EINDHOVEN UNIVERSITY OF TECHNOLOGY

A survey of literature on controller scheduling

Citation for published version (APA):
Westerhof, R. (2000). *A survey of literature on controller scheduling*. (DCT rapporten; Vol. 2000.016). Technische Universiteit Eindhoven.

Document status and date:
Published: 01/01/2000

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 05. Oct. 2023

A Survey of Literature on Controller Scheduling

Roan Westerhof

WFW report no. 2000.16
CTB595-00-2097

coaches: Marc van de Wal
René van de Molengraft

June 20, 2000

Abstract

One of the research topics at Philips CFT involves the wafer-stepper, an apparatus used in the production of IC's. The mechanical part of the machine exhibits position-dependent dynamics. The performance of the system could possibly be improved using a position dependent controller designed via scheduling techniques.

Different methods for controller scheduling are described. Distinction is made between conventional controller scheduling and LPV controller scheduling. The different methods are compared in order to arrive at a sensible choice for the problem at hand.

Conventional methods cannot guarantee stability and performance. On the contrary, it is not necessary to have the global equations of motion available and there is much freedom in the design of the overall controller. LPV (Linear Parameter Varying) methods guarantee stability and performance, but this is at the cost of conservatism. Also the global equations of motion have to be known, and the controller design procedure is complicated.

Without further investigations, it is not possible to appoint which method comes out to be the best for successful implementation. Both methods have desirable and disadvantageous aspects. For further research, it is useful to pay attention to both conventional and LPV controller scheduling methods.

Contents

Chapter 1

Introduction

1.1 Background

At Philips CFT, research is done for ASM Lithography (ASML), a manufacturer of so-called wafer steppers and scanners. These are mechanical servo systems used in mass-production of integrated circuits (ICs). The technique used in a stepper/scanner is called photolithography. Light passes through an image of the IC (the mask). A lens reduces the size of the image and projects it onto a part of a silicon disc (the wafer) covered with photoresist. The photoresist reacts to the light and the exposed area is then removed with a solvent. After exposure, the mechanical part of the stepper/scanner, the wafer stage, is moved to the next IC.

1.2 Problem description

The focus of this report is the positioning of the wafer under the lens. This has to be done very fast and with high accuracy. At present, a single feedback controller is used for all different positions or areas of motion under the lens. However, the wafer stages of ASML exhibit position-dependent dynamics (see [44]). This means that (anti-)resonance frequencies and associated damping ratios may be different for different operating regions. The designed controller should thus be robust against changing plant dynamics.

The practical consequence of this is the following. When looking at a fixed position in the operating region two types of controller can be compared: A single controller, designed for the entire operating region and accounting for position-dependency via model uncertainty (a robust controller) versus a controller designed for that specific position. The latter may be outperformed by the first with respect to its performance, in terms of bandwidth and stability margins (see [44]). Possibly, performance improvement can be achieved by applying position-dependent controllers, *i.e.*, controllers that adapt their dynamics to the particular operating point.

1.3 Controller scheduling

In the literature on position-dependent control or, more generally, on adapting the controller to a *scheduling parameter* (this can for instance be the position) different definitions are used in different ways. A term often used is *gain-scheduling*. However, this term refers to many different techniques, also to a rigid-decoupling technique which is used in wafer-stepper control at ASML. Therefore, it is necessary to redefine some expressions. *Controller scheduling* is here defined as a control strategy where:

1. the scheduling parameter can be measured as an explicit known function of time;
2. there is *a priori* knowledge available on the relationship between the scheduling parameters and the plant dynamics;
3. the information from 1. and 2. is used to change the controller during the operation.

The basic idea of controller scheduling is represented in Figure 1.1.

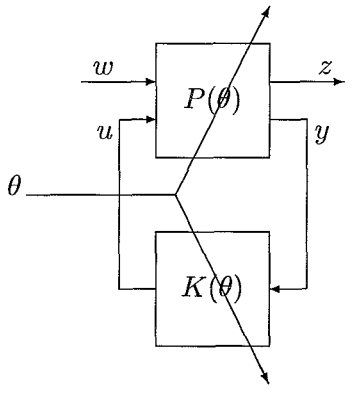

Figure 1.1: Controller scheduling with scheduling parameter θ.

The plant P is subject to exogenous variables w. The controller K feeds the plant with manipulated variables u by measuring the variables y while controlling the variables z. A parameter vector θ induces parameter-dependency into the plant. The idea of controller scheduling is to design the controller in such a way that the parameter-dependency of the plant does not, at least approximately, have any influence on the performance of the plant. Therefore, the controller must also have access to the information in the parameter vector θ.

1.4 Different types of controller scheduling

Using the definition of controller scheduling in Section 1.3, two main directions can be distinguished: conventional controller scheduling and Linear Parameter-Varying (LPV) controller scheduling. In the latter case, the LPV equations of motion of the system are explicitly used

in the controller design, whereas this is not the case for conventional controller scheduling. It is always necessary to know the equations of motion that describe the global system behavior. In principle, stability and performance can be guaranteed all over the operating range. There is much literature on LPV controller scheduling, from different authors on the same topics.

With respect to this LPV controller scheduling, one may wonder whether this technique is useful for ASML practice. At Philips CFT there is experience with the design of controllers via $\mathcal{H}_\infty$-optimization, which is a model-based method. To obtain a model for the wafer-stage, the frequency response functions are approximated by a fit (see [43] for details). This traject offers no direct opportunity for an LPV-based method, since the global equations of motion are not available. However, there are ideas for deriving an LPV model in some way. For the PAS 5500/300D wafer-stepper a position-dependent plant model is available. This offers opportunities to study position-dependent control.

Conventional controller scheduling techniques mostly follow a simpler approach. A set of different, locally designed controllers is considered. When the global system equations are available, the system can be linearized around various operating points, otherwise controllers can be obtained using local analytic or identified models. After that, the controllers are in some way glued together to form one non-linear controller. Generally speaking, disadvantages of this kind of methods are that there are various ways to construct one overall controller out of the locally designed controllers and that it is often not clear which approach is the best in a certain situation. Further, it is not clear how to design the linear controllers such that, after interpolation, the overall controlled system is stable and/or shows the desired performance. In general, these approaches are *ad hoc*. Each author usually comes up with his own approach.

There are many other, less *ad hoc*, approaches between LPV and conventional controller scheduling. One of these methods is the polytopic model approach described in Section 2.8. This method gives an overall stabilizing controller. Also, the method in Section 2.6 guarantees overall stability.

All the methods to be reviewed can be applied to the position-dependent plant model. As described above some approaches do prove stability and are more sophisticated than others, but still these methods are here categorized as conventional controller scheduling.

1.5 Goal and outline of the report

The goal of this report is to give an overview of techniques for controller scheduling that are useful for ASML applications. During the search for literature on controller scheduling it is checked whether there is some experience on the particular technique at Philips CFT. It is also studied what could be worthwhile in this respect to pay attention to. No attention is payed to concepts in literature which are not in line or too far away from current practice. For example, applications where the scheduling parameter is estimated are not taken into account. Some methods are worth mentioning but are not treated completely. For instance, no attempt is made to give a complete survey on state-feedback applications, although some methods are mentioned.

The organization of this report is as follows. Chapter 2 will treat the conventional controller scheduling methods. This chapter is divided into subsections, each of them describing a paper. Chapter 3 will start with an introduction on LPV-techniques and after that the different approaches will be presented. A comparison between both conventional and LPV controller scheduling techniques (also mutual) will be made in Chapter 4. Finally, conclusions are given will be Chapter 5.

Chapter 2

Conventional controller scheduling

2.1 Introduction

The most simple idea of controller scheduling is the following. First divide the operating range in a number of operating regions. This can for example be done by linearizations around operating points, or using an equidistant grid of points. The scheduling parameter, or when there are more scheduling parameters, the parameter vector, determines on-line which is the current operating region. The scheduling parameters can for instance be the position of a system. After constructing operating points, Linear Time Invariant (LTI) controllers are designed in these points. The number of controllers is important for both performance and implementation. Performance will be better for a large number of operating points. On the contrary, a large number of controllers is not preferable from an implementation viewpoint.

Now an overall controller can be constructed out of the different locally designed controllers. One way to do this is to switch off the controller corresponding to the operating region which is left and to switch on the controller corresponding to the operating region which is entered. This is probably not a good idea. For instance, in case of an abrupt transition, the system can become unstable or can exhibit unwanted transient effects. To avoid this kind of problems, the scheduling approach can made more advanced than the one described above.

In this chapter, several approaches of conventional controller scheduling will be described. The chapter is arranged such that first the simple methods will be treated, and after that more sophisticated methods will be described. In the last section (Section 2.9) some methods will be described that are not useful for us, but are still worth mentioning.

2.2 Controller output scheduling

In [17], a description is given of controller output scheduling, compared to an interpolation method for poles and zeros and an interpolation method for state space matrices. The lat-

ter two will be treated in Sections 2.4 and 2.5, respectively. Also in [37], controller output scheduling is used. Plant models are considered to be black boxes obtained at a finite number of operating points. There is no requirement that the controllers be of the same dimension, which is typical for controller output scheduling.

In the simplest case, two stable linear controllers K_0 and K_1 are designed at two extreme operating points represented by the scheduling variables θ_0 and θ_1, respectively. The controllers are linearly interpolated, [17]:

$$K(s) = \kappa_0 K_0(s) + \kappa_1 K_1(s), \qquad \kappa_0 + \kappa_1 = 1, \qquad \kappa_0, \kappa_1 \geq 0 \tag{2.1}$$

where κ_0 decreases linearly from 1 to 0 and κ_1 increases linearly from 0 to 1 as a function of the measured scheduling variable θ. This is also called *controller blending.* When a controller K_i is characterized by

$$K_i : \begin{cases} \dot{x}_{k_i} = A_{k_i} x_{k_i} + B_{k_i} y \\ u = C_{k_i} x_{k_i} + D_{k_i} y \end{cases}, \qquad i = 0, 1, \tag{2.2}$$

the state space description of the interpolated controller K between K_0 and K_1 is

$$K : \begin{cases} \begin{bmatrix} \dot{x}_{k_0} \\ \dot{x}_{k_1} \end{bmatrix} = \begin{bmatrix} A_{k_0} & 0 \\ 0 & A_{k_1} \end{bmatrix} \begin{bmatrix} x_{k_0} \\ x_{k_1} \end{bmatrix} + \begin{bmatrix} B_{k_0} \\ B_{k_1} \end{bmatrix} y \\ u = \begin{bmatrix} \kappa_0 C_{k_0} & \kappa_1 C_{k_1} \end{bmatrix} \begin{bmatrix} x_{k_0} \\ x_{k_1} \end{bmatrix} + \begin{bmatrix} \kappa_0 D_{k_0} & \kappa_1 D_{k_1} \end{bmatrix} y \end{cases}. \tag{2.3}$$

In this representation, the scheduling affects the output. The scheduling has no affect on the state of the controllers. An alternative to this is to apply κ_0 (κ_1) to B_{k_0} (B_{k_1}) instead of C_{k_0} (C_{k_1}). In that case, the input to the overall controller is divided to the locally designed controllers. The states of the controllers are also affected by the scheduling. It is not clear what the difference is between these two methods. This aspect has not been found in literature.

When controllers have been calculated at more than two operating points, not all the controllers need to be on-line. Only the signals from the controllers that correspond to the momentary working region are fed back into the plant. When the region is left and a new region is entered, some controllers have to be switched off and others have to be switched on. It is important that inactive controllers become active bumplessly. Unsuitable initial states of the controller lead to undesired transient effects and performance degradation. Therefore, it is necessary to implement all the controllers in parallel, see Figure 2.1. All the controllers get the output y from the parameter-dependent system $G(\theta)$ with external inputs w and external outputs z. Only the controllers K_i which are in the momentary operating range supply a part of the input u to the plant. The amount of the part of u that each controller contributes to the plant is defined by the values of the variables κ. For the locally designed controllers hold $\sum_{i=1}^{N} K_i = 1$.

Many alternatives can be applied in this framework. It is not only possible to linearly interpolate between two controllers, but also for example exponential weighing is possible and the interpolation can be done between more than two controllers. One can also doubt whether it is necessary to provide input to all the controllers. The state of controllers which are designed at points far away from the current operating point can become meaningless and so

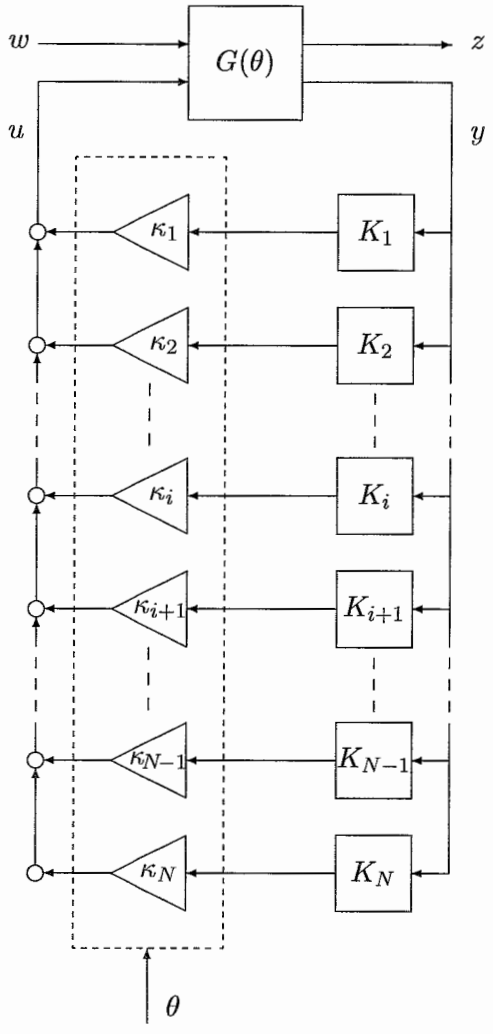

Figure 2.1: Parallel controller output-scheduling.

can yield problems when the output of such a controller has to be supplied to the plant. A solution to this problem is to provide only the controllers with the output of the system which are within or near the momentary operating range. Another solution is to only provide the controllers with the signal that are interpolated. At the moment a new region is entered, the new controller gets the state of the current controller. This alternative is only possible when all controllers have the same state.

Looking at the case of two controllers for simplicity, a problem occurs when unstable controllers are designed. It may not be possible to obtain the required performance if the LTI controllers are restricted to be stable or this may require extremely high order controllers. When an unstable controller K_1 is used and the value for κ_0 is small (so κ_1 is close to one) or when an unstable controller K_0 is used and κ_1 is small (and κ_0 is close to one), the control signal from the unstable controller will diverge. The approach from [37] for scheduling unstable controllers is summarized below.

It is assumed the real plant P at any operating point can be described as

$$P(s)_\kappa = \sum_{i=0}^{N-1} \kappa_i P_i(s), \qquad \sum_{i=0}^{N-1} \kappa_i = 1 \tag{2.4}$$

where $P_i(s)$, $i = 0, ..., N\text{–}1$ is the transfer function associated with each of the N operating points. For simplicity, only the case of two systems is considered;

$$P_\kappa(s) = \kappa_0 P_0(s) + \kappa_1 P_1(s), \qquad \kappa_0 + \kappa_1 = 1, \qquad \kappa_0, \kappa_1 \geq 0 \tag{2.5}$$

Introducing the two known (stable) plant models internally in the controller, it is possible to obtain internal stability for all interpolated values of the two original systems even for unstable controllers. This can be seen in Figure 2.2. The unstable controllers are always in closed-loop with the plant model they were designed for.

This figure could also be represented by the system P_κ in closed-loop with the system below the dashed line, the overall controller $K(s)$. The overall controller can also be described by:

$$K(s) = \frac{K^*}{1 - K^* P_\kappa}, \qquad K^* = \frac{K_0}{1 + K_0 P_0}(1 - \kappa I) + \frac{K_1}{1 + K_1 P_1}\kappa. \tag{2.6}$$

When the overall controller $K(s)$ is examined for an extreme position $\kappa = 0$ or $\kappa = 1$, the original controller K_0 or K_1 occurs. For $\kappa = 0$:

$$K(s) = \frac{\frac{K_0}{1+K_0P_0}}{1 - \frac{K_0}{1+K_0P_0}P_0} = \frac{K_0}{1 + K_0P_0 - K_0P_0} = K_0 \tag{2.7}$$

The results of this method can be generalized to hold for an arbitrary number of systems and controllers. A disadvantage of this approach is the complex implementation. The plant models have to be incorporated twice.

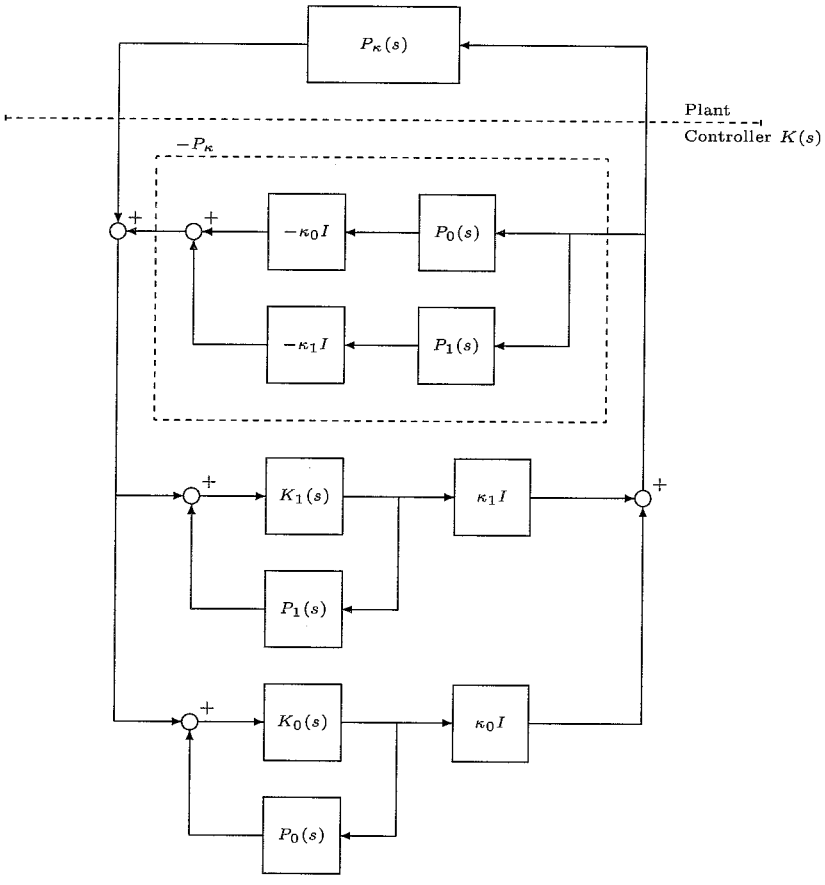

Figure 2.2: Overall structure of the controller output-scheduling approach with (possibly) unstable fixed LTI controllers.

2.3 Smooth interpolation of controller outputs

Interpolating control signals, as described in Section 2.2, is also applied in [7], but extended in such a way that the transition between controllers is smooth. The controllers are obtained with $\mathcal{H}_\infty$ techniques. Also here, there is no requirement that all controllers be of the same dimension.

In principle, at any time only one controller K_i is active. This is a difference between this method and the one in Section 2.2. While one controller is on-line the off-line controllers are being conditioned. This means that the outputs of the off-line controllers are controlled to make them equal to the output of the on-line controller. When switching between controllers from which the outputs are equal, the transfer is bumpless. This is performed using the error signal between the output of an off-line controllers $u_{i_{out}}$ and the output of the active controller u_{active}. These error signals are multiplied with a gain γ_i and this signal is subtracted from the original input to the controllers. For each inactive controller:

$$y_{i_{in}} = y_{original} - \gamma_i(u_{active} - u_{i_{out}}). \tag{2.8}$$

For the active controller it is obvious the gain is not conditioned. The setup for this method is described in Figure 2.3 for three controllers.

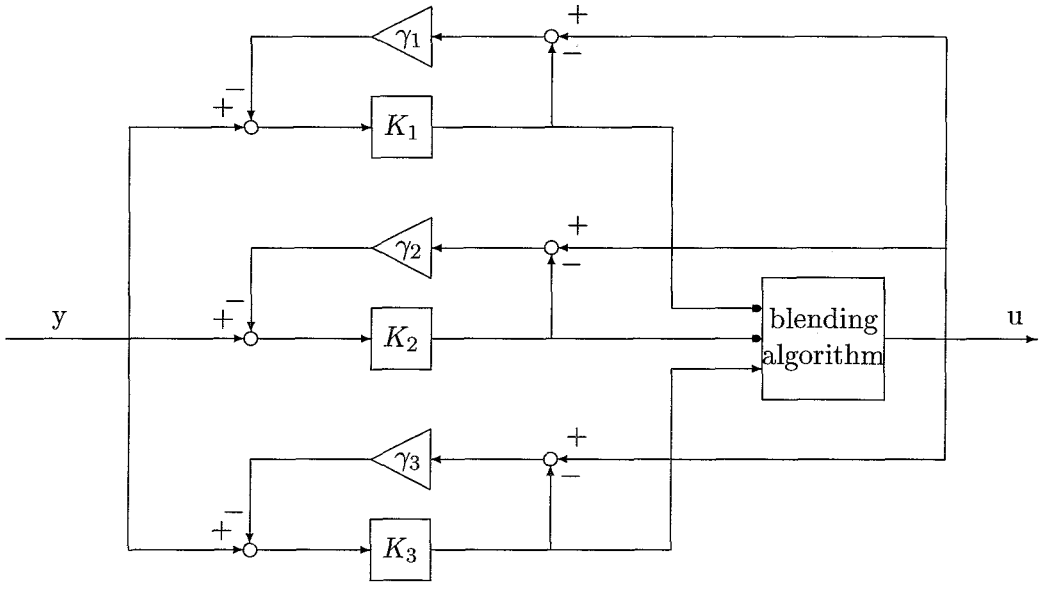

Figure 2.3: Smooth controller output-scheduling.

When the controller gets to the boundaries of its working region, it has to be switched to another controller. When transferring between controllers, a blending approach is used to avoid the bumps that could appear when the conditioning described above is not fully complete. The controllers are each valid in a certain region. The regions overlap to a certain amount. In these overlapping sectors, the outputs of the corresponding controllers are blended by linear interpolation. This is represented in Figure 2.3 by the box with 'blending algorithm'.

2.4 Interpolation of poles, zeros, and gains

In [22], linear controllers are designed at distinct operating points by $\mathcal{H}_\infty$ methods. Main idea of the paper is to interpolate these controllers between both poles and zeros and gains. It is necessary that the locally designed controllers each have the same numerator degree and the same denominator degree. Further, the operating conditions have to be sufficiently close such that migration of poles and zeros from one to the next is recognizable. Physically, this means the poles and zeros should represent the same dynamical effects.

The controller transfer functions which have been computed at different operating points are written in the following form:

$$K(s) = k \cdot \prod_{i=1}^{M} \left(\frac{s + z_i}{s + p_i} \right) \cdot \prod_{j=1}^{N} \left(\frac{s^2 + a_j s + b_j}{s^2 + c_j s + d_j} \right). \tag{2.9}$$

The controller is now in a form where all the coefficients in the numerator and denominator are real. A parameterized linear controller is computed by interpolating poles, zeros, and gains of the distinct-operating-point designs.

To satisfy the requirement that the controllers must have the same numerator degree and the same denominator degree, the $\mathcal{H}_\infty$ controller transfer functions may have to be reduced. A possible problem is that after reduction, the controller poles and zeros do not represent the same dynamical effects and recognition from one to the next is difficult. Otherwise it is sufficient for this last requirement to design sufficiently many controllers. One of the limitations in the approach is the case where the controller designs have many in- and outputs such that reducing the problem to single-input single-output SISO controller components as in (2.9) is inefficient.

2.5 Interpolation of state-space matrices

Another way to practice gain scheduling is interpolating the elements of the controller matrices. This approach is consistent with SISO laws which typically schedule proportional and integral gains. In that case, the linear controllers all have the same structure (apart from the case where notches are used). Due to this, the gains can be interpolated individually, instead of applying controller output scheduling as in Section 2.2.

In [15], a way of scheduling $\mathcal{H}_\infty$ controllers has been investigated. $\mathcal{H}_\infty$ controllers in general do not have an explicit structure and scheduling may be a problem. However, the $\mathcal{H}_\infty$ loopshaping design or coprime factor robust stabilization approach as discussed in, *e.g.* [49, Section 18.2], does produce a controller with a particular simple structure in the form of a plant observer H and state feedback F:

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + H(C\hat{x} - y) + Bu \\ u &= F\hat{x} \end{aligned} \tag{2.10}$$

The robust coprime factor stabilization procedure (see [15] and also [49, Section 5.4]) addresses robustness, but does not directly give a way of specifying performance. To specify performance,

pre- and post-compensating of the plant with shaping filters is applied;

$$P_s = W_2 P W_1. \tag{2.11}$$

The filters W_1 and W_2 specify the performance of the shaped plant P_s. The filter W_1 shapes the external inputs and filter W_2 shapes the external outputs. This actual performance problem is the same as is solved at Philips CFT.

For the controller scheduling, linear interpolation of the gains of the controller matrices is used. For example, the element (i, j) of the state feedback matrix F between the adjacent design points k and l would be calculated as:

$$F_{ij}(\alpha) = \alpha_k F_{k_{ij}} + \alpha_l F_{l_{ij}}, \qquad \alpha_k + \alpha_l = 1, \qquad \alpha_k, \alpha_l \geq 0 \tag{2.12}$$

with $\{\alpha_k = 0,\ \alpha_l = 1\}$ corresponding to operating point l and $\{\alpha_k = 1,\ \alpha_l = 0\}$ corresponding to operating point k. An alternative would be to fit polynomials through all the F's for the whole operating region, but this increases the computation effort required for each evaluation of the controller.

In order to be able to interpolate between controller gains, the shaped plant matrices A, B and C have to vary smoothly with the operating points. The weighting matrices for the linear controller also have to vary smoothly. This is because the observer applies to the weighted plant.

The complexity in terms of the number of parameters which have to be stored and updated is relatively high. Essentially, a complete parametric representation of the plant is stored, plus all of the values for the F and H matrices across the flight envelope.

2.6 Stability preserving interpolation

Interpolation of transfer functions by interpolating poles, zeros, and gains can lead to instability. This is shown in [36]. Two controllers are designed for two different (extreme) positions. The controllers *each* stabilize the frozen plant globally. When the poles, zeros, and gains are interpolated, the interpolated controller does not stabilize the plant in between the two extreme points. This problem cannot only occur for the approach in Section 2.6, but it can be a problem for every interpolated controller.

However, it is possible to perform a stability preserving interpolation of stable coprime factors (see Section A.7) of transfer functions. In [36] a coprime factor interpolation method is proposed. The interpolation problem using state-space descriptions is also addressed. For each fixed value of the scheduling-parameter, the stability is preserved. Overall stability is guaranteed to bound the rate of variation of the scheduling variable. It is necessary to have the original nonlinear equations of motion available.

The standard Jacobian linearization of the nonlinear plant is written as a function of each

scheduling parameter θ.

$$\begin{aligned} \dot{x} &= A(\theta)x + G(\theta)w + B(\theta)u \\ z &= H(\theta)x + F(\theta)w + E(\theta)u \ . \\ y &= C(\theta)x + D(\theta)w \end{aligned} \tag{2.13}$$

The transfer function from u to y for this parameter-varying system is

$$P(s,\theta) = C(\theta)(sI - A(\theta))^{-1}B(\theta). \tag{2.14}$$

Suppose a finite set of controllers,

$$K_i = \left[\begin{array}{c|c} A_i & B_i \\ \hline C_i & 0 \end{array}\right], \qquad i = 1, ..., q \tag{2.15}$$

has been designed at points $\theta_1, ..., \theta_q$, such that K_i stabilizes the parameter-varying system $P(\theta_i)$. The main assumption in [36] is a stability covering condition: for each value that the scheduling parameter θ can take there must exists at least one linear controller K_i that globally stabilizes the frozen parameter varying system $P(\theta)$. This assumption is probably not restrictive for our purposes, because it is possible to design robust controllers for the whole operating region. It is also assumed that all linear controllers have the same input and output dimension. When interpolating state space descriptions of linear controllers, it is assumed that all controllers have the same number of states and when interpolating transfer functions, it is assumed that all linear controllers have the same McMillan degree [1].

Two stability preserving interpolation methods are presented: Interpolation of transfer functions and state-space interpolation. For two transfer-functions; suppose that the controllers $K_1(s)$ and $K_2(s)$ both stabilize the system $P(s,\theta)$ for $\theta \in [a,b]$. Then there exists a parameter-varying controller $K(s,\theta)$ that stabilizes $P(s,\theta)$ for $\theta \in [a,b]$ such that $K(s,a) = K_1(s)$ and $K(s,b) = K_2(s)$. Now, the interpolated controller is stabilizing for each frozen value of the scheduling variable. This is proven in [36] using coprime factorization. Something similar can be proven for two state-space descriptions.

A bound on the rate of variation of the scheduling variable can be calculated that guarantees the closed-loop nonlinear system to be locally exponentially stable:

$$|\dot{\theta}(t)| < \|\frac{\partial}{\partial\theta}W(\theta(t))\|^{-1}, \qquad t \geq 0. \tag{2.16}$$

where $W(\theta(t))$ is a certain matrix function which meets the requirement:

$$\hat{A}^T(\theta)W(\theta) + W(\theta)\hat{A}(\theta) < -I \tag{2.17}$$

$\hat{A}(\theta)$ is a matrix depending on several plant and controller matrices:

$$\hat{A}_i(\theta) = \begin{bmatrix} A(\theta) & B(\theta)C_i \\ B_iC(\theta) & A_i \end{bmatrix} \tag{2.18}$$

With the matrices $A(\theta)$, $B(\theta)$ and $C(\theta)$ from the plant (2.13) and C_i, B_i, and A_i from the LTI controller (2.15). Given state-space descriptions of the fixed controllers, the stability preserving interpolation can be found in [36].

[1] The McMillan degree is the sum of the degree of the denominator polynomials for the elements of a transfer matrix in McMillan form. (Any real rational transfer matrix can be reduced to this form through some pre- and post- operations with square polynomial matrices.) The McMillan degree is the same as the dimension of a minimal realization of the transfer matrix. See also [49, Section 3.11]

2.7 Interpolation using free controller parameters

In [23], a general framework for handling controllers is suggested which can be applied to controller scheduling implementations. The used technique is *Youla parameterization* (see Section A.7 and [49, Section 12.6]). With the approach it is possible to switch between observer-based feedback controllers, designed at different operating points, in a stable way. A method is presented to implement a controller which can be changed without jumps. The closed loop system is guaranteed to be stable. This is due to the fact that the controller is implemented by using parameterization. Based on coprime factorization of the system and the controller, a parameterization of all controllers that stabilize the system can be given. The free parameter from this parameterization is used for controller scheduling. This technique is extended in [24], where the scheduling parameter is estimated (the scheduling parameter is measured for ASML application).

In [23], a linear MIMO system with fixed scheduling parameter $\theta(t)$ is written as a coprime factorization (see Section A.7):

$$G_{yu}(s) = NM^{-1} = \tilde{M}^{-1}\tilde{N}, \quad N, M, \tilde{N}, \tilde{M} \in R\mathcal{H}_\infty \quad \text{(real-rational functions in } \mathcal{H}_\infty)^2 \tag{2.19}$$

Let a controller for this fixed scheduling parameter $\theta(t)$ be given by:

$$K(s) = UV^{-1} = \tilde{V}^{-1}\tilde{U}, \quad U, V, \tilde{U}, \tilde{V} \in R\mathcal{H}_\infty \tag{2.20}$$

It is now possible to give a parameterization of all controllers that stabilize the system in terms of a stable parameter $Q(s)$ (see Section A.7). In fact, this is a reformulation of (2.20) in such way that not only one single controller that stabilizes the system is presented but that the whole set of controllers that all stabilize the system (2.19) is presented:

$$K(Q) = U(Q)V(Q)^{-1} \tag{2.21}$$

where

$$U(Q) := U_0 + MQ, \qquad V(Q) := V_0 + NQ, \; Q \in \mathcal{RH}_\infty \tag{2.22}$$

This can also be written as:

$$K(Q) = K + \tilde{V_0}^{-1}Q(I + V_0^{-1}NQ)^{-1}V_0^{-1}, \quad Q \in R\mathcal{H}_\infty. \tag{2.23}$$

The controller in equation (2.20) is thus extended by the second term in (2.23) by using the parameter Q. Now, suppose that several controllers K_i (as in equation (2.20)) are designed at different operating points $i = 1, ..., p$:

$$K_i(s) = U_iV_i^{-1} = \tilde{V_i}^{-1}\tilde{U_i}, \quad U_i, V_i, \tilde{U_i}, \tilde{V_i} \in R\mathcal{H}_\infty. \tag{2.24}$$

These controllers all fall in the range of equation (2.23), so they can be implemented as

$$K_i = K_0(Q) = K_0 + \tilde{V_0}^{-1}Q_i(I + V_0^{-1}NQ_i)^{-1}V_0^{-1}, \quad Q_i \in R\mathcal{H}_\infty, \quad i = 1, ..., p \tag{2.25}$$

So, it is possible to implement a controller as a stable Q parameter based on another stabilizing controller. The linear combination of the Q_i parameters is given by

$$Q = \sum_{i=1}^{p} \alpha_i Q_i \tag{2.26}$$

[2] If $F(s)$ is real-rational, then $F \in R\mathcal{H}_\infty$ if and only if F is proper ($|F(\infty)|$ is finite) and stable.

and the resulting controller K (independent of K_0) can be given by a linear combination of U_i and V_i:

$$K = \left(\sum_{i=1}^{p} \alpha_i \tilde{V}_i\right)^{-1} \sum_{i=1}^{p} \alpha_i \tilde{U}_i \quad \text{with } P_{i=1}^{p} \alpha_i = 1 \tag{2.27}$$

where α is the scaling parameter. Now it is possible to change from one controller to another by scaling the Q parameter from zero to full value in a continuous way. The closed loop system is guaranteed to be stable for all values of Q_i.

It is also possible to give an $\mathcal{H}_\infty$ solution by solving two Riccati equations (without iterative procedure). In this case, also a free parameter is obtained which can be used for controller scheduling. This approach is followed in for instance [18]. Here, the controller is scheduled by the free parameter as a function of rotational speed of a magnetic bearing. In [47] the free parameter is used in combination with fuzzy rules to schedule the controller.

A disadvantage of these methods can be that the on-line calculation time is probably bigger than for other methods. This is due to the fact that the scheduling parameter does not directly determine the controller or controller gain, as in other methods, but only indirectly via the free parameter. The free parameter then has to be fed into the controller. Another problem with this aspect is that there is no logical relation between the free parameter and the scheduling parameter. So, after the controller is designed, the relation between the two parameters has to be discovered.

Also in [21] a coprime factorization approach is followed. The situation for two controllers is described. In [20] a similar approach is followed for more than two interpolated controllers. In [21], the two controllers K_i (as in equation (2.24)), where $i = 1, 2$, are interpolated as

$$\begin{cases} U = (1-\alpha)U_1 + \alpha U_2 \\ V = (1-\alpha)V_1 + \alpha V_2 \end{cases} , 0 \leq \alpha \leq 1. \tag{2.28}$$

In addition, each of the two controllers K_i, $i = 1, 2$ can be extended with the parameter Q as in (2.21) and (2.22). The parameter Q is then used to stabilize the closed loop system. It is assumed that the variation of the operating condition does not occur so often and that the plant can be treated as a time-invariant system in a certain period. This is not elucidated.

2.8 Robust controller design and performance for polytopic models

Although this method uses state-feedback (which is not useful for us, see Section 1.5), it is treated here, because it guarantees stability *and* performance can be analyzed, unlike many of the other methods. A nonlinear dynamic system can be represented by a 'global' model which is the result of taking convex combinations of locally valid models. In literature these models have different names, such as 'local model networks' in [12] and [14]. In [2], they are called 'polytopic models'. These models can be used for controller synthesis based on LMI's. In [2], a stabilizing scheduled controller is designed, which is robust against parametric uncertainty.

In addition, performance of the closed-loop system in the state-space can be analyzed without doing simulations.

Various nonlinear systems can be described by the polytopic model

$$\Pi : \dot{x} = \sum_{j \in I_N} w_j(x, u, t)(A_j x + B_j u + c_j), \quad x(0) = x_0. \tag{2.29}$$

N_m denotes the number of separate locally valid models and $I_N := \{1, ..., N\}$. The models are parameterized by A_j, B_j, c_j. The state x is the same for all models. The so-called membership functions w_j's schedule the separate models in the operating space. A polytopic model can be derived when the nonlinear equations of motion of the system are available (see [45]) or it can be based on input-output data of the system (see [41]). For the latter case, this can be done using the Kalman filter method or the least squares method.

The state-space is partitioned in clusters. Each cluster is a region where one separate model is valid or where a certain combination of separate models is valid. For instance, there could be a region (cluster) where only the model corresponding to $j = 2$ is valid and there could be a cluster where both the model corresponding with $j = 1$ and the model corresponding to $j = 3$ is valid.

The state feedback law is of the form

$$u = K_J x + k_J \tag{2.30}$$

where J denotes in which cluster of the state-space the controller is used. In this way, the feedback laws for the several clusters form a gain-scheduled controller, so K_J is smoothly varied. The objective is now to parameterize this controller such that robust stabilization and performance of the closed loop polytopic model is achieved. The robust stabilization synthesis problem can be solved by an iterative algorithm involving LMI's. The performance of the closed loop in the state-space can then be analyzed.

2.9 State feedback methods

In [16], a set of state-feedback gains, obtained via LQR-design, is fitted to continuous functions of the state of the system using Taylor series expansion. The equations of motion of the system have to be available for this method. The fit is done for all elements of the gain matrix. This provides smooth transition between the operating regions. Using this method, no limitation on the speed of variation in the system is imposed. Simulation results in [16] illustrate the effectiveness (in this case) of the proposed approach. There is no real evidence for stability, only smooth transition is provided.

In [1], an optimal controller design for polytopic models (see Section 2.8) is proposed. It is shown that under controllability assumptions there exists a solution to a sufficient condition for optimality of the closed-loop system. A state-feedback controller is computed as a solution of a convex optimization program, *i.e.* by solving a set of LMIs.

In [32], conditions are presented which guarantee quadratic Lyapunov and robust stability when switching among a collection of state feedback controllers for an uncertain plant. The switching is based on the measured value of the system state. At each time instant, the output of only one controller is fed into the plant. Robust stability is investigated with a quadratic storage function.

In [19], a system is considered where only a few states affect the system dynamics in a nonlinear way. The system is controlled with full-state feedback. The approach guarantees stability. The plant

$$\begin{bmatrix} \dot{x}_N \\ \dot{x}_L \end{bmatrix} = \begin{bmatrix} f_N(x_N) \\ f_L(x_N) \end{bmatrix} + \begin{bmatrix} A_N(x_N) \\ A_L(x_N) \end{bmatrix} x_L + \begin{bmatrix} g_w^N(x_N) \\ g_w^L(x_N) \end{bmatrix} w + \begin{bmatrix} g_u^N(x_N) \\ g_u^L(x_N) \end{bmatrix} u. \tag{2.31}$$

has nonlinear states x_N and linear states x_L. This can also be written as

$$\dot{x} = f(x) + g_w(x)w + g_u(x)u. \tag{2.32}$$

The followed approach is compared with basic controller scheduling for a nonlinear system. In that case, controllers are designed at certain points based on the linearized systems about those points. These points are called *trim points*. For a trim point x_0, $f(x_0)+g_u(x_0)u_0 = 0$. In the proposed method, the spacing of trim points at which designs must be done is determined systematically. A control law which guarantees stability is constructed.

The approach in [19] selects *robust control Lyapunov functions* for the system based on linearizations about various trim points of the system. The main idea is to design a quadratic robust control Lyapunov function to the target equilibrium point x_0. It is possible to compute the region of stability based on this robust control Lyapunov function. After that the region of stability is expanded. This is done by designing several robust control Lyapunov functions to different trim points of the system. A trajectory starting in the region of stability of a certain trim point converges to that point. When this point also lies in the region of stability of another trim point, it is possible to switch between the associated regions of stability. By piecing together all the local stability regions, the original region of stability is expanded.

In [48], a fuzzy approach is followed. Control rules with the same fuzzy rules as the fuzzy models are scheduled by fuzzy weights. Fuzzy rules define the current measurement or the current controller output. Fuzzy weights determine in what degree the previous measurements and outputs are taken along. According to the authors, model-free fuzzy approaches cannot deal with stability, robustness and performance. Therefore, the global equations of motion are used. The controller gain matrices are computed by solving two LMIs based on a quadratic Lyapunov function, see references in [48]. In this method the system has to be written in a discrete form. The approach is based on the plant:

$$\begin{aligned} x(k+1) &= \left[\sum_{i=1}^{N} \lambda_i(k) A_i\right] x(k) + \left[\sum_{i=1}^{N} \lambda_i(k) B_i\right] u(k) \\ y(k) &= Cx(k) = [1, 0, ..., 0]x(k) \end{aligned} \tag{2.33}$$

Other fuzzy gain-scheduling approaches can be found in for instance [38], [39], [40].

Chapter 3

LPV controller scheduling

3.1 Introduction

The systems that are considered in this chapter can be represented by:

$$P(\theta) : \begin{cases} \dot{x} = A(\theta(t))x + G(\theta(t))w + B(\theta(t))u \\ z = H(\theta(t))x + F(\theta(t))w + E(\theta(t))u \\ y = C(\theta(t))x + D(\theta(t))w \end{cases} \tag{3.1}$$

where x is the state, u is the control input, y is the measured output, w and z are the variables to impose performance specifications, and $\theta(t)$ is the time-varying parameter vector.

To apply LPV controller scheduling techniques, certain assumptions have to be made:

- The nonlinear system (nonlinear with respect to the state and possibly the time-varying parameter-vector $\theta(t)$) is written as a linear state-space representation as in (3.1), where the state-space matrices still depend on the time-varying parameter-vector (see for instance [8]).
- The parameter or parameters in the parametervector θ have to appear rationally in the state-space matrices. This is not a restriction for all nonlinearities. In much cases, a parameter that appears nonlinearly can be split into two or more new linear parameters.
- It is assumed that the measurements of the system can only depend on the state and all external inputs. So, the measurements are not allowed to depend on control inputs. This is no restriction in case of the wafer-stage, where θ involves positions and where the control inputs are forces and torques. [1]
- The parameter vector $\theta(t)$ is on-line measurable.

[1] In $\mathcal{H}_\infty$ control techniques this assumption is also made, see [49, Section 17.1]. However, the assumption can be removed, that is, there is no loss of generality in this assumption. For now, it is not clear if this assumption can be removed in the LPV case.

In essence, there are three distinct approaches to solve an LPV scheduling problem:

1. A *polytopic approach* ([5]). The quadratic Lyapunov approach leads to an infinite dimensional convex problem (see Section 3.2). One way to find a solution to this problem is a polytopic approach, which is treated in Section 3.3.

2. An *LFT approach.* An LFT is a Linear Fractional Transformation (Appendix A.3).

   This approach can be followed from the point of view of the small-gain approach (see Appendix A.6). This scaled small-gain approach is applicable to LPV plants with an LFT dependence on the scheduling parameter ([4], [26]). In [50], it is shown that in general, less conservative results can be obtained using an approach with less restrictive scalings (*e.g.*, in [25],[28],[30]). These approaches ensure $\mathcal{H}_\infty$-like performance for all possible trajectories of the LPV plant, but lead to an infinite dimensional convex problem (see Section 3.2). One way to solve this is writing the problem as an LFT. Solutions can be obtained by using an LMI (Linear Matrix Inequality) approach or by using *D-K* iteration. This will be treated in Section 3.4.

3. A *Parameter-dependent Lyapunov approach.* A parameter-dependent Lyapunov approach is followed when slowly-varying systems are considered ([6],[3],[31],[46]). This is described in Section 3.5.

3.2 Preliminaries

Consider the LTI closed-loop system:

$$\begin{aligned} \dot{x}_{cl} &= A_{cl}x_{cl} + B_{cl}w \\ z &= C_{cl}x_{cl} + D_{cl}w \end{aligned} \tag{3.2}$$

where w is the disturbance input and z is the performance output of the closed-loop system.

The following statements are equivalent:

(a) $\|M\|_\infty < \gamma$ and A stable with $M(s) = D_{cl} + C_{cl}(sI - A_{cl})^{-1}B_{cl}$.
(b) there exists a solution $X > 0$ to the LMI:

$$\begin{bmatrix} A^TX + XA & XB & C^T \\ B^TX & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0. \tag{3.3}$$

This equivalence is called the Bounded Real Lemma (BRL). When the statements are valid, the system (3.2) has $\mathcal{H}_\infty$ performance γ, see [5]. The BRL is only valid for LTI (Linear Time Invariant) systems. $\mathcal{H}_\infty$ performance γ for an LPV is called *quadratic* $\mathcal{H}_\infty$ performance. This definition will become clear in the remainder of this section.

The closed-loop LPV system:

$$\dot{x}_{cl} = A_{cl}(\theta)x_{cl} + B_{cl}(\theta)w \tag{3.4}$$

$$z = C_{cl}(\theta)x_{cl} + D_{cl}(\theta)w \tag{3.5}$$

has quadratic $\mathcal{H}_\infty$ performance γ if and only if there exists a single matrix $X > 0$ such that the following BRL is valid,

$$\begin{bmatrix} A_{cl}(\theta)^T X + X A_{cl}(\theta) & X B_{cl}(\theta) & C_{cl}(\theta)^T \\ B_{cl}(\theta)^T X & -\gamma I & D_{cl}(\theta)^T \\ C_{cl}(\theta) & D_{cl}(\theta) & -\gamma I \end{bmatrix} < 0, \tag{3.6}$$

for all admissible values of the parameter vector θ. Then the single quadratic Lyapunov function

$$V(x) = x^T X x$$

establishes global asymptotic stability and the L_2 gain of the closed-loop system is bounded by γ ($\|z\|_2 < \gamma \|w\|_2$,) along all possible parameter trajectories θ. (See [5])

So, the BRL is also valid for LPV systems, but the Inequality (3.6) has to be valid for all θ. The problem with (3.6) is that an infinite number of constraints must be satisfied (there is a continuum of parameter values). In the case of *polytopic* LPV systems the condition (3.6) can be reduced to a finite set of LMI's. This will be treated in Section 3.3. Also for a plant with an LFT-structure a finite number of constraints is obtained. This will be treated in Section 3.4. An extension to the BRL in (3.6) is given in Section 3.5. Also, a rough description of the method to solve the BRL is given.

3.3 Polytopic LPV approach

For affine LPV systems with parameter values belonging to a convex (see Appendix A.4) polytope (parameter space), a polytopic approach can be applied. The state-space matrices of the plant considered in [5] are assumed to depend affinely (see Appendix A.5) on a vector θ of time-varying parameters. Also, the parameter vector has to range over a fixed polytope of vertices $\omega_1, \omega_2, ..., \omega_r$:

$$\theta \in \Theta := \mathrm{Co}\{\omega_1, \omega_2, ..., \omega_r\} \tag{3.7}$$

These assumptions mean that the state-space matrices range in a polytope of matrices. This polytope is defined as the convex hull of a finite number of matrices N_i (N_i can be read as either A_i, B_i, C_i, D_i) with the same dimensions:

$$\mathrm{Co}\{N_i,\, i = 1, ..., r\} := \left\{ \sum_{i=1}^{r} \alpha_i N_i : \alpha_i \geq 0, \sum_{i=1}^{r} \alpha_i = 1 \right\}. \tag{3.8}$$

For example, the matrices H_i in the set $\mathcal{H}$:

$$\mathcal{H} = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 0.75 \end{pmatrix} \right\} \tag{3.9}$$

all range in the convex hull

$$\mathrm{Co}\left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$$

where $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ are the vertices. The matrices H_i in (3.9) can be composed as follows:

$$H_i \in \mathcal{H} = \alpha_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad \sum_{i=1}^{3} \alpha_i = 1. \tag{3.10}$$

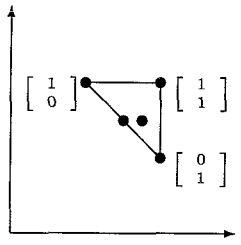

Figure 3.1: Example of a convex set

See also Figure 3.1. For instance,

$$\begin{pmatrix} 0.5 \\ 0.75 \end{pmatrix} = 0.5 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + 0.25 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 0.25 \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Using this convexity, condition (3.6) will hold for all $(A(\theta), B(\theta), C(\theta), D(\theta))$ if and only if it holds at the vertices of the state-space matrices N_i and so condition (3.6) reduces to an infinite number of constraints. A good balance has to be found between an accurate description of the polytope and the computational burden of the problem. Using many vertices results in an accurate description but in large computational burden.

Further assumptions in [5] are that the measurements of θ are available in real time and that the matrices B, E, C, D in system $P(\theta)$ (3.1) are parameter-independent:

$$\begin{aligned} \dot{x} &= A(\theta(t))x + G(\theta(t))w + Bu \\ z &= H(\theta(t))x + F(\theta(t))w + Eu \\ y &= Cx + Dw. \end{aligned} \tag{3.11}$$

The controller which is sought for has the form:

$$\Omega(\theta) : \begin{cases} x_K &= A_K(\theta)x_K + B_K(\theta)y \\ u &= C_K(\theta)x_K + D_K(\theta)y. \end{cases} \tag{3.12}$$

The closed-loop system is described by the state-space equations:

$$\begin{aligned} x_{cl} &= A_{cl}(\theta)x_{cl} + B_{cl}(\theta)w \\ q &= C_{cl}(\theta)x_{cl} + D_{cl}(\theta)w. \end{aligned} \tag{3.13}$$

Without loss of generality it can be assumed that the controller can be described as a polytope of matrices as well. If a controller $\Omega(\theta)$ has quadratic performance γ, its values $\Omega_i := \Omega(\omega_i)$ at the vertices ω_i of the parameter box must satisfy the BRL (3.6). A polytopic controller

of vertices Ω_i yields the same performance. When the parameter θ in BRL (3.6) for the closed-loop system is chosen to be ω_i, a set of matrix inequalities is obtained;

$$\begin{bmatrix} A_{cl}(\omega_i)^T X + X A_{cl}(\omega_i) & X B_{cl}(\omega_i) & C_{cl}(\omega_i)^T \\ B_{cl}(\omega_i)^T X & -\gamma I & D_{cl}(\omega_i)^T \\ C_{cl}(\omega_i) & D_{cl}(\omega_i) & -\gamma I \end{bmatrix} < 0, \qquad i = 1, 2, ..., r, \tag{3.14}$$

The core of the LPV synthesis problem is to compute the single Lyapunov matrix $X > 0$ and LTI controllers Ω_i that satisfy the system of matrix inequalities (3.14). This can be done writing the inequality (3.14) as a convex LMI problem. The derivation of these LMIs can be found in [10] and will not be repeated here. Solving the problem, the Lyapunov function X can be found. The same Lyapunov function should be used for all vertices. The LMI's can be solved with the use of the Matlab LMI Control toolbox [11]. For affine systems, a special function (`hinfgs.m`) is available to compute LPV controllers.

Once the Lyapunov matrix X has been determined, the vertex controllers Ω_i can be deduced. At the vertices of the parameter polytope (see also (3.7)), which is given by

$$\Theta = \left\{ \sum_{i=1}^{r} \alpha_i \omega_i \ : \ \alpha_i \geq 0, \ \sum_{i=1}^{r} \alpha_i = 1 \right\}, \tag{3.15}$$

linear time-invariant $\mathcal{H}_\infty$ controllers are computed by solving the corresponding BRL from (3.14). The resulting LPV controller is:

$$\Omega(\theta) = \begin{pmatrix} A_K(\theta) & B_K(\theta) \\ C_K(\theta) & D_K(\theta) \end{pmatrix} := \sum_{i=1}^{r} \alpha_i \Omega_i = \sum_{i=1}^{r} \alpha_i \begin{pmatrix} A_{Ki} & B_{Ki} \\ C_{Ki} & D_{Ki} \end{pmatrix} \tag{3.16}$$

with r the number of vertices of the polytope in which the scheduling-parameter varies. The controller enforces stability and $\mathcal{H}_\infty$ performance over the entire parameter polytope Θ and for arbitrary parameter variations.

3.4 LFT dependence

LPV systems in which the state-space matrices are rational functions of the parameters can be transformed into LFT form. This means that the LPV system is written as a linear time-invariant system enclosed by a feedback loop with the time varying parameter, see Figure 3.2. The parameter vector $\triangle$ contains a block with the scheduling parameter: $\triangle = \text{diag}(\theta_1, ...\theta_i)$. The time-varying parameter can be a function of the state. The requirement that the plant has an LFT structure does not seem to be particularly restrictive. Most practical problems can be written into LFT form. This is not always easy and the actual derivation can be quite *ad hoc*. If the system $P(\theta)$ as in (3.1) has an LFT dependence on the scheduling parameter, it can be represented as

$$\begin{bmatrix} \dot{x} \\ z_1 \\ z_2 \\ y \end{bmatrix} = \begin{bmatrix} A & G_1 & G_2 & B \\ H_1 & F_1 & F_{12} & E_1 \\ H_2 & F_{21} & F_2 & E_2 \\ C & D_1 & D_2 & 0 \end{bmatrix} \begin{bmatrix} x \\ w_1 \\ w_2 \\ u \end{bmatrix}, \ w_2 = \triangle(\theta(t)) z_2, \ \triangle(t) \in \triangle_{convex} \tag{3.17}$$

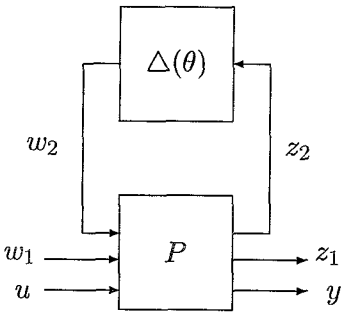

Figure 3.2: LFT-setup for an LPV-system

with the convex set

$$\triangle_{convex}(t) := \mathrm{Co}(\triangle_1, ..., \triangle_N). \tag{3.18}$$

The equation (3.17) is an extension of equation (3.1). The external inputs w and external outputs z are split up into free external inputs and outputs, w_1 and z_1, and inputs and outputs which are connected with the parameter block, w_2 and z_2. An example of a plant that can be written as an LFT is given in Section A.3.1.

For the LPV approach based on LFT dependence, also the controller K is assumed to take the LFT structure:

$$\begin{bmatrix} \dot{x}_c \\ u \\ z_c \end{bmatrix} = \begin{bmatrix} A_c & B_{c1} & B_{c2} \\ C_{c1} & D_{c11} & D_{c12} \\ C_{c2} & D_{c21} & D_{c22} \end{bmatrix} \begin{bmatrix} x_c \\ y \\ w_c \end{bmatrix}, \; w_c = \triangle_{controller}(\triangle(\theta(t)))z_c \tag{3.19}$$

with also $\triangle_c(\triangle(\theta(t))) = \triangle_{controller}(\triangle(\theta(t)))$ (defined on $\triangle_{convex}$) as a design variable. From now on, $\triangle(\theta(t))$ will be represented as $\triangle$ for notational convenience.

The approaches in [4], [13] are based on $\triangle_{controller}(\triangle) = \triangle$ such that the controller is scheduled with an identical copy of the scheduling parameters. In other approaches ([26], [25], [31]) the controller is scheduled with a function of copies of the scheduling parameters. This is explained later. The plant description following from (3.17) and (3.19) is depicted in Figure 3.3. Although this figure seems the same as the standard plant used in μ-synthesis it needs to be remarked that the most upper block is *not* an uncertainty block. The momentary value of the parameter θ need not be known, but it is on-line measurable and so not an uncertainty. This is the essential difference with robust control techniques. However, for convenience θ will be referred to as the symbol $\triangle$. The to-be-controlled plant system can be rewritten as the LTI system:

$$\begin{bmatrix} \dot{x} \\ \hline z_1 \\ \hline z_2 \\ z_c \\ \hline y \\ w_c \end{bmatrix} = \left[\begin{array}{c|c|cc|cc} A & G_1 & G_2 & 0 & B & 0 \\ \hline H_1 & F_1 & F_{12} & 0 & E_1 & 0 \\ \hline H_2 & F_{21} & F_2 & 0 & E_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ \hline C & D_1 & D_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \end{array}\right] \begin{bmatrix} x \\ \hline w_1 \\ \hline w_2 \\ w_c \\ \hline u \\ z_c \end{bmatrix} \tag{3.20}$$

which is scheduled as

$$\begin{bmatrix} w_2 \\ w_c \end{bmatrix} = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_c(\Delta) \end{bmatrix} \begin{bmatrix} z_2 \\ z_c \end{bmatrix}. \tag{3.21}$$

and interconnected with the LTI controller (3.19). The closed-loop system can then be represented by

$$\begin{bmatrix} \dot{x} \\ \hline z_1 \\ z_c \\ z_2 \end{bmatrix} = \left[\begin{array}{c|ccc} \mathcal{A} & \mathcal{B}_1 & \mathcal{B}_c & \mathcal{B}_2 \\ \hline \mathcal{C}_1 & \mathcal{D}_{11} & \mathcal{D}_{1c} & \mathcal{D}_{12} \\ \mathcal{C}_c & \mathcal{D}_{c1} & \mathcal{D}_{cc} & \mathcal{D}_{c2} \\ \mathcal{C}_2 & \mathcal{D}_{21} & \mathcal{D}_{2c} & \mathcal{D}_{22} \end{array}\right] \begin{bmatrix} x \\ \hline w_1 \\ w_c \\ w_2 \end{bmatrix}, \tag{3.22}$$

scheduled by the parameter (3.21), see Figure 3.4. In fact, Figures 3.3 and 3.4 are more

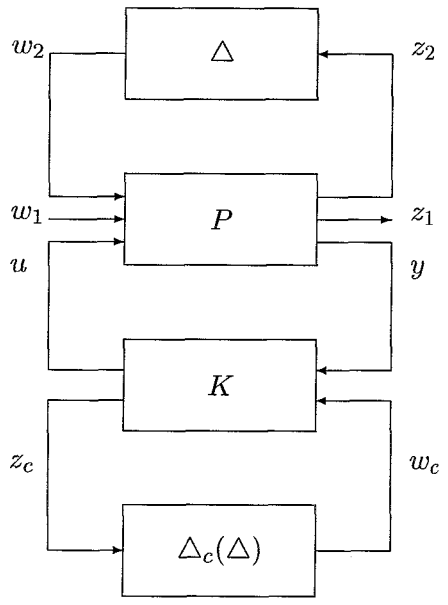

Figure 3.3: LFT setup for LPV controller scheduling.

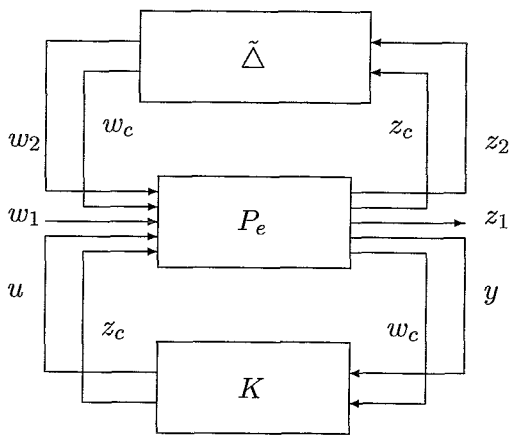

Figure 3.4: LFT-setup rewritten to standard μ-synthesis plant setup.

restrictive representations of the idea depicted in Figure 1.1. The goal is to construct an LPV controller such that for all admissible parameter curves the controlled system is exponentially stable and the quadratic performance criterion is met (see Section 3.2).

3.4.1 LMI approach

The robust performance objectives formulated in terms of a Lyapunov function can be translated into a test with multipliers or scalings (see [28]). By finding a controller, a Lyapunov function, and a multiplier that satisfy a set of LMIs, robustness and quadratic performance of the system can be guaranteed.

The analysis test is based on finding a constant quadratic Lyapunov function in order to guarantee the following properties:

- Well-posedness (see Appendix A.3) of the LFT used to describe the uncertain system.
- Uniform exponential stability (see Appendix A.9).
- Robust performance, specified as quadratic performance (such as bounding the L_2-gain, see Appendix A.10)

These properties can be translated into a sufficient LMI condition with multipliers or scalings (see Appendix A.10). This is a sufficient LMI condition and not also a necessary one because conservatism is introduced. This has to do with the scalings, see Appendix A.10. In the literature, analysis results with scalings are usually provided for block diagonal real repeated uncertainties

$$\triangle = \mathrm{diag}(\theta_1 I_{N_1}, ..., \theta_m I_{N_m}). \tag{3.23}$$

The scalings are usually restricted to have the same block diagonal structure as $\triangle$. Sometimes more restrictions are imposed, see [25]. However, unnecessary restrictions of the scalings lead to a degrading of the robust performance; the results become more conservative. On the other hand, less conservative methods require more number of variables to be solved by the LMIs what slows down the calculations. Both techniques will be described. In the next Section a method will be described which is based on small gain LTI techniques. The resulting LMIs for this method can also be obtained when following the approach described at the beginning of this section, and then applying non-full block scalings. In the succeeding section after this is extended to full block scalings, which will be treated next.

Non-full block scalings

In [4], the discrete time version of Packard [26], who first came with the idea of LPV gain scheduling, is extended to continuous time. The basis for this approach is the $\mathcal{H}_\infty$ LTI control problem. This is extended to allow for controller dependence on the parameter $\triangle$. Given an LPV plant $P(\triangle)$ as in Figure 3.3, mapping exogenous inputs w_1 and control inputs u to controlled outputs z_1 and measured outputs y, a controller $K(\theta)$ has to be found such that

- the closed-loop system is internally stable for all parameter trajectories.
- the closed-loop mapping from exogenous inputs w_1 to controlled outputs z_1 is bounded by some performance level γ.

The system in Figure 3.4 can be written as a lower LFT: $\mathcal{F}_l(P_e, K)$. From small gain theory (see Section A.6 and references in [4]), a sufficient condition for the existence of gain-scheduled controllers is as follows. If there exists scalings L and a controller K such that the nominal closed-loop system from Figure 3.4, $\mathcal{F}_l(P_e, K)$, is internally stable and satisfies

$$\left\| \begin{pmatrix} L^{1/2} & 0 \\ 0 & I \end{pmatrix} \mathcal{F}_l(P_e, K) \begin{pmatrix} L^{-1/2} & 0 \\ 0 & I \end{pmatrix} \right\|_\infty < \gamma, \tag{3.24}$$

then $\mathcal{F}_l(K, \triangle)$ is a $\mathcal{H}_\infty$ gain-scheduled controller: the closed-loop system is internally stable and the $\mathcal{H}_\infty$ norm of $\mathcal{F}_l(K, \triangle)$ is strictly less than γ. $L \in \mathcal{L}$ is a positive definite similarity

scaling:

$$\mathcal{L} = \begin{pmatrix} L_1 & L_2 \\ L_2^T & L_3 \end{pmatrix} > 0 : L_1, L_3 > 0,\ L_i\Delta = \Delta L_i,\ i = 1,2,3. \tag{3.25}$$

According to [25], this is the same as using the scalings as in equation (A.37) in Section A.10 with the restrictions $Q < 0$ *and* $S = 0$. The last one is the unnecessary restriction that makes this method more conservative than the method with full block scalings.

Applying the Bounded Real Lemma on condition (3.24), this scaled $\mathcal{H}_\infty$ problem can be written as LMIs. These LMIs guarantee the existence of a gain-scheduled $\mathcal{H}_\infty$ controller and are used to compute this controller. The LMIs are convex because the time-varying parameter has access to the controller. The computation and implementation of the controller is described in [4]. This is done in parallel with the algorithm described in [10] for pure $\mathcal{H}_\infty$ control.

Full block scalings

When using the approach of full block scalings (see [28]) the more general scheduling function $\Delta_c = \Delta_c(\Delta(t))$ instead of $\Delta_c = \Delta(t)$ is necessary (the scheduling function will turn out to be a quadratic function, see [29]). The robust quadratic performance problem as in Section 3.2 can equivalently be described (using Lyapunov) by a matrix inequality, see, for instance, [29]. In order to make the computation of the matrix inequality less hard, this inequality can equivalently be described by a more explicit LMI condition that makes use of multipliers or scalings, see also Section A.10.

The multipliers adjusted to the parameter block in (3.21) are denoted by W_e. This structure (3.21) has been *extended* (compared with standard μ-synthesis) with respect to the parameter-dependence of the controller, that is, the considered parameter block is $\tilde{\Delta} = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_c(\Delta) \end{bmatrix}$ instead of Δ. Therefore, these multipliers W_e are called *extended* multipliers:

$$W_e = \begin{bmatrix} Q_e & S_e \\ S_e^T & R_e \end{bmatrix} = \begin{bmatrix} Q & Q_{12} & S & S_{12} \\ Q_{21} & Q_{22} & S_{21} & S_{22} \\ S^T & S_{21}^T & R & R_{12} \\ S_{12}^T & S_{22}^T & R_{21} & R_{22} \end{bmatrix} \quad \text{with } Q_e < 0,\ \ R_e > 0, \tag{3.26}$$

see also Section A.10. Now, first the new LMI condition has to be solved, which results in numerical values for $Q, S, R, \tilde{Q}, \tilde{S}, \tilde{R}$ (the multiplier blocks in W_e and $\tilde{W}_e$ without indices). After that, the scalings are extended in such way that $\tilde{W}_e = W_e^{-1}$. Then the scheduling function can be constructed. The inequality (A.45) gives the solution using Schur-complement argument (this is an algebraic operation). Now, the solvability condition for the existence of the scheduling function can be written as an explicit formula. After the scheduling function is obtained, the LTI part of the controller has to be computed. This is done solving the nominal quadratic performance problem (see [27] and Appendix A.10):

$$\int_0^\infty \begin{bmatrix} w_2 \\ w_c \\ z_2 \\ z_c \end{bmatrix} \begin{bmatrix} -\gamma I & 0 & 0 & 0 \\ 0 & Q & 0 & S \\ 0 & 0 & \frac{1}{\gamma} I & 0 \\ 0 & S^T & 0 & R \end{bmatrix} \begin{bmatrix} w_2 \\ w_c \\ z_2 \\ z_c \end{bmatrix} dt \leq 0 \tag{3.27}$$

3.4.2 D-K approach for LPV systems

In [13, Chapter 8] and [35], also an LFT description is used to approach the controller scheduling problem. The idea for this approach can also be described by the Figures 3.3 and 3.4. This setup is still useful when some of the uncertainty blocks are not available to the controller. In that case, the calculation of the controller is a nonconvex problem. In that case, D-K like iterations can be applied.

The standard D-K algorithm uses dynamic D-scalings together with $\mathcal{H}_\infty$-synthesis for finding stabilizing controllers for systems with dynamic linear time-invariant (LTI) uncertainties [2]. A D-K iteration consists of two steps. In the first step the scaling D is computed for a fixed K, in the second step the controller K is computed for a fixed D in order to improve performance. $\mathcal{H}_\infty$ optimization is used to determine the controller.

The standard D-K iteration has to be modified to an iteration for gain-scheduled systems with uncertainties. In [35], the modified D-K iteration is based on the method in [4]. In [13, Section 8.5] the basis for the modified D-K iteration is a comparable method, also described in [13].

Standard D-K iterations can be extended to time-varying parameters that have a bound on the rate of variation. In [13, Chapter 9], frequency dependent scalings are used. This is possible if the variations are sufficiently slow (see also the next section). In order to reduce conservatism the use of frequency dependent scalings is attractive. One way to do this is using a particular multiplier structure: the original structure $\triangle$ can then be replaced by a linear expression in $\triangle$ and its time derivative $\dot{\triangle}$. Using upperbound μ-analysis the problem can be solved. The associated solvability conditions are non-convex (even though the plant is in LFT form), so there is no guarantee finding an adequate controller (even when one exists). The problem is non-convex due to the joint presence of true uncertainties and the time-varying parameters. Theory and an example can be found in [13, Section 9.2].

3.5 Parameter-dependent Lyapunov function

Uncertain systems with linear time-varying uncertainties that have bounded rate of variation are also called slowly time-varying systems. This class of systems lies between LTI-uncertainties and uncertainties with arbitrary rate of variation. Examples of slowly varying parameters are the velocity and altitude of an aircraft. Based on the engine and aerodynamic performance the bounds on the rate of change can be determined. This is not the case, for instance, for the angle of attack. This is part of the dynamics of the aircraft and can vary almost arbitrarily fast. With respect to the wafer stage, the scheduling parameter has position-dependency. The position cannot change arbitrarily fast. The wafer stage can thus be regarded as a slowly time-varying system.

[2] In the case of linear time-varying (LTV) uncertainties constant scalings should be used (for slow LTV uncertainties constant scalings are not necessary, only for fast varying uncertainties), but when the uncertainties are LTI, conservatism can be reduced by using dynamic scalings.

Controllers for slowly-time varying systems can be synthesized using a parameter-dependent Lyapunov function $X(\theta)$. When such function is used, the BRL (3.6) takes the form:

$$\begin{bmatrix} A_{cl}(\theta)^T X(\theta) + X(\theta) A_{cl}(\theta) & X(\theta) B_{cl}(\theta) & C_{cl}(\theta)^T \\ B_{cl}(\theta)^T X(\theta) & -\gamma I & D_{cl}(\theta)^T \\ C_{cl}(\theta) & D_{cl}(\theta) & -\gamma I \end{bmatrix}, \qquad X(\theta) > 0. \tag{3.28}$$

In case of a parameter-dependent Lyapunov approach ([6],[3],[31],[46]) the controller matrices depend both on the scheduling parameter and its time derivative. Also, the scheduling parameter is assumed to be bounded. This will become clear in the remainder of this section. The scheduling parameter and the derivative both have to be known on-line. The dependence of the controller matrices on $\dot{\theta}$ can be removed by further restriction to a specific sub-class of Lyapunov functions. This is often necessary since the derivative of the scheduling parameter can frequently neither be measured nor estimated.

The parameter is thus assumed to be bounded; both θ and $\dot{\theta}$ are bounded. When controllers are obtained that are valid for arbitrary variations, as in previous methods, results are also directly applicable to systems where the scheduling parameter depends on the state of the system, as for the wafer stage (quasi-LPV systems). When looking at parameter-dependent Lyapunov functions, this is more difficult. The scheduling parameter can depend on the state by restricting the input and the initial conditions of the state. When slow variation of the state of the closed-loop system is ensured, also slow variation of the scheduling parameter is ensured. However, the problem that arises now is that the restrictions on the input and initial conditions are dependent on the choice of the controller. An aggressive controller might need stronger restrictions on the input and initial conditions than a weak controller (to ensure that the magnitude of the scheduling parameter is less than a particular value). This coupling has an implication for the stability characteristics at the closed-loop system.

Conditions for the solvability are derived by a set of LMI's, based on the BRL (3.28). In finding a solution two problems occur: there is an infinite number of constraints that must be satisfied to meet the solvability conditions (this is the same problem as mentioned in Section 3.2) and there is the infinite dimensional nature of the Lyapunov matrix $X(\theta)$. In [31], with regard to the infinite-dimensional nature of the Lyapunov matrix, the search is restricted to a finite dimensional subspace, instead of searching over the set of all continuous functions. The synthesis inequalities (the infinite number of constraints with the infinite dimensional nature of the Lyapunov matrix) then turn out to be (still infinitely many) LMI's. This is done by replacing the unknown functions in the synthesis inequalities with functions spanned by certain basis functions. It is also possible ([3]) to introduce, in an affine fashion, copies of the plant's nonlinear functions (differential functions of the scheduling parameter) into the functions related to the Lyapunov matrix.

The problem of the infinite number of constraints in [3] and [31] is solved by gridding. A finite subset of the infinite set is chosen to solve the LMI's. This *ad hoc* gridding approach can be used to obtain an approximate solution when there are a small number of parameters, as for the wafer stage (two or three scheduling variables). This approach is only valid when the gridding points are chosen sufficiently dense. When the number of parameters is too high, the LMI's that have to be solved become too large and numerical problems arise. Alternatively, the constraints reduce to a finite number for the specific class of affine LPV plants with

parameters belonging to a convex polytope ([6]). The details of the parameter-dependent Lyapunov method are not included in this report because it is too complicated to describe the full derivation of the LMIs. The precise description of the method can be found in, for instance, [6],[3],[31],[46].

Chapter 4

Comparison

4.1 Intention

The intention of this chapter is to qualify the main differences between the treated approaches. The advantages and disadvantages of the methods from Chapters 2 and 3 are discussed so that it is possible to give a good comparison. The most important properties are checked for each method. In the first column of Table 4.1 the various criteria are included. Most of the criteria speak for themselves.

In the first row the different methods are included. For all criteria, a plus sign means that the method is positively judged for this aspect. A minus sign is negative and a circle is neutral. An attempt is made to give a quantitative measure of the usefulness of the methods. If a method is judged positively (with a plus sign) for a certain criterion, the method gets the mark 2. If a method is judged neutrally, the mark is one. If the method is negatively judged for the method, nothing happens. The sum for each method has been divided by the maximum obtainable value times ten. This results in a value between '0' and '10' for each method. It needs to be stressed that it is very difficult to extract the judgements from the text of the articles. This is often a matter of interpretation and always subjective. A criteria as 'ease of modeling' is in a certain sense related to the skills and techniques available at Philips CFT.

4.2 Discussion

'Controller output-scheduling' (Section 2.2) is quite an *ad hoc* approach. It is simple, but stability and performance cannot be guaranteed. There are many controller synthesis techniques possible. For small conservatism the judgement is positive, since at the different operating points for which the controllers are designed, no conservatism is introduced. It is always possible to apply a denser grid. Small conservatism is typical for conventional methods but this comes with the cost of non-guaranteed global stability and performance. A disadvantage of almost all conventional controller scheduling methods is that all the different controllers have

	conventional							LPV			
assessment criterion section	2.2	2.3	2.4	2.5	2.6	2.7	2.8	3.3	3.4.1	3.4.2	3.5
ease of modeling	+	+	+	+	o	−	−	o	−	−	−
small conservatism	+	+	+	+	−	o	+	−	−	o	o
ease of controller design	+	+	−	o	o	−	o	o	o	−	−
ease of controller implementation	o	o	−	o	o	−	o	o	+	+	+
guaranteed global stability	−	−	−	−	+	+	+	+	+	+	+
guaranteed global performance	−	−	−	−	−	−	−	+	+	+	+
expected usefulness	5.8	5.8	3.3	5.0	4.2	2.5	5.0	5.8	5.8	5.8	5.8

+ = positive judgement
o = neutral
− = negative judgement

section	method
2.2	Controller output-scheduling
2.3	Smooth interpolating of controller outputs
2.4	Interpolation of poles, zeros, and gains
2.5	Interpolation of state-space matrices
2.6	Stability preserving interpolation
2.7	Interpolation using free controller parameters
2.8	Robust controller design and performance for polytopic models
3.3	Polytopic LPV approach
3.4.1	LFT dependence - LMI approach
3.4.2	LFT dependence - D-K approach for LPV systems
3.5	Parameter-dependent Lyapunov approach

Table 4.1: Comparison of methods.

to be implemented, instead of the implementation of only one controller for most LPV methods. For some other conventional methods, the implementation is more difficult (see further in this section), so this criterion is judged neutrally for controller output scheduling.

The advantage of the smooth interpolation of controller outputs in Section 2.3 compared to the previous method, is that there is more effort done in making the switches between the controllers smoothly. The interpolation of the controller is somewhat more difficult. The differences with the previous method are not so big.

For 'Interpolation of poles, zeros and gains' partially the same remarks hold as for controller output scheduling. One of the limitations in the approach is the case where the controller designs have many in- and outputs such that reducing the problem to single-input single-output SISO controller components is inefficient or impossible. The numerator degree of all the controllers must be the same. This also holds for the denumerator degree. If this is the case depends on the controller synthesis method. In particular the implementation of the transfer functions seems difficult. Therefore, ease of controller design and ease of controller implementation are both negatively judged.

For interpolation of state-space matrices the same problems occur as for interpolation of transfer functions, except that there is no problem with MIMO controllers. Therefore, controller design is easier for this method. Implementation seems also easier.

For stability preserving interpolation, the complexity of the overall controller synthesis does not seem to be easier than for the LPV methods. It is necessary to have the global equations of motion available. Ease of modeling is neutrally judged. The controller implementation with transfer functions or state space matrices can be compared with conventional methods treated above: the different controllers have to be implemented together with an interpolation function. Also for this method the controllers all must have the same state dimension, which makes controller design more difficult. The method can guarantees global stability (each controller has to stabilize the whole plant) and as a consequence, conservatism is introduced.

In Section 2.7, use is made of Youla parameterization. This is well-founded theory, but the computation of the controller is difficult. Furthermore, the method is not 'finished'. The free parameter still has to be connected to the controller-schedule variable. This means modeling is difficult. Stability can be guaranteed, performance can not. Conservatism seems smaller than for LPV methods, but larger than for the conventional methods.

The method in 'Robust controller design and performance for polytopic models' (Section 2.8) requires an identification procedure to obtain the models. This is also necessary for the other methods, but with the identification procedure in Section 2.8, no experience exists at Philips CFT. Conservatism seems small. Performance can be analyzed, but it cannot be guaranteed.

For the LPV methods, stability and performance can be guaranteed. For the polytopic approach, controller implementation will be more difficult than the other LPV approaches because multiple controllers have to be implemented, as in conventional methods. Modeling is more difficult than for simple conventional methods, due to the need for a global system description. In principle, controller design is not difficult. The controllers are derived by solving LMIs (which is in fact the case for all LPV methods). However, this is expected to give

problems due to long computation time when the system is of higher order. Therefore, ease of controller design is judged neutrally. Conservatism is introduced because of approximating the true operating region by a convex hull.

Ease of modeling is moderate for LFT dependency, because the reformulation from an LPV to an LFT system is possible, but non-trivial. Also, it may involve considerable increase in the order of the parameter dependent block, which can cause problems when the LMIs have to be solved. The controller implementation seems much easier than for the conventional methods: it is no longer necessary to implement different controllers, but it suffices to implement one controller.

Most criteria for the D-K approach for LPV systems are equivalently judged as for the LMI approach, because both methods are based on an LFT description. Different is that the D-K approach can lead to less conservatism, when use is made of frequency-dependent scalings. Ease of modeling has been judged negatively, because the method can only be used when the scheduling parameter varies slowly.

In the parameter-dependent Lyapunov approach, conservatism is probably less, due to the fact that the scheduling parameter is included in the Lyapunov function. However, the method is substantially more difficult than the other LPV-methods. The computation of the controller consists of several difficult steps. When the scheduling-parameter depends on the state, extra constraints on the state have to be imposed, which are coupled with the choice of the controller.

4.3 Summary

In the last row of the table the 'expected usefulness' of the methods has been reviewed. This is based on the assessment of the criteria in the table. Note that the figures are not very high. This is due to the fact that the conventional methods at the left side of the table are positively judged for the first criteria and negative for the last and the opposite hold for the LPV methods. This is elucidated in Table 4.2.

criterion / method	Conventional	LPV
ease of modeling	+	−
small conservatism	+	−
ease of controller design	+	−
ease of controller implementation	o	o
guaranteed stability	−	+
guaranteed performance	−	+

Table 4.2: Difference between useful conventional and LPV methods.

In Table 4.2, the results from Table 4.1 are summarized for the LPV methods and for the relevant conventional methods (the methods rated 5.0 and higher, except the state-feedback method in Section 2.8). An important conclusion that can be drawn from Table 4.2 is that performance and stability can only be guaranteed at the cost of conservatism. Also ease of modeling and ease of controller design is difficult. These last three criteria are exact the advantages of the conventional methods. This makes clear the low figures in Table 4.1: both methods score good for half of the criteria, both for the other half.

Chapter 5

Conclusions

Different methods for controller scheduling have been described. Controller scheduling covers many approaches, from very simple *ad hoc* techniques to approaches that use the linear parameter varying equations of motion of the plant and LMI-based controller synthesis. To get more insight in what approaches could be useful, a distinction is made between conventional controller scheduling and LPV controller scheduling. Because there are so many different approaches, which differ in degree of difficulty, the borderline between these two classes of controller scheduling is not always clear.

Not all the approaches that can be found in literature have been described. A selection is made regarding the fact that the method must be useful for the wafer stage application described in Chapter 1. Methods that do not fulfil this requirement have not been included in the survey or have only been described shortly. The methods which seem to be useful have been studied.

In Chapter 4, a comparison is made between the described methods. On the basis of a list of criteria, the methods are judged. This gives a good overview in the (im)possibilities of the different techniques. No attempt is made to pick one best method: only after a practical application well-founded pronouncements can be made about the usefulness of a method. According to the comparison in Chapter 4, the following conclusions can be drawn:

- Conventional techniques as described in Sections 2.2, 2.3, 2.4 and 2.5 are simple in essence. Stability and performance cannot be guaranteed. There is much freedom in the design and gridding.
- LPV techniques guarantee stability and performance but they are more conservative than conventional methods. The way the controllers are computed is completely different from current practice at Philips CFT. Also the modeling of the LPV system is difficult.
- Methods in between, as described in Sections 2.6, 2.7 and 2.8, can guarantee stability, but performance cannot be guaranteed. The methods are more difficult than other conventional techniques and are sometimes not well suited.

Based on these conclusions, it is justified to perform more research on both conventional

techniques and LPV techniques. Both from the conventional methods and the LPV methods two methods have been chosen which seem useful for further investigation by simulation and/or experiments:

- Controller output scheduling (Section 2.2).
- Interpolation of state-space matrices (Section 2.5).
- Polytopic LPV approach (Section 3.3).
- LFT dependence (Section 3.4).

The conventional controller methods are chosen because of the transparency of theory. In case of the wafer-stage, the existing experience at Philips CFT can be used to design the controllers. The design of the scheduling and the implementation of the controllers and the scheduling are new. For the LPV methods, the procedure is totally new. Problems can arise in the computation of the controllers with LMIs. However, these methods guarantee stability and performance and therefore they are also interesting to implement.

Appendix A

A.1 Matrix operators

A^T	The transpose of matrix A.
A^*	The complex conjugate of matrix A.
A^{-1}	The inverse of the nonsingular matrix A.
$A^{1/2}$	For $A > 0$, $A^{1/2}$ is the unique $Z = Z^T$ such that $Z > 0$, $Z^2 = A$.
$\text{diag}[A_1, ..., A_n]$	A block diagonal matrix composed of the matrices $A_1, ..., A_n$.

A.2 Positive definite

A matrix $A = A^*$ is said to be positive definite (semi-definite), denoted by $A > 0$ (≥ 0), if $x^*Ax > 0$ (≥ 0) for all $x \neq 0$ ([49, Section 2.10]).

A.3 Linear fractional transformations for uncertainty modeling

A linear fractional transformation (LFT) is a matrix function (for more general information on LFT's see [49, Chapter 10]). The basic principle of an LFT in modeling uncertainty is *'pulling out the delta's'*. These delta's represents the uncertainty of a system. A model of a system can never be exactly the same as the physical system. Differences between a model and reality can be expressed by a kind of *representation of uncertainty*. For the analysis of unstructured model uncertainty in the frequency domain there are three commonly used uncertainty models (additive uncertainty, output multiplicative uncertainty, and input multiplicative uncertainty, see Figure A.1).

To study the stability properties of a closed-loop system subject to unstructured model uncertainty, one can also consider the system in Figure A.2. M represents the closed-loop transfer matrix for a plant controlled by a feedback controller, with external entries w and controllable output z.

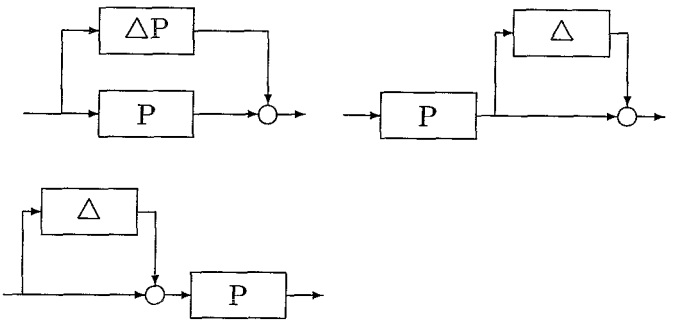

Figure A.1: Uncertainty models.

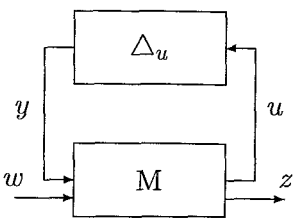

Figure A.2: An LFT $\mathcal{F}_u(M, \Delta_u)$.

The transfer can be written as:

$$\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \begin{pmatrix} u \\ w \end{pmatrix} \tag{A.1}$$

$$y = \Delta_u u \tag{A.2}$$

Or, equivalently, as an upper linear fractional transformation,

$$\mathcal{F}_u(M, \Delta_u) = M_{22} + M_{21}\Delta_u(I - M_{11}\Delta_u)^{-1}M_{12}, \tag{A.3}$$

which lays the closed-loop relation between w and z. It is also possible to define a lower LFT, when a lower loop is closed instead of an upper loop. By definition, an LFT, $\mathcal{F}_u(M, \Delta_u)$, is said to be well defined (or well-posed) if $(I - M_{11}\Delta_u)$ is invertible. See also [49, Section 10.1]. An example of an LFT to describe parameter-dependence is given below.

A.3.1 LFT example

An example of a plant that can be written as an LFT is a simple mass-spring-damper system as in Figure A.3, with mass and damping coefficient varying on a bounded interval. This example is taken from [42]. Suppose the state-space description of the plant with mass coefficient m, damping coefficient b, stiffness k, state x containing the position and velocity and a force acting on the mass represented by the controller output u is given by:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x \end{aligned} \tag{A.4}$$

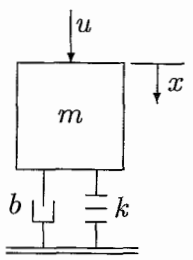

Figure A.3: Simple mass-spring-damper system.

This system is in the form of (3.1). There are no external inputs, so G, F and D are zero and there are no external outputs, so H and E are also zero. In this example, the parameter vector θ contains the mass and damping coefficients. Suppose that the mass m varies between an upper and lower bound, $m \in [m_l, m_u]$. The same holds for the damping coefficient, $b \in [b_l, b_u]$. Define $\alpha_m = \frac{m_l+m_u}{2}$ and $\alpha_b = \frac{b_l+b_u}{2}$, then m and b can be written as:

$$\begin{aligned} m &= m_0 + \alpha_m \delta_m, & \delta_m &\in [-1, 1] \\ b &= b_0 + \alpha_b \delta_b, & \delta_b &\in [-1, 1] \end{aligned} \tag{A.5}$$

It is now possible to reformulate the state-space description in (A.4) as an LFT as in (3.17). First, denote the new inputs corresponding to the parameter block by w_2 and the new outputs by z_2. Second, split the state-space matrices with parameter-dependency, $A(\theta)$ and $B(\theta)$, into parameter dependent and parameter independent parts. The parameter independent parts are now denoted by A and B. The resulting parameter independent part of the first equation of (A.4) has to be reformulated as $G_2 w_2$, where $w_2 = \triangle z_2$ defines the connection with the parameter block. This formulation yields an expression for z_2. The total system can now be written as

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m_0} & -\frac{b_0}{m_0} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{m_0} \end{bmatrix} u + \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} w_2 \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x \\ w_2 &= \begin{bmatrix} \delta_b & 0 \\ 0 & \delta_m \end{bmatrix} z_2 \\ z_2 &= \begin{bmatrix} 0 & -\frac{\alpha_d}{m_0} \\ -\frac{k\alpha_m}{m_0^2} & -\frac{b_0\alpha_m}{m_0^2} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{\alpha_m}{m_0^2} \end{bmatrix} u + \begin{bmatrix} 0 & 0 \\ -\frac{\alpha_m}{m_0} & -\frac{\alpha_m}{m_0} \end{bmatrix} w_2, \end{aligned} \tag{A.6}$$

or, equivalently, as

$$\begin{bmatrix} \dot{x} \\ \hline z_2 \\ \hline y \end{bmatrix} = \left[\begin{array}{cc|cc|c} 0 & 1 & 0 & 0 & 0 \\ -\frac{k}{m_0} & -\frac{b_0}{m_0} & 1 & 1 & \frac{1}{m_0} \\ \hline 0 & -\frac{\alpha_d}{m_0} & 0 & 0 & 0 \\ -\frac{k\alpha_m}{m_0^2} & -\frac{b_0\alpha_m}{m_0^2} & -\frac{\alpha_m}{m_0} & -\frac{\alpha_m}{m_0} & -\frac{\alpha_m}{m_0^2} \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x \\ \hline w_2 \\ \hline u \end{bmatrix}. \tag{A.7}$$

A.4 Convexity

A set $\mathcal{H}$ in a linear vector space is said to be *convex* if

$$\{x_1, x_2 \in \mathcal{H}\} \rightarrow \{x := \alpha x_1 + (1-\alpha)x_2 \in \mathcal{H} \text{ for all } \alpha \in (0,1)\}. \tag{A.8}$$

In geometric terms, this states that for any two points of a convex set also the line segment connecting these two points belongs to the set. In general, the empty set is considered to be convex. The point $\alpha x_1 + (1-\alpha)x_2$ with $\alpha \in (0,1)$ is called a convex combination of the points x_1 and x_2. More generally, convex combinations are defined for any finite set of points as follows.

Let $\mathcal{H}$ be a subset of a normed vector space and let $x_1, ..., x_n \in \mathcal{H}$. If $\alpha_1, ..., \alpha_n$ is a set of non-negative real numbers with $\Sigma_{i=1}^n \alpha_i = 1$ then

$$x := \sum_{i=1}^{n} \alpha_i x_i \tag{A.9}$$

is called a convex combination of $x_1, ..., x_n$. An example of a convex set is given in Section 3.3.

For a convex optimization problem, convexity implies that a local minimum is the global one. When one solution to the problem is found, this solution is the only solution.

A.5 Affine dependence

A matrix function $Y(\theta)$ is said to depend affinely on the parameter vector

$$\theta^T = [\theta_1, \theta_2, ..., \theta_n]$$

if $Y(\theta)$ can be written as

$$Y(\theta) = Y_0 + \theta_1 Y_1 + ... + \theta_n Y_n.$$

The dependence is also linear when $Y_0 = 0$, or equivalently, $Y(\theta = 0) = 0$.

A.6 Small gain theorem

Assume that $\triangle(s) \in \mathcal{RH}_\infty$ and $M(s) \in \mathcal{RH}_\infty$. This means that $M(s)$ and $\triangle(s)$ are real-rational proper and stable transfer functions. Let $\gamma > 0$. The closed-loop system $\mathcal{F}_u(\triangle, M)$ is well-posed and internally stable with

$$\begin{array}{lll} \text{(a)} & \|\triangle\|_\infty \leq 1/\gamma & \text{if and only if } \|M(s)\|_\infty < \gamma \\ \text{(b)} & \|\triangle\|_\infty < 1/\gamma & \text{if and only if } \|M(s)\|_\infty \leq \gamma \end{array} \tag{A.10}$$

See also [49, Section 9.2].

A.7 Coprime factorization

Coprime factorization can be summarized as follows. A linear MIMO system, for example with fixed scheduling parameter $\theta(t)$, can be written as:

$$G_{yu}(s) = NM^{-1} = \tilde{M}^{-1}\tilde{N}, \quad N, M, \tilde{N}, \tilde{M} \in R\mathcal{H}_\infty \quad \text{(real-rational functions in } \mathcal{H}_\infty)^1 \tag{A.11}$$

There exists a controller (for the fixed scheduling parameter $\theta(t)$):

$$K(s) = UV^{-1} = \tilde{V}^{-1}\tilde{U}, \quad U, V, \tilde{U}, \tilde{V} \in R\mathcal{H}_\infty \tag{A.12}$$

The coprime factorizations is chosen to satisfy the double Bezout equation (this is generally possible):

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} M & U \\ N & V \end{bmatrix} = \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} = \begin{bmatrix} I & O \\ O & I \end{bmatrix}. \tag{A.13}$$

This can be explained as follows. Let the controller $K(s)$ be an observer-based feedback controller:

$$K(s) = \left[\begin{array}{c|c} A + B_uF + HC_y + HD_{yu}F & -H \\ \hline F & 0 \end{array}\right] \tag{A.14}$$

Suppose $G_{yu}(s)$ is a proper real-rational matrix and

$$G_{yu} = \left[\begin{array}{c|c} A & B_u \\ \hline C_y & D_{yu} \end{array}\right]$$

is a *stabilizable* and *detectable* realization. The pair (A, B) is said to be stabilizable if there exists a state feedback $u = Fx$ such that $A + B_uF$ is stable. The pair (C, A) is detectable if $A + HC_y$ is stable for some H. See [49, Section 3.2]. Let F and H be such that both $A + B_uF$ and $A + HC_y$ are stable. It is then possible to construct the eight matrices from equations (A.11) and (A.12) as

$$\begin{bmatrix} M & U \\ N & V \end{bmatrix} = \left[\begin{array}{c|cc} A + B_uF & B_u & -H \\ \hline F & I & 0 \\ C_y + D_{yu}F & D_{yu} & I \end{array}\right] \tag{A.15}$$

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} = \left[\begin{array}{c|cc} A + HC_y & -(B_u + HD_{yu}) & H \\ \hline F & I & 0 \\ C_y & -D_{yu} & I \end{array}\right]. \tag{A.16}$$

The left and right side of equation (A.15) represent the same dynamic system. This also holds for equation (A.16). Now, $G_{yu} = NM^{-1} = \tilde{M}^{-1}\tilde{N}$ are *right-coprime* and *left-coprime factorizations*, respectively. See also [49, Section 4.5] or [9]). Youla parameterization via coprime factorization is treated next.

[1] If $F(s)$ is real-rational, then $F \in R\mathcal{H}_\infty$ if and only if F is proper ($|F(\infty)|$ is finite) and stable.

A.7.1 Youla parameterization via coprime factorization

It is now possible to give a parameterization of all controllers that stabilize the system in terms of a stable parameter $Q(s)$ (see [49, Section 12.6]). In fact, this is a reformulation of (A.12) in such way that not only one single controller that stabilizes the system is presented but that the whole set of controllers that all stabilize the system (A.11) is presented:

$$K(Q) = U(Q)V(Q)^{-1} \tag{A.17}$$

where

$$U(Q) := U_0 + MQ, \qquad V(Q) := V_0 + NQ, \; Q \in \mathcal{RH}_\infty \tag{A.18}$$

or by using a left factored form:

$$K(Q) = \tilde{V}_0(Q)^{-1}\tilde{U}_0(Q) \tag{A.19}$$

where

$$\tilde{U}_0(Q) = \tilde{U}_0 + Q\tilde{M}, \qquad \tilde{V}_0(Q) = \tilde{V}_0 + Q\tilde{N}, \; Q \in \mathcal{RH}_\infty.$$

The parameter Q is called the free parameter. So, the controller is

$$K(Q) = (U_0 + MQ)(V_0 + NQ)^{-1} \tag{A.20}$$

This is a linear fractional transformation in the parameter Q and can also be written as:

$$K(Q) = \mathcal{F}_l(J_K, Q) = J_{K_{11}} + J_{K_{12}}Q(I - J_{K_{22}}Q)^{-1}J_{K_{21}} \tag{A.21}$$

where J_K is

$$J_K = \begin{bmatrix} U_0V_0^{-1} & \tilde{V}_0^{-1} \\ V_0^{-1} & -V_0^{-1}N \end{bmatrix} = \begin{bmatrix} \tilde{V}_0^{-1}\tilde{U}_0 & \tilde{V}_0^{-1} \\ V_0^{-1} & -V_0^{-1}N \end{bmatrix}. \tag{A.22}$$

So, the controller given by either (A.17) or (A.19) can be written as

$$K(Q) = U_0V_0^{-1} + \tilde{V}_0^{-1}Q(I + V_0^{-1}NQ)^{-1}V_0^{-1}, \;\; Q \in R\mathcal{H}_\infty. \tag{A.23}$$

Using (A.12), this can also be written as:

$$K(Q) = K + \tilde{V}_0^{-1}Q(I + V_0^{-1}NQ)^{-1}V_0^{-1}, \;\; Q \in R\mathcal{H}_\infty. \tag{A.24}$$

A.8 $\mathcal{H}_\infty$ loop-shaping

$\mathcal{H}_\infty$ loop-shaping or normalized coprime factor stabilization is a procedure based on $\mathcal{H}_\infty$ robust stabilization combined with classical loop shaping. See [33, Section 9.4].

The open-loop plant is augmented by pre- and post-compensators to give a desired shape to the singular values (see [49, Section 2.8]) of the open-loop frequency response. Then the resulting shaped plant is robustly stabilized with respect to coprime factor (see Section A.7) uncertainty using $\mathcal{H}_\infty$ optimization. A plant G with coprime factorization is given by

$$G = NM^{-1}. \tag{A.25}$$

A perturbed plant can then be written as

$$G_p = (N + \triangle_N)(M + \triangle_M)^{-1}, \tag{A.26}$$

where $\triangle_N$, $\triangle_M$ are stable unknown transfer functions which represent the uncertainty in the nominal model G. The objective of robust stabilization is to stabilize the family of plants described by G_p.

This approach produces a controller in the form of a plant observer H and state feedback F:

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + H(C\hat{x} - y) + Bu \\ u &= F\hat{x} \end{aligned} \tag{A.27}$$

See also [49, Section 18].

A.9 Stability

An unforced dynamical system $\dot{x} = Ax$ is said to be *stable* if all the eigenvalues of A are in the open left half plane, *i.e.*, $Re\lambda(A) < 0$. A matrix A with such a property is said to be stable or Hurwitz ([49, Section 3.2]).

An equilibrium point $\mathbf{0}$ is *asymptotically stable* if it is stable and if in addition there exists some $r > 0$, such that $\|x(0)\| < r$ implies that $x(t) \to 0$ as $t \to 0$ ([34, Section 3.2]).

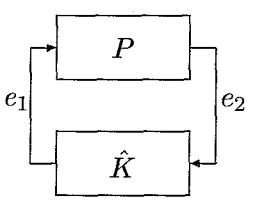

Figure A.4: Internal stability analysis.

Consider the system in Figure A.4 and assume the system is well-posed (see Section A.3) and that the realizations

$$P(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \qquad \text{and} \qquad \hat{K}(s) = \left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array}\right]$$

are stabilizable and detectable. There are no external inputs to the system. Let x and $\hat{x}$ denote the state vectors for P and $\hat{K}$, respectively. Now, the system in Figure A.4 is said to be *internally stable* if the origin $(x, \hat{x}) = (0, 0)$ is asymptotically stable, *i.e.*, the states $(x, \hat{x})$ go to zero from all initial states ([49, Section 5.3]).

Now consider a controlled system with LFT dependence on the uncertainty $\triangle(t)$. The system has state x, external inputs $w_2, ..., w_m$, external outputs $z_2, ..., z_m$ and the input w_1 and output z_1 connected with the uncertainty block $\triangle(t)$. The system is *uniformly exponentially stable* if the system is well-posed and there exist constants K and $\alpha > 0$ such that, for every uncertainty $\triangle(\cdot)$ and for every unforced ($w_2 = 0, ..., w_m = 0$) system trajectory $x(\cdot)$,

$$\|x(t)\| \leq Ke^{-\alpha(t-t_0)}\|x(t_0)\| \quad \forall \quad t \geq t_0 \geq 0, \quad \alpha > 0. \tag{A.28}$$

A.10 Performance

Consider the dynamical system:

$$\dot{x} = Ax + Bu \qquad x(t_0) = x_0. \tag{A.29}$$

The objective is to find a control function defined on the interval $[t_0, T)$ such that the state $x(t)$ is driven to a small neighborhood of the origin at time T. This is the regulator problem. When a system is controllable, this can be trivially solved for any $T > t_0$. However, in practice, limitations have to be imposed on the control input and the transient response.

The constraints on control u and transient response $x(t)$ can be measured using the weighted L_2-norm:

$$\int_{t_0}^{T} \|W_u u\|^2 dt \qquad \text{and} \qquad \int_{t_0}^{T} \|W_x x\|^2 dt$$

for some weighting matrices W_u and W_x. Hence the regulator problem can be posed as an optimal control problem with certain combined performance index on u and x. When focusing on the infinite time regulator problem, *i.e.*, $T \to \infty$, and, without loss of generality assume $t_0 = 0$, the problem is as follows: Find a control $u(t)$ defined on $[0, \infty)$ such that the state $x(t)$ is driven to the origin at $t \to \infty$ and the performance index $W = \begin{bmatrix} Q_p & S_p \\ S_p^* & R_p \end{bmatrix}$ is minimized:

$$\min_u \int_0^\infty \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^* \begin{bmatrix} Q_p & S_p \\ S_p^* & R_p \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \tag{A.30}$$

for some $Q_p = Q_p^*$, S_p, and $R_p = R_p^* > 0$. This is called an integral quadratic constraint (IQC).

Robust stability and robust performance for an uncertain system can also be characterized in this way. This is done by introducing scalings that characterize the nature of the uncertainties Δ_i affecting the plant in terms of IQC's. For $w_i = \Delta_i z_i$, The robust quadratic performance specification on the channel i is as follows: there exists an $\epsilon > 0$ such that the IQC

$$\int_0^\infty \begin{bmatrix} w_i \\ z_i \end{bmatrix}^* \begin{bmatrix} Q_{pi} & S_{pi} \\ S_{pi}^* & R_{pi} \end{bmatrix} \begin{bmatrix} w_i \\ z_i \end{bmatrix} dt \leq -\epsilon \int_0^\infty w_i(t)^* w_i(t) dt \tag{A.31}$$

holds for any trajectory of the system with $x(0) = 0$. The scalings can be collected into block-diagonal matrices $Q_p = \text{diag}(Q_{p0}, Q_{p1}, ..., Q_{pk})$, $R_p = \text{diag}(R_{p0}, R_{p1}, ..., R_{pk})$, $S_p = \text{diag}(S_{p0}, S_{p1}, ..., S_{pk})$.

For instance, taking the L_2-gain of the channel $w_0 \to z_0$ as a measure for performance, this is bounded by the value γ if the IQC (A.31) holds with the fixed scalings

$$Q_{p0} = -\frac{1}{\gamma}, \qquad R_{p0} = \gamma I, \qquad S_{p0} = 0. \tag{A.32}$$

(see also [27], [49] and [46]).

The robust quadratic performance problem can also be written as an LMI. It is briefly described below how this is done. For more information, see [29] and [31]. The quadratic performance specification (A.31) can, based on Lyapunov arguments, equivalently be described by

an LMI. Assume there exists an LTI system with time-varying parametric uncertainties with $\triangle(t) \in \triangle_{convex}$ and $\triangle_c$ the parameter block corresponding to the controller. It can be proven that, if the corresponding closed-loop system

$$\begin{bmatrix} \dot{x} \\ \hline z_1 \\ z_c \\ z_2 \end{bmatrix} = \left[\begin{array}{c|ccc} \mathcal{A} & \mathcal{B}_1 & \mathcal{B}_c & \mathcal{B}_2 \\ \hline \mathcal{C}_1 & \mathcal{D}_{11} & \mathcal{D}_{1c} & \mathcal{D}_{12} \\ \mathcal{C}_c & \mathcal{D}_{c1} & \mathcal{D}_{cc} & \mathcal{D}_{c2} \\ \mathcal{C}_2 & \mathcal{D}_{21} & \mathcal{D}_{2c} & \mathcal{D}_{22} \end{array}\right] \begin{bmatrix} x \\ \hline w_1 \\ w_c \\ w_2 \end{bmatrix} \tag{A.33}$$

$$\begin{bmatrix} w_2 \\ w_c \end{bmatrix} = \begin{bmatrix} \triangle & 0 \\ 0 & \triangle_c(\triangle) \end{bmatrix} \begin{bmatrix} z_2 \\ z_c \end{bmatrix}, \tag{A.34}$$

is well-posed and there exists an $\mathcal{X}$ satisfying

$$\mathcal{X} > 0, \begin{pmatrix} * \\ * \\ \hline * \\ * \end{pmatrix}^T \left(\begin{array}{cc|cc} 0 & \mathcal{X} & 0 & 0 \\ \mathcal{X} & 0 & 0 & 0 \\ \hline 0 & 0 & Q_{pi} & S_{pi} \\ 0 & 0 & S_{pi}^T & R_{pi} \end{array}\right) \left(\begin{array}{cc} I & 0 \\ \mathcal{A}(\triangle) & \mathcal{B}_i(\triangle) \\ \hline 0 & I \\ \mathcal{C}_i(\triangle) & \mathcal{D}_{ii}(\triangle) \end{array}\right) < 0 \tag{A.35}$$

for all $\triangle$, then the system is uniformly exponentially stable and satisfies the robust quadratic performance specification for the channel $w_i \to z_i$ (see, for instance, [29] and [31]).

On the basis of the theory behind the full block S-procedure ([28]), $\mathcal{X}$ satisfies the inequality (A.35) if and only if there exists scalings $W \in \mathcal{W}$;

$$\mathcal{W} := \left\{ W \in \mathbf{R}^{(k+l)\times(k+l)} : \; W = W^T, \begin{pmatrix} \triangle \\ I \end{pmatrix}^T W \begin{pmatrix} \triangle \\ I \end{pmatrix} > 0 \text{ for all } \triangle \in \triangle \right\}, \tag{A.36}$$

$$W = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}, \tag{A.37}$$

such that

$$\mathcal{X} > 0, \begin{pmatrix} * \\ * \\ \hline * \\ * \\ \hline * \end{pmatrix}^T \left(\begin{array}{cc|cc|cc} 0 & \mathcal{X} & 0 & 0 & 0 & 0 \\ \mathcal{X} & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & Q & S & 0 & 0 \\ 0 & 0 & S^T & R & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & Q_{pi} & S_{pi} \\ 0 & 0 & 0 & 0 & S_{pi}^T & R_{pi} \end{array}\right) \left(\begin{array}{ccc} I & 0 & 0 \\ \mathcal{A}(\triangle) & \mathcal{B}_1 & \mathcal{B}_i \\ \hline 0 & I & 0 \\ \mathcal{C}_1 & \mathcal{D}_{11} & \mathcal{D}_{1i} \\ \hline 0 & 0 & I \\ \mathcal{C}_i & \mathcal{D}_{i1} & \mathcal{D}_{ii} \end{array}\right) < 0. \tag{A.38}$$

Ideally, to have a non-conservative representation of the parameter-set, one has to determine the set of all scalings that satisfy (A.36). Unfortunately, the exact description of this set is in general hard, if not impossible. This is the reason to work with subsets, for instance, to work with diagonal multipliers. The price to pay is introducing conservatism. A larger subset of all scalings that satisfy (A.36) can be implicitly parameterized by a finite number of inequalities:

$$\begin{pmatrix} \triangle_j \\ I \end{pmatrix}^T W \begin{pmatrix} \triangle_j \\ I \end{pmatrix} > 0 \;\; \forall \; j = 1, ..., k. \tag{A.39}$$

Now, an additional constraint on W has to be imposed such that the finitely many inequalities (A.39) imply the condition (A.36). The simplest possible restriction is $Q < 0$.

By *dualization* and *explicit solvability tests* (see [29] and [31]), it is possible to eliminate variables in the LMI in order to reduce computation time. By doing so, the basis matrices K_1 and K_2 of the kernels of $\ker\begin{pmatrix} B^T & E_1^T & E_2^T \end{pmatrix}$ and $\ker\begin{pmatrix} C & F_1 & F_2 \end{pmatrix}$ respectively appear in the matrix inequalities. Then the following equivalent synthesis test is obtained: Find X, Y and multipliers $W \in \mathcal{W}$, $\tilde{W} \in \tilde{\mathcal{W}}$ that satisfy

$$\begin{pmatrix} Y & I \\ I & X \end{pmatrix} > 0, \tag{A.40}$$

$$K_2^T \begin{pmatrix} * \\ * \\ \hline * \\ * \\ \hline * \\ * \end{pmatrix}^T \left(\begin{array}{cc|cc|cc} 0 & I & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & Q & S & 0 & 0 \\ 0 & 0 & S^T & R & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & Q_p & S_p \\ 0 & 0 & 0 & 0 & S_p^T & R_p \end{array}\right) \left(\begin{array}{ccc} I & 0 & 0 \\ XA & XB_1 & XB_2 \\ \hline 0 & I & 0 \\ C_1 & D_{11} & D_{12} \\ \hline 0 & 0 & I \\ C_2 & D_{21} & D_{22} \end{array}\right) K_2 < 0, \tag{A.41}$$

$$K_1^T \begin{pmatrix} * \\ * \\ \hline * \\ * \\ \hline * \\ * \end{pmatrix}^T \left(\begin{array}{cc|cc|cc} 0 & I & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \tilde{Q} & \tilde{S} & 0 & 0 \\ 0 & 0 & \tilde{S}^T & \tilde{R} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \tilde{Q}_p & \tilde{S}_p \\ 0 & 0 & 0 & 0 & \tilde{S}_p^T & \tilde{R}_p \end{array}\right) \left(\begin{array}{ccc} -YA^T & -YC_1^T & -YC_2^T \\ I & 0 & 0 \\ \hline -B_1^T & -D_{11}^T & -D_{21}^T \\ 0 & I & 0 \\ \hline -B_2^T & -D_{12}^T & -D_{22}^T \\ 0 & 0 & I \end{array}\right) K_1 < 0, \tag{A.42}$$

and the duality coupling condition

$$\begin{pmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^T & \tilde{R} \end{pmatrix} = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}^{-1} \tag{A.43}$$

with multipliers

$$W_e = \begin{bmatrix} Q_e & S_e \\ S_e^T & R_e \end{bmatrix} = \begin{bmatrix} Q & Q_{12} & S & S_{12} \\ Q_{21} & Q_{22} & S_{21} & S_{22} \\ S^T & S_{21}^T & R & R_{12} \\ S_{12}^T & S_{22}^T & R_{21} & R_{22} \end{bmatrix} \qquad \text{with } Q_e < 0, \;\; R_e > 0 \tag{A.44}$$

and they satisfy

$$\begin{bmatrix} \Delta & 0 \\ 0 & \Delta_c(\Delta) \\ I & 0 \\ 0 & I \end{bmatrix}^T W_e \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_c(\Delta) \\ I & 0 \\ 0 & I \end{bmatrix} \geq 0 \qquad \text{for all } \Delta \in \Delta_{convex}. \tag{A.45}$$

The corresponding dual multipliers $\tilde{W}_e = W_e^{-1}$ are partitioned similarly as

$$\tilde{W}_e = \begin{bmatrix} \tilde{Q}_e & \tilde{S}_e \\ \tilde{S}_e^T & \tilde{R}_e \end{bmatrix} = \begin{bmatrix} \tilde{Q} & \tilde{Q}_{12} & \tilde{S} & \tilde{S}_{12} \\ \tilde{Q}_{21} & \tilde{Q}_{22} & \tilde{S}_{21} & \tilde{S}_{22} \\ \tilde{S} & \tilde{S}_{21} & \tilde{R} & \tilde{R}_{12} \\ \tilde{S}_{12} & \tilde{S}_{22} & \tilde{R}_{21} & \tilde{R}_{22} \end{bmatrix} \qquad \text{with } \tilde{Q}_e < 0, \;\; \tilde{R}_e > 0. \tag{A.46}$$

Bibliography

[1] Georgo Angelis, Ramidin Kamidi, René van de Molengraft, and Henk Nijmeijer. Optimal polytopic control system design. In *ISIC*, Patras, Greece, July 2000.

[2] G.Z. Angelis, M.G.J. van de Molengraft, R.J.P. van de Linden, and J.J. Kok. Robust controller design and performance for polytopic models. In *Proceedings of the European Control Conference*, Karlsruhe, Germany, August-September 1999.

[3] P. Apkarian and R.J. Adams. Advanced gain-scheduling techniques for uncertain systems. *IEEE Transactions on Control Systems Technology*, 6(1):21–32, January 1998.

[4] P. Apkarian and P. Gahinet. A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers. *IEEE Transactions on Automatic Control*, 40(5):853–864, May 1995.

[5] P. Apkarian, P. Gahinet, and G. Becker. Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: a design example. *Automatica*, 31(9):1251–1261, 1995.

[6] Pierre Apkarian and Richard J. Adams. Advanced gain-scheduling techniques for uncertain systems. In *ACC Proceedings of the American Control Conference*, pages 3331–3335, Albuquerque, New Mexico, June 1997.

[7] Harald Buschek. Full envelope missile autopilot design using gain scheduled robust control. *Journal of Guidance, Control, and Dynamics*, 22(1):115–122, January-February 1999.

[8] M. Dettori and C.W. Scherer. Gain scheduling control for a CD player using LPV techniques. *Selected Topics in Identification, modeling and Control*, 11:9–16, December 1998.

[9] B.A. Francis. *A course in $\mathcal{H}_\infty$ control theory.* Number 88 in Lecture notes in control and information sciences. Springer Verlag, 1987.

[10] Pascal Gahinet and Pierre Apkarian. A linear matrix inequality approach to $\mathcal{H}_\infty$ control. *International Journal of Robust and Nonlinear Control*, 4:421–448, 1994.

[11] Pascal Gahinet, Arkadi Nemirovski, Alan J. Laub, and Mahmoud Chilali. *LMI Control Toolbox.* The Mathworks Inc., 1995.

[12] P.J. Gawthrop. Continuous-time local state local model networks. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 852–857, Vancouver, Canada, 1995.

[13] A. Helmersson. *Methods for robust gain scheduling.* PhD thesis, Department of electrical engineering, Linköpings Universitet, Sweden, 1995.

[14] K.J. Hunt and T.A. Johansen. Design and analysis of gain-scheduled control using local controller networks. *International Journal of Control*, 66(5):619–651, 1997.

[15] Richard A. Hyde and Keith Glover. The application of scheduled $\mathcal{H}_\infty$ controllers to a VSTOL aircraft. *IEEE Transactions on automatic control*, 38(7):1021–1039, July 1993.

[16] M.A. Jarrah and O.M. Al-Jarrah. Position control of a robot manipulator using continuous gain scheduling. In *Proceedings of the 1999 IEEE International Conference on Robotics & Automation*, Detroit, Michigan, May 1999.

[17] Joy H. Kelly and Johnny H. Evers. An interpolation strategy for scheduling dynamic compensators. *AIAA paper 97-3763*, pages 1682–1690, 1997.

[18] Fumio Matsumura, Toru Namerikawa, Kazuhiko Hagiwara, and Masayuki Fujita. Application of gain scheduled $\mathcal{H}_\infty$ robust controllers to a magnetic bearing. *IEEE transactions on control systems technology*, 4(5):484–493, September 1996.

[19] Marc W. McConley, Brent D. Appleby, Munther A. Dahleh, and Eric Feron. A computationally efficient lyapunov-based scheduling procedure for control of nonlinear systems with stabiltiy guarantees. *IEEE Transactions on Automatic Control*, 45(1):33–49, January 2000.

[20] Eiichi Muramatsu. Stabilization of a plant with variable operating conditions: Interpolation of more than two representative models. In *Proceedings of the 4th European Control Conference*, 1997.

[21] Eiichi Muramatsu. An interpolated controller for stabilization of a plant with variable operating conditions. *IEEE transactions on automatic control*, 44(1):76–81, January 1999.

[22] R.A. Nichols, R.T. Reichert, and W.J. Rugh. Gain scheduling for $\mathcal{H}_\infty$ controllers: a flight control example. *IEEE transactions on control systems technology*, 1(2):69–79, June 1993.

[23] H. Niemann and J. Stoustrup. An architecture for implementation of multivariable controllers. In *Proceedings of the American Control Conference*, pages 4029–4033, San Diego, California USA, June 1999.

[24] H. Niemann and J. Stoustrup. Gain-scheduling using the Youla parameterization. In *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 2306–2311, Phoenix, Arizona, December 1999.

[25] R.G.E. Njio, C.W. Scherer, and S. Bennani. Parametrically varying flight control system design with full block scalings. *Selected Topics in Identification, Modeling and Control*, 9:113–120, December 1996.

[26] A. Packard. Gain scheduling via linear fractional transformations. *Systems & Control Letters*, 22:79–92, October 1994.

[27] Carsten Scherer, Pascal Gahinet, and Mahmoud Chilali. Multiobjective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, 42(7):896–911, July 1997.

[28] C.W. Scherer. A full block S-procedure with applications. In *Proceedings of the 36th Conference on Decision & Control*, San Diego, California USA, December 1997.

[29] C.W. Scherer. Robust mixed control and LPV control with full block scalings. In Laurent El Ghaoui and Silviu-Iulian Niculescu, editors, *Advances in Linear Matrix Inequality Methods in Control*, SIAM, Philapelphia, 1999.

[30] C.W. Scherer, R.G.E. Njio, and S. Bennani. Parametrically varying flight control system design with full block scalings. In *Proceedings of the 36th conference on Decision & control*, San Diego, California USA, December 1997.

[31] C.W. Scherer and S. Weiland. *Lecture notes DISC course on Linear Matrix Inequalities in Control*, April 2000.

[32] Efstratios Skafidas, Andrey V. Savkin, Robin J. Evans, and Ian R. Petersen. Robust stabilization via state feedback controller switching. In *2nd Asian Control Conference*, volume 3, pages 235–239, Seoul, Korea, 1997.

[33] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control.* John Wiley & Sons Ltd, 1996.

[34] Jean-Jacques E. Slotine and Weiping Li. *Applied nonlinear control.* Prentice-Hall Inc., 1991.

[35] Mark Spillman, Paul Blue, Siva Banda, and Lawton Lee. A robust gain-scheduling example using linear parameter-varying feedback. In *Proceedings of the 13th IFAC World Congress*, volume H, pages 221–226, San Francisco, 1996.

[36] Daniel J. Stilwell and Wilson J. Rugh. Stability preserving interpolation methods for the synthesis of gain scheduled controllers. *Automatica*, 36:665–671, May 2000.

[37] Jakob Stoustrup, Palle Andersen, Tom S. Pedersen, and Mads E. Hangstrup. Gain-scheduling procedures for unstable controllers: a behavioral approach. In *Proceedings of the American Control Conference*, San Diego, California, June 1999.

[38] Jawad Talaq and Fadel Al-Basri. Adaptive fuzzy gain scheduling for load frequency control. *IEEE Transactions on Power Systems*, 14(1), February 1999.

[39] S. Tan and J. Chai. Gain scheduling: from conventional to neuro-fuzzy. *Automatica*, 33(3):411–419, 1997.

[40] K. Tanaka and T. Ikeda H.O. Wang. Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: quadratic stabilizability, $\mathcal{H}_\infty$ control theory, and linear matrix inequalities. *IEEE Transactions on Fuzzy Systems*, 4(1):1–13, February 1996.

[41] R. Tebbens. Polytopic linear modeling of nonlinear mechanical systems: A data based p.l.m. generating method. Technical Report WFW 99.030, Eindhoven University of Technology, Department of Mechanical Engineering, section Systems and Control, 1999.

[42] J.C. Terlouw and P. Lambrechts. A matlab toolbox for parametric uncertainty modeling. Technical report, Technical University Delft and National Aerospace Laborotories, September 1992.

[43] Marc van de Wal. Advanced feedback control of electromechanical servo systems: Theory, tools, and applications to ASML stages. Technical Report CTB595-00-xxxx, Philips Centre for Manufacturing Technology (CFT), Mechatronics Motion, December 2000. To appear.

[44] Marc van de Wal and Gregor van Baars. Position-dependent control of the SIRE-T5 wafer stage: An exploratory study. Technical Report CTB595-99-2109, Philips Centre for Industrial Technology (CFT), Mechatronics Motion, 1999.

[45] J. Verstraete. Polytopic linear modeling of nonlinear mechanical systems. Technical Report WFW 99.019, Eindhoven University of Technology, Department of Mechanical Engineering, section Systems and Control, 1999.

[46] Dehlia M.C. Willemsen, Samir Bennani, and Carsten W. Scherer. Robust linear parametrically varying flight control system design with bounded rates. *Selected Topics in Identification, Modeling and Control*, 9:105–112, December 1996.

[47] C. Yang, T Kuo, and H. Tai. $\mathcal{H}_\infty$ gain scheduling using fuzzy rules. In *IEEE Proceedings of the 35th conference on Decision and control*, Kobe, Japan, December 1996.

[48] J. Zhao, V. Wertz, and R. Gore. Fuzzy gain scheduling controllers based on fuzzy models. In *Proceedins of the Fifth IEEE International Conference on Fuzzy systems*, volume 3, pages 1670–1676, 1996.

[49] Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and optimal control.* Prentice-Hall Inc., 1995.

[50] Kemin Zhou, Pramod P. Khargonekar, Jakob Stoustrup, and Has Henrik Niemann. Robust stability and performance of uncertain systems in state space. In *Proceedings of the IEEE Conference on Decision and Control*, pages 662–665, Tucson, Arizona, December 1992.