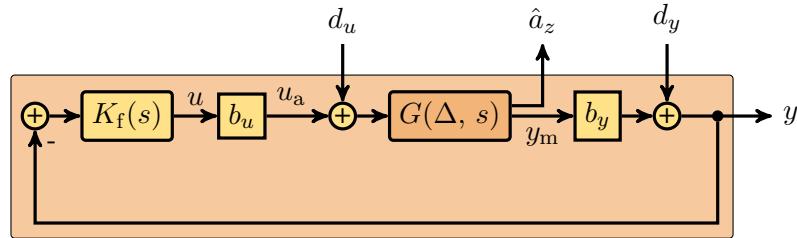


Figure 5.15: Closed flutter control loop with the uncertain flexible aircraft model.

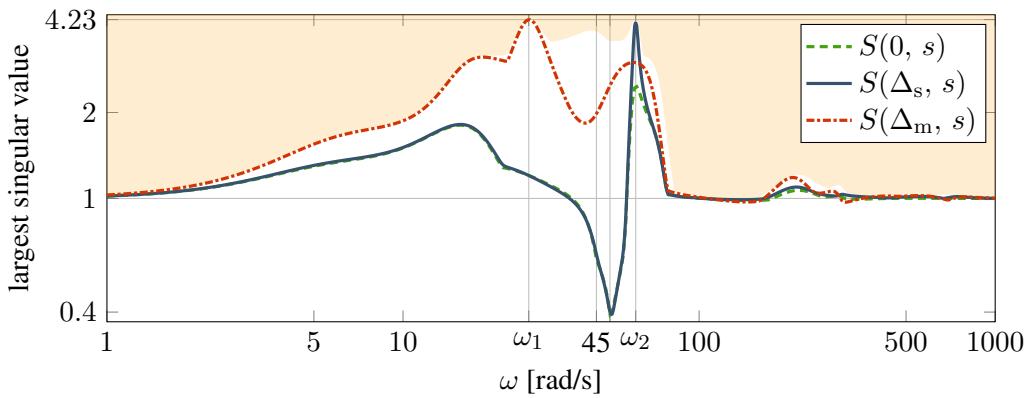


Figure 5.16: Largest singular value of the closed flutter control loop sensitivity function for different samples of the uncertainty. The boundary of the shaded area is $L(\omega)$.

where all the right half plane poles p_k of the open-loop system $b_y G(\Delta, s) b_u K_f(s)$ are summed [67, Section 6.2.3]. The sum of the unstable poles in (5.22) depends on the uncertainty since the uncertain parameters in $S(\Delta, s)$ move the poles of $G(\Delta, s)$. Most notably, the poles corresponding to the flutter modes migrate with the variation in the airspeed (δ_V) as depicted in Figure 5.9. Therefore, different uncertainty samples can produce different Bode integral values. $G(0, s)$ is stable, therefore the integral of $S(0, s)$ evaluates to zero. The classical uncertainty also produces a stable $G(\Delta_s, s)$, which means that the integral of $S(\Delta_s, s)$ is also zero. The multi-peak uncertainty pushes the poles corresponding to the flutter modes into the right hand side of the complex plane which makes

$$\int_0^\infty \ln |\det S(\Delta_m, j\omega)| d\omega = 96.95. \quad (5.23)$$

This is why such a sizable difference is observed between the gain of $S(\Delta_s, s)$ and $S(\Delta_m, s)$ in Figure 5.16.

Since we are interested in the damping the flutter control provides, we study the closed-loop transfer function from the input disturbance

$$d_u = \begin{bmatrix} d_{u,L} \\ d_{u,R} \end{bmatrix} \quad (5.24)$$

to the relative acceleration of the wing tips \hat{a}_z in Figure 5.15. More specifically, let us denote the transfer function from $d_{u,L}$ to $a_{z,L}$ by $T(\Delta, s)$. The response of $T(\Delta, s)$ to a 1° step input is illustrated in Figure 5.17 for the nominal value and the two worst-case uncertainties. Notice that next to the high frequency harmonic observable in both worst-cases, there is an additional low frequency harmonic due to $\Delta_m(s)$. The relative acceleration also clearly assumes higher values for this uncertainty, therefore the use of Algorithm 3 is deemed necessary in the evaluation of the closed-loop performance. The response of the closed-loop from any input in d_u to any output y_m or \hat{a}_z is considerably worse due to the uncertainty sample $\Delta_m(s)$ compared to $\Delta_s(s)$. The relative vertical acceleration was chosen for demonstration because it is related to the forces acting on the wing. According to these results, the acceleration of the wingtip under normal flight conditions does not exceed $3g$. Hence, the flutter controller passed this preliminary analysis step.

5.5 Validation of the flutter controller using the non-linear high fidelity model

This section details the evaluation of the flutter controller described in Section 5.3. First, the baseline control architecture, taken from [39], is presented that governs the rigid body motion of the aircraft. Then, the stability, performance, and robustness analysis is described with the baseline and flutter suppression control loops closed. These results are also contrasted with a design based on a single D-scale instead of the D-K iteration. Finally, two time domain simulations are conducted to confirm the frequency domain results. All tests are performed using the linearized versions of the nonlinear high fidelity model in Section 5.2.1.

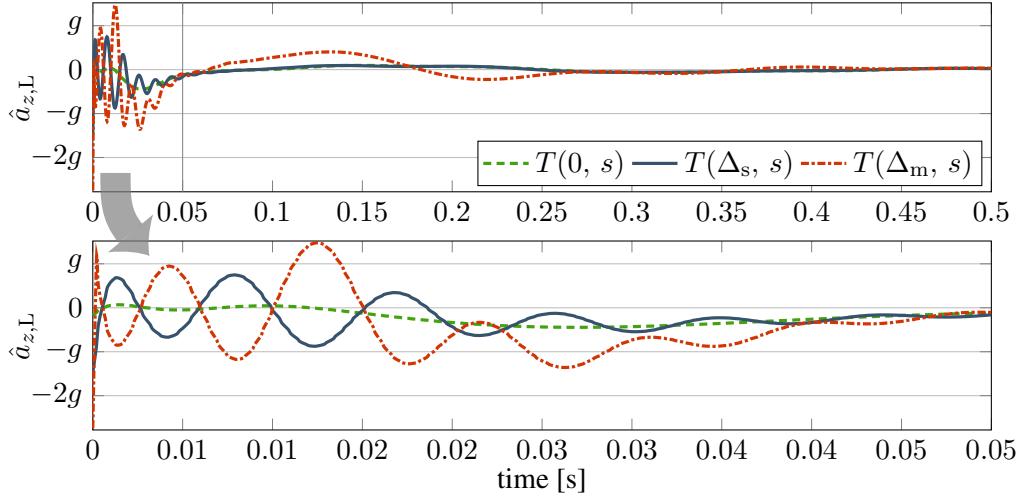


Figure 5.17: Response of the closed flutter control loop to a 1° step input disturbance for different samples of the uncertainty.

5.5.1 Baseline control architecture

The rigid body motion of this aircraft is described by a standard nonlinear six-degrees-of-freedom flight mechanics model (e.g., [33]) in terms of translational velocities u, v, w and angular velocities p (roll), q (pitch), r (yaw) in the body-fixed frame. Orientation in the earth-fixed reference frame is described in terms of Euler angles Φ (bank), Θ (pitch), and Ψ (heading). The angles between the body-fixed frame and the wind axes are angle of attack α and side-slip angle β . The flight path is described with respect to the earth by the path angle γ and the course angle χ .

For the actuation of the rigid body motion, four ruddervators on the V-tail of the aircraft are used, two on the left ($u_{rv,L1}, u_{rv,L2}$) and two on the right side ($u_{rv,R1}, u_{rv,R2}$) as illustrated in Figure 5.18. These ruddervators are combining the functionalities of classical rudders and elevators. The symmetric deflections of the ruddervator correspond to classical elevator deflections, while asymmetric deflections exhibit rudder deflections. Additionally, four control surfaces are available on each wing. As discussed in Section 5.3, the outermost pair ($u_{f,L}, u_{f,R}$) is used for flutter control while the innermost pair is used as high lift devices during takeoff and landing. The remaining two pairs ($u_{a,L2}, u_{a,R2}, u_{a,L3}, u_{a,R3}$) are utilized in the baseline control law as ailerons to actuate the roll motion of the aircraft.

The structure of the baseline controller is a classical cascaded setup depicted in Figure 5.19. The lateral-directional control problem is necessarily multivariable and requires the coordinated use of aileron command u_a and rudder command u_r . The innermost loop features roll-attitude (Φ) tracking, roll-damping augmentation via the roll rate (p), and coordinated turn capabilities, i.e., turns without side-slip, via feedback of the side-slip angle (β). The outer loop establishes control of the course angle (χ). All controllers are scheduled with velocity to increase performance over the velocity range. Within the fully

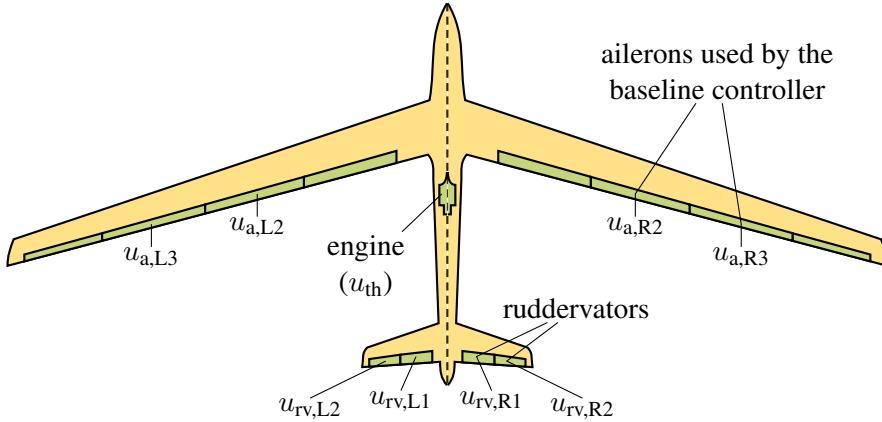


Figure 5.18: Control surface configuration of the baseline control architecture. The control inputs are marked at the corresponding control surface.

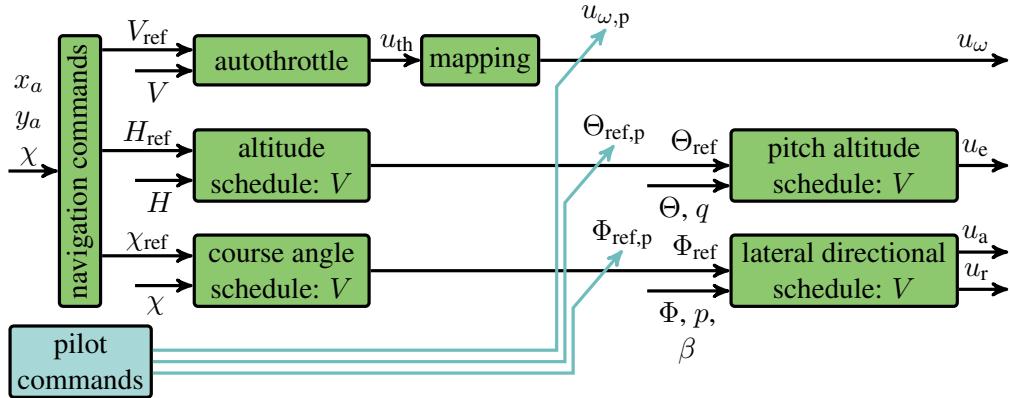


Figure 5.19: Baseline control architecture for fully automated flight, and augmented flight (indicated in blue).

automated flight mode, the reference signals for the velocity (V_{ref}), altitude (H_{ref}), and course angle (χ_{ref}) are provided by a dedicated navigation algorithm. It uses the GPS longitudinal and lateral position of the aircraft (x_a and y_a) as well as the current course angle (χ) to provide the commands. More details on the algorithm can be found in [39].

Structure-wise, the control loops use scheduled elements of PID controllers with additional roll-off filters in the inner loops to ensure that no aeroelastic mode is excited by the baseline controller. A scheduling in dependence of the airspeed V is used to ensure an adequate performance over the velocity range from 32 m/s to 70 m/s. For the scheduling, a first or second order polynomial in V is applied. The free parameters are directly included in a structured controller optimization problem. A comprehensive summary of the used controller structures for each cascaded loop is provided in Table 5.1, including the channel description in the controller architecture and the implemented scheduling.

Note that these controller outputs u_e , u_a , and u_r defer from the actual control surface inputs to ease the control design task. Thus, they need to be transformed to physical actuator commands via an adequate control allocation. The commands to the actuators

control loop	channel	structure	scheduling polyn.
pitch attitude control	$(\Theta_{\text{ref}} - \Theta)$	$\rightarrow u_e$	PI
pitch damping	q	$\rightarrow u_e$	P
roll attitude control	$(\Phi_{\text{ref}} - \Phi)$	$\rightarrow u_a$	P
roll damping	p	$\rightarrow u_a$	P
yaw control	β	$\rightarrow u_r$	PID
autothrottle	$(V_{\text{ref}} - V)$	$\rightarrow u_{\text{th}}$	2 DOF-PID
altitude	$(H_{\text{ref}} - H)$	$\rightarrow \Theta_{\text{ref}}$	PI
course angle	$(\chi_{\text{ref}} - \chi)$	$\rightarrow \Phi_{\text{ref}}$	PID

Table 5.1: Summary of the control loops of the baseline flight control system with the inner loop functions (first part) and autopilot functions (second part). 2 DOF stands for two degrees of freedom.

of the two aileron pairs are determined by

$$\begin{aligned} u_{a,L2} &= u_{a,L3} = 0.5u_a, \\ u_{a,R2} &= u_{a,R3} = -0.5u_a \end{aligned} \quad (5.25)$$

to generate the required differential aileron deflections for roll motion control. For the ruddervators, superposition of the elevator command u_e and the rudder command u_r is applied by

$$\begin{aligned} u_{rv,L1} &= u_{rv,L2} = u_e + 0.5u_r, \\ u_{rv,R1} &= u_{rv,R2} = u_e - 0.5u_r. \end{aligned} \quad (5.26)$$

Thus, symmetric deflections on the left and right of the ruddervators correspond to elevator commands while differential deflections establish rudder commands. The free parameters of the control laws are tuned to satisfy certain performance and robustness criteria as explained in [39].

5.5.2 Closed-loop stability and performance evaluation

The pole trajectories of the closed-loop (with both the flutter and baseline controller) are depicted in Figure 5.20. The damping of both flutter modes are increased significantly. The asymmetric flutter mode is stabilized up to 70 m/s while the symmetric flutter mode crosses over to the right half plane at 68 m/s. Since the closed-loop is unstable at 68 m/s, this is called the absolute flutter speed. Several other modes are influenced by the two controllers but none becomes unstable, therefore no adverse interaction between the flutter and baseline controller is observed.

Let us verify the minimization of the sensitivity functions that is the control objective in Section 5.3.2. To this end, only the flutter controller is applied and the closed-loop is opened loop-at-a-time to get the sensitivity function of the two SISO loops separately. The magnitudes from 30 m/s to 60 m/s are depicted in Figure 5.21. The regions where the sensitivity functions are greater than one are concentrated to low frequency due to the bandwidth of the flutter controller. The sensitivity functions are also reasonably flat: the peaks are 2.41 and 2.08 for the symmetric and asymmetric loop respectively. This means,

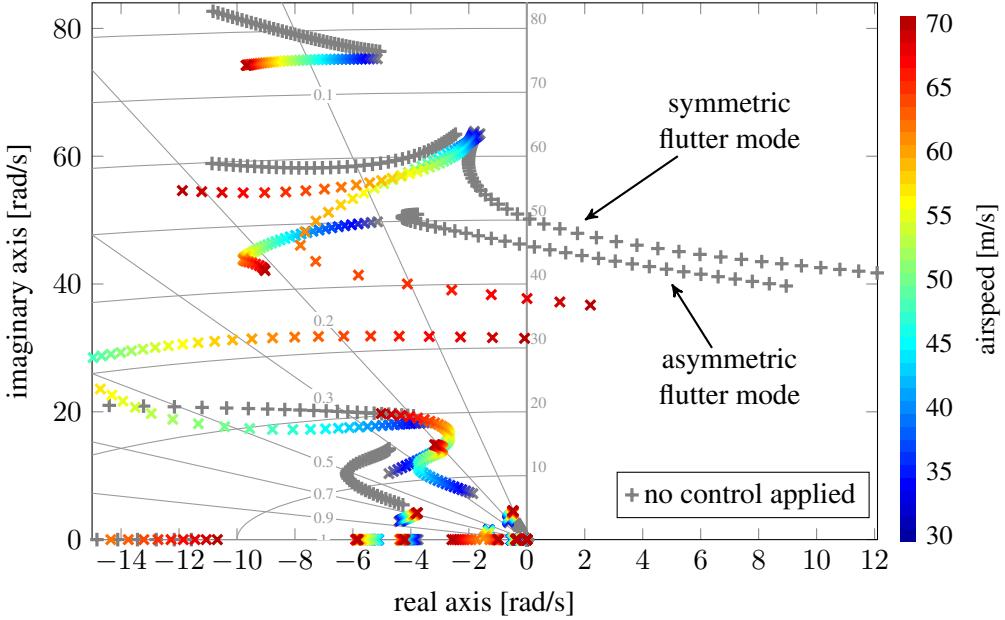


Figure 5.20: Change of pole trajectories with the flutter and baseline controller applied.

that the objective of pushing the sensitivity functions below two was not completely accomplished. The resulting stability margins are computed in Section 5.5.3.

To demonstrate the increase in damping, we compare the gain of the system with and without flutter control. Specifically, Figure 5.22 depicts the gain from an additive disturbance on $u_{f,L}$ to the vertical acceleration measured by the IMU on the left wing (see Figure 5.3). The figure shows significant decrease in damping around the flutter frequencies. Comparing Figure 5.22 to Figure 5.21, we observe that the gain increased in the low frequency range, where the sensitivity values are high, and remained close to the original in the high frequency range where the sensitivity values are close to one.

5.5.3 Robustness analysis

The robustness of the closed-loop is analyzed using disk margins. To define the disk margins, consider the interconnection in Figure 5.23. Here, $\alpha_k \in \mathbb{C}$ are complex scalar uncertainties of the form

$$\alpha_k = \frac{1 + \delta_k}{1 - \delta_k}, \quad (5.27)$$

where $\delta_k \in \mathbb{C}$, $k = 1, \dots, 5$. We differentiate between loop-at-a-time and multi-loop disk margins. The loop-at-a-time disk margin is the largest simultaneous gain and phase variation in a single channel for which the closed-loop remains stable. It is the largest value of m_l , such that, the closed-loop in Figure 5.23 is well posed and stable for $|\delta_l| < m_l$ while $\delta_k = 0$ for $k \neq l$. For multi-loop-margins, the uncertainties in several loops are allowed to vary simultaneously and independently.

The classical gain and phase margin interpretation of the loop-at-a-time margins for each input and output channel in Figure 5.23 is depicted in Figure 5.24. For four channels,

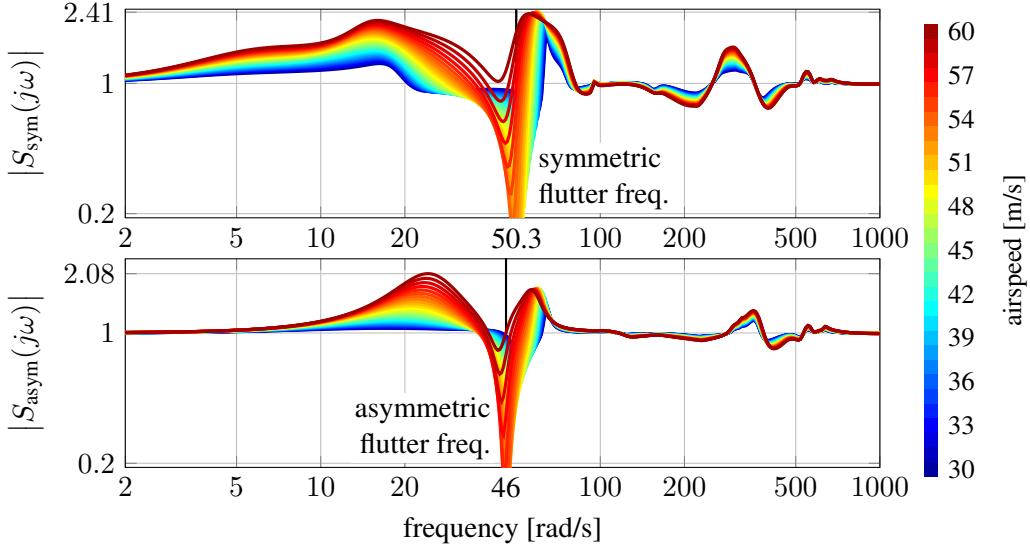


Figure 5.21: SISO sensitivity functions of the closed flutter control loop.

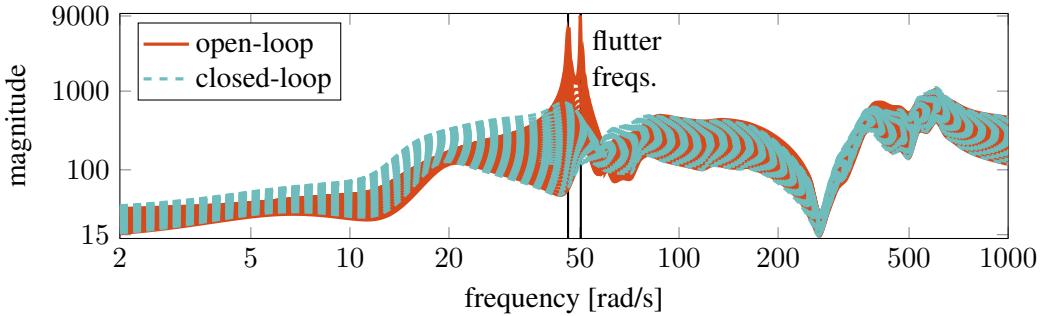


Figure 5.22: Gain of the flexible aircraft from input disturbance to wingtip acceleration in open and closed-loop.

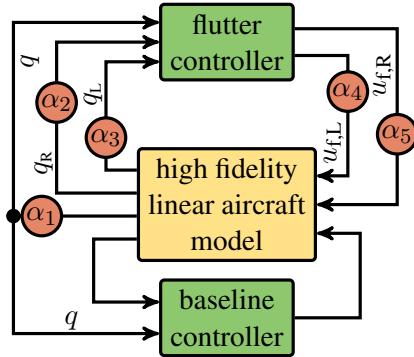


Figure 5.23: Interconnection with the injected uncertainties for the disk margin analysis.

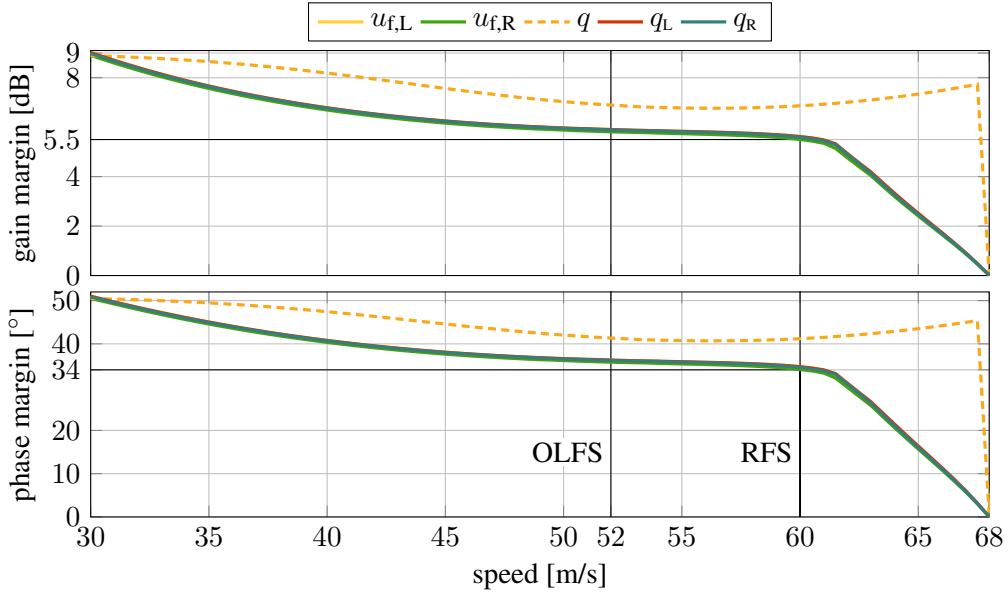


Figure 5.24: Loop-at-a-time margins of the closed flutter control loop.
(OLFS: open-loop flutter speed, RFS: robust flutter speed)

the margins are nearly the same. Since the q channel is shared by the two control loops, it shows greater robustness. The loop-at-a-time margins have acceptable values up to 60 m/s where for all channels, the gain and phase margins are 5.5 dB and 34° respectively. The margins also degrade slowly after 60 m/s therefore there is a sufficiently wide safety region around 60 m/s. In view of these facts, the robust flutter speed, i.e., the speed at which the demonstrator may fly safely, is determined to be 60 m/s. The loop-at-a-time margins became zero at 68 m/s which is the absolute flutter speed already established in Section 5.5.2.

The multi-loop-margins of the closed-loop are depicted in Figure 5.25. These margins also start to significantly degrade after 60 m/s. The most notable is the simultaneous input/output margin for which all five uncertainties are allowed to vary at the same time. At the robust flutter speed, the simultaneous input/output margin is 2.2 dB and 14° . The rate of degradation for these margins is similar to the loop-at-a-time margins. This confirms the robust flutter speed to be 60 m/s.

The critical frequency corresponding to a loop margin is the frequency where the Nyquist curve of the open loop touches the critical point (-1 or 0). The critical frequency for all margins and for all channels are less than 65 rad/s. This ensures that these robustness measures are meaningful because it shows that the perturbation causes instability in the low frequency range where the accuracy of the modeling is acceptable. At higher frequencies, any model is more uncertain therefore larger gain and phase margin are required.

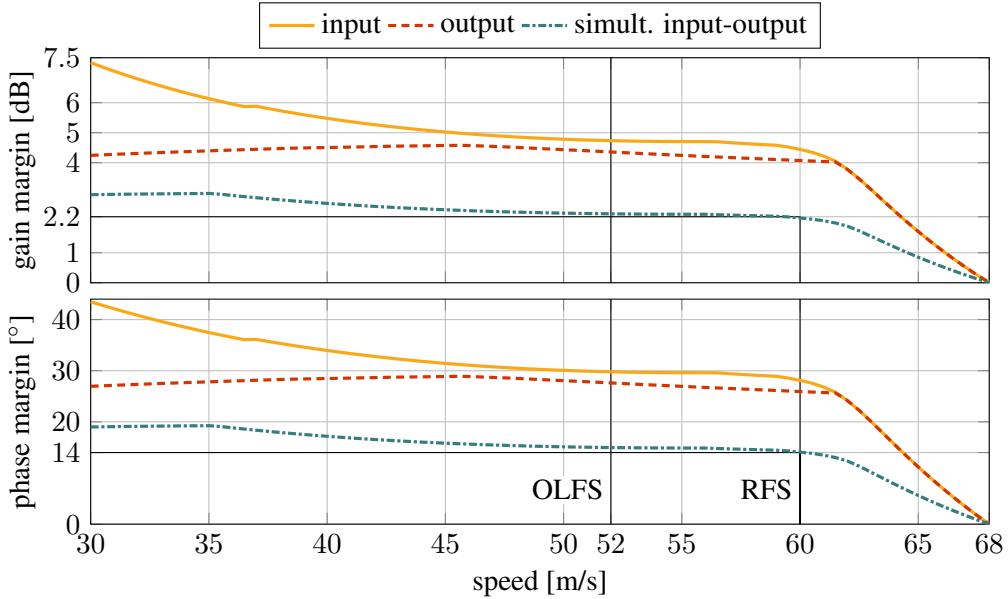


Figure 5.25: Multi-loop margins of the closed flutter control loops.
(OLFS: open-loop flutter speed, RFS: robust flutter speed)

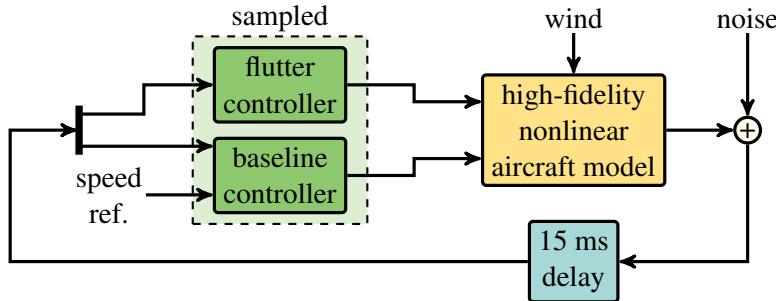


Figure 5.26: Nonlinear simulation setup.

5.5.4 Time domain simulations

Two time domain simulations are conducted to verify the frequency domain results in Sections 5.4, 5.5.2, and 5.5.3. In both cases, the baseline controller described in Section 5.5.1 and the flutter controller from Section 5.3 are operating at the same time. The setup for both simulations is depicted in Figure 5.26. The controllers are implemented in discrete time (with 200 Hz sampling frequency) and 15 ms output delay is included to simulate the delay present in the physical system. Gaussian measurement noise is added to the plant output. The only reference command input of the baseline controller used is the speed reference to demonstrate the behavior of the closed-loop at different speed values.

In the first simulation, the speed is increased starting at 46 m/s and following a staircase reference. No wind disturbance is used in this simulation. The vertical acceleration from

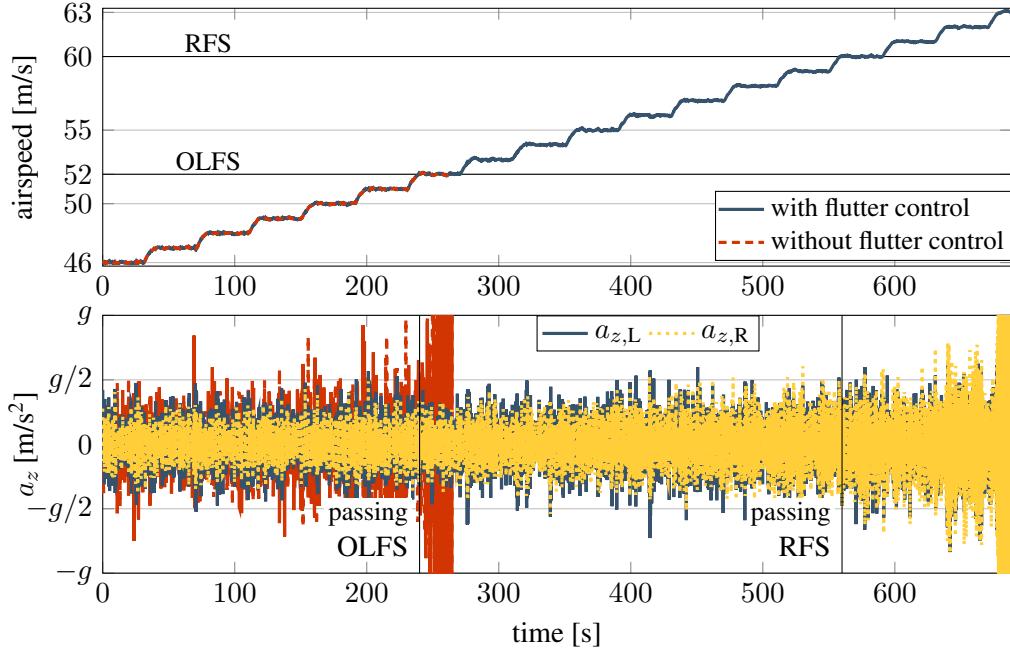


Figure 5.27: Simulation results of the flexible aircraft following a staircase speed reference with and without flutter control applied. On the bottom diagram, $g = 9.81 \text{ m/s}^2$. (OLFS: open-loop flutter speed, RFS: robust flutter speed)

the two IMUs on the wing in Figure 5.3 is recorded to demonstrate the load caused by the oscillations. These are denoted by $a_{z,L}$ and $a_{z,R}$ for the left and right wing respectively. The results are illustrated in Figure 5.27. In the graph on the top, the airspeed is depicted as a function of time while the graph on the bottom shows the vertical acceleration. If the flutter suppression controller is not activated and the aircraft passes the open-loop flutter speed, the acceleration of the wingtip grows beyond acceptable values indicating structural failure. With the flutter controller however, the demonstrator safely passes the open-loop flutter speed. The wingtip acceleration values are below critical at the robust flutter speed as well. The oscillations become damaging at 63 m/s, i.e., 3 m/s above the robust flutter speed. This is achieved with $|u_{f,L}| < 0.5^\circ$ and $|u_{f,R}| < 0.5^\circ$.

In the second simulation, we test whether it is indeed possible to fly at the robust flutter speed even in the presence of continuous Dryden wind gusts. The aircraft is accelerated from 43 m/s to 60 m/s and then it follows a constant 60 m/s speed reference command. This is illustrated in the top diagram of Figure 5.28. Without flutter control, the wing tip accelerations become critical a few seconds after the open-loop flutter speed was passed. When the flutter controller is used, the robust flutter speed is reached and acceptable acceleration values are maintained even if the wind gusts continually push the speed beyond the robust flutter speed. This demonstrates the robustness of the flutter controller discussed in Section 5.5.3. The control input is depicted in Figure 5.29. The control surface deflections increase when the open-loop and the robust flutter speed is passed. For the entire simulation, the control input is within $\pm 1.5^\circ$ bounds.

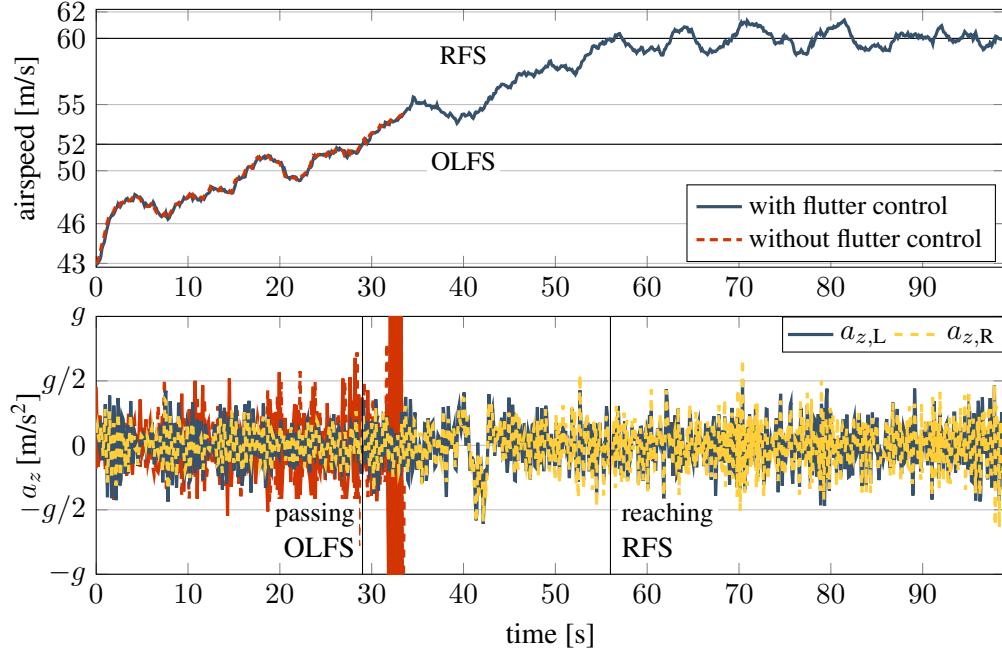


Figure 5.28: Simulation results of the flexible aircraft accelerating to the RFS with and without flutter control applied. On the bottom diagram, $g = 9.81 \text{ m/s}^2$. (OLFS: open-loop flutter speed, RFS: robust flutter speed)

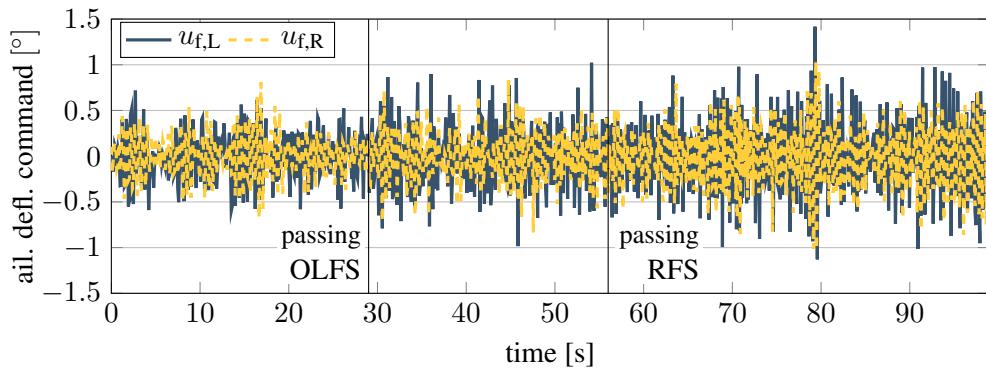


Figure 5.29: Control input in the simulation in Figure 5.28.

5.5.5 Comparison to potential alternatives

The results achieved using the structured synthesis method in Chapter 3 are compared to two alternative methods. First, structured H_∞ synthesis (`hinfstruct`) is employed on a single LTI system. This is obtained by setting the uncertain parameters to their nominal values in the design model in Section 5.3.1, and extracting the dynamic uncertainty block using LFT. The resulting controller can only increase the damping of the flutter modes at low speeds. This design consistently fails to extend the robust flutter speed beyond the open-loop flutter speed even if it is attempted for different values of the uncertain parameter describing the variation in the airspeed (δ_V).

A variation of Algorithm 1 is also used to design a flutter controller. In this version, instead of the D-K iteration in Steps 5-8, a single parameterized D-scale is optimized together with the controller, similarly to the hybrid relaxation method in [2]. Note that this is not a complete recreation of the hybrid relaxation approach since there are further differences between the two algorithms as pointed out throughout Chapter 3. The D-scales are defined to be state-space systems with state order equal to the number of basis functions in Section 5.3.3 (i.e., 5 for the synthesis of $K_{\text{asym}}(s)$ and 9 for $K_{\text{sym}}(s)$). The 'companion' parameterization option of the `tunableSS` function is chosen for both. The performance setup and controller structure is identical to what is described in Section 5.3.2. The resulting controller guarantees 2 dB simultaneous input-output gain margin and 13° phase margin up to 60 m/s. Comparing these values to the 2.2 dB and 14° obtained when using Algorithm 1 reveals the advantage of having separate D-scales for the samples in the synthesis.

Thesis 3: I propose a flutter suppression controller design technique for a flexible aircraft. In this approach, the aircraft dynamics are described by both a high fidelity and a reduced-order model. Based on the latter, an uncertain design model is constructed that takes the variations in the flutter modes and the dynamics neglected by the model reduction into account. The synthesis objective is robust stabilization by the minimization of the sensitivity function. This is achieved by the structured design method in Thesis 1. This yields a single, low order, linear time invariant controller suitable for implementation on the flight control computer aboard the aircraft. Robustness analysis applying the worst-case uncertainty construction technique in Thesis 2, as well as disk margin analysis and time-domain simulations based on the high fidelity model are provided to demonstrate that the controller expands the safe flight envelope by 15%.

Related publications: [P47, P48, P69, P46]

6

Conclusions

The present thesis details three main points. First, a novel structured optimal control design algorithm is proposed that minimizes the worst-case gain of systems with mixed uncertainty. The method samples the parametric uncertainty and employs a D-K iteration to obtain a controller for the samples containing dynamic uncertainty. In contrast to the hybrid relaxation in [2], separate D-scales are chosen for the different samples, potentially reducing conservatism. A convex optimization-based technique was presented for the construction of the scalings. The advantage of the individual D-scales over a single one was illustrated using a numerical example, and the flutter suppression control design problem. A possible future direction for this algorithm is its extension to linear parameter varying systems.

Secondly, a worst-case perturbation construction method is provided in this thesis for systems with mixed uncertainty. As opposed to the classical approach, this technique maximizes the gain of the system at multiple frequency points simultaneously. A non-linear multi-dimensional interval search is combined with a gradient ascent algorithm to find the value of the uncertain parameters that maximizes the sum of the largest singular values of the system at the given frequencies. The corresponding complex matrix samples of the dynamic uncertainty are interpolated using the BNP theorem. The resulting uncertainty is stable, has the correct block structure and is norm bounded by one. The example of a hard disk drive control system is presented to demonstrate that the method produces an uncertainty sample which elicits bad performance in a time domain simulation.

This method can be extended in several ways in future research. It can be useful as part of control design methods that require bad samples of the uncertainty, such as the robust synthesis algorithm. Selecting uncertainty samples which provide large gain at multiple frequencies could improve the convergence speed of such synthesis methods. An alternative objective is to consider the H_2 norm. The maximization of the H_2 norm increases the gain of the system over all frequencies which could degrade time-domain performance further. The inclusion of a nonlinear element is also a worthwhile direction as indicated by [66]. The possible replacement of the cube splitting algorithm that finds the worst-case value of uncertain parameters by simulated annealing, pattern search, or the Nelder-Mead simplex method is also the subject of future research.

The third main point of the thesis is the design and analysis of a flutter suppression controller for the flexible aircraft T-FLEX. The robust stabilization problem is articulated as the minimization of the sensitivity function and control effort for two SISO loops. The controller is synthesized employing the structured robust synthesis method in Chapter 3. The resulting controller is low order and LTI therefore easy to implement on the

on-board computer of the aircraft. Preliminary analysis using the multi-frequency worst-case uncertainty construction method reveals that the closed-loop with the uncertain design model produces acceptable performance. In further investigation, disk margins are computed using the baseline controller and the high fidelity model of the aircraft. Based on these margins and on time domain simulations, the open-loop flutter speed of 52 m/s is extended to 60 m/s which is a 15% increase of the flight envelope.

The next future step of the validation process is subjecting the flutter controller to hardware-in-the-loop simulations. If the controller still provides sufficient robust stabilization, it is planned to be flight tested eventually. Experimental data could prove valuable in further fine tuning the performance articulation and the synthesis process.

Acknowledgements

First of all, I express my profound gratitude to Peter Seiler. Despite the fact that he was not my formal supervisor, he took considerable time and effort to teach me key notions and to help me improve in several ways. His guidance has proven vital in all areas relevant in scientific research: in mathematics, in coding, in writing papers, and in presenting talks. His ideas and views have been deeply influential over the course of our work together. I am grateful for his advices and for our many discussions over the years that lead to the results in Chapters 3 and 4. I am also thankful for his gracious hospitality during my visits to the United States.

I thank my supervisor, Bálint Vanek, for his assistance in the research that is presented in this thesis. I acknowledge Tamás Luspay for his comments on several of my papers, presentations, posters, and on Chapter 2. I thank György Lipták for his help in the flutter control design problem that gave considerable momentum to my work and ultimately lead to the results in Chapter 5. I am grateful to Béla Takarics for constructing the model of the T-FLEX aircraft that made Chapter 5 possible. He also regularly helped me with using the model in my work.

I also thank Bálint Kiss, my supervisor during my Bachelor and Masters studies, and co-author of my first papers. Even though he was not directly involved in my PhD research, our previous work together provided me with experience and skills that have proven valuable here as well.

Bibliography

Publications of the author

- [P13] Laurent Clos-Cot, Julien Templai, Bálint Vanek, and Bálint Patartics. “Vision-based control and guidance structure for automatic landing simulations”. In: *2018 IEEE Aerospace Conference*. 2018, pp. 1–15.
- [P44] Bálint Patartics and Bálint Kiss. “The application of unknown input estimators to damp load oscillations of overhead cranes”. In: *Hungarian Journal of Industry and Chemistry* 42.2 (2014), pp. 71–77.
- [P45] Bálint Patartics, Yağız Kumtepe, Béla Takarics, and Bálint Vanek. “On the necessity of flexible modelling in fault detection for a flexible aircraft”. In: *IFAC-PapersOnLine* 54.20 (2021). Modeling, Estimation and Control Conference MECC 2021. Video presentation: <https://www.youtube.com/watch?v=iFiW7CyKkrI>, pp. 663–668.
- [P46] Bálint Patartics, György Lipták, Tamás Luspay, Peter Seiler, Béla Takarics, and Bálint Vanek. “Application of structured robust synthesis for flexible aircraft flutter suppression”. In: *IEEE Transactions on Control Systems Technology* 30.1 (2021), pp. 311–325. URL: <https://github.com/Patartics-Balint/wcunc>.
- [P47] Bálint Patartics, Tamás Luspay, Tamás Péni, Béla Takarics, Bálint Vanek, and Tiemo Kier. “Parameter varying flutter suppression control for the BAH jet transport wing”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 8163–8168.
- [P48] Bálint Patartics, Tamás Péni, and Bálint Vanek. “Flexibilis Szárnyú Repülőgép Aktív Flatter Szabályozása”. In: *Repüléstudományi Közlemények* 30.1 (2018), pp. 99–110.
- [P49] Bálint Patartics, Peter Seiler, Joaquin Carrasco, and Bálint Vanek. “Construction of a destabilizing nonlinearity for discrete-time uncertain Lurye systems”. In: *IEEE Control Systems Letters* 6 (2022), pp. 2605–2610.
- [P50] Bálint Patartics, Peter Seiler, Béla Takarics, and Bálint Vanek. “Worst-Case Uncertainty Construction via Multifrequency Gain Maximization With Application to Flutter Control”. In: *IEEE Transactions on Control Systems Technology* (2022).

- [P53] Bálint Patartics, Bálint Vanek, and Peter Seiler. “Construction of an Uncertainty to Maximize the Gain at Multiple Frequencies”. In: *2020 American Control Conference (ACC)*. Video presentation: https://www.youtube.com/watch?v=_uZ34faCVcU. IEEE. 2020, pp. 2643–2648.
- [P54] Bálint Patartics, Bálint Vanek, and Peter Seiler. “Structured Robust Synthesis with Parameter-Dependent D-Scales”. In: *2019 American Control Conference (ACC)*. IEEE. 2019, pp. 1800–1805.
- [P55] Bálint Patartics and Bálint Kiss. “Application of laser slot sensors for the state estimation of overhead cranes”. In: *2015 20th International Conference on Process Control (PC)*. IEEE. 2015, pp. 387–392.
- [P69] Béla Takarics, Bálint Patartics, et al. “Active flutter mitigation testing on the FLEXOP demonstrator aircraft”. In: *Proceedings of the AIAA Scitech 2020 Forum*. 2020, p. 1970.

References

- [1] Daniel Abramovitch and Gene Franklin. “A brief history of disk drive control”. In: *IEEE Control Systems Magazine* 22.3 (2002), pp. 28–42.
- [2] Raquel Stella da Silva de Aguiar, Pierre Apkarian, and Dominikus Noll. “Structured Robust Control Against Mixed Uncertainty”. In: *IEEE Transactions on Control Systems Technology* (2017).
- [3] Abdullah Al Mamun, GuoXiao Guo, and Chao Bi. *Hard disk drive: mechatronics and control*. CRC press, 2017.
- [4] P. Apkarian. “Tuning controllers against multiple design requirements”. In: *Proceedings of the 2013 American Control Conference*. 2013, pp. 3888–3893.
- [5] P. Apkarian, V. Bompard, and D. Noll. “Non-smooth structured control design with application to PID loop-shaping of a process”. In: *International Journal of Robust and Nonlinear Control* 17 (2007), pp. 1320–1342.
- [6] Pierre Apkarian. “Tuning controllers against multiple design requirements”. In: *2012 16th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE. 2012, pp. 1–6.
- [7] Pierre Apkarian and Dominikus Noll. “Nonsmooth H_∞ Synthesis”. In: *IEEE Trans. on Automatic Control* 51.1 (2006), pp. 71–86.
- [8] Pierre Apkarian, Dominikus Noll, and Alexandre Falcoz. *How we saved the Rosetta mission*. (Accessed on 03. 2022.) 2017. URL: <http://pierre.apkarian.free.fr/papers/rosettaProbe.pdf>.
- [9] Gary Balas, Richard Chiang, Andy Packard, and Michael Safonov. *Robust Control Toolbox User’s Guide R2014b*. 2016.
- [10] Joseph A Ball, Israel Gohberg, and Leiba Rodman. *Interpolation of Rational Matrix Functions*. Springer Basel AG, 1990.
- [11] Randal W Beard and Timothy W McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.

- [12] Richard P Braatz, Peter M Young, John C Doyle, and Manfred Morari. “Computational complexity of μ calculation”. In: *IEEE Transactions on Automatic Control* 39.5 (1994), pp. 1000–1002.
- [14] Brian P Danowsky, Aditya Kotikalpudi, David Schmidt, Chris Regan, and Peter Seiler. “Flight testing flutter suppression on a small flexible flying-wing aircraft”. In: *2018 Multidisciplinary Analysis and Optimization Conference*. 2018, p. 3427.
- [15] Stéphane Delannoy. “Design and validation of the A350 flexible structure control law”. In: *Oral presentation at the 2015 EURO-GNC* (2015).
- [16] John C Doyle, Bruce A Francis, and Allen R Tannenbaum. *Feedback control theory*. Courier Corporation, 1990.
- [17] Flight Phase Adaptive Aero-Servo-Elastic Aircraft Design Methods (FliPASED). *Horizon 2020 research and innovation programme of the European Union, grant agreement No 815058*. 2019. (Accessed on 03. 2022.) URL: <https://flipased.eu/>.
- [18] Flutter Free Flight Envelope eXpansion for ecOnomical Performance improvement (FLEXOP). *Horizon 2020 research and innovation programme of the European Union, grant agreement No 636307*. 2015. (Accessed on 11. 2019.) URL: <https://flexop.eu/>.
- [19] Hugo Fournier, Paolo Massioni, Minh Tu Pham, Laurent Bako, Robin Verney, and Michele Colombo. “Robust gust load alleviation of flexible aircraft equipped with lidar”. In: *Journal of Guidance, Control, and Dynamics* 45.1 (2022), pp. 58–72.
- [20] Pascal Gahinet and Pierre Apkarian. “Decentralized and fixed-structure H_∞ control in MATLAB”. In: *50th IEEE Conference on Decision and Control and European Control Conference*. 2011, pp. 8205–8210.
- [21] R. J. Guyan. “Reduction of stiffness and mass matrices”. In: *AIAA Journal* 3.2 (1965), pp. 380–380.
- [22] Simon Hecker, Andras Varga, and Jean-François Magni. “Enhanced LFR-toolbox for MATLAB”. In: *Aerospace Science and Technology* 9.2 (2005), pp. 173–180.
- [23] Rod Holland, Peter Young, and Chuanjiang Zhu. “Development of a skew μ lower bound”. In: *International Journal of Robust and Nonlinear Control* 15.11 (2005), pp. 495–506.
- [24] Rod Holland, Peter Young, and Chuanjiang Zhu. “Development of a skew μ upper bound”. In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 15.18 (2005), pp. 905–921.
- [25] M. Honda. “Temperature Dependent Robust Control of Hard Disk Drives Using Parameter Varying Techniques”. PhD thesis. University of Minnesota, Twin Cities, 2016.
- [26] The Mathworks Inc. *Robust Control Toolbox*. 2018.

- [27] Rudolf E Kalman. “Canonical structure of linear dynamical systems”. In: *Proceedings of the National Academy of Sciences of the United States of America* 48.4 (1962), p. 596.
- [28] A. Kotikalpudi. “Robust Flutter Analysis for Aeroservoelastic Systems”. PhD thesis. University of Minnesota, Twin Cities, 2017.
- [29] Huibert Kwakernaak, Raphael Sivan, and Bjar N D Tyreus. “Linear optimal control systems”. In: (1972).
- [30] Constantino M Lagoa, Xiang Li, Maria Cecilia Mazzaro, and Mario Sznaier. “Sampling random transfer functions”. In: *Probabilistic and Randomized Methods for Design under Uncertainty*. Springer, 2006, pp. 331–363.
- [31] Tamás Luspay, Tamás Péni, István Gőzse, Zoltán Szabó, and Bálint Vanek. “Model reduction for LPV systems based on approximate modal decomposition”. In: *International Journal for Numerical Methods in Engineering* 113.6 (2018), pp. 891–909.
- [32] Jan Marian Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.
- [33] Duane T McRuer, Dunstan Graham, and Irving Ashkenas. *Aircraft Dynamics and Automatic Control*. Vol. 740. Princeton University Press, Princeton, NJ, 2014.
- [34] Yasser M. Meddaikar, Johannes Dillinger, et al. “Aircraft Aeroservoelastic Modelling of the FLEXOP Unmanned Flying Demonstrator”. In: *Proceedings of the AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, 2019.
- [35] Claudia Moreno, Abhineet Gupta, Harald Pfifer, Brian Taylor, and Gary Balas. “Structural Model Identification of a Small Flexible Aircraft”. In: *Proceedings of the 2014 American Control Conference*. 2014, pp. 4379–4384.
- [36] Blaise G Morton. “New applications of mu to real-parameter variation problems”. In: *1985 24th IEEE Conference on Decision and Control*. IEEE. 1985, pp. 233–238.
- [37] Diego Navarro-Tapia, Andres Marcos, Pedro Simplício, Samir Bennani, and Christophe Roux. “Legacy recovery and robust augmentation structured design for the VEGA launcher”. In: *International Journal of Robust and Nonlinear Control* 29.11 (2019), pp. 3363–3388.
- [38] Diego Navarro-Tapia, Andrés Marcos, and Samir Bennani. “The VEGA launcher atmospheric control problem: A case for linear parameter-varying synthesis”. In: *Journal of the Franklin Institute* 359.2 (2022), pp. 899–927.
- [39] D. Ossmann, T. Luspay, and B. Vanek. “Baseline Flight Control System Design for an Unmanned Flutter Demonstrator”. In: *Proceedings of the 2019 IEEE Aerospace Conference*. IEEE, 2019.
- [40] A. Packard and J. Doyle. “The complex structured singular value”. In: *Automatica* 29.1 (1993), pp. 71–109.

- [41] A. Packard, J. Doyle, and G. Balas. “Linear, Multivariable Robust Control With a μ Perspective”. In: *Transactions of the ASME* 115 (1993), pp. 426–438.
- [42] Andy Packard, Gary Balas, Richard Liu, and Jong-Yeob Shin. “Results on worst-case performance assessment”. In: *American Control Conference*. Vol. 4. 2000, pp. 2425–2427.
- [43] Andy Packard, Michael KH Fan, and John C Doyle. *A Power Method for the Structured Singular Value*. Tech. rep. 1988.
- [51] Bálint Patartics, Peter Seiler, and Bálint Vanek. *GitHub repository of the MATLAB implementation of the multi-frequency worst-case uncertainty construction algorithm*. (Accessed on 07. 2020.) URL: <https://github.com/Patartics-Balint/wcunc>.
- [52] Bálint Patartics, Peter Seiler, and Bálint Vanek. *GitHub repository of the MATLAB implementation of the structured synthesis algorithm*. (Accessed on 11. 2019.) URL: <https://github.com/Patartics-Balint/wcgmin>.
- [56] Manuel Pusch and Daniel Ossmann. “ H_2 -Optimal Blending of Inputs and Outputs for Modal Control”. In: *IEEE Transactions on Control Systems Technology* (2019), pp. 1–8.
- [57] Manuel Pusch and Daniel Ossmann. “ H_2 -Optimal Blending of Inputs and Outputs for Modal Control”. In: *IEEE Transactions on Control Systems Technology* (2019), pp. 1–8.
- [58] Anders Rantzer. “On the Kalman-Yakubovich-Popov lemma”. In: *Systems & Control Letters* 28.1 (1996), pp. 7–10. ISSN: 0167-6911. DOI: [https://doi.org/10.1016/0167-6911\(95\)00063-1](https://doi.org/10.1016/0167-6911(95)00063-1).
- [59] Syed Z. Rizvi, Javad Mohammadpour, Roland Tóth, and Nader Meskin. “A Kernel-Based PCA Approach to Model Reduction of Linear Parameter-Varying Systems”. In: *IEEE Transactions on Control Systems Technology* 24.5 (2016), pp. 1883–1891.
- [60] Christian Roessler, Philipp Stahl, et al. “Aircraft Design and Testing of FLEXOP Unmanned Flying Demonstrator to Test Load Alleviation and Flutter Suppression of High Aspect Ratio Flexible Wings”. In: *Proceedings of the AIAA Scitech 2019 Forum*. AIAA, 2019.
- [61] Clément Roos. “Systems Modeling, Analysis and Control (SMAC) toolbox: An insight into the robustness analysis library”. In: *IEEE Conference on Computer Aided Control System Design*. 2013, pp. 176–181.
- [62] Jack J Ryan and John T Bosworth. “Current and future research in active control of lightweight, flexible structures using the X-56 aircraft”. In: *52nd Aerospace Sciences Meeting*. 2014, p. 0597.
- [63] Carsten Scherer and Siep Weiland. *Linear Matrix Inequalities in Control*. Lecture notes. 2000.
- [64] D. K. Schmidt. *Modern Flight Dynamics*. ISBN 9780073398112. McGraw-Hill, 2012.

- [65] Peter Seiler. "Lightweight Adaptive Aeroelastic Wing for Enhanced Performance Across the Flight Envelope". In: *Year 3 Progress Report* (2017). Period Covered: August 2016 through August 2017.
- [66] Peter Seiler and Joaquin Carrasco. "Construction of periodic counterexamples to the discrete-time Kalman conjecture". In: *IEEE Control Systems Letters* 5.4 (2020), pp. 1291–1296.
- [67] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*. Vol. 2. Wiley New York, 2007.
- [68] Bela Takarics, Balint Vanek, Aditya Kotikalpudi, and Peter Seiler. "Flight control oriented bottom-up nonlinear modeling of aeroelastic vehicles". In: *Proceedings of the 2018 IEEE Aerospace Conference*. IEEE, 2018.
- [70] J. Theis, H. Pfifer, and P. Seiler. "Robust Control Design for Active Flutter Suppression". In: *AIAA Science and Technology Forum and Exposition*. 2016, Paper No. AIAA–2016–1751.
- [71] G. Vinnicombe. "Measuring Robustness of Feedback Systems". PhD thesis. Univ. Cambridge, Cambridge, 1993.
- [72] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Pearson, 1995.
- [73] Kemin Zhou, John Comstock Doyle, and Keith Glover. *Robust and optimal control*. Vol. 40. Prentice hall New Jersey, 1996.