Data-driven Fixed-Structure Controller Design for Linear Parameter-Varying Systems

A Frequency-Domain Approach

A. Khalik

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Data-driven Fixed-Structure Controller Design for Linear Parameter-Varying Systems

A Frequency-Domain Approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

A. Khalik

Supervisors:
dr. ir. R. Tóth
ir. T.A.H Bloemers

October 18, 2020

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Copyright © Delft Center for Systems and Control (DCSC)
All rights reserved.

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Data-driven Fixed-Structure Controller Design for Linear Parameter-Varying Systems

by

A. Khalik

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: October 18, 2020

Supervisor(s):

dr.ir. R. Toth

ir. T.A.H Bloemers

Reader(s):

prof.dr.ir. T. Keviczky

dr.ir. S. Wahls

Abstract

With data-driven control it is possible to design a controller for systems with non-parametric models. The intermediate step of modelling or identification of the system is not necessary, because a non-parametric model of the system can be obtained by means of experimental data. In the linear time-invariant (LTI) framework, these non-parametric models are well defined in the frequency-domain, they are represented as frequency response functions (FRFs). Data-driven control in the frequency-domain has gained momentum in the past decades, but still relatively few methods have been developed for multiple-input-multiple-output (MIMO) systems. One of the objectives of this thesis is to develop a controller synthesis method for MIMO systems.

While the LTI framework has many advantages, such as the vast available literature and the relatively simple theory, it has its limitations. In reality, all systems have nonlinearities and some systems, especially position dependent systems (which are common in mechatronics), are less suitable to be modelled as LTI systems. The nonlinear dynamics are most likely interpreted as an uncertainty for which a robust controller has to be designed at the cost of performance. By taking into account the scheduling variable, in this case the position dependency, it is possible to improve the performance. To do so, the system is modelled as a linear parameter-varying (LPV) system for which an LPV controller is designed that takes the scheduling variable into account.

This thesis is concerned with developing a controller synthesis approach in the LPV framework using non-parametric models of MIMO systems using the local approach. This implies that an LPV controller is designed for the nonlinear system at the operating points of interest, at these points the system is assumed to exhibit LTI behaviour. What follows is an interpolation of the multiple LTI controllers to obtain a global parametrisation of the controller in the entire operating range, such that local stability and performance is guaranteed at every operating point.

Two novel data-driven LPV controller synthesis methods, based on LTI methods, are presented in this thesis. These methods improve the performance of a parameter dependent system compared to an LTI approach. The improvements are shown through simulations with nonlinear systems. From these simulations it can be concluded that the LPV controllers improve the performance compared to a similar LTI controller. Furthermore, an example is shown where an LPV controller is crucial to guarantee closed-loop stability of the nonlinear system.

Table of Contents

Acknowledgements

During my master thesis project I have learned a lot of new things in (robust) control theory. I would like to express my gratitude towards my TU Eindhoven supervisor Roland Tóth for giving me a chance to do a master thesis project that I have thoroughly enjoyed working on the past year. I appreciate the great support and feedback from both Roland and my daily supervisor Tom Bloemers that I have gotten during my project.

I would also like to thank my TU Delft supervisor Sander Wahls and coordinator Ton van den Boom for allowing me to do my master thesis project in TU Eindhoven. During the first half year I noticed that it took quite some administrative work from both TU Delft and TU Eindhoven to make it all right. I am grateful that both parties took the time and effort to do that.

Delft, University of Technology A. Khalik
October 18, 2020

Chapter 1

Introduction

1-1 Background

Due to the competitive nature of the industry, performance specifications of mechatronic systems in terms of positioning speeds and accuracy are becoming increasingly demanding. As a result of these requirements, mechatronic systems are further developed and become progressively more complex. The non-linearities that emerge as a result of further development may have different forms for mechatronic systems, such as positional or orientation dependence. These non-linearities place limitations on the performance of the system. However, in general the property of any nonlinear systems is that they show linear time-invariant (LTI) behaviour around a certain operating point. Consequently, this allows for controller design in the LTI framework.

Data-driven and model-based control In general, we can roughly state that there are two ways to design a controller: either data-driven control, which includes heuristic tuning methods, or model-based control. Data-driven control requires a non-parametric model, which can be obtained by means of experimental data. Model-based control requires first-principles modelling or identification of the system. Besides the distinction between data-driven or model-based control, one can also choose to design a fixed-structure or unstructured controller in either the frequency-domain or the time-domain. In my internship project the underlying goal was to get familiar with fixed-structure data-driven control design in the frequency-domain. To understand the benefits of data-driven control in the frequency-domain, in the internship report [1], the differences between data-driven and model-based control in the time-domain or frequency-domain were studied, along with the benefits of fixed-structure controllers. Based on the results of the internship report, the motivation follows for data-driven fixed-structure controllers in the frequency domain.

Motivation Data-driven control has the advantage that no intermediate identification or modelling steps are necessary to design a controller. In [2] it is stated that model-based

control methods are inherently less robust due to unmodeled dynamics, these parametric uncertainties are irrelevant for data-driven control methods and the only source of uncertainty comes from the measurement process [3]. Furthermore, since data is constantly stored in industrial processes, it is relatively easy and cheap to obtain a non-parametric description of the system [4]. The reasons why we prefer to design in the frequency-domain over the time-domain is because the dynamics of the system can be obtained through a frequency response function (FRF), without any information on the transfer function. The FRFs can be viewed in both the Nyquist and Bode plot, these plots are used to determine stability and performance specifications of the system. Furthermore, the open-loop frequency response can be used to determine closed-loop stability, for example, by using the Nyquist plot. Finally, frequency-domain analysis of nonlinear systems is also possible by using FRFs. The motivation for fixed-structure controllers is due to the fact that they are typically of lower order and therefore relatively easier to implement in practice than unstructured controllers, such as higher order $\mathcal{H}_\infty$ controllers. However, it is well known that fixed-structure controllers suffer from non-convexity of the synthesis, resulting in serious optimization problems with multiple local minima.

State-of-the-art LTI control design methods An overview of state-of-the-art methods in the LTI framework helps us to see what is available and what the shortcomings are in terms of frequency-domain data-driven control design. A novel robust data-driven fixed-order controller design method that satisfies the $\mathcal{H}_\infty$ criterion for LTI single-input-single-output (SISO) systems is proposed in [5]. Necessary and sufficient conditions for the existence of a controller, which is linearly parametrized using basis functions, are presented by a set of convex constraints. The main advantage of this method is that the optimization problem is convex, therefore a global optimum can be found. The shortfalls are that for a lower order controller the results are highly dependent on choice of basis functions and the method is only applicable to SISO systems. Lower order controllers, typically fixed-structure controllers, such as a PID, have the advantage of being relatively easy to describe and retune once implemented in the system. Therefore, to improve on the shortfalls of [5], an approach based on the same theory has been formulated in [6] to synthesise fixed-structure controllers. Consequently, the optimization problem becomes non-convex. Several data-driven optimization algorithms, such as particle swarm optimization (PSO), are proposed to find local minima of the optimization problem. A data-driven controller synthesis method that satisfies the Nyquist stability criterion and achieves $\mathcal{H}_\infty$ performance specifications is used by ASML. Stability of the closed-loop system is guaranteed by evaluating the encirclements of the loop transfer function around the critical point. The disadvantage of the method is that prior knowledge of the system, such as number of open right half-plane poles, is required to apply the Nyquist stability criterion. A more recent approach which uses FRFs to optimize tunable controller parameters of linear parametrizable controllers is proposed in [7]. The generalized Nyquist criterion is used to impose stability and $\mathcal{H}_\infty$ performance requirements on the closed-loop system. An advantage of this method is that it is applicable to multiple-input-multiple-output (MIMO) systems, since it is in generalized plant form. This form is exploited to absorb the controller structure inside the plant, which results in a diagonal matrix with the controller parameters. The disadvantages of this method are that the synthesis is non-convex, only stable systems and controllers are permitted and finally, the constraints are conservative. The constraints prohibit the Nyquist curve from making encirclements around the origin, while that may

be necessary in case there is a right half-plane pole present in the system that has to be cancelled. For more data-driven controller design methods in the frequency-domain see for example [8, 9, 10].
In summary, the state-of-the-art data-driven control design methods are able to synthesise fixed-structure controllers using only FRFs. However, designing for MIMO systems is clearly still a challenge. The one method [7] that is able to design for MIMO systems has a number of serious limitations that have yet to be overcome. While the objectives with regards to stability and performance specifications remain the same in the MIMO case as in the SISO case, the relations become more complex. Deriving conditions for stability and performance is not as easy in the MIMO setting, because one has to take matrix computations into account, which are not as trivial as scalar computations for SISO systems.

1-2 Project motivation

Limitations of LTI framework So far, we discussed methods of the LTI framework, because an LTI controller is an attractive option in practice due to its simplicity in terms of design and implementation. Furthermore, robust and optimal control techniques are well developed with easy to apply design schemes and they provide reliable performance in practice based on many successful applications reported in the literature. However, there is a need to meet with increasing accuracy and efficiency of processes, but the LTI framework has its limitations. In practice, nonlinear dynamics, such as position dependency, unavoidably need conservative designs to guarantee stability. By using robust control theory it is possible to stabilize such position dependent systems, but at the cost of performance. This is a challenge in the LTI framework that may be too difficult to solve.

Birth of LPV framework In the 1980s, it was suggested that instead of delving further into the research of nonlinear and time-varying systems, a model class should be developed that includes nonlinear and time-varying aspects and the existing LTI approaches should be able to be extended to this new model class [11]. As a result, the linear parameter-varying (LPV) framework was developed. LPV systems are nonlinear systems capable of describing systems with parameter-varying dynamics. An LPV system is characterized by a linear input-output (IO) map, similar to an LTI system. The difference is that a linear map in case of an LPV system depends on certain measurable time-varying variables, which are often referred to as scheduling variables. This type of modelling can be seen as an embedding of the nonlinear behaviour in the solution set of a linear structure [12]. An LPV model is either an approximation of a true non-linear system (in case of a local approach) or an embedding of the system (in case of a global approach), but either way, the modelling accuracy is improved compared to an LTI model. Furthermore, due to the linearity of LPV representations it is possible to extend LTI controller design techniques, such as robust or optimal controllers, allowing the design of an LPV controller (see e.g. [13, 14]). As it was mentioned before, LPV systems can be modelled in two ways: locally and globally. The latter is an embedding of nonlinear behaviour through a proxy description. In short, this implies that the nonlinear system is modelled as an LPV system that depends on either an external or internal signal which is referred to as the scheduling variable. The former interpolates local linear behaviours at a set of operating points, these are referred to as the scheduling variable. The local approach

is based on the design of local LTI controllers for a specified grid of operating points, resulting in a gain-scheduled controller. Through a global parametrization of the dependency of the controller on the operating points stability and performance can be guaranteed locally.

State-of-the-art LPV control design methods Over the past two decades, many methods have been developed for time-domain model-based LPV controller synthesis, in which state-space representations are used to describe system dynamics. Model-based LPV control design has many applications, such as in aerospace, automotive systems, diabetes control, the chemical industry and more (see e.g. [15, 16, 12, 17, 18]). Data-driven control design methods using input-output (IO) representations have been developed for LPV systems, such as the extension of the LTI VRFT method to the LPV framework by [19], but the method was limited because it was only applicable to slowly varying scheduling trajectories, the controller had to be linearly parametrized and an LTI reference behaviour was required. This work inspired a direct data-driven method proposed by [20] for optimization of LPV controller parameters without the need of a model of the system to be controlled. The aforementioned approach takes the scheduling variable into account, which enables the possibility to guarantee stability and performance in the operation region of interest.

The design of data-driven LPV controllers in the frequency-domain is by no means as much researched as the model-based methods, while there lies some potential in extending data-driven LTI approaches to the LPV framework. The local approach goes in conjunction with the gain-scheduling method. The aim is to design a gain-scheduled controller for a specified set of steady-state operating points of the system, these are referred to as the frozen dynamics of the system. The latter can be seen as an LTI system, hence the LPV system is assembled from frozen (LTI) FRF estimates. These frozen FRFs are used for fixed and non-fixed structure controller synthesis in the LPV framework. For example, a method is developed in [22] to directly synthesize a fixed-order LPV controller based on measurement data using the local approach. The downside is that only stable systems can be designed for and it is a conservative design method. In [23] the aforementioned disadvantages have been tackled. An LPV control design method inspired by [5] has been developed in the LPV framework using the local approach. The LTI stability and performance conditions have been reformulated such that the local approach can be used to design a fixed-order LPV controller. The downside of the aforementioned methods are that they are only applicable to SISO systems. Furthermore, it should be stressed that with the local approach, only local stability and performance guarantees are provided. So, since no extensive research has been done into data-driven fixed-structure controller tuning in the LPV setting with local stability and performance guarantees for MIMO systems it is worth investigating the possibilities.

1-3 Problem formulation & Main goal of thesis

Given a nonlinear MIMO system whose dynamics are unknown or too difficult to model. It is assumed that the system can be described as an LPV system, i.e. the scheduling variable that describes the change of the operating point is known or can be measured. Furthermore, it is assumed that FRF data of the system is available at an operating region, in other words, "frozen" FRFs at each respective given operating point are readily available. By only using measuring data, the goal is to design a MIMO fixed-structure controller, which depends on

the scheduling variable, such that at each operating point in the operating region of interest: local (internal) stability is guaranteed and local performance specifications are met for the closed-loop system without using an explicit model of the system. The main goal of this thesis is formulated below.

> *Develop an LPV data-driven control methodology that guarantees stability and performance locally for MIMO systems.*

During the internship project the state-of-the-art LTI control design methods were implemented and compared, an overview of the methods is given in [1]. Based on the preliminary investigations of the internship project and the discussion in Section 1-1, the following methods are discussed further in this thesis: the Feedback Auto Tuner (FBA) method by ASML, the $\mathcal{H}_\infty$ controller design by [5, 6] and the MIMO controller design by [7].

The main contributions of this thesis are the development of two novel MIMO LPV data-driven control design methods based on the method in [5] and the FBA by ASML. These two methods provide local stability and performance guarantees. Furthermore, compared to LTI controllers, these methods show improvements with respect to the performance of the closed-loop system.

1-4 Outline of the report

This thesis is organized as follows. Chapter 2 introduces the relevant basic elements of the data-driven LTI framework, these are the building blocks for the remainder of the thesis. Chapter 3 discusses three state-of-the-art methods from the literature. The workings of these methods are briefly explained and they are compared to each other, followed by a conclusion on which method should be extended to the LPV framework. Chapter 4 presents basic concepts of the LPV framework and gives an overview of what is available in the LPV framework in terms of controller design. In Chapter 5 two novel LPV controller synthesis methods are presented, these are based on the fixed-structure controller with $\mathcal{H}_\infty$ performance by [6] and the Feedback Auto Tuner by ASML. In Chapter 6 the novel LPV control design methods are demonstrated on a two mass-spring-damper system with a nonlinear spring. The LPV controllers are compared to each other and to their robust LTI counterparts. Finally, in Chapter 7 the conclusions and recommendations are stated.

Chapter 2

Preliminaries

The goal of this chapter is to provide an overview of the basic theory that is necessary to build up the theory for the remainder of this thesis.

2-1 System descriptions

The state-space (SS) representation is a common parametric description for a system. For a continuous linear time-invariant (LTI) system with p inputs, q outputs and n state variables the SS representation is given by:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{2-1}$$

where $x(t) \in \mathbb{R}^n$, $y(t) \in \mathbb{R}^q$, $u(t) \in \mathbb{R}^p$ and $\dot{x} = \frac{dx}{dt}$. The state matrix is A, B the input matrix, C the output matrix and D is the feed-through matrix, these all have appropriate dimensions according to the state, input and output signals. An SS description is not unique, whereas a Transfer Function (TF) representation is unique and it can be directly derived from the SS description by taking the Laplace transform of (2-1) and assuming a zero initial state $x(0) = 0$:

$$Y(s) = \underbrace{(C(sI - A)^{-1}B + D)}_{G(s)}U(s), \tag{2-2}$$

where $G(s)$ is the transfer function in case of a single-input-single-output (SISO) system and a transfer function matrix (TFM) if the system is multiple-input-multiple-output (MIMO). $G(s)$ is a matrix whose elements are real-rational functions of s which are proper. Furthermore, $U(s)$ and $Y(s)$ are Laplace transforms of u and y, defined on their corresponding region of convergence.

Definition 2.1. Consider a TFM $G(s)$:

$G(s)$ is *strictly proper* if $G(s) \to 0$ as $s \to \infty$.
$G(s)$ is *bi-proper* if $G(s) \to D \neq 0$ as $s \to \infty$.
$G(s)$ is *proper* if it is strictly proper or bi-proper.
$G(s)$ is *improper* if $G(s) \to \infty$ as $s \to \infty$.

2-1-1 Frequency Response Functions

A non-parametric description of a system is necessary for data-driven control. If a parametric model $G(s)$ of an LTI system is available in the form of a TF, the Frequency Response Function (FRF) can simply be acquired by evaluating s over $j\omega$ to get $G(j\omega)$. Otherwise, an FRF has to be directly measured from the system. This is typically done using a (combination) of sinusoidal waves at varying frequencies ω to measure the system's response. The output is a measure of the magnitude and phase of the system as a function of frequency. The advantages of using an FRF of a system are that they provide insight into benefits and trade-offs of feedback control [24]. Furthermore, if a system is unstable, measurements can be done in a closed-loop setting to obtain FRFs. FRF measurements can be plotted in a Bode or Nyquist plot, where the characteristics (such as resonance peaks) can be seen and stability can be evaluated, respectively.

2-1-2 Coprime Factorisation

Let $\mathcal{RH}_\infty$ represent the set of real rational proper and stable transfer functions that are bounded at infinity. $\mathcal{RH}_\infty$ is closed under addition and multiplication, i.e. if $N, M \in \mathcal{RH}_\infty$, then $\{N + M, NM\} \in \mathcal{RH}_\infty^{p\times q}$, where p and q are the inputs and outputs, respectively. The following definition for the coprime factorization is adopted from [25].

Definition 2.2. Let $G(s)$ be well-posed transfer function matrix. A left coprime factorization of $G(s)$ over $\mathcal{RH}_\infty$ is a factorization

$$G(s) = \tilde{M}(s)^{-1}\tilde{N}(s), \tag{2-3}$$

where $\tilde{N}(s), \tilde{M}(s) \in \mathcal{RH}_\infty$, such that $\tilde{M}(s)$ is invertible and there exist $Y, X \in \mathcal{RH}_\infty$ such that

$$\tilde{M}(s)X(s) - \tilde{N}(s)Y(s) = I. \tag{2-4}$$

A right coprime factorization

$$G(s) = N(s)M^{-1}(s) \tag{2-5}$$

is defined in the same manner. A factorization

$$G(s) = N(s)M^{-1}(s) = \tilde{M}(s)^{-1}\tilde{N}(s) \tag{2-6}$$

is a doubly coprime factorization of G over $\mathcal{RH}_\infty$ if (M, N) and $\tilde{M}, \tilde{N}$ are a right- and left-coprime factorization of G, respectively and there exist $X, Y, \tilde{X}, \tilde{Y} \in \mathcal{RH}_\infty$ such that

$$\begin{bmatrix} M & Y \\ N & X \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{X} & -\tilde{Y} \\ -\tilde{N} & \tilde{M} \end{bmatrix}. \tag{2-7}$$

Since coprime factorizations are not unique, they can be normalized such that, for example, the left coprime factorization is

$$\tilde{M}\tilde{M}^* + \tilde{N}\tilde{N}^* = I, \tag{2-8}$$

where $\tilde{M}^*$ denotes the complex conjugate transpose of $\tilde{M}$. The stability of $N(s)$ and $M(s)$ implies that $N(s)$ contains the zeros of $G(s)$ and the poles of $G(s)$ are contained in the zeros of $M(s)$. Due to the fact that N and M are coprime factors, there are no pole-zero cancellation when forming NM^{-1} [24].

2-1-3 Linear Fractional Representation and Transformations

Another useful system description, mostly used in robust control theory, are linear fractional representations (LFRs) for uncertain systems. The LFRs are used for LTI systems that are based on linear fractional transformations (LFTs). LFRs allow for manipulation of the uncertain system the same way as a standard LTI system. Furthermore, they allow for generalizations of plant interconnections. First, the definition for linear fractional transformations (LFTs) is given, which originates from [26].

Definition 2.3. Let $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \in \mathbb{C}^{p_1+p_2 \times (q_1+q_2)}$, and let the complex matrices $K \in \mathbb{C}^{q_2 \times p_2}$ and $\Delta \in \mathbb{C}^{q_1 \times p_1}$. A *lower* LFT with respect to K as the map

$$\mathcal{F}_\ell(P, \bullet) : \mathbb{C}^{q_2 \times p_2} \mapsto \mathbb{C}^{q_1 \times p_1}$$

with

$$\mathcal{F}_\ell(P, K) \triangleq P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \tag{2-9}$$

provided that the inverse $(I - P_{22}K)^{-1}$ exists. Similarly, the *upper* LFT can be defined with respect to Δ as

$$\mathcal{F}_u(\bullet, P) : \mathbb{C}^{q_1 \times p_1} \mapsto \mathbb{C}^{q_2 \times p_2}$$

with

$$\mathcal{F}_u(\Delta, P) \triangleq P_{22} + P_{21}\Delta(I - P_{11}\Delta)^{-1}P_{12} \tag{2-10}$$

provided that the inverse $(I - P_{11}\Delta)^{-1}$ exists.

Remark. The lower LFT $\mathcal{F}_\ell(P, K)$ also corresponds to $P \star K$ and the upper LFT $\mathcal{F}_u(\Delta, P)$ to $\Delta \star P$, where $\star$ is the Redheffer star product [26].

The terminology of *lower* and *upper* LFTs follow from the diagrams in Figure 2-1. The diagrams show general interconnections of control loops. An arbitrary interconnection can be represented similarly. The equations that represent the left diagram are given by

$$\begin{aligned} \begin{bmatrix} z_1 \\ y \end{bmatrix} &= \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w_1 \\ u \end{bmatrix} \\ u &= Ky, \end{aligned} \tag{2-11}$$

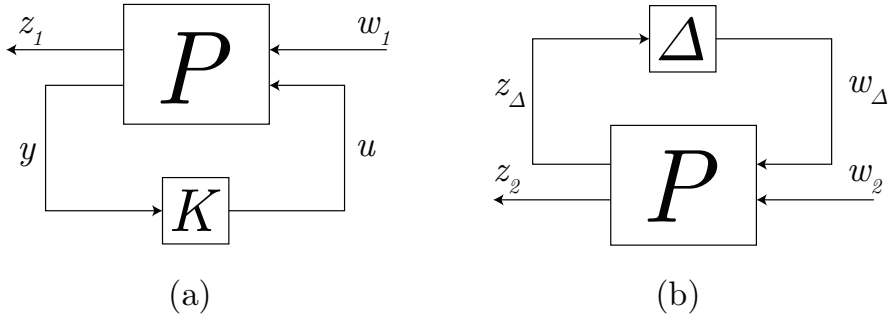

Figure 2-1: Diagram for the lower LFT (a) and the upper LFT (b).

where P is the controlled plant, K is the controller, z_1 is the performance channel, w_1 is the generalized disturbance, u denotes the control input and y is the measured output. In the same way, the equations that represent the right diagram are given as

$$\begin{bmatrix} z_\Delta \\ z_2 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w_\Delta \\ w_2 \end{bmatrix} \tag{2-12}$$
$$w_\Delta = \Delta z_\Delta,$$

where Δ is the uncertainty. An input-output description of the left figure can be found by working out (2-11) and then by eliminating the signals u, y in

$$z_1 = P_{11}w_1 + P_{12}u,\ y = P_{21}w_1 + P_{22}u,\ u = Ky$$

the closed-loop transfer function $w_1 \mapsto z_1$ can be found

$$z_1 = (P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21})w_1. \tag{2-13}$$

Notice that (2-9) and (2-13) describe the exact same transfer function. Hence, the physical interpretation of LFTs in control theory is that $\mathcal{F}_\ell(P, K)$ represents the closed-loop transfer function. The same procedure can be followed to find the map $w_2 \mapsto z_2$ for the closed-loop transfer function of the right diagram, or simply use the upper LFT. Finally, in accordance with the well-posedness condition for a feedback loop, where it is required that all closed-loop transfer functions are well-defined and proper, the following definition is given:

Definition 2.4. An LFT, e.g. $\mathcal{F}_\ell(P, K)$, is well-posed if $(I - P_{22}K)$ is invertible.

A parametric model of a system usually contains unmodeled dynamics. Rather than trying to improve the accuracy of the model, which leads to a more complex description, one can choose a simplified model and take the uncertainty into account using LFRs. This is a problem in model-based control design approaches, however, LFRs will also prove to be useful for LPV systems as well. Consider an uncertain LTI system shown in Figure 2-2a, it can be reformulated as an LTI uncertainty Δ and an LTI system (b). The new auxiliary output and input signals w_Δ and z_Δ, respectively, are due to "pulling out" the uncertainty Δ. Finally, the system and uncertainty are connected into a new uncertain system (c). The transfer function as "seen" by Δ is found using (2-12), it is simply P_{11}. The closed-loop interconnection of this uncertain system can be found using the general upper LFT formula in (2-10). In Chapter 4 linear parameter varying (LPV) system are introduced and LFRs can be used to model the system, where the scheduling variable can be seen as Δ.

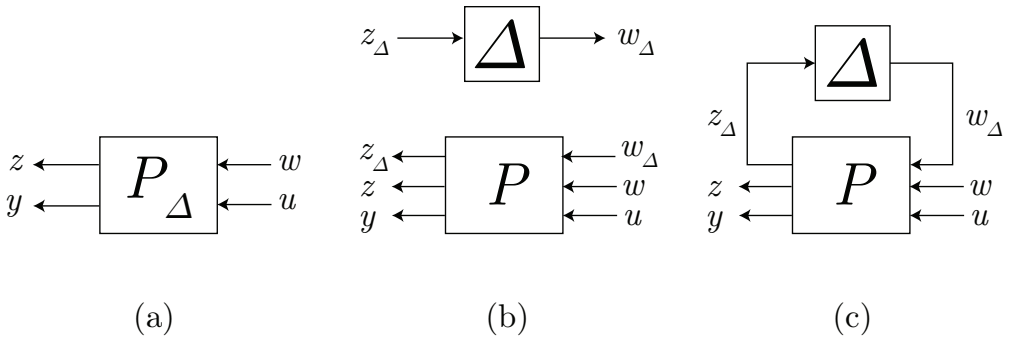

Figure 2-2: An uncertain system is reformulated in an LFR representation.

2-2 Stability

2-2-1 BIBO Stability

The transfer function and state-space representation can be used to asses stability. Let Σ be a system whose stability we want to determine. If the transfer function representation H is used, then the poles of H are used to asses the stability of the system by checking if all the poles are in the open left half-plane $\mathbb{C}_-$. Equivalently, a state-space representation (A, B, C, D) can be used, where the eigenvalues of a (minimal) A are evaluated to determine the stability of Σ. Thus, these notions of stability depend on the description of the system, nevertheless for LTI systems, these are equivalent to the bounded-input bounded-output stability (BIBO) properties.

Definition 2.5. A system is BIBO stable if a bounded input u results in a bounded output y:

$$\|u\| < \infty \implies \|y\| < \infty,$$

where $\|\bullet\|$ is a signal norm of choice.

The notion of stability depends on the norm that is chosen. For our purposes, the $\mathcal{L}_2$ and the $\mathcal{L}_\infty$-norm are the most interesting, since they represent the signal norms:

$$\|x(t)\|_p = \begin{cases} \sqrt{\int_0^\infty \|x(t)\|^2 \, dt}, & p = 2, \\ \sup\limits_{t>0} \|x(t)\|, & p = \infty, \end{cases} \tag{2-14}$$

where $\|x\|$ is the Euclidean norm $\sqrt{x(t)^T x(t)}$ of the vector $x(t)$, which simply describes the distance of $x(t)$ to the origin. The $\mathcal{L}_\infty$-norm corresponds to the maximum amplitude of the signal $x(t)$. Hence, using Definition 2.5, we can state that a transfer matrix $G(s)$ is BIBO stable if $\|u\|_\infty < \infty \implies \|y\|_\infty < \infty$. A similar statement on the stability of a system can be done using the $\mathcal{L}_2$-norm, which corresponds to the energy of the signal. The qualitative property of stability stays the same with both measures of size of the signals, however, the choice of the type of norm to measure the system's gain, does matter for characterizing its performance. In the next section, regarding performance concepts, the discussion on the two types of norms will continue.

2-2-2 Internal Stability

Besides input-to-output stability, stable behaviour of the interconnected components, i.e. internal stability, is also required. A standard feedback interconnection is shown in Figure 2-3. The latter can be represented in a more general form using the LFT. Additional disturbances v_1 and v_2 are injected into the loop. Then, all internal and external signal relations are stable if and only if the dynamic map

$$\begin{bmatrix} w \\ v_1 \\ v_2 \end{bmatrix} \rightarrow \begin{bmatrix} z \\ u \\ u_k \end{bmatrix} \tag{2-15}$$

is stable. First, consider the case where the performance channels w and z are absent. In this case, we are only interested in map from u to y, so consider P_{yu}. For internal stability, the transfer function matrix from (v_1, v_2) to (u, u_k) is analysed. The transfer function matrix is given by

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} I & -K \\ -P_{\text{yu}} & I \end{bmatrix} \begin{bmatrix} u \\ u_k \end{bmatrix}. \tag{2-16}$$

The inverse map is used to formulate a theorem for internal stability.

Theorem 2.1. Suppose that P_{yu} and K are proper. Then K renders the feedback interconnection shown in Figure 2-3 internally stable if and only if $(I + P_{\text{yu}}K)$ is well-posed (has a proper inverse), and the transfer function matrix

$$\begin{bmatrix} I & -K \\ -P_{\text{yu}} & I \end{bmatrix}^{-1} = \begin{bmatrix} (I - KP_{\text{yu}})^{-1} & K(I - P_{\text{yu}}K)^{-1} \\ (I - P_{\text{yu}}K)^{-1}P_{\text{yu}} & (I - P_{\text{yu}}K)^{-1} \end{bmatrix} \tag{2-17}$$

belongs to $\mathcal{RH}_\infty$ [26]. This is called the *4-block* test.

Next, we also consider the performance channels when assessing internal stability. In this case, the plant is given by

$$\begin{aligned} z &= P_{\text{zw}}w + \begin{bmatrix} P_{\text{zu}} & 0 \end{bmatrix} \begin{bmatrix} u \\ u_k \end{bmatrix} \\ \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} &= \begin{bmatrix} 0 \\ -P_{\text{yw}} \end{bmatrix} w + \begin{bmatrix} I & -K \\ -P_{\text{yu}} & I \end{bmatrix} \begin{bmatrix} u \\ u_k \end{bmatrix} \end{aligned} \tag{2-18}$$

The inverse map (2-15) is used to formulate the following theorem.

Theorem 2.2. Suppose that P and K are proper. Then, K stabilizes P if and only if $(I + P_{\text{yu}}K)$ is well-posed and the transfer function matrix

$$\begin{bmatrix} P_{\text{zw}} & P_{\text{zu}} & 0 \\ 0 & I & 0 \\ P_{\text{yw}} & P_{\text{yu}} & I \end{bmatrix} + \begin{bmatrix} P_{\text{zu}} \\ I \\ P_{\text{yu}} \end{bmatrix} K(I - P_{\text{yu}}K)^{-1} \begin{bmatrix} P_{\text{yw}} & P_{\text{yu}} & I \end{bmatrix} \tag{2-19}$$

is stable. This is called the *9-block* test.

It should be noted that it is not possible to find a stabilizing controller for any plant. Therefore, the following definition is useful to describe controllers that do in fact stabilize a plant.

Definition 2.6. If there exists at least one controller K that stabilizes the open-loop interconnection P Figure 2-1a, we call P a *generalized plant* [27].

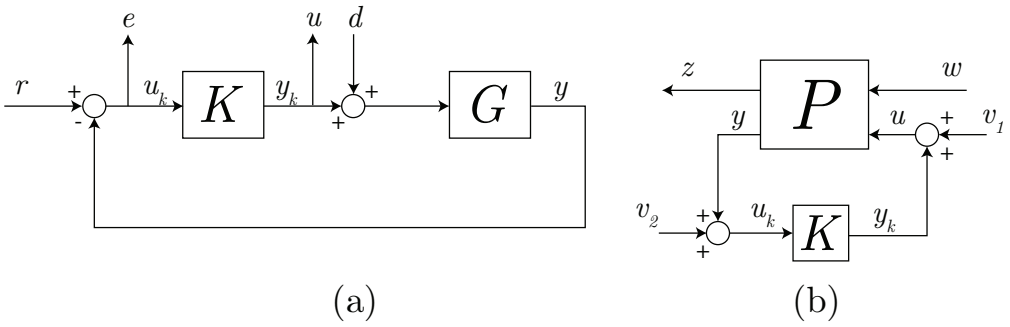

Figure 2-3: Standard feedback interconnection (a) redesigned as an internal stability analysis diagram (b).

2-2-3 Nyquist Stability Criterion

Stability of SISO and MIMO systems can be evaluated using the Nyquist stability criterion. The open-loop transfer function $L(s) = G(s)K(s)$ is used to asses the stability of the closed-loop system. The Nyquist plot of L is defined as the image of a chosen Nyquist contour Γ under L. To determine stability of the closed-loop system, we can count the encirclements of $1 + L(s)$ around 0. However, for SISO systems it is convenient to shift the Nyquist plot and count the encirclements of $L(s)$ around -1. Cauchy's argument principle is applied to the Nyquist contour, which gives the following theorem:

Theorem 2.3. (Nyquist stability theorem (SISO)) Let $L(s)$ be a proper transfer function and Γ a Nyquist contour for $L(j\omega)$ such that the Nyquist plot of $L(j\omega)$ does not pass through -1, i.e. $(I + L) \neq 0$. The number of unstable zeros Z equals the net number of clockwise encirclements N around $(-1, 0j)$ by $L(j\omega)$ plus the number of open-loop poles P_{ol}:

$$Z = N + P_{ol}. \tag{2-20}$$

The system is stable if the Nyquist curve does not go through $(-1, 0j)$ and if $Z = 0$.

In other words, to avoid zeros in the RHP one counter-clockwise encirclement around -1 is required for each open-loop pole in the RHP. Finally, it should be emphasized that we use the Nyquist stability theorem because an exact mathematical model is not required, i.e. it can also be applied to FRFs.

For MIMO systems the Nyquist stability criterion is different, the most significant difference is that the determinant has to be taken into account. Furthermore, in the MIMO setting the contour is defined differently. Poles on the imaginary axis can be moved by feedback in the SISO case, however this is not guaranteed in the MIMO case. In Figure 2-4 the MIMO contour is shown when there is a pole at $s = 0$. For any other poles on the imaginary axis the same procedure is carried out. Hence, the poles on the imaginary axis have to be taken into account when counting encirclements to get a correct stability criterion. The generalized Nyquist stability criterion for MIMO systems is given next.

Theorem 2.4. (Generalized Nyquist stability criterion (MIMO)) Let $L(s)$ be the loop transfer function matrix of a square MIMO system and let P_{ol} be the number of unstable open-loop poles in the RHP. The closed-loop transfer function matrix with negative feedback is stable if and only if the Nyquist plot of $\det(I + L(j\omega))$

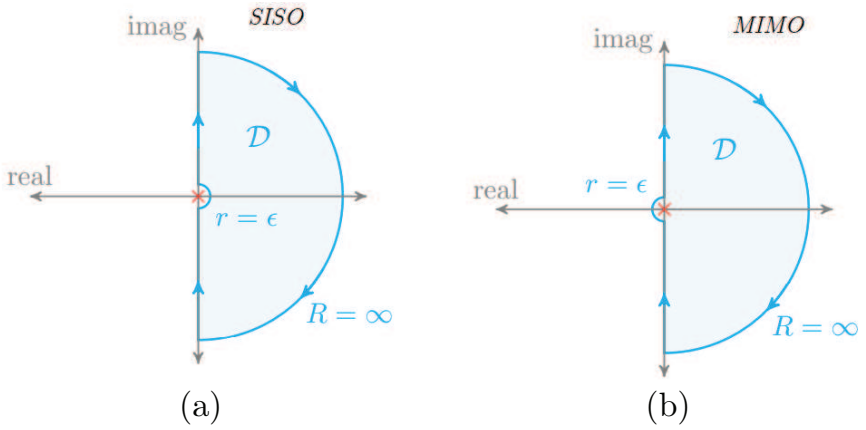

Figure 2-4: The Nyquist contour Γ for SISO systems (a) and for MIMO systems (b) in case there is a pole at $s = 0$ from the lecture notes by R. Smith [28].

1. does not go through the origin and,
2. makes P anti-clockwise encirclements of the origin.

A more elaborate description and proof of the theorem can be found in [24].

2-2-4 Robust Stability

The idea of robust stability is that a robust controller K is designed for an uncertain system P_Δ, such that stability is guaranteed for all uncertainties in $\Delta \in \mathbf{\Delta}$. The unstructured uncertainty set $\mathbf{\Delta}$ is defined as:

$$\mathbf{\Delta} := \{\Delta \in \mathcal{RH}_\infty^{p\times q} | \Delta(j\omega) \in \mathbf{\Delta}_c \, \forall \omega \in [0, \infty]\}, \tag{2-21}$$

where $\mathbf{\Delta}_c = \{\Delta_c \in \mathbb{C}^{p\times q} | \, \|\Delta_c\|_{2,2} < 1\}$ is the value set of $\mathbf{\Delta}$. This is the set of proper and stable uncertainty transfer functions. For robust stability, the assumptions are that P is a generalized plant, $\mathcal{F}_u(\Delta, P)$ is well-posed and that K stabilizes the nominal system P. Robust stability is also a relevant notion for LPV systems, because the varying parameter can be viewed as an uncertainty. The diagram for robust stability analysis is shown in Figure 2-5. The interconnection can be described in a general form:

$$\begin{aligned} \begin{bmatrix} z_\Delta \\ z \\ y \end{bmatrix} &= \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} \begin{bmatrix} w_\Delta \\ w \\ u \end{bmatrix} \\ w_\Delta &= \Delta z_\Delta \\ u &= Ky. \end{aligned} \tag{2-22}$$

The controller is collapsed into the plant with the use of lower LFT $N = \mathcal{F}_\ell(P, K)$:

$$N = \begin{bmatrix} M & N_{12} \\ N_{21} & N_{22}, \end{bmatrix} \tag{2-23}$$

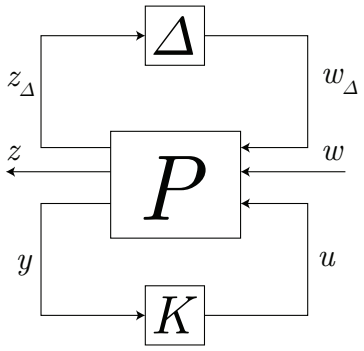

Figure 2-5: Generalized plant with uncertainty Δ.

where the transfer function seen by the uncertainty is

$$M = P_{11} + P_{13}K(I - P_{33}K)^{-1}P_{31}. \tag{2-24}$$

It is required that K is a stabilizing controller, i.e. $P \star K$ is internally stable. So the 4-block test (2-17) can be performed with P_{33} to test if K stabilizes P_{33} and if $\begin{pmatrix} I & -K \\ -P_{33} & I \end{pmatrix}^{-1}$ is stable and proper. If $P \star K$ is internally stable, then P is a generalized plant. Therefore all the transfer matrices in N are proper and stable. It suffices to check if $(I - M\Delta)^{-1}$ is proper and stable.

Theorem 2.5. If K stabilizes P, and if $(I - M\Delta)$ has a proper and stable inverse for all $\Delta \in \mathbf{\Delta}$, then K robustly stabilizes $\mathcal{F}_u(\Delta, P)$ against Δ [27].

The inverse of $(I - M\Delta)$ is stable and proper if and only if $\det(I - M(s)\Delta(s)) \neq 0$, $\forall\, s \in \mathbb{C}_= \cup \mathbb{C}_+ \cup \{\infty\}$ and $\forall \Delta \in \mathbf{\Delta}$. This result is quite complicated, since it is required to check all uncertainties and the whole RHP plane. This is not a computable condition, yet. An extra real parameter $0 < c < 1$ is introduced to scale uncertainties as $c\Delta$. subsequently, it is equivalent to verifying that $\det(I - M(s)(c\Delta(s)))$ has no zeros on the upper part of the imaginary axis for all $c \in (0, 1]$. Suppose that M is stable and $\mathbf{\Delta} \subset \mathcal{RH}_\infty$ such that $\Delta \in \mathbf{\Delta}$, then $c\Delta \in \mathbf{\Delta}$ for all $0 < c < 1$. If $\det(I - M(j\omega)\Delta(j\omega)) \neq 0$, $\forall \omega \in [0, \infty]$, $\forall \Delta \in \mathbf{\Delta}$ then $(I - M\Delta)^{-1}$ is stable and proper for all $\Delta \in \mathbf{\Delta}$ [27, see Thereom 4.5]. Now the robust stability theorem can be reformulated, such that it suffices to test if $(I - M(s)\Delta_c)$ is non-singular for $s = j\omega$ with $\omega \in \mathbb{R} \cup \{\infty\}$.

Theorem 2.6. Suppose that K stabilizes P and that Δ_c is unstructured. If

$$\|M(j\omega)\|_\infty \leq 1 \iff \det(I - M(j\omega)\Delta_c) \neq 0,\ \forall \Delta_c \in \mathbf{\Delta_c},\ \omega \in \mathbb{R} \cup \{\infty\},$$

then K robustly stabilizes $\mathcal{F}_u(\Delta, P)$ as well. The proof for this theorem can be found in [27].

Small Gain Theorem The small gain theorem is a general result which can be seen as a generalization of the Nyquist stability theorem for (time-varying) MIMO systems.

Theorem 2.7. Consider a system with a stable loop transfer function $L(s)$. Then the closed-loop system is stable if

$$\|L(j\omega)\| < 1, \forall\omega. \tag{2-25}$$

The norm can be any (induced) norm that satisfies the multiplicative property [24]

$$\|AB\| \leq \|A\| \cdot \|B\|. \tag{2-26}$$

2-3 Performance

2-3-1 Performance shaping

In case there are no uncertainties, the performance of a control system can be shaped for nominal performance. The idea is to shape the FRF of the closed-loop transfer functions using weighting filters such that the weighted system satisfies the performance specifications. An $\mathcal{H}_\infty$ control problem is formulated for this purpose in this section.

Nominal performance Consider the standard feedback interconnection in Figure 2-3, where it is assumed that P is a generalized plant and K is a stabilizing controller. Let $e = r - y$, then the interconnection is put in the general PK form, as in Figure 2-1a, where $z = [\, e,\, u \,]^T$ and $w = [\, r,\, d \,]^T$. Weighting filters for performance and disturbance specifications are used for the output signal z and generalized disturbance signal w, respectively. W_w acts as a shaping filter for the disturbance inputs to capture variation in size and direction over frequency and W_z allows to put different emphasis on size and directional attenuation at different frequencies. The weighted generalized plant is shown in Figure 2-6, where $W_z = \left(\begin{smallmatrix} W_e & 0 \\ 0 & W_u \end{smallmatrix}\right)$ and $W_w = \left(\begin{smallmatrix} W_r & 0 \\ 0 & W_d \end{smallmatrix}\right)$. The reformulated signals due to the weighting filters are $w = W_w\tilde{w}$ and $\tilde{z} = W_z z$. The weighted closed-loop transfer function $\tilde{w} \mapsto \tilde{z}$ is given by

$$\tilde{z} = (W_z N W_w)\tilde{w}, \tag{2-27}$$

where $N = P \star K$. The weighted plant is denoted as $\tilde{P} = W_z P W_w$. It should be noted that the weights only affect the performance channels $w \to z$, which means that if a controller stabilizes P, it also stabilizes $\tilde{P}$.

Definition 2.7. For a given generalized plant P and stabilizing controller K, the controller satisfies the nominal performance requirements interpreted as a weighted plant $\tilde{P}$, if

$\tilde{P}$ is a generalized plant (K internally stabilizes $\tilde{P}$),

$$\|\tilde{P} \star K\|_\infty = \|\tilde{N}\|_\infty \leq 1. \tag{2-28}$$

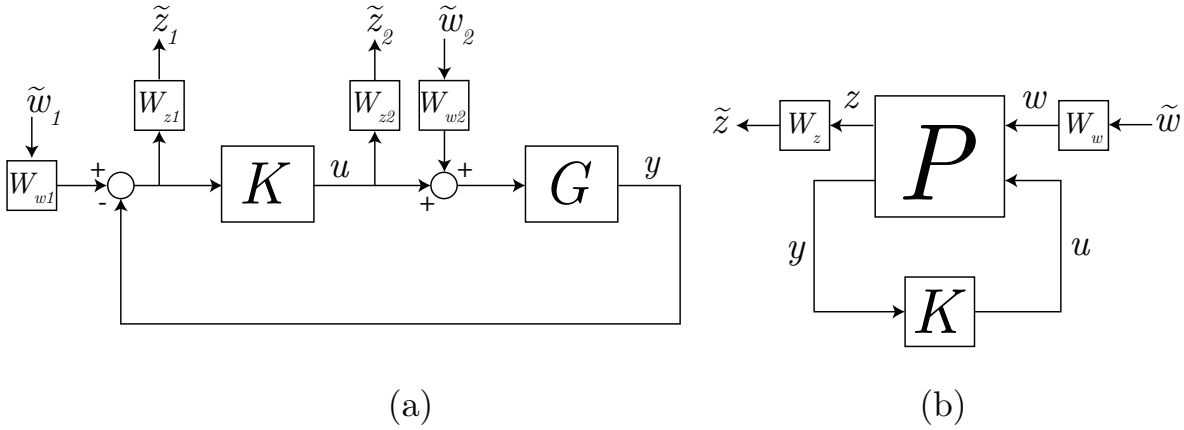

Figure 2-6: Weighted 4-block interconnection and generalized plant with weighting filters W_z and W_w.

Main loop theorem Consider the full complex block virtual perturbation $\hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}$, with

$$\mathbf{B}\hat{\mathbf{\Delta}} := \{\hat{\Delta} \in \mathcal{RH}_\infty \,|\, \sup \bar{\sigma}\left(\hat{\Delta}(j\omega)\right) < 1,\, \forall \omega \in \mathbb{R} \cup \{\infty\}\}. \tag{2-29}$$

Performance specifications can be evaluated by placing $\hat{\Delta}$ in feedback with the performance channel $w \to z$, which represents the $\mathcal{H}_\infty$ performance. In Figure 2-7 the generalized plant with the virtual uncertainty is shown, where $N = P \star K$. The non-singularity condition that should be checked is:

$$\det(I - N\hat{\Delta}) \neq 0,\, \forall \hat{\Delta} \in \hat{\mathbf{\Delta}}. \tag{2-30}$$

This condition is similar to the robust stability non-singularity condition in Theorem 2.6. When a virtual uncertainty is connected directly to the performance channel, it can be considered as a special case of robust performance or it can be seen as a generalisation of robust stability.

Theorem 2.8. The Main Loop theorem states that the following statements are equivalent

1. $\|N(j\omega)\|_\infty \leq 1$,
2. $\det(I - N(j\omega)\hat{\Delta}) \neq 0,\, \forall \hat{\Delta} \in \hat{\mathbf{\Delta}},\, \omega \in \mathbb{R} \cup \{\infty\}$.

2-3-2 4-block objective

In Section 2-2-2 the 4-block test in (2-17) was briefly mentioned to explain internal stability. This 4-block is also used in performance shaping for mixed-sensitivity controller design. The transfer functions in (2-17) are referred to as sensitivity functions, and they can be found by considering a few mappings in the interconnection in Figure 2-3. The mapping $S : r \to e$ is the output sensitivity function

$$S_\mathrm{o} := (I + PK)^{-1}. \tag{2-31}$$

The input complementary sensitivity function is the mapping $T : r \to y$

$$T_\mathrm{i} := K(I + PK)^{-1}P \tag{2-32}$$

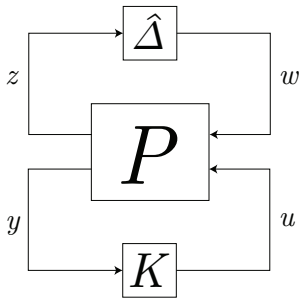

Figure 2-7: Generalized plant with virtual uncertainty block $\hat{\Delta}$.

The process sensitivity PS and controller sensitivity KS can be found similarly, their mappings are $PS := d \to y$ and $KS := r \to u$.

$$PS := (I + PK)^{-1}P, \tag{2-33}$$
$$KS := K(I + PK)^{-1}. \tag{2-34}$$

For mixed-sensitivity design they are designed simultaneously for the closed-loop system given by:

$$\begin{bmatrix} e \\ u \end{bmatrix} = \begin{bmatrix} S_{\mathrm{o}} & SP \\ KS & T_{\mathrm{i}} \end{bmatrix} \begin{bmatrix} r \\ d \end{bmatrix}. \tag{2-35}$$

The performance objectives for each of the closed-loop transfer functions are summarized below.

1. $\bar{\sigma}(S_{\mathrm{o}}(j\omega))$ should be small at low frequencies for good low-frequent reference tracking and disturbance rejection.

2. $\bar{\sigma}(PS(j\omega))$ should be small at high/low frequencies for good disturbance rejection at high/low frequencies.

3. $\bar{\sigma}(KS(j\omega))$ should be small to limit control action.

4. $\bar{\sigma}((T_{\mathrm{i}}j\omega))$ should be small at high frequencies for high frequency noise attenuation.

The control design objective is to find a controller K that internally stabilizes the closed-loop plant and is optimal with respect to the $\mathcal{H}_\infty$ norm. The performance specifications are formulated in the PK (general) form and weighting filters are used to shape the closed-loop behaviour such that certain frequencies are emphasised. In this section we assume that the controller satisfies the condition for internal stability, as presented in Theorem 2.1. The 4-block objective is to design a controller K which stabilizes the interconnection in Figure 2-6 and minimizes the $\mathcal{H}_\infty$ norm of $\begin{pmatrix} \tilde{w}_1^T & \tilde{w}_2^T \end{pmatrix}^T \to \begin{pmatrix} \tilde{z}_1^T & \tilde{z}_2^T \end{pmatrix}^T$, i.e.,

$$\min_K \left\| \begin{bmatrix} W_{z1}S_oW_{w1} & W_{z1}S_oPW_{w2} \\ W_{z2}KS_oW_{w1} & W_{z2}KS_oPW_{w2} \end{bmatrix} \right\|_\infty. \tag{2-36}$$

The weighting filters are used to individually shape the sensitivities.

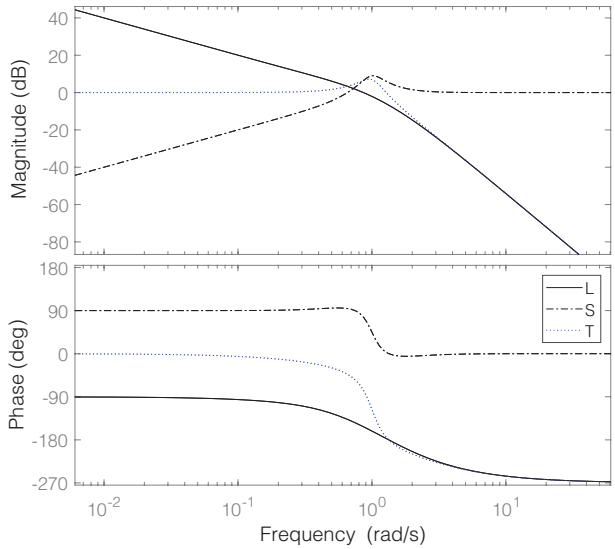

Figure 2-8: Desired loop gain, sensitivity and complimentary sensitivity.

2-3-3 Weighting filter selection

Shaping the sensitivity We assume a normalized plant and strictly proper systems, which is typically the case for physical systems. Let $L_o(j\omega)$ be the FRF of the open-loop system, then as $\omega \to \infty : L_o(j\omega) \to 0 \implies S_o(j\omega) = I$. The performance improves in the frequency range where $\bar{\sigma}(S_o(j\omega)) < 1$ and degrades when $\bar{\sigma}(S_o(j\omega)) > 1$. From Figure 2-8 we know that when

$$\begin{aligned} \bar{\sigma}(L_o(j\omega)) \gg 1 &\implies S_o \approx L_o^{-1}, T_o \approx I \\ \bar{\sigma}(L_o(j\omega)) \ll 1 &\implies S_o \approx I, T_o \approx L_o. \end{aligned} \tag{2-37}$$

Hence, at the frequency range where the system operates we would like to shape the sensitivity such that the gain is below 1.

Definition 2.8. The frequency ω_B where $\bar{\sigma}(S_o(j\omega))$ first crosses -3dB from below is the *bandwidth* of the closed-loop response.

There is a trade-off between the response time and robustness for increasing ω_B. Based on the latter a weight filter for the sensitivity can be designed, which typically takes the following form:

$$W_S(s) = \frac{s/M_s + \omega_B^*}{s + \omega_B^* \varepsilon_A}, \tag{2-38}$$

where $M_s > 1$. The maximum sensitivity $M_s = \|S_o\|_\infty$ influences the overshoot of the response, $M_s = 2$ causes around 20% overshoot. ε_A is the upper bound on $W_S(s)^{-1}$. At low frequencies we want $\bar{\sigma}(S_o(j\omega)) < \varepsilon_A$ for maximum steady-state tracking error. The inverse of the weighting filter resembles the ideal sensitivity, it is shown in Figure 2-9. It is assumed that the same weight filter can be used for the process sensitivity PS.

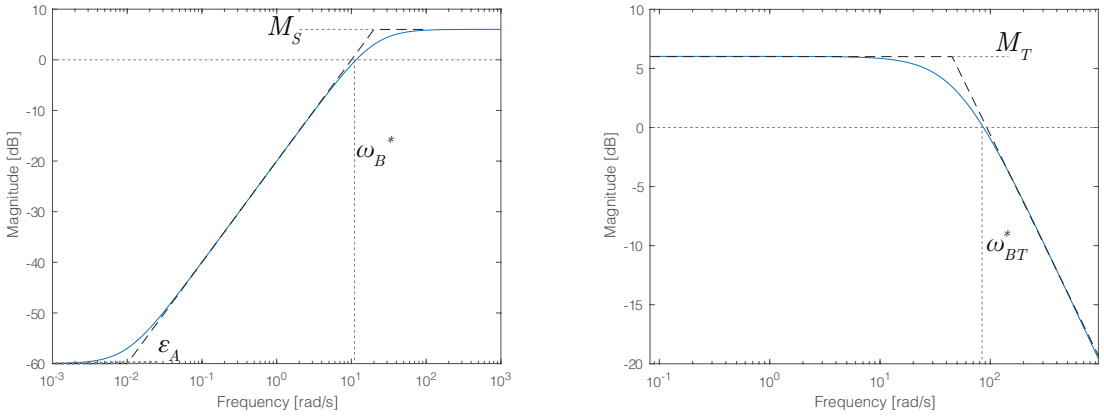

Figure 2-9: The inverse of the weighting filter for the sensitivity (a) and complimentary sensitivity (b).

Shaping the complimentary sensitivity By using the relations in (2-37) the complimentary sensitivity can be shaped.

Definition 2.9. The frequency ω_{BT} where $\bar{\sigma}(T_o(j\omega))$ first crosses -3dB from above is the *bandwidth* of the closed-loop response.

The bandwidth of the complimentary sensitivity ω_{BT} is closely related to ω_B as

$$\begin{aligned} |1-\bar{\sigma}(S_o(j\omega))| \leq \bar{\sigma}(T_o(j\omega)) \leq 1+\bar{\sigma}(S_o(j\omega)), \\ |1-\bar{\sigma}(T_o(j\omega))| \leq \bar{\sigma}(S_o(j\omega)) \leq 1+\bar{\sigma}(T_o(j\omega)). \end{aligned} \tag{2-39}$$

ω_{BT}, however, is not as good a performance indicator as ω_B. While the controller affects the response in the range $[\omega_B, \omega_{BT}]$, it does not affect performance as much because $S_o(j\omega) \approx I$. A useful weighting filter for the complimentary sensitivity is

$$W_T(s) = \frac{s+\omega_{BT}^*/M_t}{\varepsilon_A s+\omega_{BT}^*}, \tag{2-40}$$

where $M_t = \|T_o\|_\infty$ is the worst-case total variation of the output step response. The inverse of the weighting filter is shown in Figure 2-9. It is assumed that the same weighting filter can be used for the controller sensitivity KS. If that is not sufficient, the input weighting filter W_{w1} can be used to further shape KS. The weighting filters provided in (2-38) and (2-40) are good to start with, but should be tuned or higher order to further increase performance.

Waterbed effect Finally, during the weight design the relation between the sensitivity and complementary sensitivity should be kept in mind: $S_o + T_o = I$. In terms of the maximum singular values this can be rewritten as $|\bar{\sigma}(S_o) + \bar{\sigma}(T_o)| \leq 1$. This implies that the maximum difference between the singular values is 1 at all frequencies. This relation also leads to the waterbed effect constraint: the intersection of $W_S(j\omega)$ and $W_T(j\omega)$ must be below 0 dB, otherwise the controller synthesis problem is infeasible. In short, the waterbed effect states that if the sensitivity is pushed down at a certain frequency range it must come up at another. This means the designer has to choose where the singular values of the sensitivity are below 1, we typically want small singular values at the frequency range that matters for the system.

2-4 Fixed-structure control

Sometimes a controller may be restricted in its complexity due to hardware constraints. Fixed-structure controllers have predefined structure and are typically of low order, hence they are easier to implement. The disadvantage of these type of controllers is that their synthesis does not correspond to a convex optimization problem. Nevertheless, fixed-structure controllers are widely used in the industry. One of the most frequently used controller is the PID controller.

2-4-1 PID control

The PID controller has a simple structure and allows for on site tuning of the controller to improve performance. The ideal PID controller is defined by the relation between the output of the controller u and the error between the system output and reference input:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\mathrm{d}\tau + k_d \frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{2-41}$$

$$u(s) = e(s)\big(k_p + \frac{k_i}{s} + k_d s\big) \tag{2-42}$$

$$K_{PID}(s) = \frac{u(s)}{e(s)} = k_p + \frac{k_i}{s} + k_d s, \tag{2-43}$$

where k_p, k_i and k_d are the gains of the proportional, integral and derivative action, respectively. Each of these elements can be used separately or in combinations with each other, depending on the application.

Proportional action The gain k_p is proportional to the current error. If the error increases then so does corresponding control action k_p. The steady state error is the difference between the final output value and the reference input. By increasing the gain the steady state error decreases, but it can also result in a oscillatory system behaviour. In the frequency domain, a higher k_p simply increases the gain of the feedback system, which can improve the bandwidth of the system. However, if the gain increases even further, it may cause the system to be unstable. On the other hand, if the gain is too low, the controller might be unresponsive if the system is disturbed by exogenous signals.

Integral action In case there is a steady state error, an integral action is required, because the integrator ensures zero steady state error. Moreover, an integrator has the beneficial properties where it has low gain at low frequencies and high gain at high frequencies. An integrator on its own is not stable, since we get a ramp as an output to a step input. The integrator works best in combination with the proportional, which forms a PI controller, because the integrator on its own is slow.

Derivative action The derivative action provides system prediction and improves the response of the system by adding damping to reduce oscillatory behaviour and adds a phase lead to compensate for the phase lag of the integral action. The downside of the derivative action is that it makes the system sensitive to noise at high frequencies.

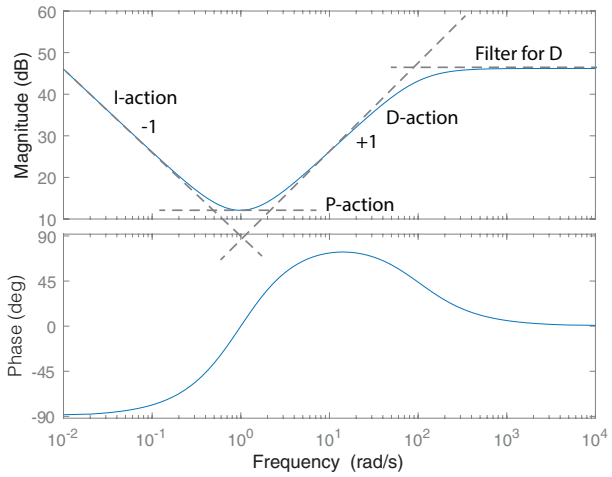

Figure 2-10: Bode plot of a PID controller demonstrating the effects of each element.

The PID controller in (2-43) is not physically realizable due to the fact that the derivative term $k_d s$ is not causal. Hence, the term is replaced by a low-pass filtered derivate:

$$\frac{k_d s}{(T_f s + 1)},$$

where T_f is a time constant. It can be interpreted as an ideal derivative filter with two modes: at low frequencies it is simply a derivative term $k_d s$ and at higher frequencies it acts as a constant gain k_d/T_f. The filtered PID controller is given by:

$$K_{PID} = k_p + \frac{k_i}{s} + \frac{k_d s}{T_f s + 1}. \tag{2-44}$$

In Figure 2-10 a Bode plot of a PID controller is shown to demonstrate the effect of each term.

2-4-2 Basic filters

Besides the PID controller, there are a few more basic filters that are frequently used. The low pass filter:

$$K_{LPF}(s) = \frac{1}{\frac{s}{\omega_c} + 1}. \tag{2-45}$$

It passes low frequencies and attenuates high frequencies, typically noise, from the corner frequency ω_c with a magnitude roll off of -20 dB per decade. A second order low pass filter:

$$K_{LPF}(s) = \frac{1}{(\frac{s}{\omega_c})^2 + \frac{s}{Q\omega_c} + 1} \tag{2-46}$$

has a magnitude roll off of -40 dB per decade. With Q it is possible to create or cancel a peak at the corner frequency. The Bode plot of these filter is shown in Figure 2-11.

A lead or lag filter is given by

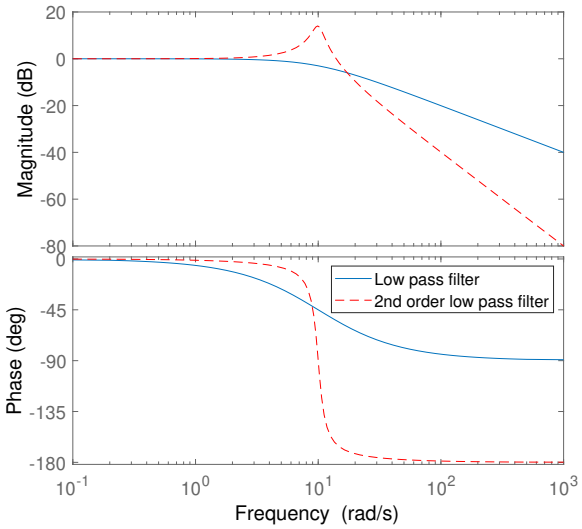

Figure 2-11: Low pass filter and second order low pass filter bode plot.

$$K_{lead/lag}(s) = \frac{\frac{s}{\omega_z} + 1}{\frac{s}{\omega_p} + 1}, \tag{2-47}$$

where ω_z is the frequency of the zero and ω_p is the frequency of the pole. If $\omega_z < \omega_p$ it is a lead filter and if $\omega_z > \omega_p$ it is a lag filter. The Bode plot in Figure 2-12 shows that a lead filter adds positive phase to the frequency response at a given frequency range and increases the gain at high frequencies, which increases the bandwidth of the system. The lag filter adds negative phase and attenuates high frequencies over a specified frequency range, but it decreases the bandwidth of the system. A lead-lag filter combines the effect of both filters.

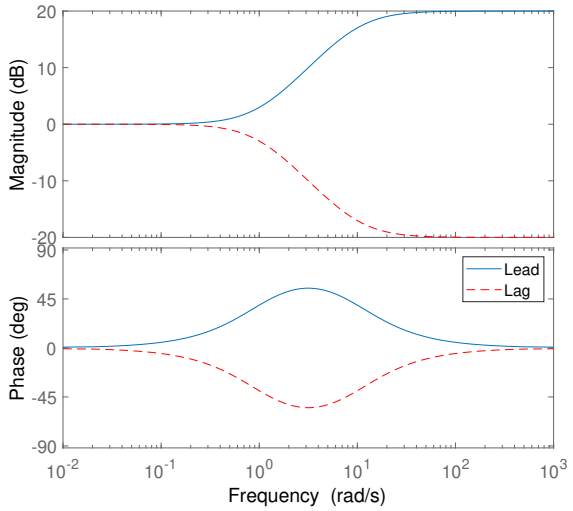

Figure 2-12: Bode plot of lead and lag filters.

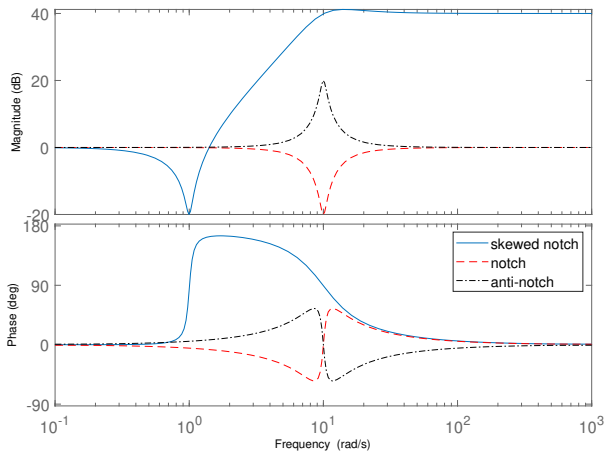

Figure 2-13: Bode plot of skewed notch, notch and anti-notch filters.

A (skewed) notch filter is given by

$$K_{notch}(s) = \frac{(\frac{s}{\omega_1})^2 + \frac{s}{Q_1\omega_1} + 1}{(\frac{s}{\omega_2})^2 + \frac{s}{Q_2\omega_2} + 1}, \tag{2-48}$$

where ω_1 and ω_2 are frequencies where the peak is placed. Q_1 and Q_2 are the damping factors, their magnitude determines the width of the peak. The choices for the parameters determine the type of notch that is placed on the frequency response, three types are shown in Figure 2-13. If $\omega_1 = \omega_2$ and $Q_1 > Q_2$ it is a notch filter which attenuates resonance peaks in the frequency response function at a given frequency. The resonance peak frequency has to be tuned almost perfectly, otherwise the notch filter does not work. It is also possible to use an anti-notch filter, which creates a resonance peak by choosing $Q_1 < Q_2$. A skewed notch is requires less precision, because it attenuates over a frequency range.

2-5 Particle swarm optimization

Particle swarm optimization (PSO) is an optimization technique to solve non-convex problems. By choosing for fixed-structure controllers, we need to solve a non-convex optimization problem for their performance optimization. The optimization problem is typically in the following form:

$$\min_x : f(x) \tag{2-49}$$

$$\begin{aligned} \text{subject to: } Ax &\leq b \\ A_{eq}x &= b_{eq} \\ c(x) &\leq 0 \\ c_{eq}(x) &= 0 \\ lb \leq x &\leq ub, \end{aligned} \tag{2-50}$$

where x, b, b_{eq}, lb and ub are vectors, and A and A_{eq} are matrices. The functions $c(x)$ and c_{eq} can be non-linear constraints and $f(x)$ is the objective function, which can be non-linear as well. In this report PSO is used for the fixed-structure controller synthesis.

PSO algorithm PSO is an iterative stochastic search method introduced by [29]. It is inspired by the movement of animals, such as a flock of birds. The swarm or the population consists of particles, each of whose position and velocity is known. The change in the direction of their position and velocity is the effect of cognitive, social and stochastic influence [29]. The particles look for the optimal solution in the specified search region. Each particle compares its personal best position to the global best amongst all the particles. The position and velocity of the particles can be described mathematically by the following two equations:

$$v_i^{k+1} = \phi^k v_i^k + \underbrace{\alpha_1\left[\gamma_{1,i}(P_i - x_i^k)\right]}_{\text{cognitive component}} + \underbrace{\alpha_2\left[\gamma_{2,i}(G - x_i^k)\right]}_{\text{social component}} \tag{2-51}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \tag{2-52}$$

respectively. The vectors x_i^k and v_i^k are the current position and velocity of the i-th particle in the k-th generation, respectively [30]. The amount of particles is N, i.e., $i = \{1, ..., N\}$. The first component of (2-51) maintains the current velocity, where ϕ is the particle inertia weight which control the way the particles search. The inertia weight is typically between $[0.4, 0.9]$ [29]. The cognitive component is where each particle computes the distance to their personal best location P_i. The social component is named so because the particles compute the distance between its current position and the best position G found by the swarm. The parameters γ_1 and γ_2 determine the influence of the cognitive and social component on the direction and velocity of the particles. These parameters are uniformly distributed random values in $[0, 1]$. The importance of each component can be increased via the acceleration constants $\alpha_{1,2}$. In Figure 2-14 a graphical interpretation of the PSO algorithm is shown. The particle has its own personal best solution P, shown in yellow, and there is a global best solution G, shown in blue. The direction of these vectors is used to form the new velocity vector v_i^{k+1}, which is the sum of the cognitive and social component. Note that the length of the vectors is not necessarily the same as the distance to the personal best and global best,

since the cognitive and social component are multiplied by random numbers. Finally, the new position x^{k+1} is the sum of the current position x^k and the velocity vector v^{k+1}.

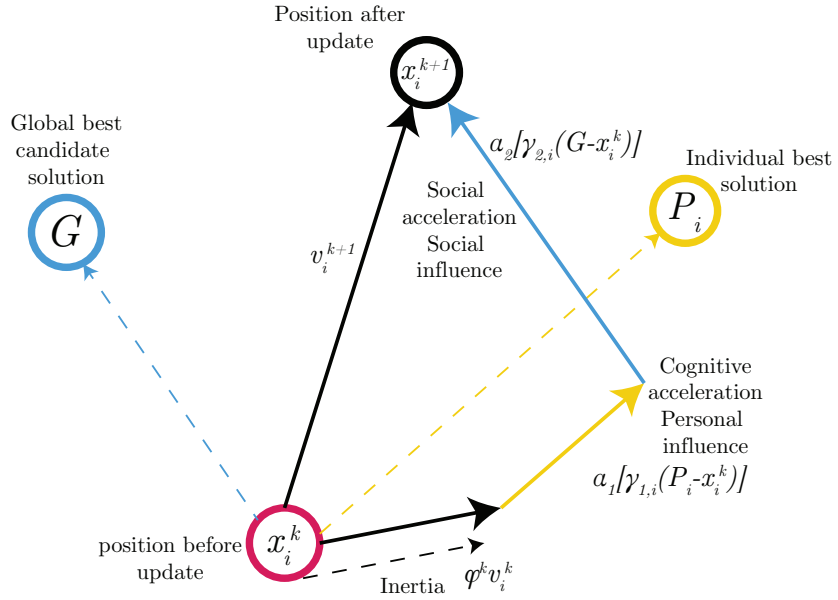

Figure 2-14: Graphical interpretation of the PSO algorithm.

Stability & Constraints In order to guarantee that the PSO algorithm converges to a stable equilibrium, necessary and sufficient conditions for the inertia weight and acceleration constants to ensure stability have been derived in [31]. The latter are given by:

$$\alpha_1 + \alpha_2 < 4 \tag{2-53}$$

$$\frac{\alpha_1 + \alpha_2}{2} - 1 < \phi < 1. \tag{2-54}$$

There are several ways to solve a constrained optimization problem with PSO. The PSO function that was used in this report has three methods to handle constraints: penalize, absorb, and nearest [30]. Penalize assigns a high objective function value to any particle that violates a constraint. This value has to be bigger than the maximum feasible objective function in the specified region. Particles can move freely around the constraints, but in the end they are attracted to feasible solutions forcing them towards the feasible region. The particles can also be absorbed to the boundary of the feasible region that is defined by the constraints. This requires the algorithm to check the objective function value of a particle, which takes relatively more time but it also might give better results if the global minimum is near a constraint. Finally, "nearest" pulls the particles that are outside the feasible region back to the nearest boundary defined by the constraints. The objective function value has to be evaluated as well for this method. A graphical interpretation of these constraint handling method is shown in Figure 2-15. The white dots indicate how the method move the particle to the feasible region.
The optimization problem does not need to be differentiable, as opposed to methods that use the gradient of the problem, e.g. gradient descent. Furthermore, PSO barely makes any assumptions on the optimization problem and can look for a solution in an user specified

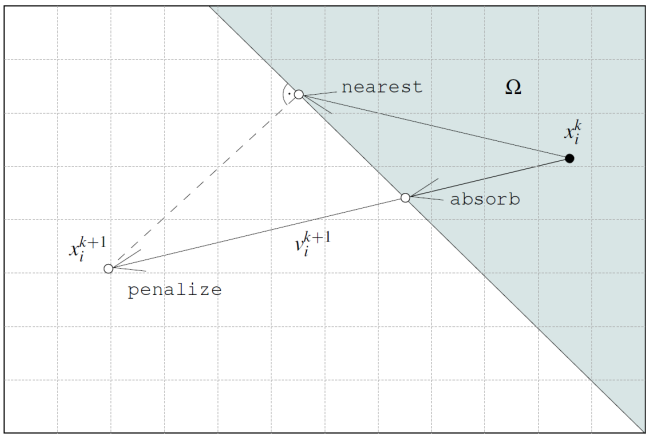

Figure 2-15: Graphical interpretation of the constraint handling methods by S. Ebbesen in [30].

search region, which can be relatively big. However, the downside is that there is no guarantee that a global minimum is found.

2-6 Summary

In this chapter, some basic theory has been presented that is necessary to build up the theory of LTI and LPV systems in this thesis. Among the discussed topics are system representations, notions of stability and performance, fixed-structure controllers and the particle swarm optimization algorithm. The theory of these topics is applied in the next chapters to formulate new stability and performance theorems. The next chapter discusses state-of-the-art LTI controller synthesis methods, where the focus is on data-driven frequency domain controller design.

Chapter 3

LTI Data-Driven Controller Design

This chapter deals with the state-of-the-art controller synthesis methods in the LTI framework. A short recap of the state-of-the-art from Section 1-1 is given in Section 3-1, then the distinctive properties of each method are briefly discussed to motivate why they are reviewed in this chapter. Four methods stand out from the state-of-the-art, these are explained in Sections 3-2 to 3-4. These methods are compared to each other in Section 3-5. Finally, the limitations of LTI controller design methods in general are discussed.

3-1 State-of-the-art

The LTI framework has been developed extensively the past decades because it allows for relatively simple controller design. In the industry it is often attempted to design a system that behaves like an LTI system around its operating points, such that theory from the LTI framework can be applied. The LTI framework in combination with data-driven control in the frequency-domain has sparked the interest of researchers in the control community. The main advantage of data-driven control is the fact that the tedious process of modelling a system is not necessary any more. The frequency-domain has some advantages as well, such as the fact that there is no need for a transfer function, as it is possible to acquire knowledge of the dynamics of the system experimentally through the frequency response. Furthermore, the open-loop frequency response can be used to assess closed-loop stability. If these advantages are put to use, a versatile controller synthesis method can be developed.
In Chapter 1 the state-of-the-art of LTI data-driven controller design methods was briefly discussed. During the internship and master thesis several methods were implemented and tested to gain some insight in the state-of-the-art data-driven controller design methods. The following four methods stand out:

1. Robust $\mathcal{H}_\infty$ controller design using frequency-domain data via convex optimization by Karimi et al. [5];
2. A data-driven method for computing fixed-structure low-order; controllers with $\mathcal{H}_\infty$ performance by Nicoletti et al. [6];

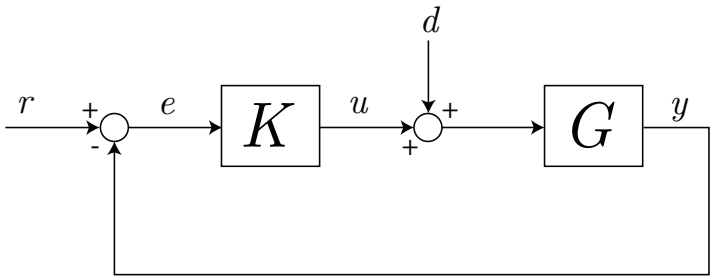

Figure 3-1: Standard feedback interconnection.

3. Frequency-domain optimization of fixed-structure controllers by van Solingen et al. [7].
4. Feedback Autotuner (FBA) by van der Meulen from ASML.

Why do these methods stand out? Each of these methods has a unique approach to design a controller, their interesting properties are discussed next. The first two methods use the same conditions to find a stabilizing controller that satisfies $\mathcal{H}_\infty$ specifications. The conditions for stability and performance are derived using coprime factorizations, these conditions make it possible to design a stabilizing controller for an unstable system. The convexity of the optimization problem depends on the type of controller that is designed. If a fixed-structure, typically of lower order, controller is designed, then a non-convex optimization problem has to be solved. If instead a fixed-order controller is designed, then the optimization problem is convex. Convexity is of course a very attractive property, because optimal control parameters can be found globally. However, the reason for the development of the second method is due to the fact that lower order controllers are easier to implement. The third method (van Solingen et al.) also aims to design a fixed-structure controller using the generalized Nyquist theorem to impose stability and $\mathcal{H}_\infty$ performance specifications on the closed-loop system. By using the method, it is possible to optimize tunable controller parameters of linear parametrizable controllers [7]. The method is advantageous because the general plant-controller (PK) form is used. The structure of the controller is absorbed by the system and the tunable control parameters are put into a diagonal matrix as the structure and parameters are separated. This renders a multilinear feasibility problem that can be solved to design controllers for MIMO systems, which is also unique from the other three methods. The fourth method (FBA) exploits the Nyquist theorem for SISO systems to find stabilizing control parameters for fixed-structure controllers. This is done by counting the number of encirclements around the critical point, assuming that the number of open-loop right half-plane poles are known. Once a range of stabilizing control parameters has been found, the controller is optimized in that range using the particle swarm optimization algorithm. The stability test combined with the optimization allow for easy (auto) tuning of fixed-structure controllers. These methods are explained in detail in the following sections.

3-2 Control Design Method with $\mathcal{H}_\infty$ Performance for SISO Systems

The first two methods aim to design a data-driven controller for SISO LTI systems that satisfy the $\mathcal{H}_\infty$ criterion for a closed-loop transfer function in the frequency-domain. Depending

on whether the controller is fixed-order or fixed-structure, different optimization methods are proposed for linear parametrized controllers.
Consider the interconnection in Figure 3-1, where $G(s)$ is a strictly proper transfer function of a SISO system. In reality, the transfer function of the system is unknown and all that is available are experimentally obtained FRFs. The plant $G(s)$ and controller $K(s)$ are represented by coprime factorizations similar to (2-5). The system and controller are factorised into factors that are stable, i.e. in $\mathcal{RH}_\infty$. The fact that the set $\mathcal{RH}_\infty$ is closed under addition and multiplication is exploited to derive conditions that impose stability and $\mathcal{H}_\infty$ performance specifications on the closed-loop system. The FRF of the plant is given by

$$G(j\omega) = N(j\omega)M^{-1}(j\omega),\ \omega \in \mathbb{R} \cup \{\infty\}. \tag{3-1}$$

The controller is given by $K(s) = X(s)Y^{-1}(s)$, where $\{X, Y\} \in \mathcal{RH}_\infty$. The objective is to design a controller that meets $\mathcal{H}_\infty$ norm specifications of the weighted sensitivity functions, i.e. we want to achieve nominal performance, see Definition 2.7. The sensitivity functions can be found by substituting the coprime factors N, M, X, Y in (2-31)-(2-34). The sensitivity function $S = (I+GK)^{-1}$ is used to explain the workings of this method, it is given in coprime factorization representation by

$$S(j\omega) = M(j\omega)Y(j\omega)D_H^{-1}(j\omega), \tag{3-2}$$

where $D_H(j\omega) = N(j\omega)X(j\omega) + M(j\omega)Y(j\omega)$. Let $W(j\omega)$ be a weighting filter with a stable frequency response function. The condition for nominal performance $|W(j\omega)S(j\omega)| < \gamma$ can be rewritten such that there are no inverses in the expression for $S(j\omega)$ in (3-2). The condition is given by

$$\gamma^{-1}|WMY|(j\omega) < |D_H(j\omega)|,\ \forall\omega \in \mathbb{R} \cup \{\infty\}, \tag{3-3}$$

where $\gamma > 0$. The following lemma is presented in [5] and is derived from [32]. This lemma is the backbone of the workings of the first two methods as it gives the necessary and sufficient condition for nominal performance (and internal stability).

Lemma 3.1. Consider the sensitivity function in (3-2) and let $H(j\omega)$ denote any sensitivity function. Suppose that $H(j\omega) = W(j\omega)M(j\omega)Y(j\omega)D_H^{-1}(j\omega)$ is the FRF of a bounded analytic function in the RHP. Then, the following constraint

$$\sup_{\omega\in\Omega} |H(j\omega)| < \gamma \tag{3-4}$$

is met if and only if there exists a stable transfer function $\alpha \in \mathcal{RH}_\infty$ that satisfies

$$\Re\{(D_H(j\omega) - \gamma^{-1}|WMY|(j\omega))\alpha(j\omega)\} > 0,\ \forall\omega \in \mathbb{R} \cup \{\infty\}. \tag{3-5}$$

The proof of this lemma can be found in [5], but it is important to note that this lemma uses the results of the Main Loop Theorem (see Theorem 2.8). Because the step from (3-4) to (3-5) is the same as from step 1 to step 2 in Theorem 2.8. The graphical interpretation of (3-5) is shown in Figure 3-2. The disc centred at D_H with radius $\gamma^{-1}WMY$ should not include the origin, thus $|NX + MY| > |\gamma^{-1}WMY|$. Lemma 3.1 states that there exists a complex function $\alpha(j\omega)$ for every frequency point such that the disc does not include the origin. This is equivalent to a line going through the origin that does not intersect with the disc. Hence, α can rotate the disc such that all the values on, and in the circle have positive real parts [6]. On the one hand, Lemma 3.1 can be used to analyse if a controller is stable and meets specific performance requirements, on the other hand it is suitable to synthesise a controller. The latter can be done in two ways: by designing a fixed-order or fixed-structure controller.

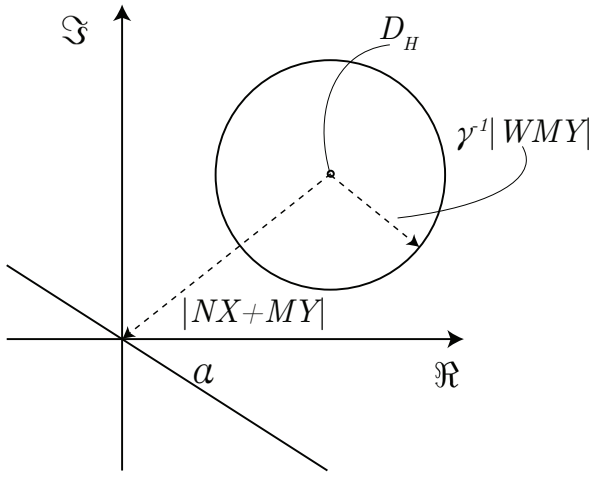

Figure 3-2: Graphical interpretation of the condition in (3-5). To satisfy (Nyquist) stability, the disc centred at D_H with radius $|\gamma^{-1}WMY|$ should not encircle the origin for every frequency point ω. The complex function α acts as a line going through the origin, the disc should not intersect with this line.

3-2-1 Fixed-order Controller Synthesis

The set of all stabilizing controllers that meet nominal performance specifications defined by the weighted norm of sensitivity functions is given in the following theorem as presented in [5].

Theorem 3.1. Given the FRFs of a system $G(j\omega)$ and weighting filter $W(j\omega)$, the following statements are equivalent:

a. There exists a controller K that stabilizes G and

$$\sup_{\omega\in\mathbb{R}\cup\{\infty\}} |W(I+GK)^{-1}| < \gamma. \tag{3-6}$$

b. There exist $\{X,Y\} \in \mathcal{RH}_\infty$ with $K = XY^{-1}$, such that

$$\gamma^{-1}|WMY|(j\omega) < \Re\left\{[NX+MY](j\omega)\right\}, \forall\omega \in \mathbb{R}\cup\{\infty\}. \tag{3-7}$$

The proof of this theorem can be found in [5]. The theorem follows from Lemma 3.1, but there is an important difference: the transfer function α is absorbed by the controller. Consider the condition in (3-5), this can be rewritten as follows such that α is absorbed. Let the controller be given by $K = \bar{X}\bar{Y}^{-1}$ with $\{\bar{X},\bar{Y}\} \in \mathcal{RH}_\infty$, then we can write

$$\Re\{N(j\omega)\bar{X}(j\omega)\alpha(j\omega) + M(j\omega)\bar{Y}(j\omega)\alpha(j\omega) - \gamma^{-1}|WM\bar{Y}\alpha|(j\omega)\} > 0. \tag{3-8}$$

Let $X = \bar{X}\alpha$ and $Y = \bar{Y}\alpha$. It follows that $\{X,Y\} \in \mathcal{RH}_\infty$ because $\{\alpha,\bar{X},\bar{Y}\} \in \mathcal{RH}_\infty$,

$$\Re\{N(j\omega)X(j\omega) + M(j\omega)Y(j\omega) - \gamma^{-1}|WMY|(j\omega)\} > 0. \tag{3-9}$$

The trade-off is that the order of the controller parametrization increases, because part of α's structure is inside of the controller. However, α cancels during realization of the controller.

The condition in (3-7) is used to formulate the following optimization problem to synthesise a fixed-order controller:

$$\begin{aligned} &\min_{\gamma,X,Y} \gamma \\ &\text{s.t. } \Re\{N(j\omega)X(j\omega) + M(j\omega)Y(j\omega) - \gamma^{-1}|WMY|(j\omega)\} > 0, \forall\omega \in \mathbb{R} \cup \{\infty\}. \end{aligned} \tag{3-10}$$

This is a non-convex optimization problem, but by linearly parametrizing the controller's coprime factors X, Y the constraint in (3-10) becomes quasi-convex. The coprime factors $X(s)$ and $Y(s)$ are linearly parametrized as follows:

$$\begin{aligned} X(s,\rho_x) &= \rho_x^T \phi(s,\xi), \\ Y(s,\rho_y) &= \rho_y^T \phi(s,\xi), \end{aligned} \tag{3-11}$$

where $\xi > 0$ is the pole location $\phi(s,\xi)$ and $\rho_x \in \mathbb{R}^{n_x}, \rho_y \in \mathbb{R}^{n_y}$ are the control parameters, and $\phi(s,\xi)$ are vectors of stable transfer functions chosen from any set of orthogonal basis functions (OBF), such as pulse basis (discrete-time only), Kautz basis, generalized OBFs or Laguerre basis [33]. The latter are given by

$$\begin{aligned} \phi_0(s) &= 1, \\ \phi_n(s,\xi) &= \frac{\sqrt{2\xi}(s-\xi)^{n-1}}{(s+\xi)^n}, \end{aligned} \tag{3-12}$$

where $n \geq 1$ is the order of the basis function. The quasi-convex optimization problem is then solved iteratively by fixing γ and implementing a bisection algorithm over γ [6]. Furthermore, the set for which the constraint in (3-10) has to be satisfied is infinite, while there are a limited number of optimization variables. This is a semi-infinite programming (SIP) optimization problem, but it can be turned into a semi-definite programming (SDP) problem. In the data-driven setting there is a limited set of data available (FRFs), so there is a finite grid over which the constraints have to be satisfied. This set of FRFs can be obtained experimentally through the frequency responses of the system. The proof for the existence of a solution to this convex optimization problem can be found in [5]. It is proven that as the order of the controller increases, the convex optimization converges to the global minimum. For higher order controllers the choice of the basis functions does not matter for the optimization problem. However, for low order (fixed-structure) controllers, the choice of the basis functions is important to reach an optimal solution. Hence, a three-step procedure is presented in [5] for low order controller design.

3-2-2 Fixed-structure Controller Synthesis

The motivation for developing the method in [6] is due to the fact that high-order controllers are sometimes difficult or even impossible to implement in practice. For low-order controller design the previous method may find an optimal solution that is far from the global solution, and it is sensitive to the pre-set values of the basis function parameters [6]. An alternative method for low-order, i.e. fixed-structured, controllers is presented by using the results of Lemma 3.1. The controller parametrization is similar to (3-11). However, the basis function $\phi(s,\xi)$ is not necessary for the controller structure, since a fixed-structure controller is

designed. The controller $K(s) = X(s)Y^{-1}(s)$ is parametrized as follows:

$$\begin{aligned} X(s, \rho_x) &= \rho_x^T \zeta(s), \\ Y(s, \rho_y) &= \rho_y^T \zeta(s), \end{aligned} \tag{3-13}$$

where ρ_x and ρ_y are the control parameters and $\zeta(s)$ is the transfer function of the structure of the controller. For example, a PI controller $K(s) = k_p + k_i/s$ can be parametrized as in (3-13) by taking

$$\begin{aligned} \rho_x^T &= \begin{bmatrix} k_p & k_i \end{bmatrix}, \\ \rho_y^T &= \begin{bmatrix} 1 & 0 \end{bmatrix}, \\ \zeta(s) &= \begin{bmatrix} s & 1 \end{bmatrix} (s+1)^{-1}. \end{aligned} \tag{3-14}$$

Because the structure is predetermined and fixed, it is not possible to absorb the structure of α in condition (3-5) inside the structure of the controller. Therefore, it is not possible to formulate a convex optimization problem as opposed to the fixed-order controller design. The results of Lemma 3.1 are used to formulate the following theorem given in [6].

Theorem 3.2. The local optimal solution for obtaining $\mathcal{H}_\infty$ performance and closed-loop stability using the fixed-structure controllers X and Y is achieved if α is parametrized with a set of stable orthogonal basis functions and the following optimization problem is realized:

$$\begin{aligned} &\min_{\gamma, X, Y} \gamma \\ &\text{s.t. } \gamma^{-1} |WMY\alpha|(j\omega) < \Re\{D_H(j\omega)\alpha(j\omega)\}, \forall \omega \in \mathbb{R} \cup \{\infty\}. \end{aligned} \tag{3-15}$$

The proof is provided in [6]. The transfer function $\alpha(s)$ can be parametrized with any OBF, take for example the Laguerre basis functions (3-12), such that $\alpha(s, \rho_\alpha) = \rho_\alpha^T \phi(s, \xi_\alpha)$ with $\rho_\alpha \in \mathbb{R}^{n_\alpha}$. The optimization problem in (3-15) can be reformulated as:

$$\begin{aligned} &\min_{\gamma, \rho, \rho_\alpha} \gamma \\ &\text{s.t. } \Re\{(D_H(j\omega, \rho) - \gamma^{-1} |W(j\omega)M(j\omega)Y(j\omega, \rho)|)\alpha(j\omega, \rho_\alpha)\} > 0, \forall \omega \in \mathbb{R} \cup \{\infty\}. \end{aligned} \tag{3-16}$$

The proof for the existence of a solution to this optimization problem can be found in [6]. Similar to the optimization problem in (3-10), an infinite amount of constraints have to be evaluated in (3-16). The SIP optimization problem is transformation into an SDP problem. The authors propose to solve the optimization problem with either Bilinear Programming or PSO (see Section 2-5).

3-3 Fixed-structure controller design using Nyquist criterion

The preceding methods only work for SISO systems. The following method has the advantage that it is in the general PK form and works for MIMO systems as well. An $\mathcal{H}_\infty$ control design problem is formulated in [7]. This approach aims to use FRF data to design fixed structure controllers. Line constraints are introduced to satisfy Nyquist stability, and performance requirements. Consider the weighted plant in Figure 3-3, where it is assumed that G is

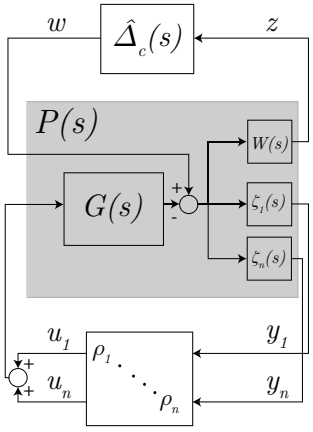

Figure 3-3: Weighted generalized plant with controller structure absorbed inside the plant.

stable. The controller is linearly parametrized such that the structure of the controller $Z(s) = \begin{bmatrix}\zeta_1(s) & \dots & \zeta_n(s)\end{bmatrix}^T$ is absorbed by the plant P. As a result the controller K is given by

$$K(\rho) = \begin{bmatrix}\rho_1 & & 0 \\ & \ddots & \\ 0 & & \rho_n\end{bmatrix}, \tag{3-17}$$

where $[\rho_1, ..., \rho_n]$ are control parameters. The diagonal controller structure is obtained using the fact that any rational, well-posed function $R(b)$ can be written as a lower LFT [34]:

$$R(b) = \mathcal{F}_\ell(M, b \otimes I), \tag{3-18}$$

where M is a fixed matrix, the symbol $\otimes$ is the Kronecker-product and $b \otimes I$ is the diagonal matrix containing the parameters b. The closed-loop system can be found using the lower LFT in (2-9). The goal is to design a controller that guarantees nominal performance:

$$||(P \star K)(j\omega)||_\infty \leq 1, \quad \forall \omega \in [0, \infty]. \tag{3-19}$$

Stability and performance requirements are achieved through Theorems 2.4 and 2.8. Let $\hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}$, with $\mathbf{B}\hat{\mathbf{\Delta}}$ as defined in (2-29), be a virtual uncertainty placed directly on the performance channel, as in Figure 2-7, that represents the $\mathcal{H}_\infty$ performance specification. Consequently, the following theorem for a condition on performance is given:

Theorem 3.3. The performance requirement in (3-19) is satisfied if, for a given generalized plant $P(j\omega, \rho)$, the Nyquist plot of

$$Q_\Delta(j\omega) = \det\left(I - P(j\omega)\left[\begin{array}{c|ccc}\hat{\Delta}(j\omega) & 0 & 0 & 0 \\ \hline 0 & \rho_1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \rho_n\end{array}\right]\right), \forall \omega, \forall \hat{\Delta}(j\omega), \tag{3-20}$$

does not encircle the origin for any stable rational transfer function $\hat{\Delta} \in \mathbf{B}\mathbf{\Delta}$ [7].

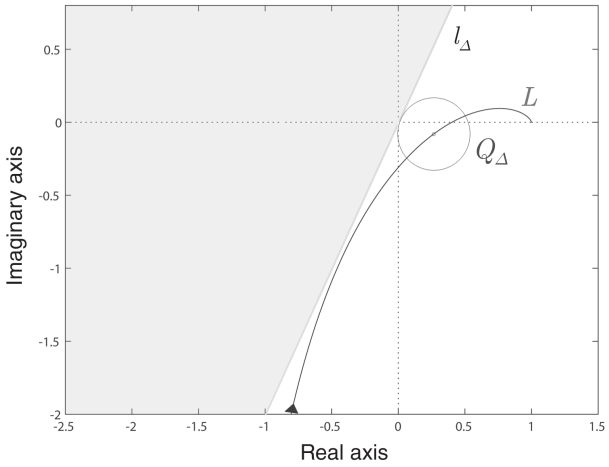

Figure 3-4: The Nyquist curve of $L(j\omega) = P(j\omega)K(j\omega)$, the line constraint $l_\Delta(j\omega, \rho)$ and the circle represents the performance $Q_\Delta(j\omega, \rho)$. The Nyquist curve and the circle are prohibited from entering the grey region by the line l_Δ.

The determinant condition can be worked out further to find that (3-20) is bilinear in ρ or multilinear in case of more control parameters [7]. This is used to formulate the following line constraint dependent on (3-20)

$$l_\Delta(j\omega, \rho,) = -\Im(Q_\Delta(j\omega, \rho)) + \kappa\Re(Q_\Delta(j\omega, \rho)) + c, \tag{3-21}$$

where κ is the slope of the line and c is the offset. This constraint prohibits the Nyquist curve from encircling the origin or going through it. The line constraints can be assigned per frequency point, thus it is possible to assign multiple lines to specific frequencies. The idea behind the line constraints is shown graphically in Figure 3-4. It should be noted that the line constraints are dependent on $\hat{\Delta}$ as well. This implies that an infinite number of constraints have to be evaluated. This problem can be solved by taking n_d realizations of $\hat{\Delta}$, denoted by $\bar{\Delta}$. The realizations are random points drawn on the unit circle, which means that $\bar{\sigma}(\bar{\Delta}) \leq 1$. The realizations of $\bar{\Delta}$ are in the set $\mathbf{B}\bar{\mathbf{\Delta}} := \{\bar{\Delta} \,|\, \bar{\sigma}(\bar{\Delta}(j\omega)) = 1\}$, with $\dim(\bar{\Delta}) = \dim(P)$. The number of line constraints that are evaluated is proportional to the number of realization n_d. In [7] it has been suggested that an amount of $n_d = 100$ is often sufficient to make the probability of missing critical uncertainties small. The line constraints depend on the controller parameters ρ and the frequency $\omega_k \in \Omega_p$, where k is the number of frequency points in the finite set Ω_p, which contains the frequency grid. For performance (and stability) the feasibility problem is as follows [7]

$$\begin{aligned} &\min_{\rho} \quad 1, \\ &\text{s.t. } l_\Delta(j\omega, \rho) < 0, \;\; \forall\omega \in \Omega_p, \;\; \forall\bar{\Delta} \in \mathbf{B}\bar{\mathbf{\Delta}}. \end{aligned} \tag{3-22}$$

The feasibility problem can be solved using readily available solvers, such as `fmincon` function in MATLAB. The line constraints have to be tuned to achieve desirable performance, since it is not an optimization problem. This requires some experience with the method and with the system for which a controller has to be designed. Designing a controller using this method

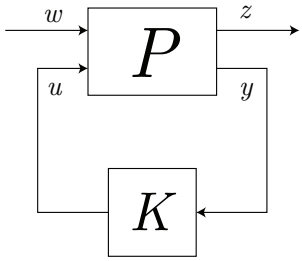

Figure 3-5: Generalized plant.

is time consuming, for me, personally. The frequency points where one wants to add a line constraint have to be selected carefully if optimal performance is desired, this process can be tedious and requires manual tuning.

3-4 Feedback Autotuner

As opposed to the previous method, the Feedback Autotuner (FBA) by ASML requires little effort to find a stabilizing controller. FBA uses the Nyquist theorem (see Theorem 2.3) to assess the stability of the closed-loop system. The stabilizing control parameters are then optimised by imposing $\mathcal{H}_\infty$ performance specifications on the closed-loop system.
Consider the generalized plant in Figure 3-5, where P is the plant and K is the controller. The plant is partitioned as follows:

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} P_{zw}(s) & P_{zu}(s) \\ P_{yw}(s) & P_{yu}(s) \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}. \tag{3-23}$$

The Nyquist theorem requires the transfer function of the open-loop interconnection, which is given by

$$L(s) = P_{yu}(s)K(s). \tag{3-24}$$

The closed-loop system is denoted as $H(s) = P(s) \star K(s)$. FBA applies a two-step controller synthesis procedure. The first step is to assess the stability of the open-loop system using the Nyquist theorem. In the data-driven setting the open-loop frequency response $L(j\omega)$ can readily be acquired through experiments. This can in turn be used to plot the Nyquist curve of $L(j\omega)$. The direction of the encirclements and the number of encirclements around the critical point are analysed to check if the condition $Z = N + P_{ol}$ in Theorem 2.3 is satisfied. The number of open-loop RHP poles (denoted as P_{ol}) of the system have to be known in advance. The direction and number of encirclements N can be observed in the Nyquist plot. The results of Theorem 2.3 are used to formulate a stability test, consider the following corollary.

Corollary 3.1. Let $L(j\omega)$ be the Nyquist curve of the SISO open-loop frequency response. The number of encirclements that $L(j\omega)$ makes around the critical point $(-1, 0j)$ is denoted as N, where clockwise encirclements have a positive sign and counter-clockwise encirclements have a negative sign. Let P_{ol} be the number of open-loop RHP poles and $Z = N + P_{ol}$ the number of closed-loop RHP poles. The closed-loop system $H(j\omega) = (P \star K)(j\omega)$ is stable if and only if

1. $L(j\omega)$ does not go through the critical point $(-1, 0j)$;
2. $Z = 0$.

The process of counting encirclements can be automated by using Corollary 3.1. The heuristic algorithm for finding the intersections is given in Algorithm 1. Let V be the vertical line going up from $(-1, 0j)$ and H_l be the horizontal line starting in $(-1, 0j)$ going to towards negative infinity. The intersections of the Nyquist curve of $L(j\omega)$ with V are denoted as $(-1, y_0)$ and the intersection of $L(j\omega)$ with H_l as $(x_0, 0)$. The index at which frequency the intersection with V occurs is denoted as i_V. The workings of the algorithm are shown graphically in Figure 3-6. The red dashed lines are the vertical and horizontal lines. The horizontal line is there to make sure that $(-1, 0j)$ is encircled by the Nyquist curve. In Figure 3-6 we can see that the solid dark grey curve has two clockwise intersections with the vertical line. However, since there is no intersection with the horizontal line we can conclude that there is no encirclement around the critical point. The dashed black Nyquist curve does intersect with both the horizontal and vertical line. The direction of the encirclement can be seen in the Nyquist plot in Figure 3-6, where clockwise encirclements are denoted as blue dots and counter-clockwise encirclements denoted as red dots. The algorithm determines the direction of the encirclement by using the index of the frequency at which the the intersection occurs: i_v, see lines 15 to 23 in Algorithm 1. If the intersection is at $L(j\omega_{i_V})$ then the difference between a point before $L(j\omega_{i_V-1})$ and after $L(j\omega_{i_V+1})$ the intersection can tell us the direction of the Nyquist curve. If the difference is positive, then the encirclement is clockwise and if the difference is negative, then the encirclement is counter-clockwise. The encirclements are denoted as N_{dir} and are evaluated at every intersection k. Finally, the stability test also takes encirclements due to integrators into account, denoted as N_{int} in Algorithm 1. Integrators cause the Nyquist curve to go to infinity, so we do not get a closed Nyquist curve and therefore it is not possible to count all the intersections. The sign and number of encirclements due to integrators are $N_{int} = \lfloor \frac{n_i}{2} \rfloor$. Once all the encirclements have been determined, they are summed up to find the total number of encirclements $N = N_{dir} + N_{int}$. Finally, N and P_{ol} are summed up to determine if $Z = 0$, i.e. if the closed-loop system is stable.

Once it has been determined that the closed-loop system is stable for a range of control parameters, the next step can take place: optimizing the control parameters for optimal performance. The goal is to achieve nominal performance:

$$\|W(j\omega)H(j\omega)\|_\infty \leq \gamma, \, \forall \omega \in \Omega_p, \tag{3-25}$$

where $W(j\omega)$ is a proper and stable weighting filter, $\gamma > 0$ and Ω_p is a finite frequency grid. FBA is used to design fixed-structure controllers, therefore the optimization problem is non-convex. The optimization problem is solved using PSO (see Section 2-5). In Algorithm 2 it is shown how PSO works in this setting. First, it looks for stabilizing set of control parameters, this is its initial population. Next, within the range of the stabilizing control parameters PSO looks for the optimal values to minimize γ on the finite frequency grid Ω_p.

At ASML they have experience with their systems and know if there are any open-loop poles and they know the typical values for the control parameters for which the system is stable. This saves time in the initialization process. Therefore, this method works best if there is prior knowledge with regards to the system.

Algorithm 1: Stability test for SISO systems.

`function` stability_test $(L(j\omega), P_{ol}, n_i)$ **Input :** $L(j\omega)$: frequency response of the open-loop transfer function
P_{ol}: number of open-loop RHP poles
n_i: number of integrators
Output: Number of closed-loop RHP poles Z
if $(x_0, y_0) = (-1, 0j)$ `//` (x_0, y_0) `are the coordinates of the intersections with vertical line` V **then**
  $L(j\omega)$ goes through $(-1, 0j)$
  stability cannot be determined
  return
end
if x_0 *or* y_0 *is empty* **then**
  No intersection with V and H_l `//` H_l `is the horizontal line`
  $N = 0$
else
  Let i_V be the index of the frequency at which the intersection with V occurs.
end
if i_V *is empty* **then**
  No intersection with V
  $N = 0$
else
  for k = *number of intersections* **do**
    if $L(j\omega_{i_V+1}) - L(j\omega_{i_V-1}) > 0$ **then**
      Sign of encirclement is positive
      $N_{dir}(k) = +1$
    else
      Sign of encirclement is negative
      $N_{dir}(k) = -1$
    end
  end
end
if $n_i > 1$ **then**
  $N_{int} = +\lfloor \frac{n_i}{2} \rfloor$
end
$N = N_{dir}(1) + ... + N_{dir}(k) + N_{int}$
$Z = N + P_{ol}$
if $Z = 0$ **then**
  No closed-loop RHP poles
  Closed-loop system is stable
else
  Closed-loop system is unstable
end

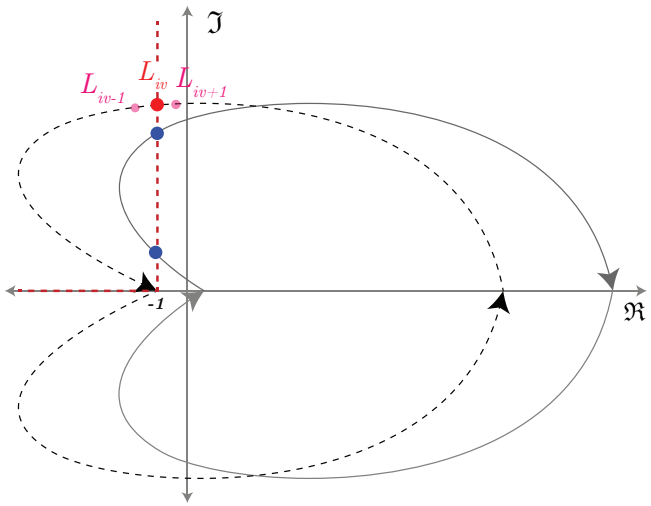

Figure 3-6: Nyquist curve of two systems. Both curves intersect with the red vertical dashed line, however only the dashed curve intersects with both red dashed lines and thus encircles the origin. The red dots denote CCW encirclements and the blue dots denote CW encirclements.

Algorithm 2: FBA PSO algorithm for SISO systems.

Initialize a population of size i control parameter $\rho = [\rho_1 ... \rho_i]$ particles within a random or known range of control parameters.

for *every set of controller parameters* ρ_k *at iteration* k **do**
  Stability test of Algorithm 1
  if ρ_k *renders the closed-loop system stable at iteration* k **then**
    Compute the performance specification in (3-25)
  else
    Flag the control parameters ρ_k at iteration k as infeasible
  end
end
for *all stabilzing control parameters* $[\rho_1 ... \rho_i]_k$ *at iteration* k **do**
  Minimize γ in (3-25)
end

Method	*1*	*2*	*3*	*4*
Convexity	✓	✗	✗	✗
Unstable plant	✓	✓	✗	✓
MIMO	✗	✗	✓	✓*
No prior knowledge required	✓	✓	✗	✗
Fixed-structure controller	✗	✓	✓	✓
PK form	✗	✗	✓	✓

Table 3-1: The properties of the methods as ordered in Section 3-1: 1 is the fixed-order controller by Karimi et al., 2 is the fixed-structure controller by Nicoletti et al., 3 is the line constraints method by van Solingen et al. and 4 is FBA by van der Meulen. * *only through sequential loop closing.*

3-5 Overview and comparison of the methods

Four methods of the state-of-the-art have been presented in the preceding sections. Each of these methods has also been implemented during the internship and master thesis project. Table 3-1 provides an overview of the properties of these methods. We can see that only the first method by Karimi et al. results in a convex optimization problem, which is of course preferred because only then it is ensured that a globally optimal controller is designed. However, this comes with the price of losing the possibility to design a fixed-structure controller. Moreover, for lower order controllers, the results depend on the choice of the basis functions and it does not necessarily give optimal results with respect to the desired performance specifications. The second method by Nicoletti et al. aims to solve this issue by using the same theorems and it is presented as a complementary method to design for fixed-structure controllers, which are typically of lower order. How do these two methods compare to the third method by van Solingen et al.? Graphically, the three methods *seem* the same: introduce a line (albeit an imaginary line for the first two methods) in the complex plane to prohibit the Nyquist curve from encircling or going through the origin. However, the constraint in (3-5) (by Karimi et al.) is much less conservative than the one in (3-21) (by Solingen et al.). The latter constraint creates an entire prohibited region, that is manually designed by the engineer, which the Nyquist curve cannot enter, thus resulting in a conservative constraint. The former constraint (by Karimi et al.) calls for the existence of a complex function α that can rotate the disc such that it lies in the RHP, which can be interpreted as an imaginary line that does not intersect the disc. Aside from the constraints, the first two methods are optimization problems and the third is a feasibility problem, which suggests that the controller is not necessarily designed for optimal performance. Furthermore, the line constraints have to be tuned manually if the performance requirements are not met.
The first three methods derive conditions such that the (generalized) Nyquist Theorem and the Main Loop Theorem are satisfied. In this respect, FBA is more of a heuristic approach to satisfy the Nyquist Theorem for stability. Its properties make it an attractive control design approach, because it is applicable for unstable plants, as opposed to the method by Solingen et al. and it is also in the general PK form. The biggest downside of the FBA method is that prior knowledge of the system is required, while with the first two methods that is not necessary.
Based on the properties in Table 3-1, the preceding discussion and my experience with the

methods, the first two methods and FBA will be used as a basis to develop new LPV control design methods. The convexity of the first method makes it an attractive approach and the second (complimentary) method is useful in case a lower order controller has to be designed. FBA stands out because it works for MIMO systems and it is formulated in the general PK form. These three methods are used as the building blocks for the new LPV methods presented in Chapter 5.

3-6 Conclusions

While the LTI framework has a vast, strong and relatively simple theory to design robust controllers, it is still limited and can lead to poor controller design in certain applications. The most obvious limitation is that LTI systems do not exist in the real world. While it is possible to design stabilizing LTI controllers for complex systems in the industry, it does not mean it is possible to design controllers that satisfy tight and more demanding performance requirements. Take for example an industrial wafer stage, which has nonlinear position-dependent dynamics. At different operating points, the resonance peaks shift and can limit the achievable bandwidth. It is possible to model these position-dependent dynamics as uncertainties in the LTI system for which a robust controller can be designed. However, this can cause conservatism with limitations on the achievable performance [35].
One way to take the position-dependent dynamics into account is by designing an LPV controller. The next chapter introduces the theory behind the LPV framework.

Chapter 4

LPV Framework

When I first got introduced to the set of Complex numbers, my mathematics teacher told me "By learning the set of Complex numbers, you learn to understand the Real numbers better". Furthermore, by using Complex numbers, we can accurately describe natural phenomena that otherwise would not be as accurate or possible with only Real numbers (e.g. wave functions). The LPV framework can be seen as the Complex numbers and the LTI framework as the Real numbers. The LTI framework is part of the LPV framework, it can be seen as a special case. This implies that the extensive LTI theory can also be used in the LPV framework to synthesise a controller. While we can design stabilizing controllers with robust performance in the LTI framework, the maximum performance we can achieve is still inherently limited, compared to the LPV framework where the system variations can be handled by a varying control structure. The limitations of the LTI framework can be overcome in the LPV framework, because the representation is more flexible to express nonlinear and time-varying system behaviour.
In this chapter, relevant theory from the LPV framework is presented, such as the local and global approach, parameter dependence and LFR representation of LPV systems. The concept of "frozen" FRFs is explained in Section 4-1-5. General steps for LPV controller synthesis are given in Section 4-2. Finally, an overview of what is available in the field of research of LPV controller synthesis is presented in 4-3.

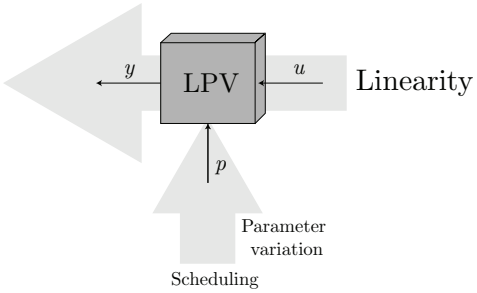

Figure 4-1: Schematic input-output signal view of an LPV system.

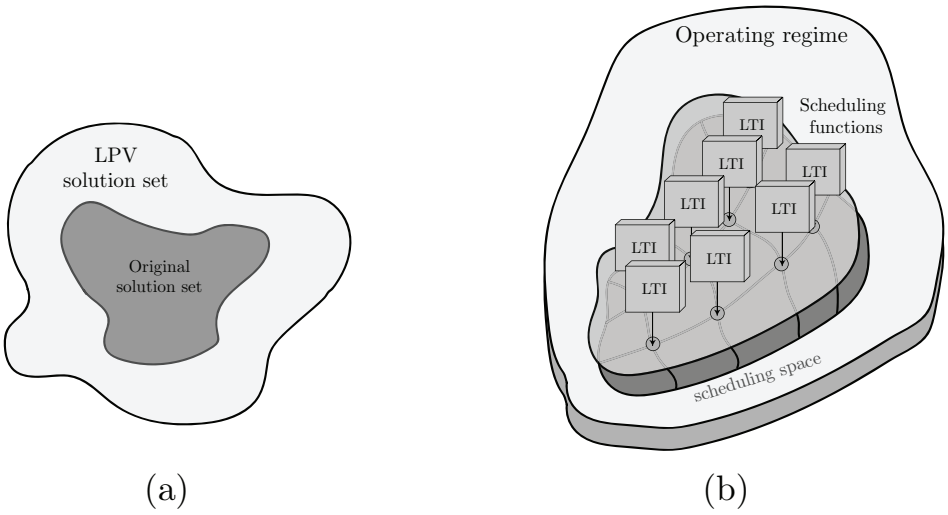

Figure 4-2: Global approach (a) and local approach (b) to model an LPV system.

4-1 System Definition

4-1-1 Local approach vs. Global approach

Given a nonlinear system which has to operate under a range of operating conditions. There are three types of controller that can be designed for this system: a robust LTI controller, a nonlinear controller or an LPV controller. In the first case, the system might require a very robust controller which results in relatively poor performance. Secondly, a nonlinear controller may provide good performance, however, nonlinear controller design is complex and mainly relies on stability, while a systematic procedure to design for performance is missing. The final option seems like the best of both worlds: a robust controller with good performance, because it takes nonlinearities in the form of varying parameters into account.
A schematic view of an LPV system is given in Figure 4-1, which depicts the linear dynamical relation between the input and output signals. In general, there are two ways to obtain an LPV description of a nonlinear plant, either with the global or local approach.

Global Approach Suppose a given system G is nonlinear or time-dependent, then in Figure 4-2 the dark grey solution set belongs to that system. All possible signals are contained in the corresponding set. The idea is to create a proxy description or an embedding by pulling out a latent variable and putting it back into the system, see Figure 4-3. In this way, a linear relationship is obtained that depends on a time-dependent scheduling variable p. Let p, for example, be a state variable. Then the underlying concept is that when p is pulled out it can take on values of the state at least in the original solution set and beyond. That is because for each (discrete) time step we take values from the whole space of $p \in \mathbb{P}$. This results in a behaviour that is at least as large as the original solution set, see the light grey LPV solution set in Figure 4-2. If we have an LPV system where the scheduling variable is a function of the state, the behaviour is always larger than the original, the original is included in the set but there is always some conservativism because p is sampled separately. If the scheduling variable is external, then there is no conservativism. The behaviour set is the same size in

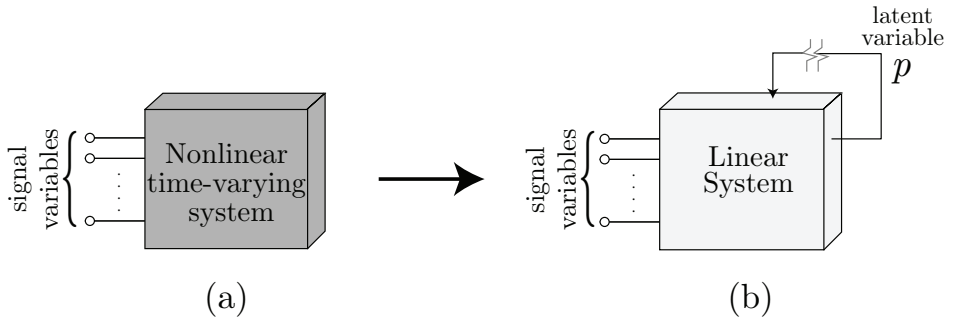

Figure 4-3: By taking out a latent variable p an embedding of the original nonlinear time-varying system (4-3a) is created (4-3b). In Figure 4-2a it is shown schematically how the LPV solution set is an embedding of the original solution set.

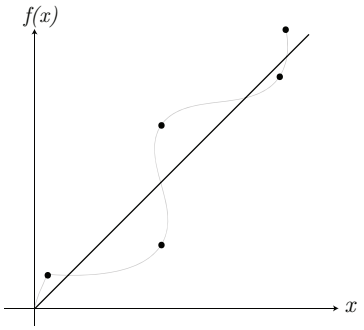

Figure 4-4: Classic gain-scheduling requires interpolation of the controllers for every grid point. During interpolation some information on the grid points may be lost, depending on the interpolation technique.

that case. The main advantage of the global approach is that we can guarantee the system is globally stable and meets the performance specifications globally as well.

Local Approach Alternatively, it is possible to look at the local behaviour of a non-linear system, this is called the local approach. The system is "frozen" at an operating point and this results in a linear behaviour, so the system is actually linearised. These are referred to as frozen behaviours. This process is repeated for a number of operating points, this way a number of LTI grid points are extracted from the whole behaviour set of the system, see Figure 4-2b. The idea is that all LTI behaviours on the grid points are then interpolated depending on the scheduling variable.

It is important to emphasize the difference between the local approach and classic gain-scheduling. The latter works as follows: given a nonlinear system which is linearised on a number of grid points. By doing so, linearised models for the system are obtained. A controller is then designed for each grid point separately. These controllers are interpolated dependent on the scheduling variable. The disadvantage is that the results depend on the interpolation techniques used and it is uncertain what happens between the grid points. It may even be undetermined what will happen on a grid point, for example with state-space matrices, see also Figure 4-4. Note that it is possible to use interpolation techniques that do pass through the grid points, but they are computationally more demanding.

The difference with the local approach is that by using LPV theory, the interpolation, as it were, is built into the controller. This results in a global parametrisation of the controller. The local behaviour of the controller is tuned using local models. Therefore, the advantage is that we can guarantee with certainty that the stability and performance requirements are met on the grid points. However, it is still uncertain whether the stability and performance requirements are met between the operating points.

Global vs. Local Approach It seems that the global approach would be favourable over the local approach, seeing as it provides an embedding of the global dynamics behaviour of the system and global stability guarantees. However, the process of acquiring a model for the global approach is more demanding, especially in the industrial setting. There has to be enough possibility to perturb the system for an informative data record and the model specifications should not be given in the frequency domain [11]. Whereas for the local approach, the model of the system consists of the frozen dynamics, which are simply LTI systems at a range of operating points. Hence, it is possible to use the full potential of the LTI framework to design an LPV controller. The LPV framework becomes an extension of the LTI framework to meet tighter performance specifications which may have been impossible to satisfy before.

4-1-2 State-space representation

While the focus is on data-driven control in the frequency-domain, it is useful to gain a better understanding of LPV systems by discussing the state-space representation. The state-space description of an LPV system is given as follows:

$$\begin{aligned} \dot{x}(t) &= A(p(t))x(t) + B(p(t))w(t) \\ z(t) &= C(p(t))x(t) + D(p(t))w(t) \\ x(0) &= x_0, \end{aligned} \tag{4-1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $z(t) \in \mathbb{R}^m$ is the output, $w(t) \in \mathbb{R}^k$ is the input and $p(t) \in \mathbb{R}^p$ is the scheduling variable vector. The scheduling variable itself is a projection of measurable variables in the system, i.e. input, output or state signals $p(t) = \mu(x(t), w(t))$. In the local setting it is assumed that during design and modelling the scheduling variable's trajectory is constant and that it is independent, consequently, this introduces conservatism into the system's description. It is assumed that p is known or measurable. Furthermore, p is always assumed to be bounded and takes values inside the convex set $p \in \mathbb{P}$, because the model targets a certain operating range. The general guideline for the local approach is that slow variation of the scheduling variable is preferable, because this ensures that the controller is not going to suffer from the scheduled operating conditions. However, this depends on the type of system that the control designer deals with. For some systems the scheduling variable's speed can be fast and some slow, see e.g. [36, 37]. The global approach, on the other hand, can guarantee stability and performance for arbitrarily fast variations.

4-1-3 Parameter dependence

There are several ways the system can depend on the scheduling variable p, of which three are presented. Affine parameter dependence is when the system matrices are defined as:

$$A(p(t)) = A_0 + A_1 p(t) + ... + A_n p_n(t). \tag{4-2}$$

Consider, for example, a mass-spring system with a varying spring parameter $\ddot{x} + k(t)x = u$, with $k \in [1, 2]$. The system matrix $A = \left[\begin{smallmatrix} 0 & 1 \\ -k(t) & 0 \end{smallmatrix}\right]$ can be written in affine form as

$$A(k(t)) = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A_0} + \underbrace{\begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}}_{A_1} k(t). \tag{4-3}$$

If the system matrix depends polynomially on the scheduling variable, it is given as:

$$A(p(t)) = A_0 + A_1 p(t) + A_2 p(t)^2 + A_3 p(t)^3 + ... + A_n p(t)^n. \tag{4-4}$$

An example of such parameter dependence can be shown using a simple example from [38]. Consider the nonlinear system

$$\dot{x}(t) = x(t)^3, \tag{4-5}$$

it can be written as an LPV system with polynomial parameter dependence by taking $p(t) = x(t)$

$$\dot{x}(t) = -p(t)^2 x(t). \tag{4-6}$$

Finally, the rational parameter dependence is when the system matrices are defined as:

$$A(p(t)) = \left[A_{n_0} + A_{n_1} p_{n_1} + ... + A_{n_N} p_{n_N}\right] \left[I + A_{d_1} p_{d_1} + ... + A_{d_D} p_{d_D}\right]^{-1} \tag{4-7}$$

For example, the following system has rational parameter dependence:

$$\dot{x}(t) = \left(\frac{p(t)}{p(t)^2 + 1} - 3\right) x(t). \tag{4-8}$$

The general dependence goes beyond the preceding types, such as meromorphic functions or local unit truncation (LUT) and more [11].

4-1-4 LFR representation of LPV systems

In Section 2-1-3 LFRs were introduced to represent uncertain systems. The same principle can be applied to model LPV systems dependent on the scheduling variable. In the case of LPV systems the LFR consists of two parts: the LTI system part and the parameter varying part. The latter can be viewed as the time-varying/nonlinear uncertainty that is extracted from the LTI system, see Figure 4-5. LPV systems in the LFR form are represented by

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} &= \begin{bmatrix} A_1 & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}, \\ w &= \Delta(p) z, \end{aligned} \tag{4-9}$$

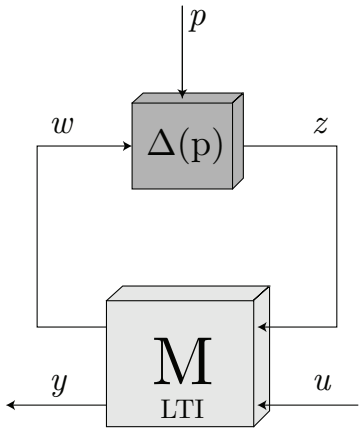

Figure 4-5: LFR form for an LPV system.

where x is the system state, w and z are the exogenous variables, $\{A_1, .., D_{22}\}$ are constant matrices and $\Delta = Ip$ is a function of p with linear static dependence [11]. Equation (4-9) can be rearranged to the standard LFT form as:

$$\begin{aligned} \left[\begin{array}{c} z \\ \hline \dot{x} \\ y \end{array}\right] &= \left[\begin{array}{c|cc} D_{11} & C_1 & D_{12} \\ \hline B_1 & A_1 & B_2 \\ D_{21} & C_2 & D_{22} \end{array}\right] \left[\begin{array}{c} w \\ \hline x \\ u \end{array}\right], \\ w &= \Delta(p) z. \end{aligned} \tag{4-10}$$

We can further simplify this, such that it resembles the structure in Figure 4-5

$$\begin{aligned} \left[\begin{array}{c} z \\ \hline y_M \end{array}\right] &= \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array}\right] \left[\begin{array}{c} w \\ \hline u_M \end{array}\right], \\ w &= \Delta(p) z. \end{aligned} \tag{4-11}$$

The LPV system can be found using the upper LFT as in (2-10)

$$\mathcal{F}_u(\Delta(p), M) = M_{22} + M_{21}\Delta(p) \left(I - M_{11}\Delta(p)\right)^{-1} M_{21}. \tag{4-12}$$

Note that $M_{11} = D_{11}$, thus (4-12) is well posed if the inverse of $(I - D_{11}\Delta(p))$ exists for all $p \in \mathbb{P}$. Moreover, if $D_{11} = 0$ and if $\Delta(p)$ is affine, then there is affine dependence on the scheduling variable, otherwise it can either be a polynomial or rational dependence.

4-1-5 Frozen Frequency Response Function

The LFR representation is useful in case a model of the system is available, for instance during the implementation and testing of LPV control design methods. In the LTI framework, FRFs are commonly used as a frequency-domain system representations (See Section 2-1-1). In the LPV framework, there are three available frequency-domain representations: instantaneous FRFs (iFRF), harmonic FRFs (hFRF) and frozen FRFs (fFRF). The iFRFs and hFRFs are dependent on the scheduling signal, as opposed to fFRFs. The iFRF provides a representation form where the FRF of the system embeds the past of the scheduling signal implicitly and

the hFRF represents the output spectrum for an arbitrary scheduling spectrum that is not directly embedded in the hFRF [39]. The fFRF is a local representation of the system and it is assumed that the scheduling variable is constant over time, for this reason it is used for (and limited to) the local approach (gain-scheduled control).
The fRF can be derived by first introducing the pulse response representation of SISO discrete-time LPV systems [11]:

$$y(t) = \sum_{\tau=0}^{\infty} \underbrace{g(p(t), p(t-1), ..., p(t-\tau), \tau)}_{g_{\mathbf{p},t}(\tau)} u(t-\tau), \tag{4-13}$$

where $u : \mathbb{Z} \to \mathbb{R}$ is the input, $y : \mathbb{Z} \to \mathbb{R}$ is the output, $p : \mathbb{Z} \to \mathbb{R}$ is the scheduling variable of the system and $g(\cdot, \tau)$ is a series of bounded functions which are convergent in case of asymptotic stability of the represented system. In case the scheduling variable is constant, $p(t) \equiv p_0$, then the pulse response $g_{\mathbf{p},t}(\tau)$ becomes time-invariant, which is denoted as $g_{p_0}(\tau)$. The input-output relation in (4-13) changes to:

$$y(t) = \sum_{\tau=0}^{\infty} g_{p_0} u(t-\tau). \tag{4-14}$$

The discrete-time Fourier transform (DTFT) is applied to (4-14) and it gives

$$Y(e^{j\omega}) = G_{p_0}(e^{j\omega}) U(e^{j\omega}), \tag{4-15}$$

where $G_{p_0}(e^{j\omega})$ can be found by applying the DTFT to g_{p_0} over the argument τ

$$G_{p_0}(e^{j\omega}) = \sum_{\tau=0}^{\infty} g_{p_0}(\tau) e^{j\omega\tau}. \tag{4-16}$$

The fFRF is represented by G_{p_0} for a constant scheduling variable. Although the fFRF offers insight into the dynamics of the LPV system, such an analysis is only valid if the scheduling variable is constant [39].

4-2 LPV gain-scheduled controller synthesis

The advantage of the local approach is that LTI controller synthesis methods, such as the ones discussed in Chapter 3, can be used to design a gain-scheduled LPV controller. In this section, a general three step procedure for LPV controller design is presented:

1. Formulate an LPV model;
2. Controller parametrization;
3. LPV controller synthesis.

Formulate an LPV model We assume that the LFR representation, discussed in Section 4-1-4, is used to describe the LPV system, in case a model of the system is available. Otherwise, we assume that a set of fFRFs of the system are available and that the scheduling variable $p \in \mathbb{P}$ is known, where $\mathbb{P}$ is the bounded set of operating points of interest.

Controller parametrization Similar to how the system can depend on the scheduling variable, the controller dependency on the scheduling variable can be affine, polynomial or rational, see Section 4-1-3. For example, a PID controller with affine dependence on p is given as:

$$K_{PID}(p) = k_{\mathrm{p}_0} + k_{\mathrm{p}_1} p + \frac{k_{\mathrm{i}_0} + k_{\mathrm{i}_1} p}{s} + \frac{(k_{\mathrm{d}_0} + k_{\mathrm{d}_1} p)s}{T_{\mathrm{f}} s + 1}. \tag{4-17}$$

LPV controller synthesis Once a non-parametric model of the system is acquired and a controller parametrization has been chosen, the synthesis step can take place. By using the local approach, we can synthesise the LPV controller similar to the LTI synthesis, as discussed in Chapter 3. Hence, the $\mathcal{H}_\infty$ criterion is used to synthesise an LPV controller. Stability and performance can be evaluated for every frozen FRF using the theorems presented in Chapter 2. The downside is that we can only guarantee local stability and performance for each frozen FRF and not globally for the nonlinear system as a whole. The issues of fixed-structure controller design remain the same in the LPV setting: a non-convex optimization problem has to be formulated and solved.

4-3 Overview of control design methods in LPV framework

Most of the control design methods developed in the LPV framework are in the time-domain and model-based. The state-space representation (4-1) is often used for LPV modelling and synthesis. An extensive survey of time-domain and model-based LPV control applications validated by experiments or high-fidelity simulations is presented in [40]. The authors briefly review and compare the discussed methods in their paper by classifying the methods into three techniques: polytopic approach introduced by [41], linear fractional transformation approach introduced by [42] and gridding-based approach introduced by [43]. In each of these approaches synthesis can be carried out as a convex optimization problem via a finite number of linear matrix inequalities (LMIs) for both parameter-independent and parameter-dependent Lyapunov functions [40]. For more information with regards to model-based time-domain LPV controller synthesis methods it is recommended to study the survey and the methods discussed within it.

Recently, the attention has shifted to frequency-domain controller synthesis in the LPV framework. This field of research is still in its infancy and therefore relatively few methods have been developed so far. The general approach of these LPV controller synthesis methods is that an LTI method is modified such that it works for LPV systems as well. The LTI setting can be seen as a "special" case of an LPV system: that is where the scheduling variable is constant.

An example of such an approach is the gain-scheduled controller design by linear programming presented in [44], which is based on the method that was developed earlier by [45]. A fixed-order controller is designed for stable SISO LPV plants. The method is based on shaping the multiple open-loop transfer functions for a number of scheduling variables in the Nyquist diagram. A controller can readily be computed from available FRFs of the LPV model and no interpolation is required. Another gain-scheduling method is presented in [46] to design an $\mathcal{H}_\infty$ controller by convex optimization. This method is also based on shaping the open-loop

transfer function in the Nyquist diagram, but now with constraints on the weighted infinity-norm. An improved version of the preceding two methods is given in [22, 47], where an FRF data-based controller synthesis approach is presented. The solution to the problem is obtained via convex optimization. It is an improvement with respect to [44, 46] in the sense that the desired control objectives are formulated in terms of a model matching problem, which allows performance specifications to be incorporated in the design through frequency domain shaping filters [47]. Finally, the LTI concept of Virtual Reference Feedback Tracking (VRFT) has been used to design gain-scheduled controllers for LPV systems, where in the first instance it was directly extended to the LPV framework by [19]. The method served as a groundwork for the improved method presented in [20], where fixed-order LPV model-reference controllers are designed using the VRFT concept for SISO LPV systems. The preceding methods only allow controller synthesis for SISO systems, which is a good start for a relatively new research field, but it is obvious that more research needs to be done to design for MIMO LPV systems as well.

The presented theory from this chapter is part of the building blocks that are necessary for the next chapter, where two novel LPV controller synthesis methods are presented.

Chapter 5

Controller Design Methods

In this chapter two novel LPV controller synthesis methods are presented. Both methods allow for the design of a fixed-structure controller for an unstable MIMO systems. The underlying idea behind the methods originates from the method by Nicoletti et al. and the Feedback Auto Tuner by S. van der Meulen (see Chapter 3). A convenient feature of these LPV methods is when the scheduling variable is unknown, it is possible to recover the LTI results. Therefore, the LTI case can be seen as a special case.

5-1 General Problem Setting

Both methods aim to solve the same control design problem. Thus, assume the following general problem setting. Consider a continuous time MIMO nonlinear system given by

$$\Gamma(t) = \begin{cases} \dot{x}(t) = f(x(t), w(t)), \\ z(t) = h(x(t), w(t)), \end{cases} \tag{5-1}$$

where $x(t) : \mathbb{R}^+ \to \mathbb{X} \subseteq \mathbb{R}^{n_x}$ is the state variable, $w(t) : \mathbb{R}^+ \to \mathbb{W} \subseteq \mathbb{R}^{n_w}$ is the generalized disturbance signal and $z(t) : \mathbb{R}^+ \to \mathbb{R}^{n_z}$ is the performance output of the system. We want to design a controller for this system not by modelling or identifying it, but rather via experimentally available data. While in the following derivations it is assumed that the data is sufficient, it should be stressed that in reality there is only limited amount of frequency response data, which is evaluated point wise.
The nonlinear system (5-1) is modelled as an LPV system that depends on a constant scheduling variable $p \in \mathbb{P}$. It is assumed that the scheduling map is known and that local frozen linear approximations of the system are available at the operating points $(x_0, w_0) \in \mathbb{X}_0 \times \mathbb{W}_0 \subset \mathbb{X} \times \mathbb{W}$. The set of frozen FRFs (fFRFs) is denoted as

$$\mathcal{G}(j\omega, p) = \{G_1(j\omega), ..., G_m(j\omega);\ \omega \in \Omega_p\}, \tag{5-2}$$

where Ω_p represents the finite available data set, m is the number of scheduling variables $p_1, \ldots, p_m$ and $G_m(j\omega)$ is an $n_z \times n_w$ matrix dependent on the convex set of scheduling

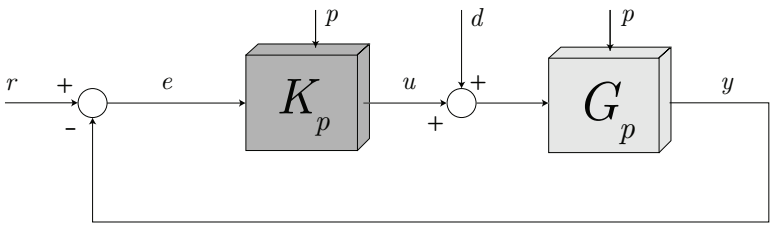

Figure 5-1: Standard LPV feedback interconnection for a 4-block problem.

variables $p = \mu(x_0, w_0) \in \mathcal{P} \subset \mathbb{P} = \mu(\mathbb{R}^{n_x}, \mathbb{R}^{n_w})$, the latter relates the operating conditions to the scheduling map $\mu(x_0, w_0)$. Whenever the scheduling variable is frozen, i.e. constant, it is assumed that local LTI behaviour of the system is obtained. This is a standard assumption for the local approach. The control design problem is approached from a local point of view by using the fFRFs, but it is important to note that the controller is a global parametrization.

From here on, the problem setting will continue in the relevant sections.

5-2 Fixed-structure LPV Controller with $\mathcal{H}_\infty$ performance

5-2-1 Problem Formulation

It is assumed that for each p the plant G_p and controller K_p can be represented as coprime factorizations over $\mathcal{RH}_\infty$ (see Definition 2.2), such that for each $p \in \mathcal{P}$ it follows that $G_p = M_p^{-1}(j\omega)N_p(j\omega)$ and $K_p = X_p(j\omega)Y_p^{-1}(j\omega)$. Here K_p represent the controller dependent on p. Due to the fact that the controller and plant are formulated in the coprime factorisation representation, design for unstable plants is possible as well (see Section 3-2). The interconnection in Figure 5-1 is used for the controller design, therefore w is given by $(w_1, w_2)^T = (r, d)^T$ and z is given by $(z_1, z_2)^T = (e, u)^T$. The input-output map $w \mapsto z$ in terms of coprime factorizations of G_p and K_p is given by

$$\begin{aligned} \mathcal{M}_p = \begin{bmatrix} I & M_p^{-1}N_p \\ -X_pY_p^{-1} & I \end{bmatrix}^{-1} &= \begin{bmatrix} S_p & G_pS_p \\ -K_pS_p & -T_p + I \end{bmatrix} \\ &= \begin{bmatrix} Y_p \\ X_p \end{bmatrix} U_p^{-1} \begin{bmatrix} M_p & -N_p \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, \end{aligned} \tag{5-3}$$

where $U_p = M_pY_p + N_pX_p$. Local internal stability is guaranteed if and only if $\mathcal{M}_p \in \mathcal{RH}_\infty$ [26]. From Theorem 2.1 we can derive that this amounts to checking if $U_p^{-1} \in \mathcal{RH}_\infty$, because all other terms in $\mathcal{M}_p$ are by definition in $\mathcal{RH}_\infty$ due to the use of coprime factorizations. Furthermore, if $\mathcal{M}_p$ is internally stable, so is H_p, which is given as

$$H_p = \begin{bmatrix} S_p & G_pS_p \\ -K_pS_p & -T_p \end{bmatrix} = \begin{bmatrix} Y_p \\ X_p \end{bmatrix} U_p^{-1} \begin{bmatrix} M_p & -N_p \end{bmatrix}, \tag{5-4}$$

because $\mathcal{RH}_\infty$ is closed under addition and multiplication. The addition of the identity $I \in \mathcal{RH}_\infty$ can therefore be dropped. Consequently, this allows for a mixed-sensitivity design. The control design problem is to synthesise an LPV controller K_p using fFRFs of the given system such that the following conditions are satisfied:

(C.I) the closed-loop system H_p in (5-4) is locally internally stable for every scheduling variable $p \in \mathcal{P}$;

(C.II) the 4-block performance objective, with $\gamma > 0$, satisfies

$$\|W_z H_p W_w\|_\infty \leq \gamma, \ \forall p \in \mathcal{P} \tag{5-5}$$

Here $W_z \in \mathcal{RH}_\infty$ and $W_w \in \mathcal{RH}_\infty$ are weighting filters, which can be used to achieve performance specifications. In Section 2-3-3 it has been explained how to design them.

5-2-2 Stability and Performance Analysis Conditions

Stability

The goal is to satisfy the 4-block test, i.e. the closed-loop transfer function matrix H_p should be internally stable for all $p \in \mathcal{P}$. Therefore, we have to check whether $U_p^{-1} \in \mathcal{RH}_\infty$ for all $p \in \mathcal{P}$ and if the latter is true, then $H_p \in \mathcal{RH}_\infty$ for all $p \in \mathcal{P}$. In [23] a theorem is presented, which gives a necessary and sufficient condition to guarantee stability for SISO systems. This theorem and its proof have been reformulated for MIMO systems such that it gives a necessary and sufficient condition on $G_p = M_p^{-1}N_p$ and $K_p = X_pY_p^{-1}$ to guarantee stability of H_p uniformly over p.

Theorem 5.1. Let $\{N_p, M_p, X_p, Y_p\} \in \mathcal{RH}_\infty$, let H_p be defined as in (5-4) and define $U_p = M_pY_p + N_pX_p$, then the following conditions are equivalent. For all $p \in \mathcal{P}$

(i) $H_p \in \mathcal{RH}_\infty$.
(ii) $U_p^{-1} \in \mathcal{RH}_\infty$.
(iii) $\det(U_p) \neq 0$, $\forall s \in \mathbb{C}^+ \cup \mathbb{C}^0 \cup \{\infty\}$.
(iv) There exists a multiplier $\{\alpha, \alpha^{-1}\} \in \mathcal{RH}_\infty$ such that

$$\Re\{\det(U_p)\alpha\} > 0, \ \forall \omega \in \mathbb{R} \cup \{\infty\}. \tag{5-6}$$

Proof. Conditions (i), (ii) and (iii) are equivalent under the condition that $\{N_p, M_p, X_p, Y_p\} \in \mathcal{RH}_\infty$, the proof follows along the lines of [48, Theorem 5.1.6 and its proof]. The proof for the equivalence between (ii) and (iv) is given as follows.
($\Rightarrow$) Assume that $\{\alpha, \alpha^{-1}\} \in \mathcal{RH}_\infty$ and that $\Re\{\det(U_p)\alpha\} > 0, \ \forall \omega \in \mathbb{R} \cup \{\infty\}$. This implies that $\det(U_p)\alpha$ is strictly passive and thus it does not have RHP poles or zeros [49]. Moreover, $\det(U_p(\infty))\,\alpha(\infty) > 0$ implies that U_p and α are bi-proper. Since $\{\alpha, \alpha^{-1}\} \in \mathcal{RH}_\infty$, U_p must have no RHP zeros as it would require $\alpha \notin \mathcal{RH}_\infty$ to cancel it under passivity of $\det(U_p)\alpha$. Because U_p does not have RHP zeros and it is bi-proper, it follows that $U_p^{-1} \in \mathcal{RH}_\infty$. Therefore, (iv) implies (ii), and consequently (i) and (iii).
($\Leftarrow$) Assume that $\{U_p^{-1}, U_p\} \in \mathcal{RH}_\infty$ as $\{N_p, M_p, X_p, Y_p\} \in \mathcal{RH}_\infty$, then there exists some bi-proper $\{\alpha, \alpha^{-1}\} \in \mathcal{RH}_\infty$ such that $\Re\{\det(U_p)\alpha\} > 0, \ \forall \omega \in \mathbb{R} \cup \{\infty\}$. This is because U_p and α are bi-proper and have no RHP zeros (due to $\{U_p^{-1}, \alpha^{-1}\} \in \mathcal{RH}_\infty$) imply that $\det(U_p)\alpha$ is strictly passive for some α. Hence, (ii) implies (iv). □

We can derive from Theorem 5.1 that if U_p^{-1} exists and is stable, then H_p is internally stable. Therefore internal stability is achieved if condition (iv) is satisfied. One should note that, because coprime factorizations for G_p and K_p are used, it does not matter whether G_p is stable or not, the theorem applies either way.

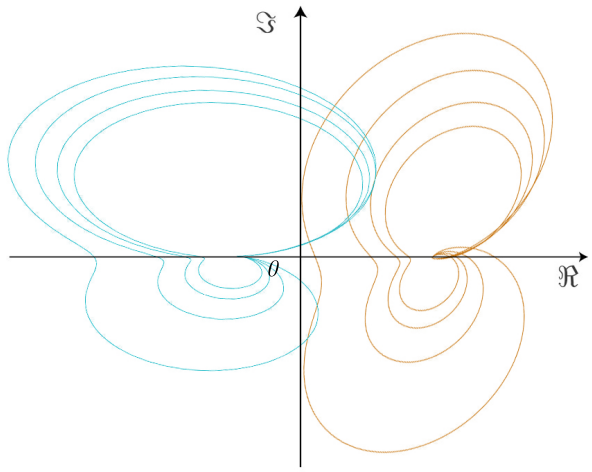

Figure 5-2: This sketch gives an interpretation of Theorems 5.1 and 5.2. The Nyquist curves of $\det(U_p)$ (blue) and $\det(U_p)\alpha$ (orange) are plotted for each operating point. Due to the multiplier α the blue curve moves to the RHP, which results in the orange curve.

Performance

Consider the interconnection in Figure 2-7 and let $\hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}$ be a virtual uncertainty, with $\mathbf{B}\hat{\mathbf{\Delta}}$ as defined in (2-29). The results of the Main Loop Theorem (Theorem 2.8) are used to formulate the following theorem.

Theorem 5.2. Consider H_p as defined in (5-4) and let $\tilde{H}_p = W_z H_p W_w$, with $\{W_z, W_w\} \in \mathcal{RH}_\infty$. Then, both the internal stability requirement $H_p \in \mathcal{RH}_\infty$ and the performance specification in (5-5) are satisfied if and only if there exists a multiplier $\alpha \in \mathcal{RH}_\infty$ such that

$$\Re\{\det(I - \gamma^{-1}\tilde{H}_p\hat{\Delta})\alpha\} \geq 0,\ \forall\omega \in \mathbb{R} \cup \{\infty\},\ \forall p \in \mathcal{P},\ \forall\hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}. \tag{5-7}$$

Proof. By applying Theorem 2.8 to (5-5) we get

$$\|\tilde{H}_p\|_\infty \leq \gamma \iff \det(I - \gamma^{-1}\tilde{H}_p\hat{\Delta}) \neq 0,\ \forall\omega \in \mathbb{R} \cup \{\infty\},\ \forall p \in \mathcal{P},\ \forall\hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}.$$

Consequently, we can apply Theorem 5.1 to derive the condition in (5-7). □

The interpretation of Theorems 5.1 and 5.2 is shown graphically in Figure 5-2. The existence of a multiplier α implies that the Nyquist curve of $\det(U_p)\alpha$ lies in the RHP, i.e. it is strictly passive. This also ensures that the Nyquist curve of $\det(U_p)$ does not go through or encircles the origin. From the Nyquist theorem it follows that U_p^{-1} is stable and therefore H_p is stable. Performance is guaranteed if the disc with centre $c_p = \det(U_p)$ and radius $r_p = \det(I - \gamma^{-1}\tilde{H}_p\hat{\Delta})$ does not include the origin for all frequencies and every scheduling variable $p \in \mathcal{P}$.

5-2-3 Fixed-structure Controller Synthesis

Local stability and performance specifications can be guaranteed through Theorem 5.2. The result is a set of controllers that satisfy the conditions (C.I) and (C.II). The controller parametrization is discussed next, after which the controller synthesis step follows.

Controller Parametrization

The controllers $K_p = X_p Y_p^{-1}$ are parametrized similar to (3-13). Any type of fixed-structure controller (or combination thereof) from Section 2-4 can be parametrized as such by separating the structure of the controller $\zeta(s)$ from the parameters ρ_x and ρ_y:

$$\begin{aligned} X_p\left(s, \rho_x(p)\right) &= \rho_x^T(p)\zeta(s) \\ Y_p\left(s, \rho_y(p)\right) &= \rho_y^T(p)\zeta(s). \end{aligned} \tag{5-8}$$

It should be noted that in the parametrization the control parameters ρ_x and ρ_y are dependent on the scheduling variable p. Consider, for example, a PID controller with affine dependence, as given in (4-17). It can be parametrized as follows:

$$\begin{aligned} \rho_x^T(p) &= \begin{bmatrix} (k_{\mathrm{p}_0} + k_{\mathrm{p}_1} p)T_{\mathrm{f}} + k_{\mathrm{d}_0} + k_{\mathrm{d}_1} p & (k_{\mathrm{i}_0} + k_{\mathrm{i}_1} p)T_{\mathrm{f}} + k_{\mathrm{p}_0} + k_{\mathrm{p}_1} p & k_{\mathrm{i}_0} + k_{\mathrm{i}_1} p \end{bmatrix} \\ \rho_y^T &= \begin{bmatrix} T_{\mathrm{f}} & I & 0 \end{bmatrix} \\ \zeta(s) &= \begin{bmatrix} s^2 & s & I \end{bmatrix}^T (s+\xi)^{-2}, \end{aligned} \tag{5-9}$$

where T_{f} is a time constant for the derivative action and $\xi > 0$. Of course, other parametrizations are possible and the dependencies of the controller on p can also vary, see Section 4-1-3. Besides fixed-structure controllers, it is also possible to design for fixed-order controllers by parametrizing the controller as follows:

$$\begin{aligned} X_p(s, \rho_x(p)) &= \rho_x(p)^T \phi(s, \xi) \\ Y_p(s, \rho_y(p)) &= \rho_y(p)^T \phi(s, \xi). \end{aligned} \tag{5-10}$$

Any stable orthogonal basis function (OBF) $\phi(s, \xi)$ can be used, such as pulse basis (discrete-time only), Katz basis, Laguerre basis and generalized OBFs. Take for example, the Laguerre basis function, it is given by:

$$\begin{aligned} \phi_0(s) &= 1, \\ \phi_n(s, \xi) &= \frac{\sqrt{2\xi}(s-\xi)^{n-1}}{(s+\xi)^n}, \end{aligned} \tag{5-11}$$

Controller synthesis

The optimization problem can be formulated using Theorem 5.2:

$$\begin{aligned} \min_{\gamma, \rho_x, \rho_y} \quad & \gamma \\ \text{s.t. } & \Re\{\det(I - \gamma^{-1}\tilde{H}_p(j\omega)\hat{\Delta})(j\omega)\alpha\} \geq 0, \\ & \forall \omega \in \mathbb{R} \cup \{\infty\},\ \forall p \in \mathcal{P},\ \forall \hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}. \end{aligned} \tag{5-12}$$

This optimization problem (5-12) is non-convex, because fixed-structure controller synthesis does not correspond to a convex optimization problem. The constraint in (5-12) has a practical issue that has to be overcome: it is not possible to check the constraint for an infinite set of frequencies. In practice, every fFRF is defined on a specific finite grid Ω_p with n_ω frequency points for a finite number of operating points dependent on the scheduling variable $p \in \mathcal{P}$. The grid Ω_p has to be defined properly, because we approximate the infinite set of frequencies by a finite set where the constraint in (5-12) is checked.

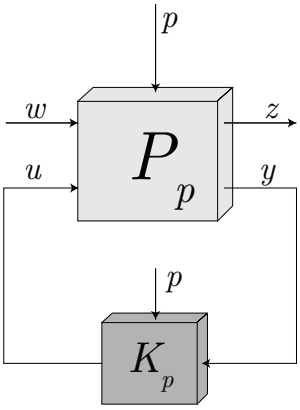

Figure 5-3: Generalized plant.

Realizations of $\hat{\Delta}$ Furthermore, similar to the performance condition in Theorem 3.3, one needs to check every possible realization (infinite amount) of $\hat{\Delta}$. These virtual perturbations can be realized in the same way as described in Section 3-3 by taking a number of random realizations on the unit circle which are in $\mathcal{RH}_\infty$. However, during the development of this method we learned that it does not work properly that way. Taking random realizations on the unit circle could result in $\|\tilde{H}_p\|_\infty > \gamma$, this is in contradiction with the condition for nominal performance (5-5). We propose a new method to find realizations of $\hat{\Delta}$ such that it is always $\|\tilde{H}_p\|_\infty < \gamma$.

The realizations of $\hat{\Delta}$ are realized through the Singular Value Decomposition (SVD). We take the SVD of the frequency response of $\tilde{H}_p(j\omega)$. At any fixed frequency ω the matrix of $\tilde{H}_p(j\omega)$ is decomposed into its SVD:

$$\tilde{H}_p = U\Sigma V^*, \tag{5-13}$$

where Σ consists of the non-negative singular values σ_i on its diagonal in a descending order. U and V are unitary, i.e. $U^*U = UU^* = I$, they are called the left and right singular vectors, respectively. The realizations of $\hat{\Delta}$ are denoted as $\bar{\Delta}$ and per frequency point a $\bar{\Delta}$ is computed using U and V:

$$\bar{\Delta}(j\omega) = \bar{v}(j\omega)\bar{u}(j\omega)^*, \tag{5-14}$$

where $\bar{v}$ and $\bar{u}$ are the first columns of V and U. The latter are linked to the maximum singular value of $\tilde{H}_p(j\omega)$ at that given frequency. Therefore, we are guaranteed that we construct the worst case perturbation, instead of a random perturbation. This way, the optimized variable γ is always equal to or slightly larger than the $\mathcal{H}_\infty$ norm. Furthermore, $\bar{\Delta} \leq 1$ because V and U are unitary. The optimization problem can then be solved using a non-convex solver, such as PSO (see Section 2-5).

5-3 LPV Feedback Autotuner

5-3-1 Problem Formulation

Consider the generalized plant in Figure 5-3, where P_p is partitioned as

$$\begin{bmatrix} z \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} P_{p_{zw}}(s) & P_{p_{zu}}(s) \\ P_{p_{yw}}(s) & P_{p_{yu}}(s) \end{bmatrix}}_{P_p(s)} \begin{bmatrix} w \\ u \end{bmatrix}. \tag{5-15}$$

It is assumed that the number of open RHP poles (P_{ol}), and integrators (n_i) of the system are known. The loop transfer is denoted as

$$L_p(s) = P_{p_{yu}}(s)K_p(s). \tag{5-16}$$

It should be noted that G_p is inside the plant P_p, hence we could reformulate $L_p(s) = G_p(s)K_p(s)$, but to keep generality of the problem, the notation in (5-16) is used. The closed-loop transfer function matrix is given by

$$H_p(s) = P_p(s) \star K_p(s). \tag{5-17}$$

H_p is internally stable if it satisfies Theorem 2.4 and the performance objectives are met if the $\mathcal{H}_\infty$ norm of the weighted closed-loop system is minimized. The control design problem is to synthesise an LPV controller K_p using fFRFs such that the following conditions are satisfied:

(c.i) the closed-loop system H_p in (5-17) is locally internally stable for all scheduling variables $p \in \mathcal{P}$;

(c.ii) the weighted closed-loop system $\tilde{H}_p = W_z H_p W_w$ achieves local nominal performance:

$$\|\tilde{H}_p\|_\infty \leq \gamma, \ \forall p \in \mathcal{P}, \tag{5-18}$$

where $\gamma > 0$.

5-3-2 Stability and Performance Analysis

Stability

If $L_p(s)$ satisfies the Generalized Nyquist Theorem for all $p \in \mathcal{P}$ then $H_p(s)$ is locally internally stable. Therefore, stability of the MIMO system can be evaluated in the complex plane by determining if $\det(I + L_p(s))$ does not go through the origin and the net encirclements around the origin are zero for every operating point. A non-parametric stability test in the frequency domain includes the following steps:

a) For every operating point construct a Nyquist plot with the given FRF.
b) Count the number of encirclements around the origin and determine the direction of each encirclement.

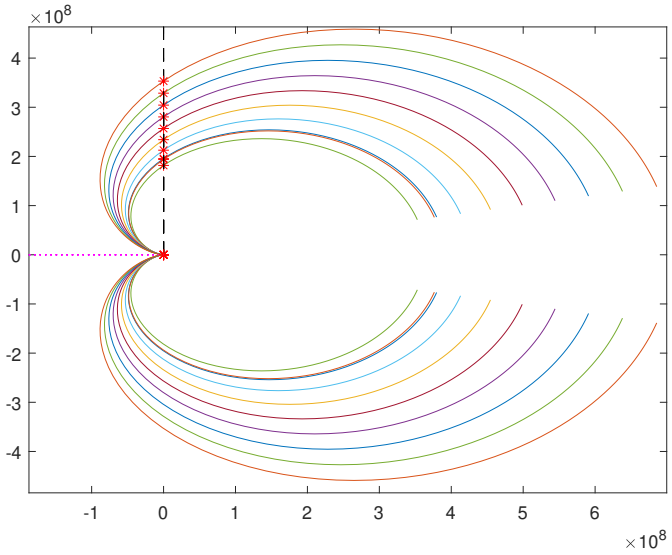

Figure 5-4: Nyquist curves of a MIMO system at 10 different operating points. Every Nyquist curve has distinct intersections with the vertical and horizontal line, these intersections are used to determine the sign and number of encirclements around the origin.

The Nyquist curve for positive frequencies is obtained by computing $\det(I + L_p(j\omega))$. Let $\mathcal{N}_p(j\omega)$ be the complete Nyquist curve that contains both the positive and negative frequencies. $\mathcal{N}_p(j\omega)$ is constructed by combining the positive Nyquist curve with its mirrored (around the Real axis) version. It is important to note that the direction of the Nyquist curve changes for negative frequencies, i.e. if the Nyquist curve goes to the origin for positive frequencies, then for negative frequencies it moves away from the origin and vice versa. This is crucial when determining the direction of the encirclements.

Stability test Consider Theorem 2.4. The results of the theorem can be used to formulate a MIMO stability test for an LPV system.

Corollary 5.1. Let $\mathcal{N}_p(j\omega)$ be the Nyquist curve of $\det(I + L_p(j\omega))$ for all $p \in \mathcal{P}$. The number of encirclements of $\mathcal{N}_p(j\omega)$ around the origin is denoted as N, where clockwise (CW) encirclements are positive and counter-clockwise (CCW) encirclements are negative. P_{ol} is the number of open RHP poles and $Z = N + P_{ol}$ is the number of closed-loop RHP poles. Then, H_p, as defined in (5-17), is locally internally stable for all $p \in \mathcal{P}$ if and only if

1. $\mathcal{N}_p(j\omega)$ does not go through the origin;
2. $Z = 0$.

The encirclements of $\mathcal{N}_p$ around the origin can be counted for every $p \in \mathcal{P}$ individually, this is shown graphically in Figure 5-4. The same principles are used as in Section 3-4, but the difference is that in this case we determine the encirclements around the origin for an MIMO LPV system. The stability test is given in Algorithm 3. Let V be the vertical line going up from $(0, 0j)$ and H_l be the horizontal line starting in $(0, 0j)$ going to the left towards negative infinity. The intersections of the Nyquist curve of $\mathcal{N}_p(j\omega)$ with V are denoted as $(0, y_0)$ and the intersection of $\mathcal{N}_p(j\omega)$ with H_l as $(x_0, 0)$. The index at which frequency the intersection with V occurs is denoted as i_V. The direction and number of the encirclements

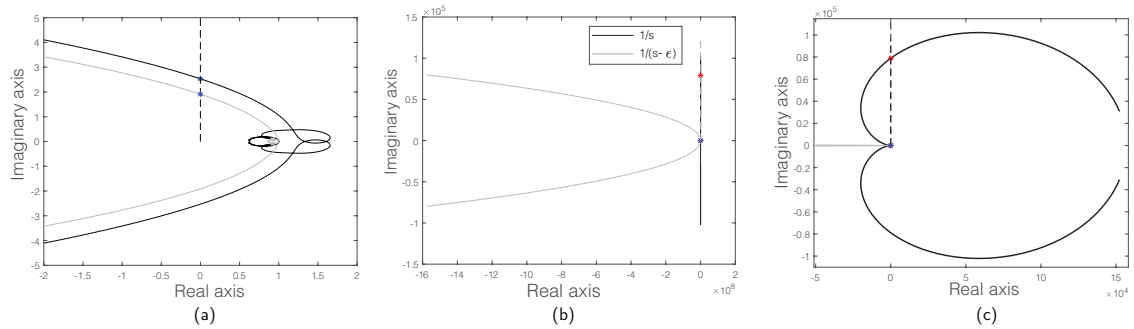

Figure 5-5: In all three Nyquist plots the same Nyquist curve of a system with integrators is plotted at different zoom levels, the light grey curve has a shifted integrator. In (a) we can see the Nyquist curves around the origin. The system with the pure integrator has one intersection with the vertical dashed line in $(0, 2j)$ and the system with the shifted integrator has an intersection at $(0, 3j)$. In (b) the Nyquist curve of the system with an integrator goes towards $-\infty$, thus there is no other intersection with the dashed vertical line, besides the one we see in (a). The number of intersection for the system with a pure integrator is therefore just 1 at $(0, 2j)$. However, by zooming in on the real axis in (b) we can see that in (c) the system with a shifted integrator shows that there is another intersection at $(0, 0.8j \cdot 10^5)$. Hence, by using a shifted integrator we can find all encirclements by counting the intersection with the vertical line, while that is not possible with a pure integrator.

at each intersection is denoted as N_{dir} and the number of encirclements due to integrators is denoted as N_{int}.

The way N_{dir} are determined is similar to the method in Algorithm 1. The changes become apparent when it comes to the determining N_{int}. The most notable difference is that integrators cause CCW encirclements, as opposed to the CW encirclements in the SISO case. This has to do with the fact that the contour of the Nyquist curve for a MIMO system is different from a SISO system, see Section 2-2-3 for the reasoning. Furthermore, during the development of the MIMO stability test it was found (heuristically) that there is a distinction between one or more than one integrators.

If the system (or controller) has integrators, the Nyquist curve starts in $+\infty$ and goes to $-\infty$ or the other way around, as shown in Figure 5-5. It can be determined how many encirclements integrators cause. Let n_i be the number of integrators of $L(j\omega)$. If $n_i \leq 1$, it accounts for no encirclement or 1 CCW encirclement, this depends on where the Nyquist curve starts. If the Nyquist curve starts in $-\infty$ and goes to $(1, 0j)$ it causes a CCW encirclement due to the integrator. Otherwise the integrator does not cause an encirclement, due to its influence the Nyquist curve starts in $+\infty$ and goes to $(1, 0j)$ without encircling the origin. If $n_i > 1$, the integrators cause $\lfloor \frac{n_i}{2} \rfloor$ CCW encirclements. If the system has got no integrators and during controller design pure integrators are avoided, it is not necessary to determine encirclements caused by integrators. In Figure 5-5 the difference is shown between a pure integrator and an integrator that is slightly shifted to the RHP. Near the origin the curves are similar (Figure 5-5a), but the system with the pure integrator seems to never encircle the origin because it goes to $-\infty$ (Figure 5-5b), while the other Nyquist plot encircles the origin origin twice, both CW and CCW (Figure 5-5c). Counting encirclements is therefore easier if there are no integrators or if they can be avoided.

Algorithm 3: The stability test for MIMO LPV systems as given in Corollary 5.1.

`function` stability_test $(\mathcal{N}_p(j\omega), P_{ol}, n_i)$ **Input** : $\mathcal{N}_p(j\omega)$: Nyquist curve of $\det(I - L_p(j\omega))$
P_{ol}: number of open-loop RHP poles
n_i: number of integrators
Output: Number of closed-loop RHP poles Z_m for every scheduling variable p_m
for m = *number of scheduling variables* p_m **do**
  if $(x_0, y_0) = (0, 0j)$ `//` (x_0, y_0) `is an intersection with the vertical line` **then**
    $\mathcal{N}_p(j\omega)$ goes through $(0, 0j)$
    stability cannot be determined
    return
  end
  if x_0 *or* y_0 *is empty* **then**
    No intersection with V and H_l, $N = 0$ `//` V `is the vertical line,` H_l `is the horizontal line and` N `is the number of intersections/encirclements`
  else
    Let i_V be the index of the frequency at which the intersection with V occurs.
  end
  if i_V *is empty* **then**
    No intersection with V, $N = 0$
  else
    for k = *number of intersections* `// count encirclements due to intersections with V` **do**
      if $\mathcal{N}_p(j\omega_{i_V+1}) - \mathcal{N}_p(j\omega_{i_V-1}) > 0$ **then**
        $N_{dir}(k) = +1$
      else
        $N_{dir}(k) = -1$
      end
    end
  end
  if $n_i = 1$ `// count encirclements due to pure integrators at` $1/s$ **then**
    if $\mathcal{N}_p(j\omega)$ *starts in LHP* **then**
      $N_{int} = -1$
    else
      $N_{int} = 0$
    end
  else if $n_i > 1$ **then**
    $N_{int} = -\lfloor \frac{n_i}{2} \rfloor$
  else
    $N_{dir} = 0$
  end
  `// algorithm continues on next page`
end

for m = *number of scheduling variables* p_m `// continuation of previous for-loop` **do**
  $N = N_{dir}(1) + ... + N_{dir}(k) + N_{int}$
  $Z = N + P_{ol}$
  if $Z = 0$ **then**
    Closed-loop system is locally stable for scheduling variable p_m
  else
    Closed-loop system is locally unstable for scheduling variable p_m
  end
end

Performance

The goal is to achieve nominal performance. The performance specifications have to be formulated in the frequency-domain by the engineer. Because the generalized plant form is used, it is possible to design for many general interconnections, such as the 4-block. In this way there is some form of freedom to shape the performance of the closed-loop system. The specifications are met by designing proper weighting filters, see Section 2-3-3 for the design procedure. Weighting filters are designed to ensure that for every scheduling variable $p \in \mathcal{P}$ the sensitivities satisfy the desired performance shape. The condition for nominal performance is given in (5-18).

5-3-3 Fixed-structure controller Synthesis

Condition (c.i) is met through the stability test as given in Algorithm 3. Condition (c.ii) is satisfied by optimizing the $\mathcal{H}_\infty$ norm of the weighted closed-loop transfer function matrix for every scheduling variable $p \in \mathcal{P}$. The $\mathcal{H}_\infty$ norm of (5-18) is evaluated for every stabilizing set of parameters for K_p to determine if $\gamma \leq 1$.

Controller Parametrization Any type of fixed-structure controller, such as the ones described in Section 2-4, can be parametrized as an LPV controller K_p. Let ρ be a control parameter and $\zeta(s)$ the controller structure. The controller is given by

$$K_p(s, \rho, p) = \sum_{i=1}^{m} \left(\rho_{i,0} + \sum_{j=1}^{n} \rho_{i,j} p^n \right) \zeta_i(s), \tag{5-19}$$

where m is the number of parameters and n determines the dependency on p. For $n = 1$ the dependency on p is affine and for $n \geq 2$ the controller is a polynomially dependent on p.

Controller Synthesis The optimization problem can be formulated as:

$$\begin{aligned} \min_{\rho} \quad & \gamma \\ \text{s.t.} \quad & \text{stability test in Algorithm 3} \\ & \|\tilde{H}_p(j\omega)\|_\infty \leq \gamma,\ \forall \omega \in \Omega_p,\ \forall p \in \mathcal{P}. \end{aligned} \tag{5-20}$$

Similar to (5-12), this is a non-convex optimization problem because a fixed-structure controller is synthesised in (5-20). In Section 5-1 it was mentioned that there is limited amount of frequency response data available in a real setting. This has its effects on how constraints can be evaluated in general for any data-driven approach, since the constraints are not actually checked for all frequencies but only for the available frequencies. For the LPV FBA approach this could mean that the Nyquist curve is incomplete due to a lack of frequency response data. It is important that the data contains enough information to plot (most of) the Nyquist curve. If the frequency range of the grid is too small or not dense enough, which depends on the system, stability cannot be assessed correctly. It is possible to verify if the frequency grid Ω_p is adequate by plotting the FRF in the complex plane to see how much of the Nyquist curve is plotted.

PSO Algorithm The optimization problem is solved using PSO, see Section 2-5. For each optimization problem the lower and upper bound for the parameter space are defined by the user or set to default, i.e. parameters which are known to provide feasible solutions. The optimization via PSO is a three-step procedure:

1. Propose parameters ρ_m for controller $K_p(j\omega, \rho)$;
2. Run Algorithm 3 to determine the local stability of the closed-loop system with $K_p(j\omega\rho)$;
3. Optimize parameters ρ_m of stabilizing controllers such that γ is minimized in (5-20) for local nominal performance.

The PSO algorithm for LPV FBA is shown in Algorithm 4.

Algorithm 4: LPV FBA PSO algorithm for MIMO systems.

for m = *number of scheduling variables* p_m **do**
  Initialize a population with size i of control parameters $\rho = [\rho_1 ... \rho_i]$ particles within a random or known range of control parameters.
  for *every set of controller parameters* ρ_k *at iteration* k **do**
    Stability test of Algorithm 3
    if ρ_k *renders the closed-loop system locally stable at iteration* k **then**
      Compute the performance specification in (5-18)
    else
      Flag the control parameters ρ_k at iteration k as infeasible
    end
  end
  for *all stabilzing control parameters* ρ_k *at iteration* k **do**
    Minimize γ in (5-20)
  end
end

5-4 Summary

Two novel control design methods have been presented in this section. Both methods aim to design a fixed-structure LPV controller using fFRFs. The first method, in Section 5-2, uses the results of the Main-Loop Theorem and coprime factorisation representation of the system

and controller to design an LPV controller that guarantees local stability and performance. The derived conditions apply specifically to the 4-block problem. FBA, on the other hand, is formulated in the general PK form, therefore it is possible to design a controller for many different interconnections. Local stability is guaranteed at each operating point through a stability test in the complex plane using the Generalized Nyquist theorem. In the next chapter these two methods are applied to synthesise LPV controllers for two nonlinear systems.

Chapter 6

Simulations

In this chapter LPV controllers are designed for a two mass-spring-damper (MSD) system and a model of a helicopter system from the literature [50, 51] using the methods as described in Chapter 5. For easier reference, the fixed-structure controller with $\mathcal{H}_\infty$ performance is abbreviated to FCH (Fixed-structure Controller H_∞). The following steps are taken to design a controller. First, FRFs of the systems are obtained, after which the design objectives are presented for a 4-block structure. A desired controller structure is chosen and finally, the controller parameters with respect to the maximum singular values of the systems are optimized. The objective of this chapter is to provide an extensive comparison between the two LPV controllers. The LPV controllers are also compared to their LTI counterparts. Based on these comparisons an overview of the advantages and disadvantages of the methods is given in the final section of this chapter.

6-1 Mass-spring-damper system model

The free body diagram of the MSD system is shown in Figure 6-1. It consists of two masses m_1 and m_2, two dampers c_1 and c_2, and two nonlinear Duffing springs k_{D_1} and k_{D_2}. The

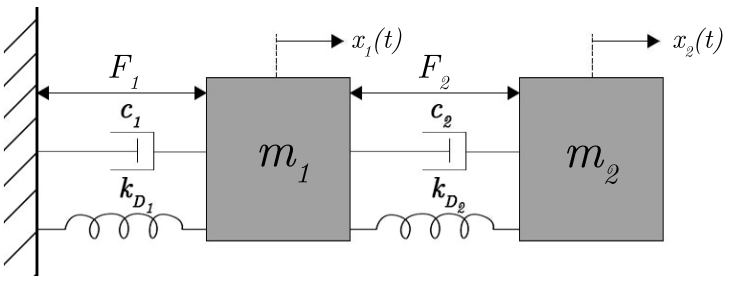

Figure 6-1: Free body diagram of a two mass-spring-damper system.

Parameter	*Symbol*	*Value*	*Unit*
Mass 1	m_1	0.30	kg
Duffing spring 1: linear stiffness	k_1	160	N/m
Duffing spring 1: nonlinear stiffness	κ_1	210	N/m
Damper constant 1	c_1	1.71	Ns/m
Mass 2	m_2	1.6	kg
Duffing Spring: linear stiffness	k_2	150	N/m
Duffing Spring: nonlinear stiffness	κ	410	N/m
Damper constant 2	c_2	0.28	Ns/m

Table 6-1: Parameters of the two mass-spring-damper system.

continuous time dynamics of the MSD is given by:

$$\begin{aligned} m_1\ddot{x}_1(t) &= F_1(t) - c_1\dot{x}_1(t) - k_1x_1(t) - \kappa_1x_1^3(t) - c_2(\dot{x}_1(t) - \dot{x}_2(t)) - k_2(x_1(t) - x_2(t)) \\ &\qquad - \kappa_2(x_1^3(t) - x_2^3(t)) - F_2(t) \quad (6\text{-}1) \\ m_2\ddot{x}_2(t) &= F_2(t) - c_2(\dot{x}_2(t) - \dot{x}_1(t)) - k_2(x_2(t) - x_1(t)) - \kappa(x_2^3(t) - x_1^3(t)), \end{aligned}$$

where $x_1(t)$ and $x_2(t)$ are the outputs describing the displacements of masses m_1 and m_2, respectively. The linear stiffness of the Duffing springs is given by k_1 and k_2, whereas the nonlinear stiffness is denoted by κ_1 and κ_2. The operating range of the Duffing springs is limited to $\{x_1(t), x_2(t)\} \in [-\sqrt{2}, \sqrt{2}]$. The inputs are denoted by $F_1(t)$ and $F_2(t)$, these are forces acting on each respective mass. The nonlinear model in (6-1) can be written as an LPV system by introducing the following two scheduling variables $p_1(t) = \mu_1(x(t)) = x_1^2(t)$ and $p_2(t) = \mu_2(x(t)) = x_1^2(t) + x_1(t)x_2(t) + x_2^2(t)$, the LPV representation is given as

$$\begin{aligned} m_1\ddot{x}_1(t) &= F_1(t) - c_1\dot{x}_1(t) - k_1x_1(t) - \kappa_1p_1(t)x_1(t) - c_2(\dot{x}_1(t) - \dot{x}_2(t)) \\ &\qquad - k_2(x_1(t) - x_2(t)) - \kappa_2p_2(t)(x_1(t) - x_2(t)) - F_2(t) \quad (6\text{-}2) \\ m_2\ddot{x}_2(t) &= F_2(t) - c_2(\dot{x}_2(t) - \dot{x}_1(t)) - k_2(x_2(t) - x_1(t)) - \kappa p(t)(x_1(t) - x_2(t)), \end{aligned}$$

with $p_1(t) \in [0,\, 2]$ and $p_2(t) \in [0,\, 6]$. For constant p_1 and p_2, these differential equations are transformed to the Laplace domain and rewritten such that we get the following matrix:

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} m_1s^2 + c_1s + k_1 + \kappa_1p_1 & m_2s^2 \\ -(c_2s + k_2 + \kappa_2p_2) & m_2s^2 + c_2s + k_2 + \kappa_2p_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (6\text{-}3)$$

By inverting the matrix in (6-3) we find the 2×2 transfer function matrix $G_p(s)$. The scheduling variables p_1 and p_2 are treated as an uncertainty, such that the LPV system can be represented in the LFR form as in Section 4-1-4. This is used to construct a set of FRF data at $N_p = 9$ frozen operating points in a linearly spaced grid $\mathcal{P}$ and a frequency grid with $N_\omega = 500$ logarithmically spaced frequency points in the range $[10^{-1},\, 10^3]$. The FRFs of the system at the 9 operating points are shown in Figure 6-2. The parameters of the MSD system are given in Table 6-1.

6-1-1 Design Objectives

Consider the weighted 4-block interconnection which is also depicted in the generalized plant form in Figure 2-6. The controller design objectives are formulated in the generalized plant

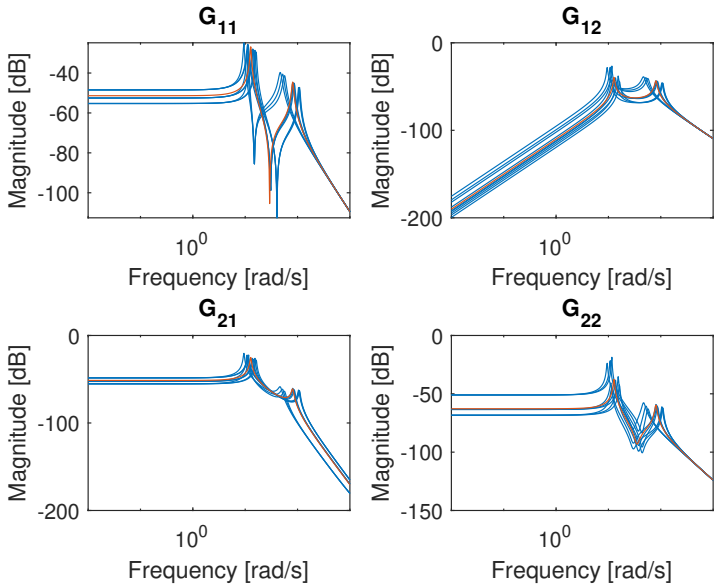

Figure 6-2: The FRFs of the MSD system at 9 different operating points in blue and the nominal system response in orange.

form. Weighting filters are used on the four sensitivities to meet the performance specifications.

Stability requirements Internal stability is a necessary requirement. The closed-loop system is internally stable if the 4-block test in Section 2-2-2 is satisfied. Both methods check if the condition is satisfied as follows:

FBA: for every frozen FRF the stability test in Corollary 5.1 is performed.

FCH: for every frozen FRF stability is guaranteed through Theorem 5.1.

Performance specifications Consider the mixed-sensitivity problem in Section 2-3-2. We want to design a controller that achieves reference tracking and satisfies the nominal performance requirement, see Definition 2.7. The sensitivity $S(j\omega)$ should be shaped such that at low frequencies $|S(j\omega)| \ll 1$ and at higher frequencies we require that the maximum sensitivity $\|S(j\omega)\|_\infty \leq 2$ for reference tracking. The complimentary sensitivity is shaped the other way around, we want high gain at low frequencies and low gain at high frequencies to attenuate noise. In terms of time-domain specifications, a necessary requirement is that the closed-loop system should be able to achieve a zero steady state error and follow the reference, where relatively fast tracking responses are preferred. The weighting filters have to be designed such that the aforementioned specifications can be met.

6-1-2 Weighting Filter Selection

The output weighting filters $W_z = \text{diag}(W_S, W_S, W_T, W_T)$ are used to shape the sensitivities and the input weighting filters W_w are used to shape the reference and disturbance signal.

Since there are no specifications on the reference and disturbance signal, we can simply choose the following input weighting filters $W_w = \text{diag}(I, I, I, I)$. The output weighting filters are designed while keeping the performance specifications in mind, this results in the following filters. The same weighting filter W_S is designed for both the sensitivity S and process sensitivity PS, since their shapes are not much different. The transfer function of weighting filter W_S is the same as in (2-38), with $\omega_B^* = 0.2$ rad/s, $\varepsilon_A = 10^{-3}$ and $M_s = 2$:

$$W_S = \frac{s/2 + 0.2}{s + 2 \cdot 10^{-4}}. \tag{6-4}$$

The weighting filter W_T is designed for both KS and T. The transfer function of W_T is similar to the one in (2-40), with $M_t = 2$, $\omega_{BT}^* = 155$ rad/s and $\varepsilon_A = 10^{-3}$:

$$W_T = \frac{10^3(s + 77.5)}{s + 55 \cdot 10^5}. \tag{6-5}$$

The weighting filters were tuned iteratively while keeping the waterbed effect in mind.

6-1-3 Controller Parametrization

The goal is to design a controller such that the $\mathcal{H}_\infty$ norm of the weighted closed-loop transfer function matrix is below 1. A PID controller with a low pass filter (LPF) is designed for the MSD system. A PI controller with a high integrator term increases the bandwidth and therefore tracking performance improves. A second order LPF ensures that signals with frequencies lower than the cut-off frequency ω_c can pass through the filter, while past the cut-off frequency there is a -40 dB per decade roll off to attenuate the signal. The controllers are parametrized differently for each method.
For FBA, the PID is parametrized as in (4-17) and the LPF as in (2-46), consequently the controller designed for FBA is given by

$$K_{FBA} = K_{PID} \cdot K_{LPF} \cdot \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix}, \tag{6-6}$$

where $k_1, ..., k_4$ are gains. The PID and LPF controller are given by:

$$K_{PID} = k_{\text{p}_0} + k_{\text{p}_1} p_1 + k_{\text{p}_2} p_2 + \frac{k_{\text{i}_0} + k_{\text{i}_1} p_1 + k_{\text{i}_2} p_2}{s} + \frac{(k_{\text{d}_0} + k_{\text{d}_1} p_1 + k_{\text{d}_2} p_2)s}{T_{\text{f}} s + 1}, \tag{6-7}$$

$$K_{LPF} = \frac{1}{(s/\omega_c)^2 + s/(Q\omega_c) + 1}. \tag{6-8}$$

For FCH, the controller $K_{FCH} = X_p Y_p^{-1}$ is given by

$$\begin{aligned} X_p(s) &= \rho_x^T \phi(s, a) \cdot \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix}, \\ Y_p(s) &= \rho_y^T \phi(s, a) \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \\ \phi &= \begin{bmatrix} s^2 & s & I \end{bmatrix}^T (s + a)^{-2}, \end{aligned} \tag{6-9}$$

where $X_p(s)$ is parametrized as in (5-9), where ρ_y is modified as a result of the LPF:

$$\begin{aligned}\rho_x(p_1,p_2) &= \begin{bmatrix}(k_{\mathrm{p}_0}+k_{\mathrm{p}_1}p_1+k_{\mathrm{p}_2}p_2)T_{\mathrm{f}}+k_{\mathrm{d}_0}+k_{\mathrm{d}_1}p_1+k_{\mathrm{d}_2}p_2\\(k_{\mathrm{i}_0}+k_{\mathrm{i}_1}p_1+k_{\mathrm{i}_2}p_2)T_{\mathrm{f}}+k_{\mathrm{p}_0}+k_{\mathrm{p}_1}p_1+k_{\mathrm{p}_2}p_2\\k_{\mathrm{i}_0}+k_{\mathrm{i}_1}p_1+k_{\mathrm{i}_2}p_2\end{bmatrix}\\ \rho_y^T &= \begin{bmatrix}T_{\mathrm{f}}/\omega_c^2 & 1/Q\omega_c & 1\end{bmatrix}\\ \zeta(s) &= \begin{bmatrix}s^2 & s & I\end{bmatrix}^T(s+\xi)^{-2},\end{aligned} \tag{6-10}$$

where ω_c is the cut-off frequency and Q is the damping coefficient. The multiplier $\alpha(s)$ is parametrized with Laguerre basis of order 6 as follows:

$$\alpha(s) = \begin{bmatrix}1 & \dfrac{\sqrt{2}(s-1)}{(s+)} & \ldots, & \dfrac{\sqrt{2}(s-1)^5}{(s+1)^6}\end{bmatrix}\begin{bmatrix}\rho_{\alpha_0}\\ \vdots \\ \rho_{\alpha_6}\end{bmatrix}, \tag{6-11}$$

where ρ_{α_i} are design parameters in the optimization problem.

6-1-4 Controller optimization

With the parametrizations of the controller ready, the control parameters can be optimized while γ is minimized for both methods. The PSO algorithm requires an initial population to set up the optimization, therefore random control parameters were fed to PSO to look for a stabilizing set of parameters, see also Algorithm 4. Once a set of stabilizing parameters was found, the optimization problems in (5-12) and (5-20) could be solved for FCH and FBA, respectively. The initial population size was 100.

6-1-5 Results

The results of the optimization problems are shown in Tables 6-2 and 6-3. We can directly see in the table that there is no gain for the derivative action, during the initialisation of the stabilising set of parameters it was determined that no derivative action was required. With regards to γ, it should be noted that γ indicates what the maximum $\mathcal{H}_\infty$ norm is of the closed-loop system at all nine operating points. For example, the $\mathcal{H}_\infty$ norm of the closed-loop system at the nine operating points for the LPV controller designed using FBA are $\begin{bmatrix}0.71 & 0.71 & 0.82 & 0.77 & 0.67 & 0.80 & 0.84 & 0.68 & 0.79\end{bmatrix}$, therefore, $\gamma = 0.84$ in this case. Furthermore, because a non-convex optimization problem is solved, it should be stressed that the results in Table 6-2 show local minima that guarantee local stability and performance specifications for every frozen FRF, i.e. the operating points of interest. For comparison the optimization problem has also been solved in case an LTI controller is designed. The LPV parameters are omitted in that case, the results for the LTI controllers are also shown in Table 6-2. What follows is a comparison of the LPV controller designs using FCH and FBA. Next, the LPV controllers are compared to their LTI counterparts.

	γ	k_{p_0}	k_{p_1}	k_{p_2}	k_{i_0}	k_{i_1}	k_{i_2}	ω_c	Q
LPV FCH	0.85	46.20	51.69	52.38	120.32	25.39	41.57	1.65	0.42
LPV FBA	0.84	104.01	30.29	86.41	52.86	49.89	40.36	1.12	0.41
LTI FCH	1.71	170.04	0	0	280.91	0	0	1.44	0.81
LTI FBA	1.53	107.39	0	0	366.39	0	0	1.92	0.25

Table 6-2: Parameters of the LPV and LTI PID controller and low pass filter designed with both FBA and FCH method. See Table 6-3 for the gains k_1, k_2, k_3, k_4.

	k_1	k_2	k_3	k_4
LPV FCH	0.76	0.45	-1.74	1.68
LPV FBA	1.20	0.64	-2.20	3.72
LTI FCH	1.17	0.47	-1.53	2.47
LTI FBA	4.02	1.15	-2.72	2.02

Table 6-3: Parameters the controllers (continued).

6-1-6 Comparison of LPV Controllers

The singular values plot of each of the 4-block sensitivities is shown in Figure 6-4, where the dashed black line is the weighting filter. The $\mathcal{H}_\infty$ norm of the weighted 4-block is less than one at each operating point for both controllers. In general the local $\mathcal{H}_\infty$ norm is slightly lower for FCH at the operating points and the FCH controlled system provides a better performance. Overall, the singular value plots are similar and most notably there are resonance peaks in the range $\omega \in [8, 18]$ rad/s. This leads to slightly oscillatory tracking response, see Figure 6-3. In Figure 6-3 the step response of the complimentary sensitivity function is shown, where in the left figure $F_1(t) = 1$ and $F_2(t) = 0$ and in the right figure $F_1(t) = 0$ and $F_2(t) = 1$. We can see that the FCH controlled system has a faster tracking response with a faster settling time and less overshoot for the position of mass 1. The maximum overshoot is 8% and the settling time of the slowest response is about 8 seconds, while FBA needs nearly 13 seconds to settle with an overshoot of 15%. For tracking of the position of mass 2, FBA seems to rise and settle faster than FCH, but again with a relatively higher overshoot of about 14%. The settling time of the slowest step response is around 10 seconds for FBA while FCH requires more than 16 seconds. These are of course local step responses at the different operating points, the nonlinear system has been simulated and the step response of the closed-loop system is shown in Figure 6-5. The most notable difference is the difference between the overshoot, where FCH has no overshoot for both positions x_1 and x_2, FBA has 14% and 25% overshoot, respectively. Furthermore, FCH also settles faster in nearly 4 seconds, while FBA needs more than 6 seconds to follow the reference signal. Both controllers have a relatively slow rise times and cannot follow the reference signal right from the start. The design requirements are met by using either one of the two controllers, however, the FCH controlled system provides better tracking performance.

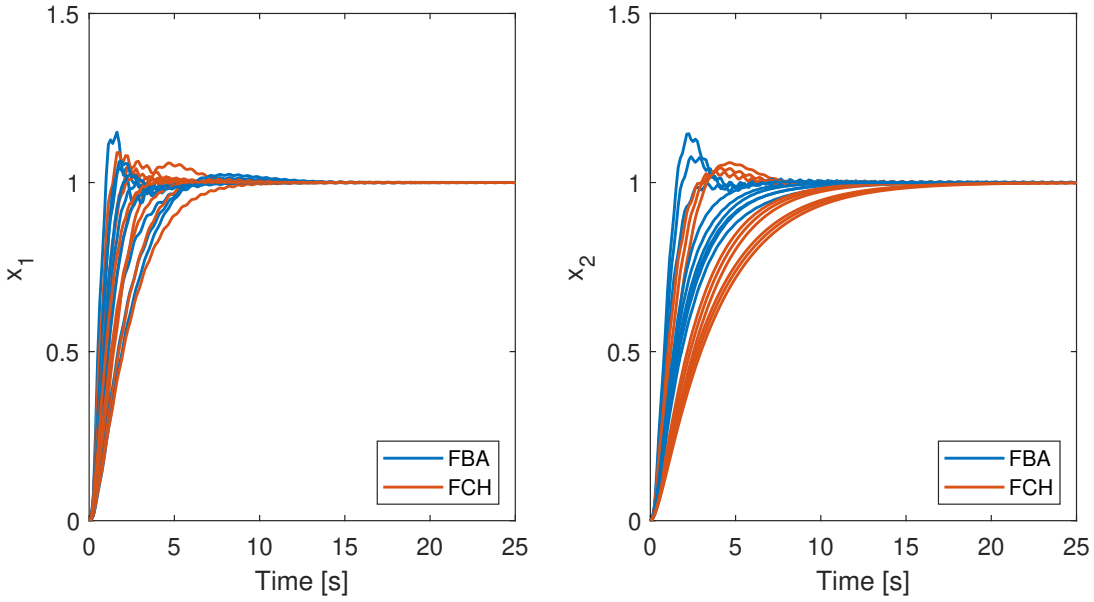

Figure 6-3: Tracking responses of the closed-loop systems (T) for FCH and FBA for nine operating points.

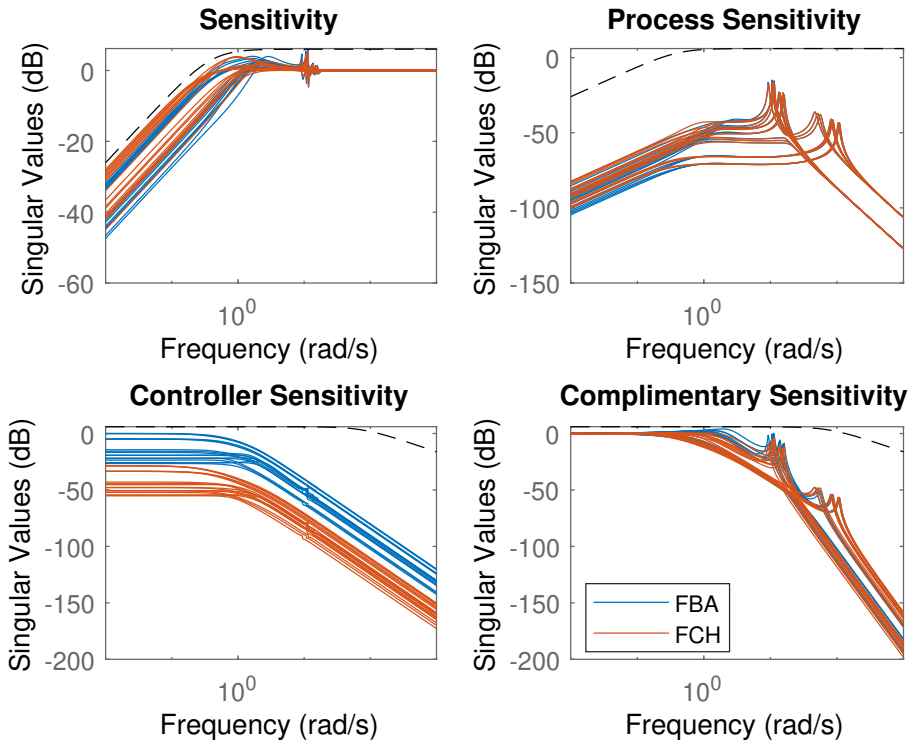

Figure 6-4: Singular values of the 4-block closed-loop transfer functions with LPV controllers for nine operating points and the corresponding weighting filters (black dashed line).

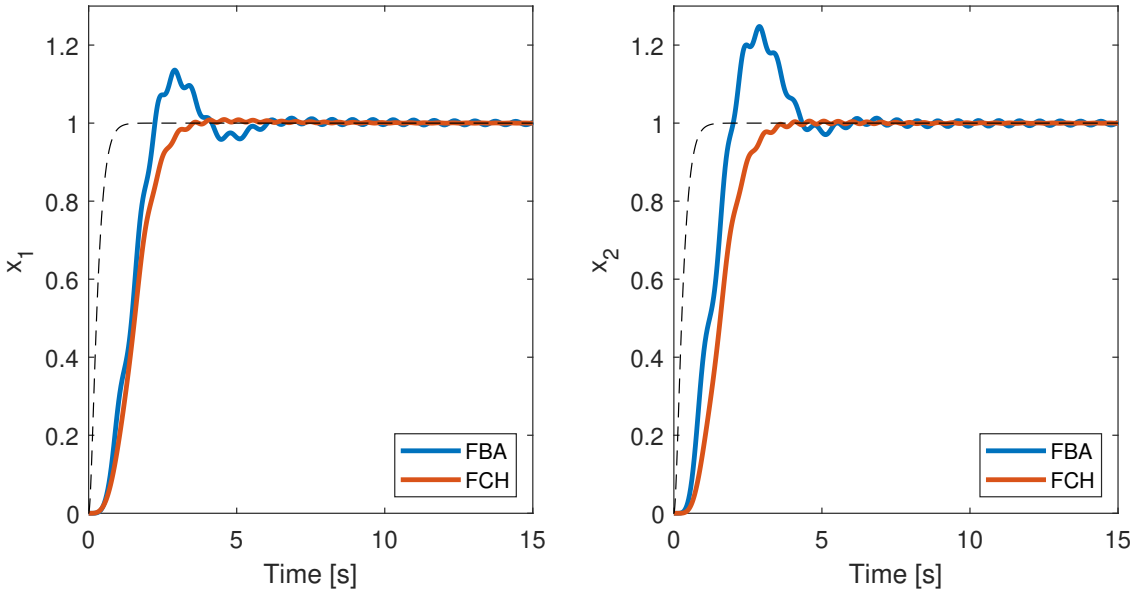

Figure 6-5: Simulation of the closed-loop nonlinear system with the designed LPV controllers. The reference signal is the dashed black line, it is a low pass filtered step.

6-1-7 Comparison of LPV and LTI Controllers

It must be emphasised that for the comparison the weighting filters have not been changed throughout the optimization problem, otherwise it would not be a fair comparison between the LPV and LTI controllers. The two LTI controllers show some similarities, therefore the differences between the LPV and LTI controllers are nearly the same for both methods. From Table 6-2 we can see notable differences in the parameters, for example, γ is below 1 with the LPV controllers, whereas the LTI controllers are above 1 and therefore do not meet the nominal performance requirement. This is also where the advantage of an LPV controller becomes apparent, because the LTI controllers apply the same proportional and integral gain to all the different operating points. Whereas the LPV controllers look for the gains such that the best performance is reached at each operating point. First, let us compare the LPV FCH and LTI FCH controllers. In Figure 6-6 we can see that the singular values of the closed-loop LTI controlled system do not meet the performance specifications, because the singular values of the sensitivity and complimentary sensitivity cross the upper bound of the weighting filters, implying that their $\mathcal{H}_\infty$ norm is higher than 2. In Figure 6-7 the tracking response of the complimentary sensitivity is shown. The overshoot of the LTI controller is higher for the positions of both masses and so is the settling time. In the simulation of the nonlinear model the difference between the LPV and LTI controller becomes much more clear, see Figure 6-8. The LTI controller has an oscillatory response and takes about 5 seconds longer to settle. The rise time of the LTI controller is higher, which is due to the relatively higher gain overall.
The same differences can be observed for the FBA control design method. In Figures 6-9 to 6-11 we can also see that the LTI FBA controller behaves slightly worse than the LTI FCH controller, as there is even more overshoot (20% in the nonlinear simulation) and longer settling times (nearly 12 seconds). The same differences discussed for the LPV FCH and LTI

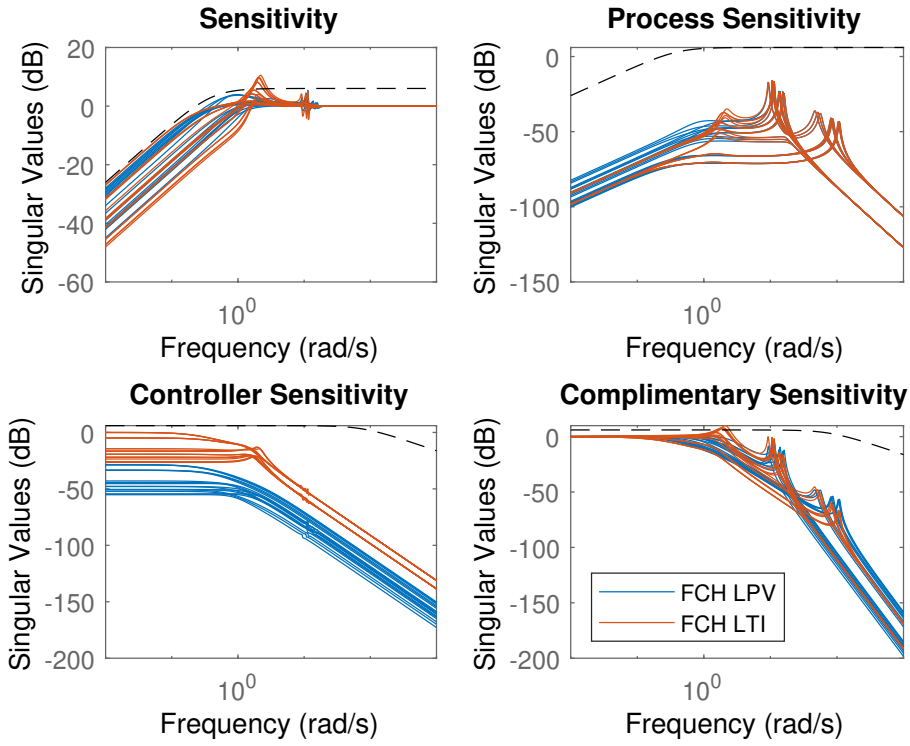

Figure 6-6: Singular values of the 4-block closed-loop transfer functions with an LPV FCH controller (blue lines) and LTI FCH controller (orange lines) for nine operating points and the corresponding weighting filters (black dashed line)

FCH controller apply to the FBA controllers.

6-1-8 Conclusions

By using either of the two LPV control design methods it is possible to design an internally stabilizing controller that meets $\mathcal{H}_\infty$ performance specifications. The characteristics of both methods can be found in Table 6-4. Both methods solve a non-convex optimization problem, however, it is possible to turn FCH into a convex optimization problem if a fixed-order controller is designed instead of a fixed-structure controller. The major difference between the methods is that FBA requires prior knowledge with respect to the open RHP poles and integrators of the plant to correctly asses the stability of the loop transfer in the complex plane. Furthermore, for FBA to work efficiently a stabilizing set of (known) control parameters are very much needed. Otherwise PSO needs a relatively long time to find a stabilizing set of parameters. During the design procedure the set of stabilizing parameters for FBA was acquired through simulations with the FCH controller. This is because the constraints for the FCH controller are more relaxed and easier to compute. Despite the mentioned disadvantages, it should be noted that with FBA it is possible to use the generalised plant form and therefore FBA has the great advantage that the plant and controller interconnection does not matter as much since many interconnections can be written in the PK form. Despite the fact that

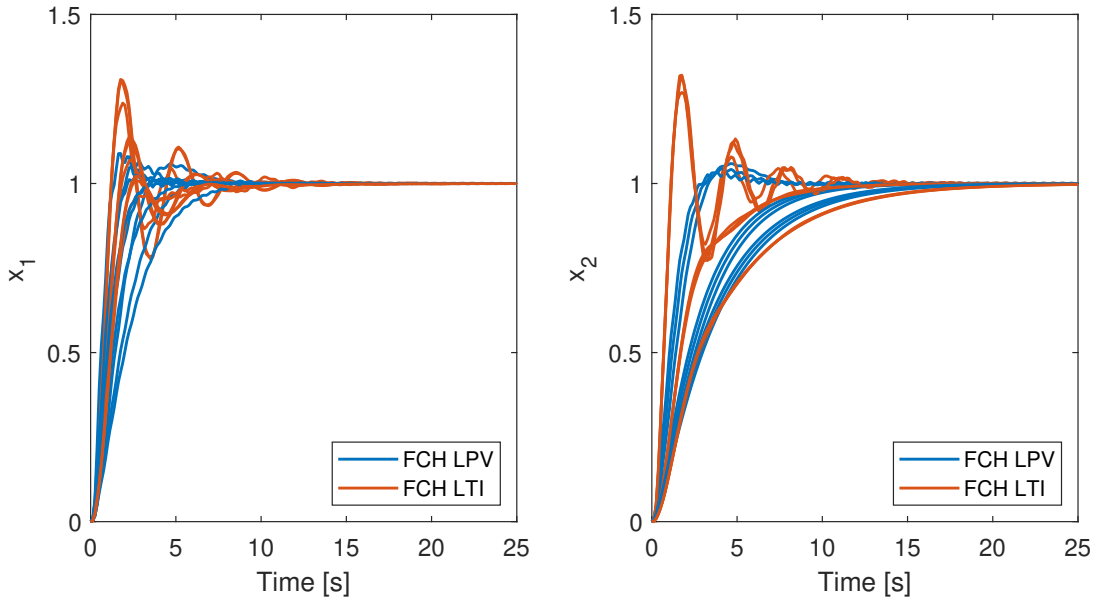

Figure 6-7: Tracking response of the closed-loop system (T) for FCH LPV and FCH LTI for nine operating points.

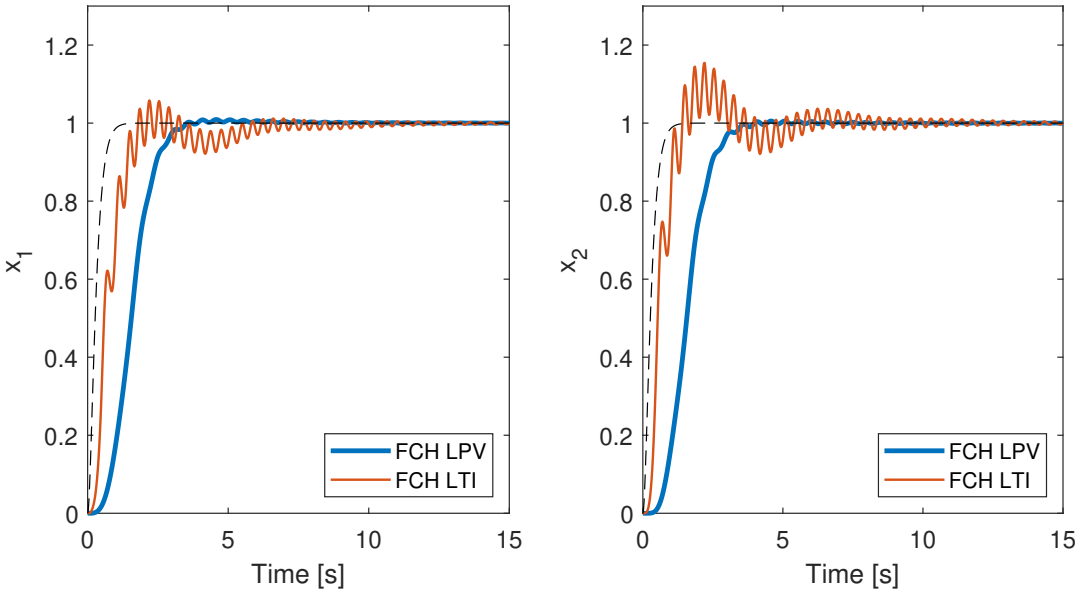

Figure 6-8: Simulation of the closed-loop nonlinear system with the designed LPV controller (blue line) and LTI controller (orange line). The reference signal is the dashed black line.

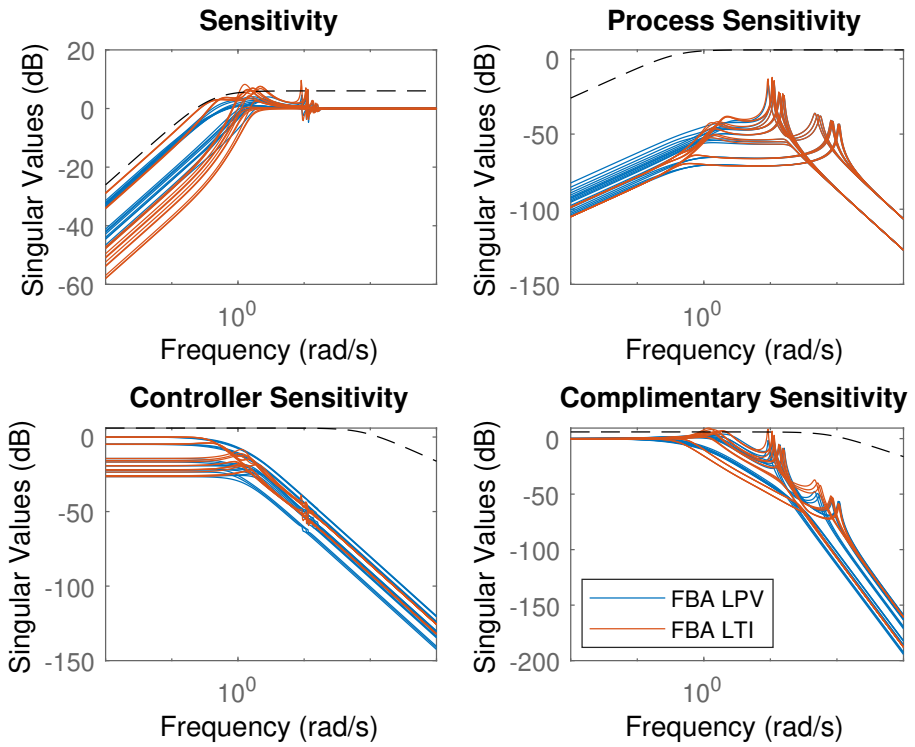

Figure 6-9: Singular values of the 4-block closed-loop transfer functions with an LPV controller (blue lines) and LTI controller (orange lines) for nine operating points and the corresponding weighting filters (black dashed line).

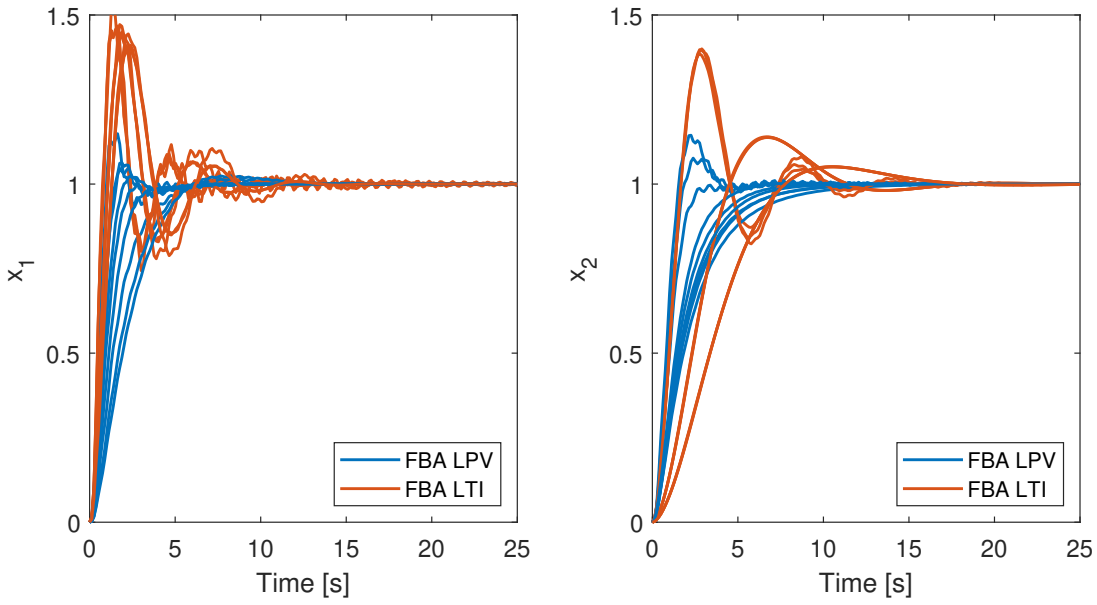

Figure 6-10: Tracking response of the closed loop system (T) for FBA LPV and FBA LTI for nine operating points.

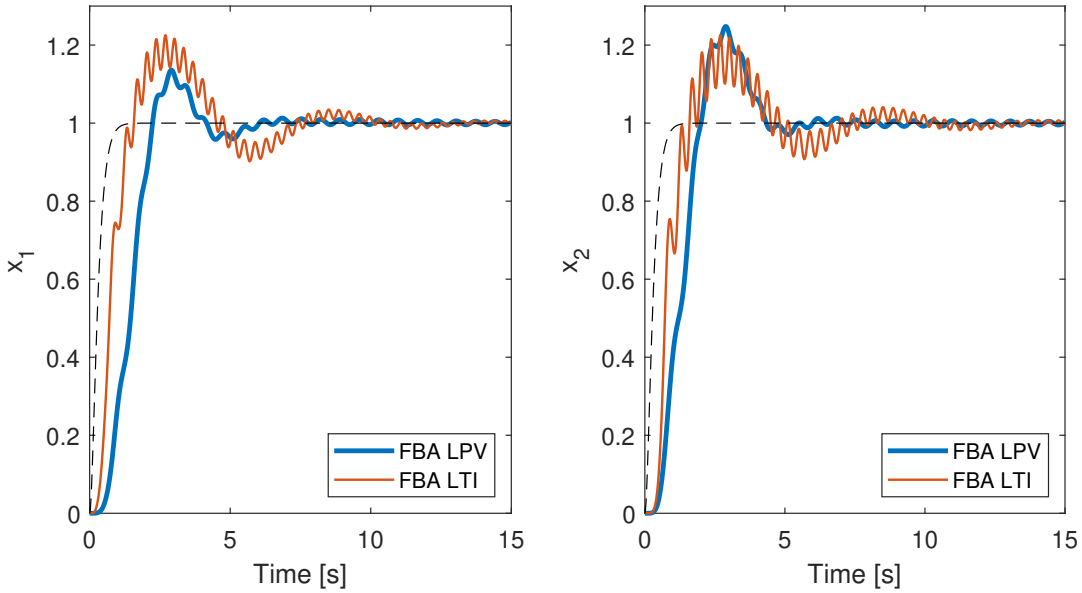

Figure 6-11: Simulation of the closed-loop nonlinear system with the FBA LPV controller and FBA LTI controller. The reference signal is the dashed black line, it is a low pass filtered step.

	FCH	*FBA*
Convexity	✗	✗
Design for unstable plant	✓	✓
Generalized plant form	✗	✓
No prior knowledge required	✓	✗
Fastest computation time	✓	✗

Table 6-4: Characteristics of FCH and FBA.

the control design procedure for FCH has been formulated in the 4-block form, it is still possible to formulate different mixed-sensitivity problems using the four available sensitivities. Finally, FCH is computationally less demanding because it has to take one constraint into account which guarantees stability and performance (see (5-12)), while FBA has to perform a stability test each iteration, flag a set of stabilizing parameters and optimize them afterwards, see Algorithms 3 and 4. The comparison between the LPV and LTI controller has shown that there is an advantage to extending LTI controllers to be LPV. The latter are just as robust as the LTI controllers, but since there is more freedom in the control design, it also performs better. Overall, it can be concluded that a gain-scheduled LPV controller is an obvious improvement that is readily realized. The two controllers do have one disadvantage in common, which is the fact that stability and performance can only be guaranteed locally. However, the controllers are global parametrizations, so they are derived from an infinite set of LTI controllers dependent on the scheduling variables and the LPV controller is optimized such that we have local stability for the infinite set of LTI systems, i.e. the frozen behaviours of the original system.

6-2 Helicopter model

The helicopter model presented in [50] is modified by the authors in [51] such that it fits the LPV system description. The state-space representation of the helicopter is given as follows:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0.75 & 2 & cos(p) & sin(p) \\ 0 & 0.5 & -sin(p) & cos(p) \\ 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 10 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{6-12}$$
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

This system represents a two-input system with identical actuator dynamics, a parameter dependent rotation matrix, and second order plant dynamics [51]. The trajectory of the scheduling variable p is in the range $-\pi \leq p \leq \pi$ and can be described as a sine wave: $p(t) = \pi sin(5t/\pi)$. In this case, seven linearly spaced operating points are picked in the operating range: $p = \begin{bmatrix} -\pi & -\frac{2}{3}\pi & -\frac{1}{3}\pi & 0 & \frac{1}{3}\pi & \frac{2}{3}\pi & \pi \end{bmatrix}$. The rotation matrix depends on the scheduling variable and therefore for every p the system dynamics change. The FRFs of the system at different operating points are shown in Figure 6-12. It should be noted that for $p = -\pi$ and $p = \pi$ we get the same rotation matrix, since $cos(\pi) = cos(-\pi) = -1$ and $sin(\pi) = sin(-\pi) = 0$, therefore we can also simply examine six operating points instead of seven.

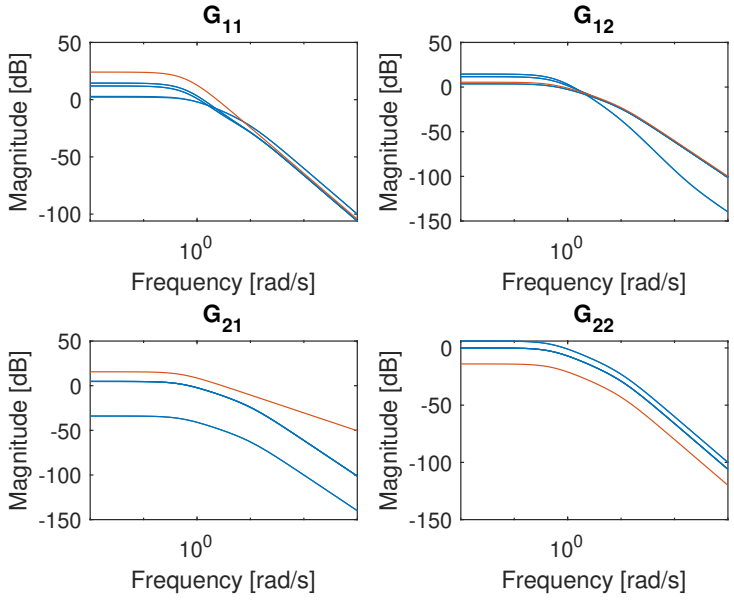

Figure 6-12: FRFs of the helicopter model at six different operating points in blue and the nominal system response in orange.

6-2-1 Design Objectives

Consider the weighted generalized plant interconnection in Figure 2-6, which is used to design a mixed-sensitivity controller. The design objectives for this system are in general the same as described for the mass-spring-damper system in Section 6-1-1. Thus, the system should be locally internally stable at every operating point and the $\mathcal{H}_\infty$ norm of the weighted closed-loop system should be smaller than 1 at every operating point for the selected weighting filters.

Weighting Filter Selection The weighting filters from [51] are used and modified to shape the sensitivity functions. The output weighting filters are $W_z = diag(W_S, W_S, W_T, W_T)$. The transfer function of W_S for the sensitivity and process sensitivity is parametrized as follows with $\omega_B^* = 0.1$ rad/s, $\varepsilon_A = 10^{-2}$ and $M_s = 2$:

$$W_S = \frac{s/2 + 0.1}{s + 10^{-3}}. \tag{6-13}$$

The weighting filter W_T is designed for both the controller and complimentary sensitivity. For this system the weighting filter is simply a static gain that acts as a bound for the control action:

$$W_T = \frac{1}{280}. \tag{6-14}$$

The input weighting filters are $W_w = diag(W_R, W_R, I, I)$, where W_R is a weighting filter for reference signal, it is given by

$$W_R = \frac{20}{s + 0.2}. \tag{6-15}$$

These weighting filters roughly imply objectives of decoupled command response and disturbance rejection with zero steady state error for tracking performance [51].

6-2-2 Controller parametrization

The system has one scheduling variable p, but because of the trigonometric functions it convenient to define two new scheduling variables $p_1 = sin(p)$ and $p_2 = cos(p)$ that depend on p. This way it is easier to take the elements of the rotation matrix into account for the controller parametrization. For the MSD example we also used two scheduling variables, therefore the controller parametrizations for FBA and FCH described in Section 6-1-3 are also directly applicable for the controller design of this system. Hence, a PID controller with a second order low pass filter is also designed for this system using both FBA and FCH, see (6-6) and (6-9), respectively.

6-2-3 Results

The resulting control parameters for the LPV controllers for both FBA and FCH are shown in Tables 6-5 and 6-6. What is striking is that there are no parameters for LTI controllers in the tables, this is because a PID and low pass filter LTI controller cannot stabilize this system like an LPV controller can. This is due to the rotation matrix, it changes the sign

	γ	k_{p_0}	k_{p_1}	k_{p_2}	k_{i_0}	k_{i_1}	k_{i_2}	k_{d_0}	k_{d_1}	k_{d_2}
LPV FCH	0.93	2.56	0.73	81.89	5.17	4.05	38.3	-0.04	0.01	10.33
LPV FBA	0.98	0.17	0.38	103.41	5.11	-0.57	56.75	0.02	0.03	12.99

Table 6-5: Parameters of the LPV PID controller. See Table 6-6 for the gains k_1, k_2, k_3, k_4 and the parameters of the low pass filter.

	ω_c	Q	k_1	k_2	k_3	k_4
LPV FCH	1.87	1.99	2.06	0	1.36	2.00
LPV FBA	0.29	1.32	1.52	0	1.14	1.77

Table 6-6: Parameters of the PID controller and the low pass filter.

of the elements in the state matrix A of the state-space representation of the system. This becomes apparent when $p_1 = sin(p)$ and $p_2 = cos(p)$ are computed for the range of p:

$$\begin{array}{lllllll} p_1 & = [& 0 & -\frac{1}{2}\sqrt{3} & -\frac{1}{2}\sqrt{3} & 0 & \frac{1}{2}\sqrt{3} & \frac{1}{2}\sqrt{3} &] \\ p_2 & = [& -1 & -\frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & -\frac{1}{2} &] \end{array}$$

Note that p_1 and p_2 are paired, so for $p = -\pi$ we have $p_1 = 0$ and $p_2 = -1$ and for $p = -\frac{2}{3}\pi$ $p_1 = -\frac{1}{2}\sqrt{3}$ and $p_2 = -\frac{1}{2}$ and so on. The dynamics of the system change and therefore the controller should adapt to the new dynamics accordingly. By choosing k_{p_2}, k_{i_2} and k_{d_2} larger the controller stabilizes the plant for all six operating points, because p_2 determines if the controller gain is overall negative or positive. This in turn switches the feedback that the controller provides at the different operating points, which should be positive at the first, second and last operating points to guarantee local stability and negative at the rest of the operating points. An LTI controller is not capable of the same dynamic change in behaviour as an LPV controller and can therefore only guarantee local stability at three operating points. This system therefore shows the obvious benefits of an LPV controller when a system is parameter dependent.

6-2-4 Comparison of LPV controllers

Based on Tables 6-5 and 6-6 there are some minor differences between the control parameters, but it seems that both controllers were optimised near a (local) minimum. This can also be seen when we look at the singular values of the closed-loop system in Figure 6-14, where there are barely any differences between the FRFs of the FCH and FBA controlled system. Some subtle differences can be seen in the step response of the nonlinear system simulation in Figure 6-15. The FBA controlled system has a slightly higher overshoot, but the responses are very similar. The peaks in the step response are due to the trajectory of the scheduling variable, which is a sinusoid $p(t) = \pi \sin(5t/\pi)$ as shown in Figure 6-13. A different reference signal is used in Figure 6-16 to examine the tracking performance for an oscillatory signal. In this case, a pulse is generated and passed through a low pass filter. Both controllers are able to track this signal with minimal errors.

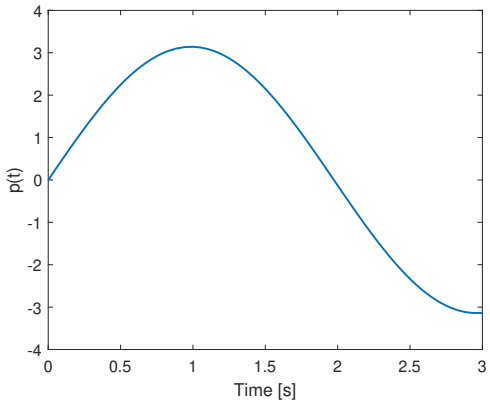

Figure 6-13: Trajectory of the scheduling variable $p(t) = \pi \sin(5t/\pi)$.

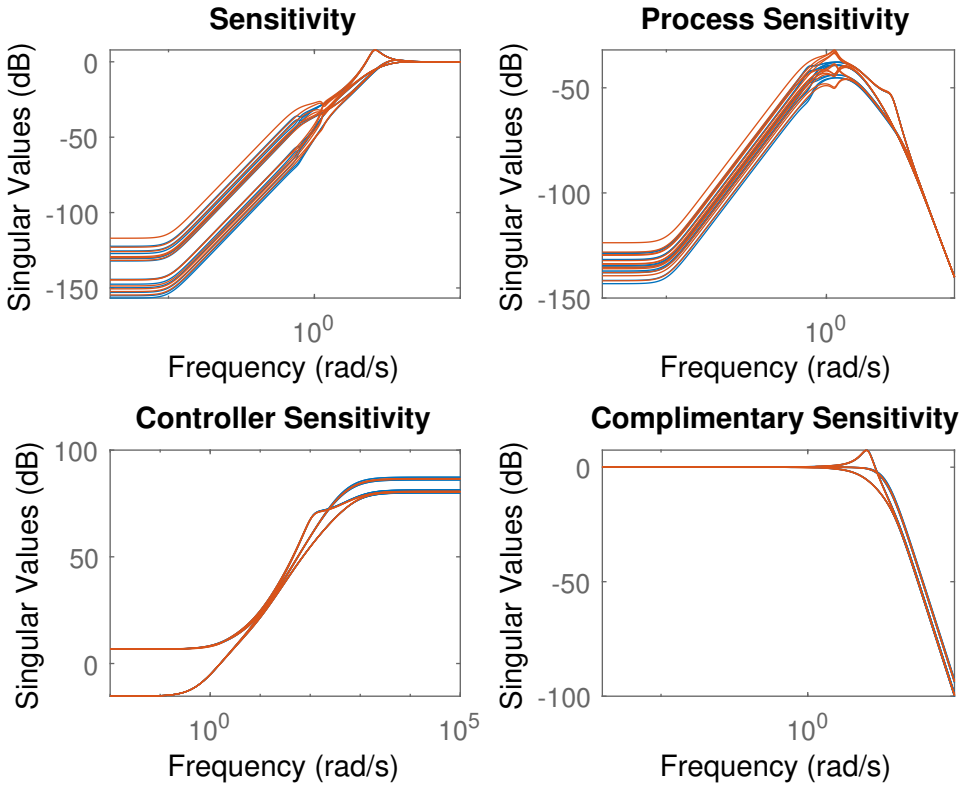

Figure 6-14: Singular values of the closed-loop sensitivity FRFs.

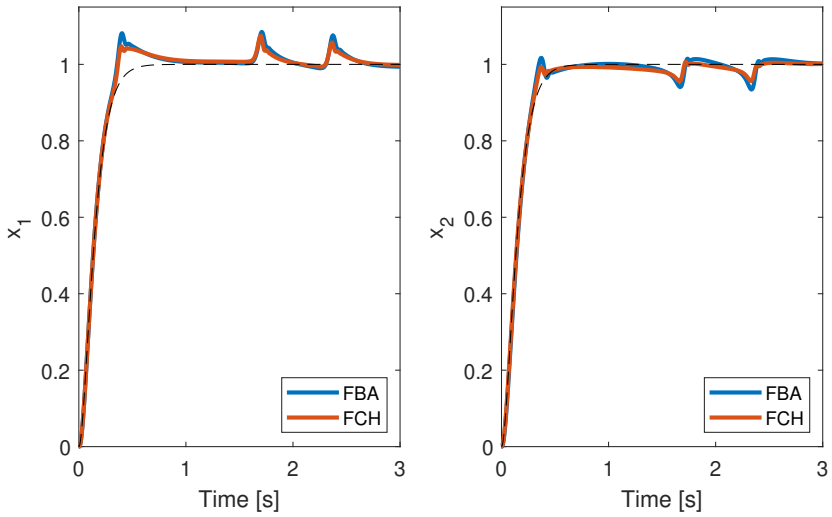

Figure 6-15: Simulation of the closed-loop nonlinear system with the FBA and FCH LPV controllers. The reference signal is a low pass filtered step.

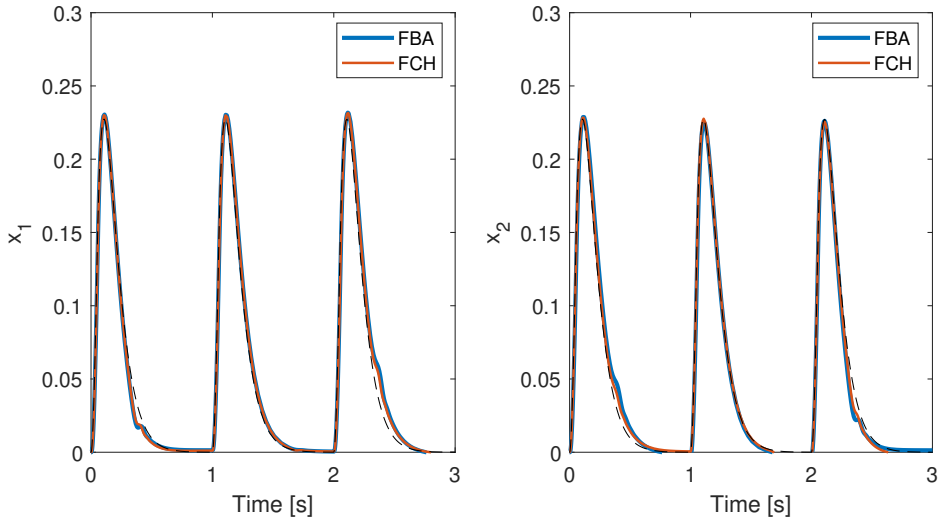

Figure 6-16: Simulation of the closed-loop nonlinear system with the FBA and FCH LPV controllers. The reference signal is a low pass filtered pulse response.

6-2-5 Conclusions

The main goal of this example system was to show the benefits of LPV control design. In that respect it served its purpose, because both controller design methods were unable to synthesise a stabilizing LTI controller. This is because the system has a strong dependency on the scheduling variable, even more so than the MSD in Section 6-1. For this reason, it was not possible to design an LTI controller that stabilizes the system locally at the operating points of interest. Only an LPV controller suffices to guarantee local internal stability, because the dynamics of the controller change accordingly with the dynamics of the system due to their shared dependency on the scheduling variable p. The differences between the performance that the FCH and FBA controllers showed in this example were not as notable as for the MSD. The two controllers both satisfied the local performance objectives.

Chapter 7

Conclusions & Recommendations

7-1 Conclusions

The LTI framework is frequently used to model mechatronic systems and design controllers. If the mechatronic system has nonlinearities, such as position dependent dynamics, then the nonlinear behaviour of the system tends to be more dominant. Consequently, it becomes increasingly difficult to describe the dynamics of the system using an LTI model. Moreover, the trade-off between robustness and performance will lean more towards the design of a robustly stabilizing controller with relatively poor performance. This is due to the fact that the LTI framework is limited in terms of control design compared to the LPV framework.

In theory, the LPV framework provides an improvement to the closed-loop performance because the controller has more degrees of freedom as it takes the nonlinearities into account. The nonlinearities of the system are contained within the scheduling variable, this enables us to apply LTI control design methods to nonlinear systems. This allows for improvement of performance of the closed-loop system when the same controller structure and weighting filters are used. This theory was put to test by performing simulations on parameter dependent systems with the two novel data-driven LPV controller synthesis methods to design (fixed) structured controllers for MIMO systems.

These methods are based on state-of-the-art control design methods in the LTI framework as discussed in Chapter 3. The methods are developed using the local approach in the LPV framework. The local approach uses frozen frequency response functions at different operating points of the process and is then followed by interpolation of the resulting controllers over the operating range of the system. This implies that only local stability and performance are guaranteed to the closed-loop system. However, as opposed to classic gain scheduling control, the LPV controllers are a global parametrization that take the scheduling dynamics into account.

In Chapter 6 we have shown that LPV control is crucial to guarantee closed-loop internal stability of the parameter dependent system. In case of the mass-spring-damper system, the LPV controller showed improvements in terms of performance compared to the LTI controller.

Moreover, the simulations on the helicopter system showed that the closed-loop system could simply not be stabilized by an LTI controller.

Besides comparing the LPV controllers to LTI controllers, we also compared the two new LPV control design methods to each other. We learned that the Fixed-structure $\mathcal{H}_\infty$ Controller (FCH) is easier to use than the Feedback Autotuner (FBA), because it does not require prior knowledge with respect to the open-loop poles and integrators of the plant. FCH is computationally also less demanding, because FBA evaluates stability and performance separately, while with FCH the optimization only requires to satisfy one constraint that guarantees both local stability and local performance.

Finally, it can be concluded that state-of-the-art LTI data-driven controller design methods can be systematically extended to the LPV framework to improve the performance of MIMO systems by means of an LPV controller. Especially the FCH control design method shows potential of being able to design a robust fixed-structure controller that satisfies performance objectives that were otherwise not achievable with an LTI controller. We can conclude that, with the development of these two data-driven MIMO LPV control design methods, the first steps have been taken into this relatively new field of control theory. Next, the recommendations for further studies are presented.

7-2 Recommendations

During this thesis it was also attempted to develop a convex controller synthesis method based on [5], but due to time restrictions the method is not fully developed. In Appendix A it is explained what steps should be taken to derive a convex controller synthesis method. Some of the steps have been taken, but the optimization problem could not be solved. Further work is required to fully develop the method and solve the optimization problem.
Furthermore, future works should aim to derive conditions that provide global stability and performance conditions for LPV systems, as opposed to the local stability and performance conditions derived in this thesis.

Appendix A

Fixed-order Controller Design

A-1 Requirements for convex optimization

In Section 5-2 a non-convex optimization problem is presented for fixed-structure controller synthesis. In theory, it should be possible to reformulate the controller synthesis problem such that a fixed-order controller can be designed via convex optimization. In case only SISO LTI systems are considered, it has already been shown that it is possible to design a fixed-order controller via convex optimization in [5]. Consider the following condition from Section 5-2-2 for easy reference:

$$\mathfrak{R}\{\det(I - \gamma^{-1}\tilde{H}_p\hat{\Delta})\alpha\} \geq 0, \forall\omega \in \mathbb{R} \cup \{\infty\}, \forall p \in \mathcal{P}, \forall\hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}. \tag{A-1}$$

This condition leads to a non-convex optimization problem, as shown in Section 5-2-3. To make the condition in (A-1) convex, the following steps must be taken:
(1) The multiplier α should be absorbed;
(2) The condition should be reformulated without the need of a determinant;
(3) Get rid of $\hat{\Delta}$.

If these steps are satisfied, then we could get an optimization problem as in (3-10). In the remainder of this chapter we will present the attempts to satisfy the steps as mentioned above.

A-2 Design for Sensitivity

Many attempts have been made to come up with a condition where α can be absorbed for the 4-block problem, but none of them were successful. However, if we choose to design just for the sensitivity, it becomes possible to absorb alpha. First, for completeness sake, the condition for stability is given, where it is not necessary (yet) to absorb α. The performance condition is then given as well, after which it will be reformulated such that α can be absorbed. After this, it is shown how to get rid of the determinant.

After the condition for stability is given, the performance condition is presented and shown that it is possible to absorb α.

A-2-1 Stability

It is assumed that for each p the plant G_p and controller K_p can be represented as coprime factorizations over $\mathcal{RH}_\infty$ (see Definition 2.2), such that for each $p \in \mathcal{P}$ it follows that $G_p = M_p^{-1}(j\omega)N_p(j\omega)$ and $K_p = X_p(j\omega)Y_p^{-1}(j\omega)$. Here K_p represent the controller dependent on p. Due to the fact that the controller and plant are formulated in the coprime factorisation representation, design for unstable plants is possible as well. Consider the interconnection in Figure 5-1. The input-output map $r \mapsto e$ gives the sensitivity function $S_p = (I + G_pK_p)^{-1}$, which is then denoted, using the coprime factorizations of G_p and K_p, as

$$S_p = (I + M_p^{-1}N_pX_pY_p^{-1})^{-1} = Y_p(M_pY_p + N_pX_p)^{-1}M_p. \tag{A-2}$$

In Section 5-2-2 a condition for stability of a 4-block system was derived. In this case, it is only required to derive a stability condition for the sensitivity function S_p. Instability can occur if $I + G_pK_p = 0$ for any $p \in \mathcal{P}$, therefore the following theorem is focussed on assuring that $I + G_pK_p \neq 0$ for all $p \in \mathcal{P}$.

Theorem A.1. Let $\{N_p, M_p, X_p, Y_p\} \in \mathcal{RH}_\infty$ and let S_p be defined as in (A-2) then the following conditions are equivalent. For all $p \in \mathcal{P}$

(i) $S_p \in \mathcal{RH}_\infty$.
(ii) $(M_pY_p + N_pX_p)^{-1} \in \mathcal{RH}_\infty$.
(iii) $\det(M_pY_p + N_pX_p) \neq 0$, $\forall s \in \mathbb{C}^+ \cup \mathbb{C}^0 \cup \{\infty\}$.
(iv) There exists a multiplier $\{\alpha, \alpha^{-1}\} \in \mathcal{RH}_\infty$ such that

$$\Re\{\det(S_p)\alpha\} > 0, \ \forall \omega \in \mathbb{R} \cup \{\infty\}, \ \forall p \in \mathcal{P}. \tag{A-3}$$

The proof is virtually the same as the proof for Theorem 5.1. In summary, the coprime factorizations over $\mathcal{RH}_\infty$ are used to guarantee stability of an (un)stable plant G_p or controller K_p. It should be noted that Theorem A.1 implies stability of S_p, therefore (5-3) is internally stable (see Theorem 2.1).

A-2-2 Performance

The performance condition $\|W_PS_p\|_\infty \leq \gamma$ is directly related to the Main Loop Theorem (Theorem 2.8), therefore its results are used to formulate the following theorem. Let $\hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}$ be defined as in Section 5-2-2.

Theorem A.2. Consider S_p as defined in (A-2) and let $W_P \in \mathcal{RH}_\infty$. Then, both the local stability requirement, $S_p \in \mathcal{RH}_\infty$, and the performance specification $\|W_PS_p\|_\infty \leq \gamma$ are satisfied if and only if there exists a multiplier $\alpha \in \mathcal{RH}_\infty$ such that

$$\begin{aligned}\Re\{\det(M_pY_p + N_pX_p - \gamma^{-1}M_p\hat{\Delta}W_PY_p)\alpha\} > 0,\\ \forall \omega \in \mathbb{R} \cup \{\infty\}, \forall p \in \mathcal{P}, \forall \hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}.\end{aligned} \tag{A-4}$$

The proof of this theorem is the same as the proof for Theorem 5.2 in Section 5-2-2.

A-2-3 Steps towards convex optimization

In Section 5-2-3 the parametrization for fixed-order controllers is given, which is necessary for convex optimization. The following theorem gives the necessary and sufficient conditions such that the sensitivity function S_p is stable and it satisfies an $\mathcal{H}_\infty$ performance bound.

Theorem A.3. Given a plant $G_p = M_p^{-1}N_p$, with $\{N_p, M_p\} \in \mathcal{RH}_\infty$ coprime, and a weighting filter $W_P \in \mathcal{RH}_\infty$, the following statements are equivalent:

(i) There exists a controller K_p that satisfies local stability and local performance requirements.

(ii) There exists a controller $K_p = X_p Y_p^{-1}$, with $\{X_p, Y_p\} \in \mathcal{RH}_\infty$ coprime, such that

$$\begin{aligned} \Re\{\det(M_p\tilde{Y}_p + N_p\tilde{X}_p - \gamma^{-1}M_p\hat{\Delta}W_P\tilde{Y}_p)\} > 0, \\ \forall\omega \in \mathbb{R}\cup\{\infty\},\ \forall p \in \mathcal{P},\ \forall\hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}. \end{aligned} \tag{A-5}$$

Proof. The difference between (A-4) and (A-5) is that α is absorbed inside the structure of the controller. The following result of the Main Loop theorem is used:
$\|(I + G_pK_p)^{-1}\|_\infty \leq \gamma \iff \det(I - \gamma^{-1}\hat{\Delta}W_P(I + G_pK_p)^{-1}) \neq 0$ to show how α can be absorbed:

$$\det(I - \gamma^{-1}\hat{\Delta}W_P(I + G_pK_p)^{-1}) \neq 0 \tag{A-6}$$

$$\det(I + G_pK_p - \gamma^{-1}\hat{\Delta}W_P) \neq 0, \tag{A-7}$$

$$\det(I + M_p^{-1}N_pX_pY_p^{-1} - \gamma^{-1}\hat{\Delta}W_P) \neq 0, \tag{A-8}$$

$$\det(M_pY_p + N_pX_p - \gamma^{-1}M_p\hat{\Delta}W_PY_p) \neq 0, \tag{A-9}$$

Equation (A-6) is equivalent to (A-9) in terms of non-singularity. We can then apply Theorem A.2 to introduce a mulitiplier α. This is done such that the Nyquist plot of (A-9) does not encircle nor go through the origin, hence (A-9) is stable according to the Generalized Nyquist Theorem.

$$\Re\{\det(M_pY_p + N_pX_p - \gamma^{-1}M_p\hat{\Delta}W_PY_p)\alpha\} > 0, \tag{A-10}$$

$$\Re\{\det(M_p\tilde{Y}_p + N_p\tilde{X}_p - \gamma^{-1}M_p\hat{\Delta}W_P\tilde{Y}_p)\} > 0, \tag{A-11}$$

where $\tilde{X} = X\alpha$ and $\tilde{Y} = Y\alpha$. □

The proof above shows that it is possible to absorb α, therefore step 1 is satisfied. The condition in (A-5) can then be used to formulate an optimization problem as follows:

$$\begin{aligned} \min_{\gamma,\rho_x,\rho_y} \quad & \gamma \\ \text{s.t. } & \Re\{\det(M_p\tilde{Y}_p + N_p\tilde{X}_p - \gamma^{-1}M_p\hat{\Delta}W_P\tilde{Y}_p)\} \geq 0, \\ & \forall\omega \in \mathbb{R}\cup\{\infty\},\ \forall p \in \mathcal{P},\ \forall\hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}. \end{aligned} \tag{A-12}$$

This optimization problem can be reformulated without taking the real part of the determinant. Consider some matrix A and the Main Loop Theorem, in which it is stated that

$\det(I - A) \neq 0$, this implies that if $\det(I - A) = 0$ then there should exist some frequency-dependent eigenvalue $\lambda(j\omega)$ and complex eigenvector $v(j\omega)$ such that

$$(I - A)v = 0 \tag{A-13}$$
$$\|A\|\|v\| = |\lambda|\|v\|. \tag{A-14}$$

This leads to the following optimization problem:

$$\begin{aligned} \min \quad & e \\ \text{s.t.} \quad & \|(I - A)v\| < e. \end{aligned} \tag{A-15}$$

It follows that if $e = 0$, then $\det(I - A) = 0$ and if $e \neq 0$, then $\det(I - A) \neq 0$. It is stressed that $v \in \mathcal{RH}_\infty$ due to a homotopy argument and that it is a frequency-dependent eigenvector. Therefore, there are similarities to $v(j\omega)$ and $\alpha(j\omega)$. This procedure is then applied to (A-12), where we then end up with the following condition:

$$\begin{aligned} \min_{e,\rho_x,\rho_y} \quad & e \\ \text{s.t.} \quad & \|\Big(M_p Y_p + N_p X_p - \gamma^{-1} M_p \hat{\Delta} W_P Y_p\Big)\, v(j\omega)\| < e \\ & \forall \omega \in \mathbb{R} \cup \{\infty\},\ \forall p \in \mathcal{P},\ \forall \hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}. \end{aligned} \tag{A-16}$$

It should be noted that $v(j\omega)$ can be absorbed by M_p and Y_p, similar to how α was absorbed in (A-9). During the implementation of this condition we found that it was not possible to simultaneously minimize e and the control parameters ρ_x and ρ_y at the same time. Therefore, convex optimization was not possible and we opted for a convex analysis condition where we just minimize the performance bound e. However, this also did not show satisfactory results because the condition is numerically unreliable. In order to really use (A-16), further research is required.

Finally, as a next step we should also absorb $\hat{\Delta}$ in (A-16) by only looking at the perimeter, which can be seen as just an arbitrary rotation. Then the question is if this rotation can be realised somehow computationally efficiently or represented as a multiplier (similar to the S-procedure). Then we get 2 LMI's. We also need to think about how to get rid off the numerical problem with the bound in (A-16).

Bibliography

[1] A. Khalik, "Data-driven controller design." unpublished report, 2020.

[2] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, p. 3–35, 2013.

[3] A. Nicoletti, "A Data-Driven Frequency-Domain Approach for Robust Controller Design via Convex Optimization," 2017. Presented 04 Dec 2017.

[4] R. Pintelon and J. Schoukens, *System identification: a frequency domain approach*. John Wiley and Sons, Inc, 2012.

[5] A. Karimi, A. Nicoletti, and Y. Zhu, "Robust $\mathcal{H}_\infty$ controller design using frequency-domain data via convex optimization," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 12, p. 3766–3783, 2016.

[6] A. Nicoletti and A. Karimi, "A data-driven method for computing fixed-structure low-order controllers with $\mathcal{H}_\infty$ performance," *2018 European Control Conference (ECC)*, 2018.

[7] E. V. Solingen, J. V. Wingerden, and T. Oomen, "Frequency-domain optimization of fixed-structure controllers," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 12, p. 3784–3805, 2016.

[8] D. Garcia, A. Karimi, and R. Longchamp, "Data-driven controller tuning using frequency domain specifications," *Industrial and Engineering Chemistry Research*, vol. 45, no. 12, p. 4032–4042, 2006.

[9] A. Karimi and G. Galdos, "Fixed-order h-infinity controller design for nonparametric models by convex optimization," *Automatica*, vol. 46, no. 8, pp. 1388–1394, 2010.

[10] H. Parastvand and M.-J. Khosrowjerdi, "Parameterised controller synthesis for siso-lti uncertain plants using frequency domain information," *International Journal of Systems Science*, vol. 47, no. 1, pp. 32–44, 2016.

[11] R. Tóth, *Modeling and identification of linear parameter-varying systems*. Springer, 2010.

[12] A. Bachnas, R. Tóth, J. Ludlage, and A. Mesbah, "A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study," *Journal of Process Control*, vol. 24, no. 4, p. 272–285, 2014.

[13] A. Packard, "Gain scheduling via linear fractional transformations," *Systems and Control Letters*, vol. 22, no. 2, p. 79–92, 1994.

[14] C. W. Scherer, "Mixed h2/hinf control for time-varying and linear parametrically-varying systems," *International Journal of Robust and Nonlinear Control*, vol. 6, no. 9-10, p. 929–952, 1996.

[15] G. Balas, I. Fialho, A. Packard, J. Renfrow, and C. Mullaney, "On the design of lpv controllers for the f-14 aircraft lateral-directional axis during powered approach," *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, 1997.

[16] I. Fialho and G. Balas, "Road adaptive active suspension design using linear parameter-varying gain-scheduling," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, p. 43–54, 2002.

[17] L. Kovacs, "Linear parameter varying (lpv) based robust control of type-i diabetes driven for real patient data," *Knowledge-Based Systems*, vol. 122, p. 199–213, 2017.

[18] N. Chen and J. T. Wen, "Lpv control design for precision motion systems," *2014 American Control Conference*, 2014.

[19] S. Formentin and S. M. Savaresi, "Virtual reference feedback tuning for linear parameter-varying systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, p. 10219–10224, 2011.

[20] S. Formentin, D. Piga, R. Tóth, and S. M. Savaresi, "Direct learning of lpv controllers from data," *Automatica*, vol. 65, p. 98–110, 2016.

[21] R. D. Rozario and T. Oomen, "Frequency response function identification of lpv systems: a global approach with application to mechanical systems," *IFAC-PapersOnLine*, vol. 51, no. 15, p. 108–113, 2018.

[22] T. Bloemers, R. Tóth, and T. Oomen, "Towards data-driven lpv controller synthesis based on frequency response functions," in *IEEE 58th Conference on Decision and Control*, pp. 5680–5685, 2019.

[23] T. Bloemers, R. Tóth, and T. Oomen, "Data-driven rational lpv controller synthesis for unstable systems using frequency response functions," in *39th Benelux Meeting on Systems and Control, 2020* (B. J. Raffaella Carloni and E. Lefeber, eds.), p. 144, 2020.

[24] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*. Wiley, 2010.

[25] R. Curtain, G. Weiss, and M. Weiss, "Coprime factorization for regular linear systems," *Automatica*, vol. 32, no. 11, p. 1519–1531, 1996.

[26] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*. Prentice Hall, 1996.

[27] C. Scherer, "Theory of robust control."

[28] R. Smith, "A mimo nyquist stability test example," 2015.

[29] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm intelligence*. Morgan Kaufmann, 2001.

[30] S. Ebbesen, P. Kiwitz, and L. Guzzella, "A generic particle swarm optimization matlab function," *2012 American Control Conference (ACC)*, 2012.

[31] R. Perez and K. Behdinan, "Particle swarm approach for structural design optimization," *Computers amp; Structures*, vol. 85, no. 19-20, p. 1579–1588, 2007.

[32] A. Rantzer and A. Megretski, "A convex parameterization of robustly stabilizing controllers," *IEEE Transactions on Automatic Control*, vol. 39, no. 9, p. 1802–1808, 1994.

[33] P. S. Heuberger, P. M. Hof, and B. Wahlberg, *Modelling and Identification with Rational Orthogonal Basis Functions*. Springer, 2005.

[34] K. Zhou and J. C. Doyle, *Essentials of robust control*. Prentice Hall, 1998.

[35] M. G. Wassink, M. V. D. Wal, C. Scherer, and O. Bosgra, "Lpv control for a wafer stage: beyond the theoretical solution," *Control Engineering Practice*, vol. 13, no. 2, p. 231–245, 2005.

[36] P. Koelewijn, R. Tóth, and H. Nijmeijer, "Linear parameter-varying control of nonlinear systems based on incremental stability," *IFAC-PapersOnLine*, vol. 52, no. 28, p. 38–43, 2019.

[37] R. Wang, R. Tóth, and I. R. Manchester, "A comparison of lpv gain scheduling and control contraction metrics for nonlinear control," *IFAC-PapersOnLine*, vol. 52, no. 28, p. 44–49, 2019.

[38] C. Briat, *Linear Parameter-Varying and Time-Delay Systems Analysis, Observation, Filtering amp; Control*. Springer Berlin, 2016.

[39] M. Schoukens and R. Tóth, "Frequency response functions of linear parameter-varying systems," *IFAC-PapersOnLine*, vol. 52, no. 28, p. 32–37, 2019.

[40] C. Hoffmann and H. Werner, "A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, p. 416–433, 2015.

[41] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled H_∞ control of linear parameter-varying systems: a design example," *Automatica*, vol. 31, no. 9, pp. 1251 – 1261, 1995.

[42] A. Packard, "Gain scheduling via linear fractional transformations," *Systems & Control Letters*, vol. 22, no. 2, pp. 79 – 92, 1994.

[43] F. Wu, *Control of linear parameter varying systems*. PhD thesis, Univ. Berkeley, 1995.

[44] M. Kunze, A. Karimi, and R. Longchamp, "Gain-scheduled controller design by linear programming," in *2007 European Control Conference (ECC)*, pp. 5432–5438, 2007.

[45] A. Karimi, M. Kunze, and R. Longchamp, "Robust controller design by linear programming with application to a double-axis positioning system," *Control Engineering Practice*, vol. 15, no. 2, pp. 197 – 208, 2007.

[46] A. Karimi and Z. Emedi, "H_∞ gain-scheduled controller design for rejection of time-varying narrow-band disturbances applied to a benchmark problem," *European Journal of Control*, vol. 19, no. 4, pp. 279 – 288, 2013. Benchmark on Adaptive Regulation: Rejection of unknown/time-varying multiple narrow band disturbances.

[47] T. Bloemers, R. Tóth, and T. Oomen, "Data-driven lpv reference tracking for a control moment gyroscope," *IFAC-PapersOnLine*, vol. 52, no. 28, p. 134–139, 2019.

[48] M. Vidyasagar and M. Vidyasagar, *5.1 Closed-loop Stability*, p. 75–76. Morgan and; Claypool, 2011.

[49] C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties*. Academic Press, 1975.

[50] G. Meyer, R. Su, and L. Hunt, "Application of nonlinear transformations to automatic flight control," *Automatica*, vol. 20, no. 1, pp. 103 – 107, 1984.

[51] F. Wu, X. H. Yang, A. Packard, and G. Becker, "Induced l2-norm control for lpv systems with bounded parameter variation rates," *International Journal of Robust and Nonlinear Control*, vol. 6, no. 9-10, pp. 983–998, 1996.