Lehrstuhl für Steuerungs- und Regelungstechnik
Technische Universität München
Ordinarius: Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss

# Generation of low order LFT Representations for Robust Control Applications

**Simon Hecker**

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

**Doktor-Ingenieurs (Dr.-Ing.)**

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Klaus Diepold

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss
2. Univ.-Prof. Dr.-Ing. habil. Boris Lohmann

Die Dissertation wurde am 04.07.2006 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 16.10.2006 angenommen.

Veröffentlicht im VDI-Verlag:

Hecker, S.: Generation of low order LFT Representations for Robust Control Applications. Fortschritt-Berichte VDI Reihe 8 Nr.1114 Düsseldorf: VDI-Verlag 2006, ISBN 978-3-18-511408-3

# Vorwort

Die vorliegende Arbeit entstand am Institut für Robotik und Mechatronik des Deutschen Zentrums für Luft- und Raumfahrt e.V. (DLR) in Oberpfaffenhofen. Herrn Dr.-Ing. Johann Bals danke ich nicht nur für die Möglichkeit, die Arbeit in seiner Abteilung durchzuführen, sondern auch für die Freiheit in der Bearbeitung der mir gestellten wissenschaftlichen Aufgabe, sowie für das Vertrauen, das er mir nicht nur auf diesem Gebiet entgegenbrachte.
Herrn Prof. Dr.-Ing./Univ. Tokio Martin Buss danke ich für die Übernahme des Erstgutachtens, sowie für die hilfreichen Vorschläge und Anmerkungen zu meiner Arbeit. Herrn Prof. Dr.-Ing. Boris Lohmann danke ich für die Übernahme des Zweitgutachtens und Herrn Prof. Dr.-Ing. Klaus Diepold danke ich für die Übernahme des Vorsitzes der Prüfung.
Mein Dank gilt allen Kolleginnen und Kollegen für das sehr gute Arbeitsklima in der Abteilung für Entwurfsorientierte Regelung. Insbesondere danke ich Herrn Dr. Andras Varga, der durch seine Ideen und Arbeiten mein Interesse an dem Forschungsthema geweckt hat. Viele der Ergebnisse und Erkenntnisse resultieren aus den zahlreichen Diskussionen mit ihm. Ich danke ihm auch für die detaillierte Durchsicht meiner Arbeit. Besonderer Dank gilt auch Herrn Gertjan Looye, der durch seine vielen Anregungen und seine Unterstützung beim RCAM Anwendungsbeispiel wesentlich zum Gelingen der Promotion beigetragen hat.
Meinen Eltern danke für die Unterstützung, durch die sie mir die Arbeit überhaupt erst ermöglichten. Die Möglichkeit meines Studiums und ihr Rückhalt haben mir die Voraussetzungen für die vorliegende Arbeit gegeben.
Bei meiner Frau Birgit möchte ich mich in ganz besonderer Weise für ihre liebevolle Unterstützung bedanken. Ihr und der vielen Freude mit unserem Sohn Leon verdanke ich die Motivation zur Durchführung meiner Promotion.

München, 2006 Simon Hecker

Für Birgit und Leon

# Contents

# Acronyms

| | |
|---|---|
| BTA | Balanced Truncation Approximation |
| DAE | Differential and Algebraic Equations |
| ETD | Enhanced Tree Decoposition |
| IQC | Integral Quadratic Constraint |
| LFR | Linear Fractional Representation |
| LFT | Linear Fractional Transformation |
| LMI | Linear Matrix Inequality |
| LPV | Linear Parameter Varying |
| LQG | Linear Quadratic Gaussian |
| LTI | Linear Time Invariant |
| MIMO | Multiple Input Multiple Output |
| OL | Open-Loop |
| ONR | Output Nulling Representation |
| PS | Parameter Space |
| PUM | Parametric Uncertainty Modelling |
| QFT | Quantitative Feedback Theory |
| RCAM | Research Civil Aircraft Model |
| SISO | Single Input Single Output |
| SLICOT | Subroutine Library in Systems and Control Theory |
| SPA | Singular Perturbation Approximation |
| TD | Tree Decomposition |
| VS | Variable Splitting |

# Notations

| | |
|---|---|
| $K$ | controller matrix |
| $G$ | plant model |
| $G_a$ | actuator model |
| $G_d$ | reference model |
| $S$ | sensitivity function |
| $T$ | complementary sensitivity function |
| $R$ | controller sensitivity function |
| $\delta$ | parameter vector |
| $\Delta$ | uncertainty matrix |
| $\Pi$ | uncertain parameter set |
| $\mathcal{F}_u(\cdot,\cdot)$ | upper LFT |
| $\mathcal{F}_l(\cdot,\cdot)$ | lower LFT |
| $I_n$ | n-dimensional identity matrix |
| $0_n$ | $n \times n$ zero matrix |
| $\mathbb{R}$ | set of real numbers |
| $\mathbb{R}^{n\times m}$ | set of $n$ by $m$ matrices with elements in $\mathbb{R}$ |
| $\mathbb{R}[\delta]$ | set of real multivariate polynomial functions |
| $\mathbb{R}[\delta,\delta^{-1}]$ | set of real multivariate Laurent polynomial functions |
| $\mathbb{R}(\delta)$ | set of real multivariate rational functions |
| $\mathbb{C}$ | set of complex numbers |
| $\mathbb{C}^{n\times m}$ | set of $n$ by $m$ matrices with elements in $\mathbb{C}$ |
| $\mathbb{Z}$ | set of integers |
| $\mathbf{R}H_\infty$ | set of real rational proper stable matrices |
| $X^T$ | transpose of matrix $X$ |
| $X^{-1}$ | inverse of matrix $X$ |
| $\ker[X]$ | null space of matrix $X$ |
| $X > Y$ | $X, Y$ hermitian, $X - Y$ positive definite |
| $X \geq Y$ | $X, Y$ hermitian, $X - Y$ positive semidefinite |
| 1-d | one-dimensional |
| n-d | n-dimensional (multi-dimensional) |
| $\overline{\sigma}$ | largest singular value |
| $\mu$ | structured singular value |

# Abstract

The Linear Fractional Transformation (LFT) is a general, flexible and powerful framework to represent uncertain systems. Linear Fractional Representations (LFRs) are the basis for the application of many modern robust control techniques (e.g., robust $H_\infty$ control design, $\mu$-synthesis/analysis). For several classes of uncertain systems, it is in principle straightforward to generate equivalent LFRs. However, the resulting LFRs are generally not unique, a theory for the generation of LFRs with minimal complexity does not exist and the pure application of existing ad-hoc realization methods generally yields LFRs of high complexity. LFR-based modern robust control methods are numerically highly demanding and of high computational complexity, e.g., many methods require to solve a large system of Linear Matrix Inequalities (LMIs). Therefore the application of these methods is restricted to LFRs of reasonable complexity, otherwise the computation time will be unacceptable or the methods may even fail.

To realize LFRs of low complexity, a three step procedure is employed in this thesis consisting of (i) symbolic preprocessing of uncertain system models using improved and newly developed decomposition techniques, (ii) object-oriented LFR realization based on a newly developed generalized/descriptor LFT, (iii) numerical multidimensional order reduction based on newly implemented numerical reliable and efficient routines.

All the techniques are implemented in version 2 of the LFR-toolbox for MATLAB and allow to realize an LFR of almost minimal complexity for one of the most complex parametric aircraft models available in the literature. Using this LFR, a reliable LFR-based robust stability analysis covering the whole flight envelope has been performed, which was not possible with earlier generated LFRs of high complexity.

An LFR of minimal complexity is generated for a parametric vehicle model. Based on this LFR, a $\mu$-synthesis controller and a gain-scheduled Linear Parameter Varying (LPV) controller are synthesized. Both controllers show better performance and are robust with respect to a considerably larger parametric uncertainty domain than recently developed controllers using the Parameter Space (PS) method.

# 1 Introduction

## 1.1 Historical Remarks

Model uncertainty and robustness have always been a central theme in the field of automatic control and the main motivation for using feedback control is to reduce the effects of uncertainty, which may appear in various forms as disturbances or as imperfections in the models used to design control laws [8].

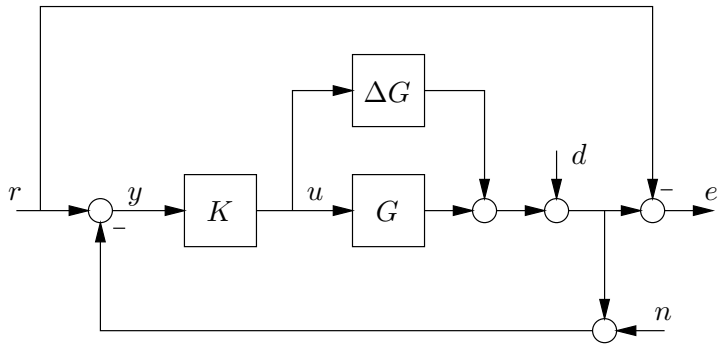


Figure 1.1: Simple block diagram of a controlled system

To facilitate the discussion, consider a simple representation of a controlled system in figure 1.1, where $K$ is the controller, $G$ is the plant to be controlled, $\Delta G$ represents additive plant model perturbations, $r$ is the reference input, $u$ is the plant input, $d$ describes external disturbances, $n$ represents sensor noise, $y$ is the signal that enters the controller, the so-called measurement output and $e$ is the tracking error [26].

Usually the controller $K$ must be designed such that the following objectives and constraints are fulfilled in some optimal form:

- **Stability:** The closed loop system must be stable
- **Tracking:** The tracking error $e$ should be zero
- **Disturbance rejection:** The disturbance $d$ should not affect the tracking error $e$
- **Sensor noise rejection:** The tracking error $e$ should not be influenced by the sensor noise $n$
- **Avoiding actuator saturation:** The actuator, which is assumed to be a part of $G$ should not be saturated by the input $u$

- **Robustness:** All aforementioned objectives and constraints - or at least the stability requirement - should be fulfilled to an acceptable level, even if the real plant may differ from the plant model $G$ by an amount $\Delta G$.

It is clear that all these requirements can only be fulfilled to some extent. Usually different requirements pose different demands on the control action and the final controller $K$ will be a compromise solution. Therefore it is important to quantify all the objectives and to weigh them in a suitable way. The robustness requirement generally reduces the achievable closed-loop performance as the controller can not be tuned only for one specific nominal plant model $G$, but the objectives must be satisfactorily fulfilled for the real plant, which may lie in a set of plant models given as $G + \Delta G$. In general, model perturbations and uncertainties described by $\Delta G$ can never be avoided due to the following facts:

- **Unmodelled dynamics:** Usually control design requires model descriptions of reasonable size and complexity. In many cases linear, time-invariant models $G$ of reduced order are used for control design. High frequency dynamics or nonlinearities are usually approximated leading to imperfections in the model.

- **Time variance:** Plant dynamics undergo changes during operation. Varying environmental conditions (e.g., temperature changes or variation in the earth magnetic field during one orbit of a satellite) or the wear caused by aging influence the plant dynamics.

- **Varying loads:** Plant dynamics change depending on their load conditions. For example the inertia and the position of center of gravity of an aircraft change depending on the distribution of passengers, cargo and fuel.

- **Manufacturing variance:** If we consider a series of plants the control design is usually performed for one or several prototypes of them. However, in a series production there will always be manufacturing variances between the individual plants and the controller must cope with these.

- **Identification errors:** Even if the model structure includes all dynamics of the real process, there will always be errors in the identification of the model parameters due to limitations of the identification hardware and methods.

Considering robustness of SISO systems to gain variations, a first systematic control design approach was developed by Bode [16, 17]. At this time linear systems were described using transfer functions or frequency responses and it was straightforward to describe model uncertainties as deviations from some nominal frequency response. The Bode diagram has proven to be very useful to graphically design controllers and measures as gain and phase margins were introduced. A generalization of Bode's work, considering robustness to arbitrary plant variations, was developed by Horowitz [48], which founded the Quantitative Feedback Design Theory (QFT) [49].

In the 1960s, the state-space theory became more popular. Systems were then described by differential equations allowing new insights [52] and initiating the development of

many new control design techniques. The LQG method [51] is probably the best-known development of this time. The control design problem for a linear system with Gaussian disturbances is formulated as an optimization problem with quadratic constraints and it admits an analytical solution. The controller structure consists of a Kalman filter and a linear state feedback. In the case that all state variables can be measured, the LQG design offers very strong robustness properties with at least 60° of phase margin and infinite gain margin as shown in [70]. Unfortunately, this does not hold for the output feedback case, where a Kalman filter is used to estimate the system states [33]. An obvious idea to achieve robustness was to modify the Kalman filter in some way to recover the loop transfer of the case where all states are measured [69]. However, the modified Kalman filter may be completely nonoptimal as far as disturbance rejection is concerned. Conclusively, the classical approach of LQG controller design as described in [51] is not the proper way to incorporate performance requirements, disturbance rejection and robustness.

A paradigm shift in robust control was represented by the paper [89]. It was a starting point of the so called $H_\infty$ theory, which has become very popular in control theory. It allows to incorporate performance specifications, disturbance rejection and robustness requirements into one optimization problem and allows to generalize the successful classical design techniques in the frequency domain from SISO to MIMO systems. Solutions of the $H_\infty$ problem in state-space were presented in [35] marking a cornerstone in the robust control theory.

## 1.2 Linear Fractional Transformations (LFTs) in robust control

To perform robust $H_\infty$ controller synthesis it is essential to quantify the uncertainties in the process model, to define a precise mathematical formulation and to describe how these uncertainties enter into the system interconnection. In general two types of uncertainties are considered:

- Model imperfections due to neglected nonlinearities or unmodelled dynamics are usually represented as additive or multiplicative norm-bounded frequency domain errors with arbitrary phase, which are related to process components or the whole process. These are called *unstructured uncertainties*.

- Uncertainties in model parameters can be described more precisely. The process model usually contains structural information of how the uncertain parameters affect the process dynamics. This often allows to achieve better closed-loop performance than in the unstructured uncertainty case. If the structural information is ignored one may obtain a conservative approximation of the true error. Therefore parametric uncertainties can be handled more systematically and they are denoted as *structured uncertainties*.

In literature the terminology structured uncertainty is usually equivalent to parametric uncertainty. However, considering a system consisting of at least two coupled components

and each component admits an unstructured uncertainty description, then it is important to note that at system level the overall uncertainty description also becomes structured, although not parametric, uncertainty. Hence, in such cases one also may reduce conservativeness by considering this structure during robust stability/performance analysis or robust controller synthesis and the terminology structured uncertainty does not only stand for parametric uncertainty.

Mathematical models of uncertainties must be explicitly included in the process model. In general the resulting overall model must be transformed/rearranged into a special structure that admits the application of robust $H_\infty$ methods. *Linear Fractional Transformations (LFTs)* are a general, flexible and powerful way to represent uncertainty in systems. The idea of using LFT-based uncertainty descriptions (denoted as Linear Fractional Representations (LFRs)) is to keep separated what is known from what is unknown by expressing the process model as a feedback connection of a nominal plant and the uncertainty description. An LFR defines a set of process models and the real process is assumed to lie inside this model set. LFRs are suitable for the application of robust $H_\infty$ design methods.

LFT-based robust control was extensively studied during the last years (see [7, 29, 28, 13, 24, 77, 65, 72, 74, 92] and references therein). Successful applications can be found in various areas, as aeronautics [58], missile control [67], control of flexible structures [9], control of CD-players [31], high-precision positioning in IC-manufacturing [78] and many others.

## 1.3 Motivation and Thesis Structure

In principle, it is straightforward to transform a linear plant model including unstructured and structured uncertainties into an LFR. However, this transformation is not unique and the blindfold application of *ad-hoc* methods [77] generally leads to LFRs of high complexity. LFT-based robust control methods are numerically highly demanding and of high computational complexity, e.g., many methods are based on solving a large system of Linear Matrix Inequalities (LMIs) [19, 73]. Therefore the application of these methods is restricted to LFRs of reasonable complexity, otherwise the computation time will be unacceptable, the numerical errors will increase or the methods may even fail.

It is the aim of the present thesis to develop new techniques and to improve existing methods for the realization of LFRs with reduced complexity for parametric plant models. A plant model may include many uncertain parameters and these generally enter the model equations in a highly structured way. The same parameter may appear in many parts of the model and parametric expressions can be very complicated (e.g., resulting from rational parametric approximations of multidimensional tables in an aircraft model [84]). Therefore it is crucial to employ systematic and structure exploiting techniques to reduce the LFR complexity resulting from parametric uncertainty. Concerning unstructured uncertainties, these are usually employed to describe general imperfections of process components or the whole process and occur only once in a model description. Hence, the transformation of models including unstructured uncertainties to an LFR can generally be done by inspection and there is low potential to reduce the LFR complexity

resulting from unstructured uncertainty. To simplify the presentation, the low order LFR realization techniques developed in this thesis are presented exclusively for systems with structured/parametric uncertainties.

In figure 1.2 the design process of LFT-based robust control is sketched. Starting from a suitable representation of an uncertain parametric process model, the LFR construction procedure consists of three steps and the resulting LFR may be used for robust stability/performance analysis (e.g., $\mu$-analysis [92], Integral Quadratic Constraint (IQC)-based analysis [61],...) or robust controller synthesis (e.g., $\mu$-synthesis, Linear Parameter Varying (LPV) control,...).

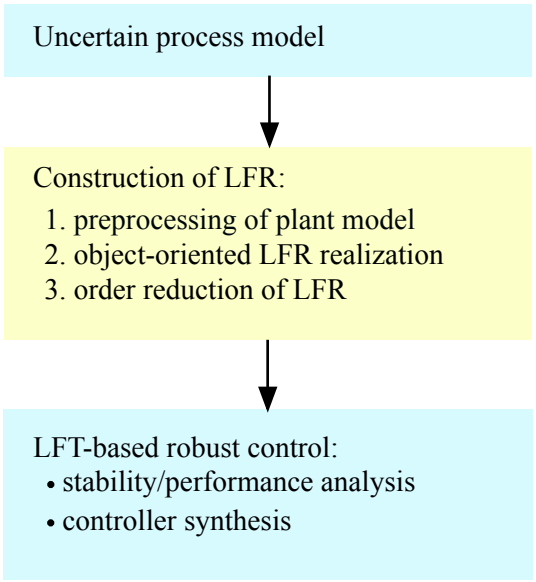


Figure 1.2: LFT-based robust control design process

The object-oriented LFR realization procedure (step 2) [77, 41, 42] is probably the most flexible and powerful way to transform a parametric model into an LFR. It can easily be automated and can be applied to arbitrary rational parametric uncertainties. However, the sole application of this method does not consider any structure in the parametric dependence of the model and may yield LFRs of high complexity (i.e., high order). To overcome this limitation a preliminary preprocessing step [44, 24, 83, 43] can be performed to find equivalent representations of the plant model that allow the generation of LFRs with low complexity using the object-oriented LFR realization approach. In a final step, order reduction can be performed to further reduce the complexity by removing redundant parts of the LFR. Alternatively, the LFR may be simplified by calculating approximate LFRs.

In this thesis, new techniques and improvements of existing methods for each of the three steps of the LFR construction procedure are presented. The applicability of the improved overall procedure is demonstrated by realizing LFRs of possibly minimal complexity for highly complex parametric uncertain systems.

In chapter 2 the class of linear parametric models considered in this thesis is defined.

Two ways how to derive such models from more general nonlinear parametric models are presented. A new representation called generalized LFR is introduced to overcome a basic limitation of the standard LFR to represent arbitrary rational parametric expressions. Formulas for the related object-oriented LFR realization technique are developed. Finally, procedures are presented for the realization of LFRs for generalized parametric state-space systems.

Chapter 3 presents an overview of symbolic decomposition methods suitable as preprocessing tools. New techniques are proposed and several enhancements of existing symbolic preprocessing methods are shown. All these symbolic methods, together with the generalized LFR and enhanced numerical order reduction techniques for LFRs are supported by the newly developed version 2 of the LFR-toolbox for MATLAB, which is briefly described in chapter 4.

Chapter 5 describes the application of the developed low order LFR realization techniques to the RCAM (Research Civil Aircraft Model), which is one of the most complicated parametric models available in literature. The order of the generated LFR was reduced by about 60% compared to former realizations reported in literature. This low order LFR allows to apply $\mu$-analysis techniques for the whole flight envelope, which was not possible before with the high-order models. Closed-loop robust stability was analyzed for 12 different controllers and some of them were indicated to not achieve closed-loop stability for the uncertain plant model. This could be confirmed by a complementary optimization-based worst-case search, where worst-case plant models leading to unstable closed-loop systems were identified, for these cases.

In chapter 6 an LFR of minimal complexity is realized for an uncertain parametric model describing the lateral dynamics of a vehicle. This LFR is employed for the realization of two a robust vehicle steering controllers. One controller is generated using the $\mu$-synthesis technique and the second controller is an automatically generated gain-scheduled controller based on robust Linear Parameter Varying (LPV) control design techniques. Both controllers provide better robust performance properties than recently developed controllers for this vehicle model.

# 2 Generalized Linear Fractional Representation (LFR) for parametric uncertain systems

## 2.1 From nonlinear to linear parametric models

Many physical dynamical systems can be described by a continuous-time, nonlinear, parametric plant of implicit form

$$\begin{aligned} 0 &= f(\dot{x}, x, u, \delta) \\ y &= g(x, u, \delta) \end{aligned} \tag{2.1}$$

where $f \in \mathbb{R}^n$ and $g \in \mathbb{R}^p$ are vector functions, $x \in \mathbb{R}^n$ is the state vector with $\dot{x}$ as its time derivative, $u \in \mathbb{R}^m$ is the input vector and $\delta \in \mathbb{R}^k$ is a vector of physical parameters. The parameter values are bounded by their minimum and maximum values and the admissible parameter value set is defined as

$$\Pi = \{\delta : \delta_i \in [\delta_{i,\min}, \delta_{i,\max}], i = 1, \ldots, k\} \tag{2.2}$$

Each parameter $\delta_i$ may be described in different units and the ranges of admissible values may differ by orders of magnitudes. For the application of LFT-based techniques for robust stability analysis and robust controller synthesis (e.g. $\mu$-analysis and $\mu$-synthesis) it is generally required that the uncertain parameters are normalized to lie in an $k$-dimensional cube centered at the origin.

To perform the normalization of parameters, $\delta_i$ can be replaced by a rational parametric function $n_i(\widetilde{\delta}_i)$ such that $\delta_i = n_i(\widetilde{\delta}_i)$ with $|\widetilde{\delta}_i| \leq 1$. The function $n_i(\widetilde{\delta}_i)$ is chosen such that, $\delta_{i,\min} = n_i(-1)$, $\delta_{i,\max} = n_i(1)$, $\delta_{i,\text{nom}} = n_i(0)$, where $\delta_{i,\text{nom}} \in [\delta_{i,\min}, \delta_{i,\max}]$ represents the nominal value of the parameter $\delta_i$. Hence the nominal system

$$\begin{aligned} 0 &= f(\dot{x}, x, u, \delta)|_{\delta=\delta_{\text{nom}}} \\ y &= g(x, u, \delta)|_{\delta=\delta_{\text{nom}}} \end{aligned}$$

can be represented as

$$\begin{aligned} 0 &= f(\dot{x}, x, u, n(\widetilde{\delta}))|_{\widetilde{\delta}=0} \\ y &= g(x, u, n(\widetilde{\delta}))|_{\widetilde{\delta}=0}. \end{aligned}$$

This is convenient because the nominal system corresponds now to a normalized system where all parameters are zero, i.e., $\widetilde{\delta} = 0$.

In robust control applications, non-linear parametric models of the form (2.1) are useful to perform parameter sensitivity studies (e.g., via simulations). However, the application of LFT-based robust control synthesis and analysis techniques requires a linear description or approximation of the nonlinear plant (2.1). Such a description can be obtained by applying Jacobian-based linearization or by reformulating the nonlinear plant in a quasi-Linear Parameter Varying (LPV) form. Both methods are briefly described in the following. For a comprehensive treatment see [68, 55, 59].

### 2.1.1 Jacobian-based linearization

Recall that a *trim point* $\dot{x}_t, x_{\text{trim}}, u_{\text{trim}}, y_{\text{trim}}, \delta_{\text{trim}}$ of the plant model (2.1) satisfies $0 = f(\dot{x}_{\text{trim}}, x_{\text{trim}}, u_{\text{trim}}, \delta_{\text{trim}})$, $y_{\text{trim}} = g(x_{\text{trim}}, u_{\text{trim}}, \delta_{\text{trim}})$ and if $\dot{x}_{\text{trim}} = 0$ it may be an equilibrium point of (2.1).

**Definition 2.1.** The functions $\dot{x}_{\text{trim}}(\delta), x_{\text{trim}}(\delta)$ and $u_{\text{trim}}(\delta)$ define a family of trim points for (2.1) on the set $\Pi$ if

$$\begin{aligned} 0 &= f(\dot{x}_{\text{trim}}(\delta), x_{\text{trim}}(\delta), u_{\text{trim}}(\delta), \delta), \quad \delta \in \Pi \\ y_{\text{trim}} &= g(x_{\text{trim}}(\delta), u_{\text{trim}}(\delta), \delta). \end{aligned} \tag{2.3}$$

The $n+p$ nonlinear equations (2.1) must be solved (trimming) to determine $n+p$ free components of these vectors selected from a total of $2n+m+p$ components. Note, that the values of the remaining $n+m$ components are set to values defining specific trim conditions.

A corresponding family of linearized parametric descriptor systems is defined by

$$\begin{aligned} E(\delta)\dot{x}_\delta &= A(\delta)x_\delta + B(\delta)u_\delta \\ y_\delta &= C(\delta)x_\delta + D(\delta)u_\delta \end{aligned} \tag{2.4}$$

where

$$\begin{aligned} E(\delta) &= -\frac{\partial f}{\partial \dot{x}}(\dot{x}_{\text{trim}}(\delta), x_{\text{trim}}(\delta), u_{\text{trim}}(\delta), \delta) \\ A(\delta) &= \frac{\partial f}{\partial x}(\dot{x}_{\text{trim}}(\delta), x_{\text{trim}}(\delta), u_{\text{trim}}(\delta), \delta) \\ B(\delta) &= \frac{\partial f}{\partial u}(\dot{x}_{\text{trim}}(\delta), x_{\text{trim}}(\delta), u_{\text{trim}}(\delta), \delta) \\ C(\delta) &= \frac{\partial g}{\partial x}(\dot{x}_{\text{trim}}(\delta), x_{\text{trim}}(\delta), u_{\text{trim}}(\delta), \delta) \\ D(\delta) &= \frac{\partial g}{\partial u}(\dot{x}_{\text{trim}}(\delta), x_{\text{trim}}(\delta), u_{\text{trim}}(\delta), \delta) \end{aligned} \tag{2.5}$$

and where

$$\dot{x}_\delta = \dot{x} - \dot{x}_{\text{trim}}(\delta),\ x_\delta = x - x_{\text{trim}}(\delta),\ u_\delta = u - u_{\text{trim}}(\delta) \text{ and } y_\delta = y - y_{\text{trim}}(\delta). \tag{2.6}$$

For each fixed value of $\delta \in \Pi$ the linearization (2.4) describes to local behavior of the nonlinear plant (2.1) about the corresponding trim point.

If the inverse of $E(\delta)$ exists it is possible to obtain an explicit parametric state-space system of the form

$$\begin{aligned} \dot{x}_\delta &= \widetilde{A}(\delta)x_\delta + \widetilde{B}(\delta)u_\delta \\ y_\delta &= C(\delta)x_\delta + D(\delta)u_\delta \end{aligned} \tag{2.7}$$

where $\widetilde{A}(\delta) = E(\delta)^{-1}A(\delta)$ and $\widetilde{B}(\delta) = E(\delta)^{-1}B(\delta)$.

The main limitation of the Jacobian-based linearization is that in general there exists no analytical solution for the functions $\dot{x}_{\text{trim}}(\delta)$, $x_{\text{trim}}(\delta)$, $u_{\text{trim}}(\delta)$, $y_{\text{trim}}(\delta)$. A common practice is to simply substitute these functions with their numerical values at the corresponding trim point, computed for the nominal value of $\delta$. The resulting Jacobian matrices in (2.5) (obtained for example using symbolic differentiation) depend explicitly on the parameters in $\delta$. However, in this case all the information about the parametric dependence of $\dot{x}_{\text{trim}}(\delta)$, $x_{\text{trim}}(\delta)$, $u_{\text{trim}}(\delta)$, $y_{\text{trim}}(\delta)$ is lost and the validity of the linear approximation (2.4) is strongly limited. To increase the validity of the linear model representation, a more sophisticated way is to calculate polynomial or rational approximations of these functions [84] based on physical knowledge of the parametric dependence in the entries of the matrices in (2.4).

### 2.1.2 Quasi-LPV models

For gain-scheduling control design a linear plant description may be obtained by rewriting the nonlinear model (2.1) in quasi-LPV form, if possible. The idea is to hide nonlinear terms, that may also depend on state variables, in newly defined parameters that can be measured and are employed as scheduling variables. This means that in various parts of the model, state variables must be relabelled, while they remain dynamical variables elsewhere. Consider for example the nonlinear system

$$\begin{aligned} \dot{x} &= A(x,\delta)x + B(x,\delta)u \\ y &= C(x,\delta)x + D(x,\delta)u, \end{aligned} \tag{2.8}$$

with $x(t)$ confined to some operating region $X \subset \mathbb{R}^n$. It is clear, that the solutions of (2.8) are a subset of the solutions of the system

$$\begin{aligned} \dot{x} &= A(\overline{\delta})x + B(\overline{\delta})u \\ y &= C(\overline{\delta})x + D(\overline{\delta})u, \end{aligned} \tag{2.9}$$

with $\overline{\delta} \in \Pi \times X$. Hence, it is possible to over-bound the nonlinear system (2.8) with a quasi-LPV system (2.9). This may introduce some conservativeness as no a priori knowledge of the arbitrarily varying parameters in $\overline{\delta}$ is assumed and the explicit dependence on the state variables is not exploited.

Both methods, the Jacobian-based linearization and the quasi-LPV approach achieve linearity in the plant description, however this is done in quite different ways.

In general, for the realization of LFRs, only rational parametric dependence is allowed in the entries of the linear plant descriptions and therefore finally all remaining nonlinearities in these matrices must be approximated by rational functions.

## 2.2 Standard LFT

This section introduces a new matrix function: the linear fractional transformation (LFT).

**Definition 2.2.** Consider the partitioned matrix

$$M = \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array}\right] \in \mathbb{C}^{(p_1+p_2)\times(m_1+m_2)} \tag{2.10}$$

and the two complex matrices $\Delta \in \mathbb{C}^{m_1 \times p_1}$ and $\Omega \in \mathbb{C}^{m_2 \times p_2}$. The *upper* LFT with respect to $M$ and $\Delta$ is defined with

$$\mathcal{F}_u(M, \Delta) = M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12} \tag{2.11}$$

provided the inverse $(I - M_{11}\Delta)^{-1}$ exists. A *lower* LFT with respect to $M$ and $\Omega$ is defined as

$$\mathcal{F}_l(M, \Omega) = M_{11} + M_{12}\Omega(I - M_{22}\Omega)^{-1}M_{21} \tag{2.12}$$

provided the inverse $(I - M_{22}\Omega)^{-1}$ exists. Furthermore, let

$$\begin{aligned} z &= M_{11}w + M_{12}u \\ y &= M_{21}w + M_{22}u \\ w &= \Delta z \end{aligned} \tag{2.13}$$

and

$$\begin{aligned} z &= M_{11}w + M_{12}u \\ y &= M_{21}w + M_{22}u \\ u &= \Omega y \end{aligned} \tag{2.14}$$

be the equation-based representations of the upper LFT and lower LFT, respectively.

The definition of *upper* and *lower* LFT should be clear from the diagram representations of $\mathcal{F}_u(M, \Delta)$ and $\mathcal{F}_l(M, \Omega)$ in figure 2.1.

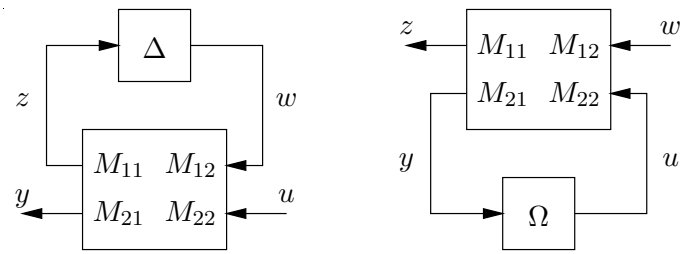


Figure 2.1: Diagram representations of $\mathcal{F}_u(M, \Delta)$ and $\mathcal{F}_l(M, \Omega)$

The *upper* LFT describes the relation between $y$ and $u$ after closing the *upper* loop, i.e. $w = \Delta z$ and $y = \mathcal{F}_u(M, \Delta)u$. Similarly, the *lower* LFT describes the relation between $z$ and $w$ after closing the *lower* loop, i.e. $u = \Omega y$ and $z = \mathcal{F}_l(M, \Omega)w$. In the literature, usually the upper LFT is used to represent uncertainty in plant models and therefore this representation will be employed in the following. Furthermore, the short term LFR (Linear Fractional Representation) will be used to denote the upper LFT representation of an uncertain parametric model.

In an LFR, $M_{22}$ typically stands for the nominal model (i.e., corresponds to $\Delta = 0$), while $M_{11}$, $M_{12}$ and $M_{21}$ describe the structure of how the uncertainty affects the nominal model. Hence the LFR can be employed to represent an uncertain model as a feedback interconnection of the model uncertainties described by $\Delta$ and a nominal plant model.

**Remark 2.1.** The LFT is a direct generalization of the notion of state-space realizations, since any LTI system can be represented as an LFR with

$$M = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right], \quad \Delta = \tfrac{1}{s} I, \tag{2.15}$$

and the transfer function between $y$ and $u$ (see figure 2.2), is given by

$$\mathcal{F}_u(M, \Delta) = D + C\tfrac{1}{s}(I - A\tfrac{1}{s})^{-1}B = D + C(sI - A)^{-1}B.$$

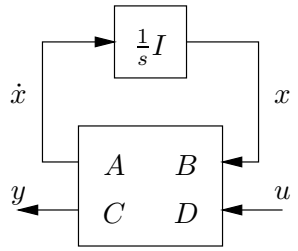


Figure 2.2: LFR of state-space system

**LFR realization problem:** Given a $p_2 \times m_2$ real matrix $G(\delta)$ depending rationally on $k$ parameters grouped into the real vector $\delta = (\delta_1, \ldots, \delta_k)$. Find matrices $M, \Delta$ such that

$$G(\delta) = \mathcal{F}_u(M, \Delta) \tag{2.16}$$

where $M \in \mathbb{R}^{(p_1+p_2)\times(p_1+m_2)}$ and

$$\Delta = \operatorname{diag}(\delta_1 I_{r_1}, \ldots, \delta_k I_{r_k}) \tag{2.17}$$

with lowest possible order $r$, defined as $r = p_1 = \sum_{i=1}^{k} r_i$. Note, that the matrix $\Delta$ has block-diagonal structure, motivating the terminology of structured uncertainty.

Hence the ultimate goal is to find LFRs of minimal order. However, representing a parameter dependent matrix as an LFR is basically equivalent to a multidimensional realization problem [18], where a minimal realization theory is still lacking. For 1-dimensional systems as (2.15) (or equivalently, systems with only one diagonal block in $\Delta$) a minimal realization can always be obtained by eliminating unobservable and uncontrollable parts from the state-space representation (2.15). Unfortunately for multidimensional systems similar techniques cannot be employed (see [18] and section 3.1 for a brief description).

**Example 2.1.** Consider the simple scalar parametric expression $G(\delta) = \delta_i$, which can be directly realized as

$$\delta_i = \mathcal{F}_u(M, \Delta) = \mathcal{F}_u\left(\left[\begin{array}{c|c} 0 & 1 \\ \hline 1 & 0 \end{array}\right], \delta_i\right) \tag{2.18}$$

and will be denoted as *elementary* LFR. Starting from elementary LFRs for uncertain parameters, an object-oriented approach [77] can be employed to realize an LFR for a rational parametric matrix $G(\delta)$. However, there is a basic limitation for the realization of arbitrary rational matrices via standard LFRs. As an example, the expression $G(\delta) = 1/\delta_i$ cannot be represented as an LFR with $\Delta$ of the form (2.17). This is equivalent to the case that it is not possible to represent non-proper transfer functions as for example $G(s) = s$ in standard state-space form (2.15). One way to represent $G(\delta) = 1/\delta_i$ as an LFR is to use in (2.18) $\Delta = 1/\delta_i$. However, $G(\delta) = \delta_i + 1/\delta_i$ does not have an LFR as both $\delta_i$ and $1/\delta_i$ enter the expression.

In practice, to overcome the above difficulties, a normalization of uncertainties is performed. Assuming, for example that $\delta_i \in [\delta_{i,\min}, \delta_{i,\max}]$ and $\delta_{i,\text{nom}} := (\delta_{i,\max} + \delta_{i,\min})/2 \neq 0$, then with $\delta_{i,\text{sl}} := (\delta_{i,\max} - \delta_{i,\min})/2$ one obtains $\delta_i = \delta_{i,\text{nom}} + \delta_{i,\text{sl}}\overline{\delta}_i$, with $\overline{\delta}_i \in [-1, 1]$. With this normalization, $\overline{G}(\overline{\delta}) := 1/(\delta_{i,\text{nom}} + \delta_{i,\text{sl}}\overline{\delta}_i) = 1/\delta_i$ can be represented as

$$\overline{G}(\overline{\delta}) = \mathcal{F}_u\left(\left[\begin{array}{c|c} -\delta_{i,\text{sl}}\delta_{i,\text{nom}}^{-1} & -\delta_{i,\text{sl}}\delta_{i,\text{nom}}^{-1} \\ \hline \delta_{i,\text{nom}}^{-1} & \delta_{i,\text{nom}}^{-1} \end{array}\right], \overline{\delta}_i\right).$$

A negative aspect of this approach is that the normalization must be performed as a preliminary symbolic operation before the LFR realization and this often leads to an increase of the overall order of the LFR (see [24] and next example).

**Example 2.2.** Consider $G(\delta) = \delta_i^2$ and its normalized and expanded form $\overline{G}(\overline{\delta}) = \delta_{i,\text{nom}}^2 + 2\delta_{i,\text{nom}}\delta_{i,\text{sl}}\overline{\delta}_i + \delta_{i,\text{sl}}^2\overline{\delta}_i^2$. The object-oriented LFR realization procedure of [77] yields an LFR of minimal order 2 for $G(\delta)$, whereas an LFR of order 3 is obtained for $\overline{G}(\overline{\delta})$, where this procedure generates an LFR by interconnecting a second order LFR for $\delta_{i,\text{sl}}^2\overline{\delta}_i^2$ with a first order LFR for $2\delta_{i,\text{nom}}\delta_{i,\text{sl}}\overline{\delta}_i$ and the constant term $\delta_{i,\text{nom}}^2$. The result is a third order LFR.

This simple example clearly illustrates that it is desirable to perform the normalization as the last step in any LFR generation. This is particularly important when employing symbolic preprocessing techniques as described in section 3 or in [24], as these techniques may start with an expansion of the symbolic expressions in $G(\delta)$. Expanding normalized symbolic functions usually yields large expressions, which may result in LFRs of high order, as this artificially introduced complexity cannot be fully recovered by the symbolic factorization and decomposition techniques.

## 2.3 Generalized LFT

In this section the generalized LFT is introduced, which is a natural extension of the standard LFT. It may also be called descriptor LFT in analogy to the generalized state space realizations via descriptor systems [25].

### 2.3.1 Definition

The generalized upper LFT is defined with the partitioned matrix

$$M = \left[\begin{array}{c|c|c} M_{10} & M_{11} & M_{12} \\ \hline 0 & M_{21} & M_{22} \end{array}\right] \tag{2.19}$$

as

$$\mathcal{F}_u(M, \Delta) = M_{22} + M_{21}\Delta(M_{10} - M_{11}\Delta)^{-1}M_{12}, \tag{2.20}$$

where the square submatrix $M_{10}$ is allowed to be generally singular but the inverse $(M_{10} - M_{11}\Delta)^{-1}$ must exist. For $\Delta$ we assume the more general structure

$$\Delta = \operatorname{diag}(\delta_0 I_{r_0}, \delta_1 I_{r_1}, \ldots, \delta_k I_{r_k}), \tag{2.21}$$

where $\delta_0$ is a nonzero constant, which is set to 1 in the following. The equation-based definition of the generalized upper LFT is given as

$$\begin{aligned} M_{10}z &= M_{11}w + M_{12}u \\ y &= M_{21}w + M_{22}u \\ w &= \Delta z, \end{aligned} \tag{2.22}$$

and the order of a generalized LFR $(M, \Delta)$ is defined as $r = \sum_{i=1}^{k} r_i$. Note that the standard LFR (2.11) corresponds to $M_{10} = I$ and $r_0 = 0$.

With the generalized LFR we can represent $G(\delta) = 1/\delta$ as

$$\frac{1}{\delta} = \mathcal{F}_u\left(\left[\begin{array}{cc|cc|c} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & -1 & 0 & 0 \end{array}\right], \begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix}\right). \tag{2.23}$$

There is a clear analogy to generalized state-space systems, where for example the differentiator $G(s) = s$ is described in descriptor form as

$$\begin{aligned} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} -1 & 0 \end{bmatrix} x, \end{aligned} \tag{2.24}$$

consisting of a combination of differential ($\dot{x}_2 = x_1$) and algebraic ($x_2 = u$) equations (DAE system).

**Remark 2.2.** In the generalized LFR the algebraic equations are formally described as an additional dimension - the block $I_{r_0}$ in $\Delta$ - in a multidimensional system representation. The generalized LFR may seem to be more complex as the standard LFR, as for example a 2-dimensional system representation (2.23) (two diagonal blocks in $\Delta$) is used to represent $G(\delta) = 1/\delta$. However, it will be shown that the object-oriented LFR realization procedure based on generalized LFRs is actually simpler than for standard LFRs, as explicit matrix inversions can be avoided. Furthermore, it will be shown that the artificially introduced additional dimension ($I_{r_0}$ in $\Delta$) will usually vanish after a normalization of the generalized LFR and finally a standard LFR is obtained. It is important to emphasize that the generalized LFR allows to perform the normalization step as the last step in the LFR realization procedure and this generally leads to LFRs of lower order as a preliminary symbolic normalization is avoided.

### 2.3.2 Algebraic Properties

Since LFRs are similar to transfer-function matrix representations of linear state-space systems, the basic matrix operations like addition/subtraction, multiplication, transposition, inversion as well as column/row concatenation correspond to similar operations performed on the transfer-function matrices of linear systems. These are operations underlying the methods used to generate LFRs of parametric matrices [77, 41, 42]. The following results for generalized LFRs generalize similar results for standard LFRs. To simplify notation, in the following lemmas, the partitioned matrix $M$ is written as

$$M = \left[\begin{array}{c|c|c} E & A & B \\ \hline 0 & C & D \end{array}\right],$$

where $E = M_{10}$, $A = M_{11}$, $B = M_{12}$, $C = M_{21}$ and $D = M_{22}$.

**Lemma 2.1.** (without proofs) Let $M_1$, $M_2$ and $M$ be partitioned matrices

$$M = \left[\begin{array}{c|c|c} E & A & B \\ \hline 0 & C & D \end{array}\right], M_1 = \left[\begin{array}{c|c|c} E_1 & A_1 & B_1 \\ \hline 0 & C_1 & D_1 \end{array}\right], M_2 = \left[\begin{array}{c|c|c} E_2 & A_2 & B_2 \\ \hline 0 & C_2 & D_2 \end{array}\right]$$

and let $\Delta$, $\Delta_1$, $\Delta_2$ be the corresponding uncertainty matrices.

(i) Parallel connection:

$$\mathcal{F}_u(M, \Delta) := \mathcal{F}_u(M_1, \Delta_1) \pm \mathcal{F}_u(M_2, \Delta_2),$$

with

$$M = \left[\begin{array}{cc|cc|c} E_1 & 0 & A_1 & 0 & B_1 \\ 0 & E_2 & 0 & A_2 & \pm B_2 \\ \hline 0 & 0 & C_1 & C_2 & D_1 \pm D_2 \end{array}\right], \Delta = \left[\begin{array}{cc} \Delta_1 & 0 \\ 0 & \Delta_2 \end{array}\right].$$

(ii) Series/Cascade connection:

$$\mathcal{F}_u(M, \Delta) := \mathcal{F}_u(M_1, \Delta_1)\mathcal{F}_u(M_2, \Delta_2),$$

with

$$M = \left[\begin{array}{cc|cc|c} E_1 & 0 & A_1 & B_1C_2 & B_1D_2 \\ 0 & E_2 & 0 & A_2 & B_2 \\ \hline 0 & 0 & C_1 & D_1C_2 & D_1D_2 \end{array}\right], \Delta = \left[\begin{array}{cc} \Delta_1 & 0 \\ 0 & \Delta_2 \end{array}\right].$$

(iii) Column concatenation:

$$\mathcal{F}_u(M, \Delta) := \left[\begin{array}{cc} \mathcal{F}_u(M_1, \Delta_1) & \mathcal{F}_u(M_2, \Delta_2) \end{array}\right],$$

with

$$M = \left[\begin{array}{cc|cc|cc} E_1 & 0 & A_1 & 0 & B_1 & 0 \\ 0 & E_2 & 0 & A_2 & 0 & B_2 \\ \hline 0 & 0 & C_1 & C_2 & D_1 & D_2 \end{array}\right], \Delta = \left[\begin{array}{cc} \Delta_1 & 0 \\ 0 & \Delta_2 \end{array}\right].$$

(iv) Row concatenation:

$$\mathcal{F}_u(M, \Delta) := \left[\begin{array}{c} \mathcal{F}_u(M_1, \Delta_1) \\ \mathcal{F}_u(M_2, \Delta_2) \end{array}\right],$$

with

$$M = \left[\begin{array}{cc|cc|c} E_1 & 0 & A_1 & 0 & B_1 \\ 0 & E_2 & 0 & A_2 & B_2 \\ \hline 0 & 0 & C_1 & 0 & D_1 \\ 0 & 0 & 0 & C_2 & D_2 \end{array}\right], \Delta = \left[\begin{array}{cc} \Delta_1 & 0 \\ 0 & \Delta_2 \end{array}\right].$$

(v) Transposition:

$$\mathcal{F}_u(M_{tr}, \Delta_{tr}) := \mathcal{F}_u(M, \Delta)^T,$$

with

$$M_{tr} = \left[\begin{array}{c|c|c} E^T & A^T & C^T \\ \hline 0 & B^T & D^T \end{array}\right], \Delta_{tr} = \Delta^T.$$

(vi) Consider

$$\left[\begin{array}{cc} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{array}\right] = \mathcal{F}_u\left(\left[\begin{array}{c|c|cc} \tilde{E} & \tilde{A} & \tilde{B}_1 & \tilde{B}_2 \\ \hline 0 & \tilde{C}_1 & \tilde{D}_{11} & \tilde{D}_{12} \\ 0 & \tilde{C}_2 & \tilde{D}_{21} & \tilde{D}_{22} \end{array}\right], \tilde{\Delta}\right).$$

Then

$$\mathcal{F}_u(M, \Delta) := \mathcal{F}_u\left(\left[\begin{array}{c|c|c} E & A(\delta) & B(\delta) \\ \hline 0 & C(\delta) & D(\delta) \end{array}\right], \overline{\Delta}\right),$$

with

$$M = \left[\begin{array}{cc|c|c} \tilde{E} & 0 & \tilde{A} & \tilde{B}_1 & \tilde{B}_2 \\ 0 & E & \tilde{C}_1 & \tilde{D}_{11} & \tilde{D}_{12} \\ \hline 0 & 0 & \tilde{C}_2 & \tilde{D}_{21} & \tilde{D}_{22} \end{array}\right], \Delta = \left[\begin{array}{cc} \tilde{\Delta} & 0 \\ 0 & \overline{\Delta} \end{array}\right].$$

**Lemma 2.2.** Similarity transformations: Let $Q$ and $Z$ be invertible matrices such that $Z\Delta = \Delta Z$. Then

$$\mathcal{F}_u(M, \Delta) = \mathcal{F}_u(\widetilde{M}, \widetilde{\Delta})$$

where

$$\widetilde{M} = \left[\begin{array}{c|c|c} QEZ & QAZ & QB \\ \hline 0 & CZ & D \end{array}\right], \ \widetilde{\Delta} = \Delta$$

**Proof.**

$$\mathcal{F}_u(\widetilde{M}, \widetilde{\Delta}) = D + CZ\Delta(QEZ - QAZ\Delta)^{-1}QB$$

with commutativity of $Z$ and $\Delta$ we obtain

$$\mathcal{F}_u(\widetilde{M}, \widetilde{\Delta}) = D + C\Delta Z(QEZ - QA\Delta Z)^{-1}QB,$$

and moving $Z$ and $Q$ into the parenthesis finally yields

$$\mathcal{F}_u(\widetilde{M}, \widetilde{\Delta}) = D + C\Delta(Q^{-1}QEZZ^{-1} - Q^{-1}QA\Delta ZZ^{-1})^{-1}B = \mathcal{F}_u(M, \Delta).$$

■

In the next lemma a more general class of transformations including similarity transformations are considered. To define this transformation the equation based representation on a generalized LFR (2.22) is formally rewritten as

$$\begin{bmatrix} 0 \\ y \end{bmatrix} = \begin{bmatrix} A\Delta - E & B \\ C\Delta & D \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix},$$

where $w$ is substituted by $w = \Delta z$.

**Lemma 2.3.** Let $Q$ and $Z$ be invertible matrices such that $Z\Delta = \Delta Z$. Furthermore $Q_k$ and $Z_k$ are matrices such that $Q_k E = 0$ and $EZ_k = 0$, i.e., $\mathrm{im}(Q_k^T) \subset \ker(E^T)$ and $\mathrm{im}(Z_k) \subset \ker(E)$. Then

$$\begin{aligned} \begin{bmatrix} 0 \\ y \end{bmatrix} &= \begin{bmatrix} A\Delta - E & B \\ C\Delta & D \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \\ &= \begin{bmatrix} Q & 0 \\ Q_k & I \end{bmatrix} \begin{bmatrix} A\Delta - E & B \\ C\Delta & D \end{bmatrix} \begin{bmatrix} Z & Z_k \\ 0 & I \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}. \end{aligned} \tag{2.25}$$

**Proof.** The matrices $Q$ and $Z$ define a similarity transformation which was shown in Lemma 2.2. Therefore in the proof only the trivial case $Q = I$ and $Z = I$ is considered.

First, the transformation related to $Q_k$ is proven and without loss of generality it is assumed that $Z_k = 0$. By definition $Q_k Ez = 0$ and therefore $Q_k(A\Delta z + Bu) = 0$. The transformation in (2.25) simply describes an addition of $Q_k(A\Delta z + Bu) = 0$ to the output equation, i.e.,

$$\begin{aligned} \begin{bmatrix} 0 \\ y \end{bmatrix} &= \begin{bmatrix} I & 0 \\ Q_k & I \end{bmatrix} \begin{bmatrix} A\Delta - E & B \\ C\Delta & D \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix} \\ &= \begin{bmatrix} A\Delta - E & B \\ Q_k A\Delta + C\Delta & Q_k B + D \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix}. \end{aligned}$$

and $y = (Q_k A\Delta + C\Delta)z + (Q_k B + D)u = C\Delta z + Du + Q_k(A\Delta z + Bu) = C\Delta z + Du$.

The dual result for a transformation with $Z_k$, assuming that $Q_k = 0$, follows by considering the transposed system

$$\begin{bmatrix} 0 \\ \overline{y} \end{bmatrix} = \begin{bmatrix} \Delta^T A^T - E^T & \Delta^T C^T \\ Z_k^T \Delta^T A^T - Z_k^T E^T + B^T & D^T + Z_k^T \Delta^T C^T \end{bmatrix} \begin{bmatrix} \overline{z} \\ \overline{u} \end{bmatrix},$$

where by definition $Z_k^T E^T \overline{z} = 0$ and therefore $Z_k^T(\Delta^T A^T \overline{z} + \Delta^T C^T \overline{u}) = 0$.

■

**Lemma 2.4.** Permutation: Let $T$ be an orthogonal matrix, i.e. $TT^T = I$. Furthermore each row of $T$ consists only of zeros except one element, which is 1. Then for a diagonal matrix $\Delta$

$$\mathcal{F}_u(M, \Delta) = \mathcal{F}_u(\widetilde{M}, \widetilde{\Delta}),$$

with

$$\widetilde{M} = \left[\begin{array}{c|c|c} T^T E T & T^T A T & T^T B \\ \hline 0 & CT & D \end{array}\right], \ \widetilde{\Delta} = T^T \Delta T,$$

and $\widetilde{\Delta}$ is a diagonal matrix with the same entries as in $\Delta$, but the order of the entries is permuted.

**Proof.** The permutation property of the transformation should be clear and it is also obvious that

$$\mathcal{F}_u(\widetilde{M}, \widetilde{\Delta}) = D + CTT^T \Delta T(T^T E T + T^T A T T^T \Delta T)^{-1} T^T B = \mathcal{F}_u(M, \Delta).$$

■

**Lemma 2.5.** Inversion: Suppose $\mathcal{F}_u(M, \Delta)$ is a $p \times p$ invertible matrix with

$$M = \left[\begin{array}{c|c|c} E & A & B \\ \hline 0 & C & D \end{array}\right]$$

then

$$\mathcal{F}_u(M_{inv}, \Delta_{inv}) := \mathcal{F}_u(M, \Delta)^{-1},$$

with

$$M_{inv} = \left[\begin{array}{cc|cc|c} 0 & 0 & D & C & I_p \\ 0 & E & B & A & 0 \\ \hline 0 & 0 & -I_p & 0 & 0 \end{array}\right], \ \Delta_{inv} = \begin{bmatrix} I_p & 0 \\ 0 & \Delta \end{bmatrix}. \tag{2.26}$$

**Proof.** To show that (2.26) represents a generalized LFR for the inverse system, the product $\mathcal{F}_u(M_I, \Delta_I) = \mathcal{F}_u(M, \Delta)\mathcal{F}_u(M_{\text{inv}}, \Delta_{\text{inv}})$ is calculated using (ii) of Lemma 2.1 with

$$M_I = \left[\begin{array}{ccc|ccc|c} 0 & 0 & 0 & D & C & C & D \\ 0 & E & 0 & B & A & 0 & 0 \\ 0 & 0 & E & 0 & 0 & A & B \\ \hline 0 & 0 & 0 & -I_p & 0 & 0 & 0 \end{array}\right], \ \Delta_I = \begin{bmatrix} I_p & 0 & 0 \\ 0 & \Delta & 0 \\ 0 & 0 & \Delta \end{bmatrix}. \tag{2.27}$$

Applying the transformation defined in Lemma 2.3 with

$$Q = \begin{bmatrix} I_p & 0 & 0 \\ 0 & I & I \\ 0 & 0 & I \end{bmatrix}, Q_k = \begin{bmatrix} 0_p & 0 & 0 \end{bmatrix}, Z = \begin{bmatrix} I_p & 0 & 0 \\ 0 & I & -I \\ 0 & 0 & I \end{bmatrix}, Z_k = \begin{bmatrix} -I_p \\ 0 \\ 0 \end{bmatrix},$$

to (2.27) yields $\mathcal{F}_u(M_I, \Delta_I) = \mathcal{F}_u(\widetilde{M}_I, \widetilde{\Delta}_I)$ with

$$\widetilde{M}_I = \left[\begin{array}{ccc|ccc|c} 0 & 0 & 0 & D & C & 0 & 0 \\ 0 & E & 0 & B & A & 0 & 0 \\ 0 & 0 & E & 0 & 0 & A & B \\ \hline 0 & 0 & 0 & -I_p & 0 & 0 & I_p \end{array}\right], \widetilde{\Delta}_I = \Delta_I. \tag{2.28}$$

From a direct evaluation of $\mathcal{F}_u(\widetilde{M}_I, \widetilde{\Delta}_I)$ it follows $\mathcal{F}_u(\widetilde{M}_I, \widetilde{\Delta}_I) = I_p$.

■

Note, that by using a generalized LFR, its inverse (see (2.26)) can be determined as a generalized LFR in terms of the original matrices, without any explicit matrix inversion.

**Corollary 2.1.** If $D$ is invertible, an alternative expression for $M_{inv}$ and $\Delta_{inv}$, involving inversion of $D$ may be obtained. Therefore one may write the explicit relation defined by $M_{inv}$, $\Delta_{inv}$ as

$$\begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} D & C \\ B & A \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} I_p \\ 0 \end{bmatrix} u,$$
$$y = -I_p w_1, \tag{2.29}$$
$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & \Delta \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

From (2.29) it can be seen that $z_1 = w_1$ and one can solve for $w_1$ as $w_1 = -D^{-1}Cw_2 - D^{-1}u$. Substituting $w_1$ in (2.29) finally yields $M_{inv}$ and $\Delta_{inv}$ as

$$M_{inv} = \left[\begin{array}{c|c|c} E & A - BD^{-1}C & -BD^{-1} \\ \hline 0 & D^{-1}C & D^{-1} \end{array}\right], \Delta_{inv} = \Delta.$$

It is well known that for any rational parametric matrix $G(\delta)$, it is possible to find a left (right) polynomial fractional representation, with $G(\delta) = D^{-1}(\delta)N(\delta)$ ($G(\delta) = \widetilde{N}(\delta)\widetilde{D}^{-1}(\delta)$), where $N(\delta)$, $D(\delta)$ ($\widetilde{N}(\delta)$, $\widetilde{D}(\delta)$) are multivariate polynomial matrices. It is possible to express such fractional representations in terms of the underlying LFRs of the factors. The following results are particularly useful when realizing LFRs for rational parametric matrices in terms of polynomial factorizations.

**Lemma 2.6.** Let $\mathcal{F}_u(M, \Delta) = \begin{bmatrix} N(\delta) & D(\delta) \end{bmatrix}$ be defined with

$$M = \left[\begin{array}{c|c|cc} E & A & B_N & B_D \\ \hline 0 & C & D_N & D_D \end{array}\right]$$

and assume that $D(\delta)$ is a $p \times p$ invertible matrix. Then

$$\mathcal{F}_u(M_{\text{lf}}, \Delta_{\text{lf}}) = D^{-1}(\delta)N(\delta),$$

with

$$M_{\text{lf}} = \left[\begin{array}{cc|ccc} 0 & 0 & D_D & C & D_N \\ 0 & E & B_D & A & B_N \\ \hline 0 & 0 & -I_p & 0 & 0 \end{array}\right], \; \Delta_{\text{lf}} = \begin{bmatrix} I_p & 0 \\ 0 & \Delta \end{bmatrix}. \tag{2.30}$$

If $D_D$ is invertible, alternative representations of $M_{\text{lf}}$ and $\Delta_{\text{lf}}$ are

$$M_{\text{lf}} = \left[\begin{array}{c|c|c} E & A - B_D D_D^{-1} C & B_N - B_D D_D^{-1} D_N \\ \hline 0 & D_D^{-1} C & D_D^{-1} D_N \end{array}\right], \; \Delta_{\text{lf}} = \Delta.$$

**Proof.** Using Lemma 2.5 and (ii) of Lemma 2.1, one obtains

$$\mathcal{F}_u(M_M, \Delta_M) = D^{-1}(\delta)N(\delta),$$

where

$$M_M = \left[\begin{array}{c|c|c} E_M & A_M & B_M \\ \hline 0 & C_M & D_M \end{array}\right] = \left[\begin{array}{ccc|ccc|c} 0 & 0 & 0 & D_D & C & C & D_N \\ 0 & E & 0 & B_D & A & 0 & 0 \\ 0 & 0 & E & 0 & 0 & A & B_N \\ \hline 0 & 0 & 0 & -I_p & 0 & 0 & 0 \end{array}\right],$$

$$\Delta_M = \begin{bmatrix} I_p & 0 & 0 \\ 0 & \Delta & 0 \\ 0 & 0 & \Delta \end{bmatrix}.$$

Now, a similarity transformation is applied to $M_M$, yielding a transformed matrix $\widetilde{M}_M$. Consider the transformation matrices $Q$ and $Z$ given by

$$Q = \begin{bmatrix} I_p & 0 & 0 \\ 0 & I & I \\ 0 & 0 & I \end{bmatrix}, Z = \begin{bmatrix} I_p & 0 & 0 \\ 0 & I & -I \\ 0 & 0 & I \end{bmatrix},$$

with the identity matrix $I$ of the same size as $\Delta$. It is easy to see that $Z\Delta_M = \Delta_M Z$, thus applying Lemma 2.2 yields

$$\widetilde{M}_M = \left[\begin{array}{c|c|c} QE_MZ & QA_MZ & QB_M \\ \hline 0 & C_MZ & D_M \end{array}\right] = \left[\begin{array}{ccc|ccc|c} 0 & 0 & 0 & D_D & C & 0 & D_N \\ 0 & E & 0 & B_D & A & 0 & B_N \\ 0 & 0 & E & 0 & 0 & A & B_N \\ \hline 0 & 0 & 0 & -I_p & 0 & 0 & 0 \end{array}\right].$$

By evaluating $\mathcal{F}_u(\widetilde{M}_M, \widetilde{\Delta}_M)$ directly, the expression reduces to $\mathcal{F}_u(M_{\text{lf}}, \Delta_{\text{lf}})$.

The result for invertible $D_D$ can easily be derived from (2.30) or it can be proven similarly as done in [14]. ∎

The following lemma (given without proof) is the dual result for a right fractional representation.

**Lemma 2.7.** Let $\mathcal{F}_u(M,\Delta) = \left[\begin{array}{c} N(\delta) \\ D(\delta) \end{array}\right]$ be defined with

$$M = \left[\begin{array}{c|c|c} E & A & B \\ \hline 0 & C_N & D_N \\ 0 & C_D & D_D \end{array}\right],$$

and assume that $D(\delta)$ is a $p \times p$ invertible matrix. Then

$$\mathcal{F}_u(M_{\text{rf}}, \Delta_{\text{rf}}) = N(\delta)D^{-1}(\delta)$$

with

$$M_{\text{rf}} = \left[\begin{array}{cc|cc|c} 0 & 0 & D_D & C_D & -I_p \\ 0 & E & B & A & 0 \\ \hline 0 & 0 & D_N & C_N & 0 \end{array}\right], \; \Delta_{\text{rf}} = \left[\begin{array}{cc} I_p & 0 \\ 0 & \Delta \end{array}\right].$$

If $D_D$ is invertible, alternative representation of $M_{\text{rf}}$ and $\Delta_{\text{rf}}$ are

$$M_{\text{rf}} = \left[\begin{array}{c|c|c} E & A - BD_D^{-1}C_D & BD_D^{-1} \\ \hline 0 & C_N - D_N D_D^{-1} C_D & D_N D_D^{-1} \end{array}\right], \; \Delta_{\text{rf}} = \Delta.$$

### 2.3.3 Object-oriented LFR realization procedure for rational parametric matrices

Using the results of section 2.3.2, one can readily build an LFR for an arbitrary rational parametric matrix $G(\delta)$ along the lines of the following object-oriented procedure, where the realization problem is decomposed into the elementary steps:

1. Build elementary LFRs of the form (2.18) for all distinct parameters $\delta_i$, $i = 1, \ldots, k$.
2. Generate an LFR of each rational matrix element $g_{ij}(\delta)$ from the LFRs of its numerator and denominator polynomials by applying Lemma 2.6. The polynomial realizations are constructed using object-oriented realization techniques based on multiplication and addition/subtraction of LFRs (see Lemma 2.1) relying on efficient multivariate polynomial evaluation schemes.
3. Use row and column concatenations for LFRs (see Lemma 2.1) to obtain an LFR of $G(\delta)$ from the LFRs of the individual entries $g_{ij}(\delta)$.
4. employ Lemma 2.4 to reorder $(M, \Delta)$, such that $\Delta$ is of the form as in (2.21).

**Example 2.3.** Using the object-oriented realization procedure, an LFR for $G(\delta) = \left[\begin{array}{cc} \frac{\delta_2}{\delta_1} & \delta_2\delta_3 + \delta_4 \end{array}\right]$, with $g_{11}(\delta) = \delta_2/\delta_1$ and $g_{12}(\delta) = \delta_2\delta_3 + \delta_4$ is realized as follows:

- realize elementary LFRs $(M_i, \Delta_i), i = 1, \ldots, 4$, with $\delta_i = \mathcal{F}_u(M_i, \Delta_i)$, where

$$M_i = \left[\begin{array}{c|c|c} 1 & 0 & 1 \\ \hline 0 & 1 & 0 \end{array}\right], \; \Delta_i = \delta_i$$

- employ (iii) of Lemma 2.1 to calculate an LFR for the column concatenation of the numerator and denominator polynomial of $g_{11}(\delta)$, i.e.

$$\mathcal{F}_u(M_5, \Delta_5) = \begin{bmatrix} \mathcal{F}_u(M_2, \Delta_2) & \mathcal{F}_u(M_1, \Delta_1) \end{bmatrix},$$

with

$$M_5 = \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \end{array}\right], \Delta_5 = \begin{bmatrix} \delta_2 & 0 \\ 0 & \delta_1 \end{bmatrix},$$

and apply Lemma 2.6 to $(M_5, \Delta_5)$ yielding $g_{11}(\delta) = \mathcal{F}_u(M_6, \Delta_6)$, with

$$M_6 = \left[\begin{array}{ccc|ccc|c} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{array}\right], \Delta_6 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \delta_2 & 0 \\ 0 & 0 & \delta_1 \end{bmatrix}$$

- apply (ii) of Lemma 2.1 to $(M_2, \Delta_2)$ and $(M_3, \Delta_3)$ to obtain $(M_7, \Delta_7)$ such that $\delta_2\delta_3 = \mathcal{F}_u(M_7, \Delta_7)$, with

$$M_7 = \left[\begin{array}{cc|cc|c} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 0 \end{array}\right], \Delta_7 = \begin{bmatrix} \delta_2 & 0 \\ 0 & \delta_3 \end{bmatrix}$$

- apply (i) of Lemma 2.1 to $(M_7, \delta_7)$ and $(M_4, \Delta_4)$ to realize $(M_8, \Delta_8)$, with $g_{22}(\delta) = \mathcal{F}_u(M_8, \Delta_8)$, where

$$M_8 = \left[\begin{array}{ccc|ccc|c} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}\right], \Delta_8 = \begin{bmatrix} \delta_2 & 0 & 0 \\ 0 & \delta_3 & 0 \\ 0 & 0 & \delta_4 \end{bmatrix}$$

- apply (iii) of Lemma 2.1 to $(M_6, \Delta_6)$ and $(M_8, \Delta_8)$ to obtain $(M_9, \Delta_9)$, with $G(\delta) = \mathcal{F}_u(M_9, \Delta_9)$, where

$$M_9 = \left[\begin{array}{cccccc|cccccc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{array}\right]$$

$$\Delta_9 = \mathrm{diag}(1, \delta_2, \delta_1, \delta_2, \delta_3, \delta_4)$$

- finally reorder $(M_9, \Delta_9)$ to obtain $(M_{10}, \Delta_{10})$ with $G(\delta) = \mathcal{F}_u(M_{10}, \Delta_{10})$ and

$$M_{10} = \left[\begin{array}{cccccc|cccccc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{array}\right]$$

$$\Delta_{10} = \mathrm{diag}(1, \delta_1, \delta_2 I_2, \delta_3, \delta_4).$$

An alternative realization approach of a rational parametric matrix has been proposed in [14]. The LFR of $G(\boldsymbol{\delta})$ is built by starting with the symbolic calculation of a left (or right) fractional representation $G(\boldsymbol{\delta}) = D^{-1}(\boldsymbol{\delta})\, N(\boldsymbol{\delta})$ with $D(\boldsymbol{\delta})$ and $N(\boldsymbol{\delta})$ multivariate polynomial matrices. After obtaining $[N(\boldsymbol{\delta})\ D(\boldsymbol{\delta})]$ as a standard LFR, the realization of $G(\boldsymbol{\delta})$ follows by employing Lemma 2.6 (or Lemma 2.7). A potential weakness of this approach is the lack of an efficient factorization algorithm with guaranteed minimal degree of denominator factors. Since the degrees of denominators for problems with many parameters and large matrix dimensions tend to be high, the orders of realizations are frequently higher than those resulted employing the simple approach above.

### 2.3.4 Normalization

In section 2.1 the parameter set $\Pi$ was defined (2.2), where each parameter is bounded by a minimum and a maximum value, i.e. $\delta_i \in [\delta_{i,\min}, \delta_{i,\max}]$. To perform the normalization of parameters, $\delta_i$ can be replaced by a rational parametric function $n_i(\widetilde{\delta}_i)$ such that $\delta_i = n_i(\widetilde{\delta}_i)$ with $|\widetilde{\delta}_i| \leq 1$. The function $n_i$ is chosen such that, $\delta_{i,\min} = n_i(-1)$, $\delta_{i,\max} = n_i(1)$, $\delta_{i,\mathrm{nom}} = n_i(0)$. Hence, the nominal plant model $G_{\mathrm{nom}}$ can be defined as $G_{\mathrm{nom}} := G(\delta_{\mathrm{nom}}) = G(n(\widetilde{\delta}))|_{\widetilde{\delta}=0}$.

It is important that the normalization does not increase the order of the LFR and therefore the functions $n_i$ must admit a first order LFR. All these requirements are fulfilled by choosing $n_i$ as the linear fractional transformation between $\delta_i$ and $\widetilde{\delta}_i$ as [23]

$$\delta_i = \frac{\gamma_i + \alpha_i \widetilde{\delta}_i}{1 + \beta_i \widetilde{\delta}_i},\ i = 1, \ldots, k, \tag{2.31}$$

where the parameters $\gamma_i$, $\alpha_i$ and $\beta_i$ are determined by

$$\begin{aligned} \gamma_i &= \delta_{i,\mathrm{nom}} \\ \alpha_i &= \frac{\delta_{i,\max}(\delta_{i,\mathrm{nom}} - \delta_{i,\min}) - \delta_{i,\min}(\delta_{i,\max} - \delta_{i,\mathrm{nom}})}{(\delta_{i,\max} - \delta_{i,\min})} \\ \beta_i &= \frac{(\delta_{i,\mathrm{nom}} - \delta_{i,\min}) - (\delta_{i,\max} - \delta_{i,\mathrm{nom}})}{(\delta_{i,\max} - \delta_{i,\min})} \end{aligned}$$

The following result provides formulas to express $G(n(\widetilde{\delta}))$ in terms of the LFT for $G(\delta)$.

**Lemma 2.8.** Let $G(\delta) = \mathcal{F}_u(M, \Delta)$ with

$$M = \left[\begin{array}{c|c|c} E & A & B \\ \hline 0 & C & D \end{array}\right]$$

and let $\Delta = \mathcal{F}_u(N, \overline{\Delta})$ be the LFR of the relation between the initial and normalized parameters, where

$$N = \left[\begin{array}{c|c|c} I & A_N & B_N \\ \hline 0 & C_N & D_N \end{array}\right],$$

with

$$\begin{aligned} A_N &= \mathrm{diag}(0_{r_0}, -\beta_1 I_{r_1}, \ldots, -\beta_k I_{r_k}) \\ B_N &= \mathrm{diag}(0_{r_0}, I_{r_1}, \ldots, I_{r_k}) \\ C_N &= \mathrm{diag}(0_{r_0}, (\alpha_1 - \gamma_1\beta_1) I_{r_1}, \ldots, (\alpha_k - \gamma_k\beta_k) I_{r_k}) \\ D_N &= \mathrm{diag}(I_{r_0}, \gamma_1 I_{r_1}, \ldots, \gamma_k I_{r_k}) = \Delta_{\mathrm{nom}}. \end{aligned} \tag{2.32}$$

If $(E - A\Delta_{\mathrm{nom}})$ is invertible, then

$$\mathcal{F}_u(M, \mathcal{F}_u(N, \overline{\Delta})) = \mathcal{F}_u(\overline{M}, \overline{\Delta}) = \mathcal{F}_u(\widetilde{M}, \widetilde{\Delta}),$$

where

$$\overline{M} = \left[\begin{array}{c|c|c} I & \overline{A} & \overline{B} \\ \hline 0 & \overline{C} & \overline{D} \end{array}\right], \ \overline{\Delta} = \mathrm{diag}(0_{r_0}, \widetilde{\delta}_1 I_{r_1}, \ldots, \widetilde{\delta}_k I_{r_k}),$$

with

$$\begin{aligned} \overline{A} &= A_N + B_N(E - AD_N)^{-1} A C_N \\ \overline{B} &= B_N(E - AD_N)^{-1} B \\ \overline{C} &= C(D_N(E - AD_N)^{-1} A + I) C_N \\ \overline{D} &= D + C D_N (E - AD_N)^{-1} B. \end{aligned} \tag{2.33}$$

Partitioning $\overline{M}$ as

$$\overline{M} = \left[\begin{array}{cc|cc|c} I_{r_0} & 0 & \overline{A}_{11} & \overline{A}_{12} & \overline{B}_1 \\ 0 & I & \overline{A}_{21} & \overline{A}_{22} & \overline{B}_2 \\ \hline 0 & 0 & \overline{C}_1 & \overline{C}_2 & \overline{D} \end{array}\right], \tag{2.34}$$

the matrices $\widetilde{M}$ and $\widetilde{\Delta}$ are given by

$$\widetilde{M} = \left[\begin{array}{c|c|c} I & \overline{A}_{22} & \overline{B}_2 \\ \hline 0 & \overline{C}_2 & \overline{D} \end{array}\right] \tag{2.35}$$

$$\widetilde{\Delta} = \mathrm{diag}(\widetilde{\delta}_1 I_{r_1}, \ldots, \widetilde{\delta}_k I_{r_k}). \tag{2.36}$$

**Proof.** In the normalized LFR, $\Delta$ is substituted by $\Delta = \mathcal{F}_u(N, \overline{\Delta})$ and it is straightforward to derive (2.33) from the relation $\mathcal{F}_u(\overline{M}, \overline{\Delta}) = \mathcal{F}_u(M, \mathcal{F}_u(N, \overline{\Delta}))$. From the particular structure of $A_N$, $B_N$ and $C_N$ it follows directly that the matrices $\overline{A}_{11}$, $\overline{A}_{12}$, $\overline{A}_{21}$, $\overline{B}_1$, and $\overline{C}_1$ in (2.34) are null.

■

**Remark 2.3.** A necessary condition for the normalization of a generalized LFR is that $(E - A\Delta_{\text{nom}})$ is invertible. This simply means that the nominal model $G_{\text{nom}} = \mathcal{F}_u(M, \Delta_{\text{nom}}) = D + C\Delta_{\text{nom}}(E - A\Delta_{\text{nom}})^{-1}B$ is well-defined, and this condition must always be fulfilled. It is essential to note, that when applying the normalization to a generalized LFR, the arbitrarily introduced additional block $I_{r_0}$ in $\Delta$ can be removed and the resulting LFR (2.35), (2.36) can be represented as a standard LFR. Therefore the generalized LFR can be seen as a convenient means to avoid the preliminary symbolic normalization of the parametric matrix $G(\delta)$. The normalization can be applied as the last step in the LFR realization procedure and the final LFR will be a standard LFR. Of course this only holds for an LFR of a rational parametric matrix $G(\delta)$. In the case of general multidimensional dynamical systems, operators like the integration operator $1/s$ or the shift operator $1/z$ may be included in several dimensions (e.g., in the $x$ and $y$ direction in 2-dimensional image processing applications) and for these operators a normalization does not make sense. If the model is non-proper in these dimensions, it is not possible to represent the system as a standard LFR.

**Remark 2.4.** In literature, the most common normalization with $\delta_{i,\text{nom}} = (\delta_{i,\max} + \delta_{i,\min})/2$, $i = 1, \ldots, k$ is generally used. This implies that the nominal values are at the center of the uncertainty intervals, which is generally not true. In this case one may simply choose $\beta_i = 0$, $\gamma_i = \delta_{i,\text{nom}} = (\delta_{i,\max} + \delta_{i,\min})/2$ and $\alpha_i = (\delta_{i,\max} - \delta_{i,\min})/2$ in (2.32).

### 2.3.5 Special form of generalized LFT

In section 2.3.3 the object-oriented LFR realization procedure was proposed to generate a generalized LFR for a rational parametric matrix $G(\delta)$. The outcome of this procedure is an LFR, which can always be expressed in the partitioned form

$$M = \left[\begin{array}{c|c|c} E & A & B \\ \hline 0 & C & D \end{array}\right] = \left[\begin{array}{cc|cc|c} 0_{r_0} & 0 & A_{11} & A_{12} & B_1 \\ 0 & I & A_{21} & A_{22} & B_2 \\ \hline 0 & 0 & C_1 & C_2 & D \end{array}\right] \tag{2.37}$$
$$\Delta = \operatorname{diag}(I_{r_0}, \delta_1 I_{r_1}, \ldots, \delta_k I_{r_k}) := \operatorname{diag}(I_{r_0}, \overline{\Delta}),$$

with $E$ as a block-diagonal matrix consisting of a $r_0 \times r_0$ zero block and an identity matrix of size $\sum_{i=1}^{k} r_i$.

Using the equation based description of the generalized LFT in (2.37), one has

$$0_{r_0} = A_{11}w_1 + A_{12}w_2 + B_1 u \tag{2.38}$$
$$z_2 = A_{21}w_1 + A_{22}w_2 + B_2 u \tag{2.39}$$
$$y = C_1 w_1 + C_2 w_2 + Du \tag{2.40}$$
$$w_1 = z_1 \tag{2.41}$$
$$w_2 = \overline{\Delta} z_2. \tag{2.42}$$

Since $w_1 = z_1$, one may simply add the term $I_{r_0} w_1$ on both sides of equation (2.38). The

corresponding LFR is defined by $\overline{M}$ and $\Delta$ with

$$\overline{M} = \left[\begin{array}{c|c|c} \overline{E} & \overline{A} & \overline{B} \\ \hline 0 & \overline{C} & \overline{D} \end{array}\right] = \left[\begin{array}{cc|cc|c} I_{r_0} & 0 & A_{11}+I_{r_0} & A_{12} & B_1 \\ 0 & I & A_{21} & A_{22} & B_2 \\ \hline 0 & 0 & C_1 & C_2 & D \end{array}\right], \tag{2.43}$$

and obviously $\mathcal{F}_u(M,\Delta) = \mathcal{F}_u(\overline{M},\Delta)$. Since $\overline{E}$ is an identity matrix, this is a standard LFT with a constant block $I_{r_0}$ in $\Delta$. The algebraic properties of Lemma 2.1, 2.2 and the normalization described by Lemma 2.8 also hold for this special form of the generalized LFT, where $E = I$ is assumed.

Only the operations for inversion and the realization of left-/right fractional representations must be adapted. The following corollaries are immediate consequences of Lemmas 2.5, 2.6 and 2.7 for this special structure of the generalized LFT.

**Corollary 2.2.** Consider the LFR described by the partitioned matrix

$$M = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

with $D \in \mathbb{R}^{p\times p}$ and the uncertainty matrix $\Delta$. Then $\mathcal{F}_u(M,\Delta)^{-1} = \mathcal{F}_u(M_{\text{inv}},\Delta_{\text{inv}})$ with

$$M_{\text{inv}} = \left[\begin{array}{cc|c} D+I_p & C & I_p \\ B & A & 0 \\ \hline -I_p & 0 & 0 \end{array}\right], \; \Delta_{\text{inv}} = \left[\begin{array}{cc} I_p & 0 \\ 0 & \Delta \end{array}\right].$$

If $D$ is invertible, alternative representations for $M_{\text{inv}}$ and $\Delta_{\text{inv}}$ are

$$M_{\text{inv}} = \left[\begin{array}{c|c} A-BD^{-1}C & -BD^{-1} \\ \hline D^{-1}C & D^{-1} \end{array}\right], \; \Delta_{\text{inv}} = \Delta.$$

**Corollary 2.3.** Let $\mathcal{F}_u(M,\Delta) = \left[\begin{array}{cc} N(\delta) & D(\delta) \end{array}\right]$ be defined with

$$M = \left[\begin{array}{c|cc} A & B_N & B_D \\ \hline C & D_N & D_D \end{array}\right]$$

and assume that $D(\delta)$ is a $p \times p$ invertible matrix. Then

$$\mathcal{F}_u(M_{\text{lf}},\Delta_{\text{lf}}) = D^{-1}(\delta)N(\delta)$$

with

$$M_{\text{lf}} = \left[\begin{array}{cc|c} D_D+I_p & C & D_N \\ B_D & A & B_N \\ \hline -I_p & 0 & 0 \end{array}\right], \; \Delta_{\text{lf}} = \left[\begin{array}{cc} I_p & 0 \\ 0 & \Delta \end{array}\right]. \tag{2.44}$$

If $D_D$ is invertible, alternative representations of $M_{\text{lf}}$ and $\delta_{\text{lf}}$

$$M_{\text{lf}} = \left[\begin{array}{c|c} A-B_DD_D^{-1}C & B_N-B_DD_D^{-1}D_N \\ \hline D_D^{-1}C & D_D^{-1}D_N \end{array}\right], \; \Delta_{\text{lf}} = \Delta$$

**Corollary 2.4.** Let $\mathcal{F}_u(M,\Delta) = \left[\begin{array}{c} N(\delta) \\ D(\delta) \end{array}\right]$ be defined with

$$M = \left[\begin{array}{c|c} A & B \\ \hline C_N & D_N \\ C_D & D_D \end{array}\right],$$

and assume that $D(\delta)$ is a $p \times p$ invertible matrix. Then

$$\mathcal{F}_u(M_{\rm rf}, \Delta_{\rm rf}) = N(\delta)D^{-1}(\delta)$$

with

$$M_{\rm rf} = \left[\begin{array}{cc|c} D_D + I_p & C_D & -I_p \\ B & A & 0 \\ \hline D_N & C_N & 0 \end{array}\right], \; \Delta_{\rm rf} = \left[\begin{array}{cc} I_p & 0 \\ 0 & \Delta \end{array}\right].$$

If $D_D$ is invertible, alternative representations of $M_{\rm rf}$ and $\Delta_{\rm rf}$ are

$$M_{\rm rf} = \left[\begin{array}{c|c} A - BD_D^{-1}C_D & BD_D^{-1} \\ \hline C_N - D_N D_D^{-1} C_D & D_N D_D^{-1} \end{array}\right], \; \Delta_{\rm rf} = \Delta.$$

### 2.3.6 Relation to Behavioral Representations

A behavioral representation for systems with structured uncertainty has been introduced in [30], having the form

$$\begin{aligned} z &= A\Delta z + Bw \\ 0 &= C\Delta z + Dw \end{aligned} \tag{2.45}$$

This description will be referred to as *output nulling representation* (ONR). In this representation, the vector $z$ is the state and $w$ includes all system variables like inputs, outputs or some so-called latent variables. When manipulating such models there is no need for an a priori choice of input and output variables. In contrast, the explicit generalized LFRs are input-output type representations. Since conversions between the two representations are straightforward (see below), both representations are suitable to represent arbitrary expressions with rational dependency on uncertain parameters. However, as we will see later, the capabilities of these representations to obtain low order LFRs (e.g., suitable for LFT-based robust stability analysis) are quite different.

For the realization of the input-output dependence $y = G(\delta)u$, consider the following ONR

$$\begin{aligned} z &= A\Delta z + B_1 y + B_2 u \\ 0 &= C\Delta z + D_1 y + D_2 u \end{aligned}$$

which corresponds to (2.45) with $B = [\, B_1 \; B_2 \,]$, $D = [\, D_1 \; D_2 \,]$ and $w = [y^T u^T]^T$. Assuming that $D_1 \in \mathbb{R}^{p\times p}$, an explicit generalized LFR of $G(\delta)$ is the following one

$$y = \mathcal{F}_u\left(\left[\begin{array}{cc|c} D_1 + I_p & C & D_2 \\ B_1 & A & B_2 \\ \hline -I_p & 0 & 0 \end{array}\right], \left[\begin{array}{cc} I_p & 0 \\ 0 & \Delta \end{array}\right]\right) u \tag{2.46}$$

If $D_1$ is invertible, a standard LFR is given by

$$y = \mathcal{F}_u\left(\left[\begin{array}{c|c} A - B_1 D_1^{-1} C & B_2 - B_1 D_1^{-1} D_2 \\ \hline -D_1^{-1} C & -D_1^{-1} D_2 \end{array}\right], \Delta\right) u \tag{2.47}$$

Conversely, assume that we have for $G(\delta)$ an explicit generalized LFR of the form

$$y = \mathcal{F}_u\left(\left[\begin{array}{cc|c} A_{11} + I_p & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & C_2 & D \end{array}\right], \left[\begin{array}{cc} I_p & 0 \\ 0 & \Delta \end{array}\right]\right) u \tag{2.48}$$

An ONR is given by

$$\begin{aligned} z &= \left[\begin{array}{cc} A_{11} + I_p & A_{12} \\ A_{21} & A_{22} \end{array}\right] \left[\begin{array}{cc} I_p & 0 \\ 0 & \Delta \end{array}\right] z + \left[\begin{array}{cc} 0 & B_1 \\ 0 & B_2 \end{array}\right] \left[\begin{array}{c} y \\ u \end{array}\right] \\ 0 &= \left[\begin{array}{cc} C_1 & C_2 \end{array}\right] \left[\begin{array}{cc} I_p & 0 \\ 0 & \Delta \end{array}\right] z + \left[\begin{array}{cc} -I_p & D \end{array}\right] \left[\begin{array}{c} y \\ u \end{array}\right] \end{aligned} \tag{2.49}$$

From the above relations it follows that the generalized LFR and the ONR are mathematically equivalent formalisms to represent rational parametric matrices.

The basic aspect of generating LFRs is the efficient representation of interconnected systems. When interconnecting two ONRs, a basic requirement is (see [86]), that the two representations have the same signal space. To ensure this condition, the resulting interconnected system typically contains latent variables and it may be necessary to introduce additional variables to describe the interconnection constraints. The presence of a large number of latent variables (very common for complex ONRs) makes the behavioral approach less suitable for an efficient LFT-based model building. In contrast, object-oriented approaches like that described in section 2.3.3 or in [77], produce explicit LFRs with a "minimal" amount of data. The following simple example will make this aspect clear.

**Example 2.4.** For the input-output dependence $y = (\delta_1 + \delta_2)u$ we build an ONR to obtain via (2.47) an LFR suitable for $\mu$-analysis. ONRs to represent $y_i = \delta_i u_i$ for $i = 1, 2$ are given by

$$\begin{aligned} z_i &= u_i \\ 0 &= \delta_i z_i - y_i \end{aligned}$$

To represent $y = (\delta_1 + \delta_2)u$ the interconnection constraints

$$\begin{aligned} u_1 &= u_2 (= u) \\ y &= y_1 + y_2 \end{aligned}$$

must be fulfilled. To obtain the final ONR, we collect all states in $z = [\, z_1 \; z_2 \,]^T$ and all variables in $w = [\, y \; y_1 \; u_1 \; y_2 \; u \,]^T$ and write down the above equations in the standard

ONR form

$$z = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} z + \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} w$$
$$0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} z + \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ -1 & 1 & 0 & 1 & 0 \end{bmatrix} w$$

With $\widetilde{y} = [\, y \; y_1 \; u_1 \; y_2 \,]^T$ as output variable and $u$ as input variable, we apply now (2.47), with

$$D_1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix}$$

to obtain the explicit LFR

$$\begin{bmatrix} y \\ y_1 \\ u_1 \\ y_2 \end{bmatrix} = \mathcal{F}_u \left( \left[\begin{array}{cc|cc|c} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{array}\right], \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} \right) u.$$

To obtain the LFR of the input-output dependence $y = (\delta_1 + \delta_2)u$ we simply omit the output equations corresponding to the latent variables $y_1, u_1, y_2$, yielding

$$y = \mathcal{F}_u \left( \left[\begin{array}{cc|cc|c} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 & 0 \end{array}\right], \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} \right) u.$$

Note that the resulting LFR is just that one which is directly obtained by using LFR manipulations as those described in sections 2.3.2, 2.3.5 and the possibly ill-conditioned inversion of a usually large matrix $D_1$ in (2.47) is avoided.

This simple example shows that because of the presence of latent variables, the ONRs have a certain data redundancy, which is not present in standard or generalized LFRs. The direct elimination of latent variables in ONRs is quite involved even for 1-dimensional systems (see [27]) and it is probably an open problem in the general multidimensional case.

**Remark 2.5.** The formulas for the normalization of parameters for a plant model in LFT-form as described in section 2.3.4 also hold for the special form of the generalized LFR by substituting $E$ with $I$ in (2.33). The resulting normalized LFR can be represented as a standard LFR, without constant block in $\Delta$.

## 2.4 LFR realization for parametric descriptor system

In section 2.1, parametric descriptor systems of the form

$$\begin{aligned} E(\delta)\dot{x} &= A(\delta)x + B(\delta)u \\ y &= C(\delta)x + D(\delta)u \end{aligned} \tag{2.50}$$

were introduced and in this section, the application of the proposed object-oriented LFR realization approach will be presented for these systems. We assume that the matrices $E(\delta)$, $A(\delta)$, $B(\delta)$, $C(\delta)$, $D(\delta)$ depend rationally on $\delta$, $E(\delta)$ and $A(\delta)$ are square matrices and $E(\delta) \in \mathbb{R}^{n\times n}$ has a constant rank $r \leq n$ for all $\delta \in \Pi$.

The transfer function matrix $G(s, \delta)$ (with $s$ as the Laplace variable) of the descriptor system (2.50) is given by

$$G(s, \delta) = D(\delta) + C(\delta)(sE(\delta) - A(\delta))^{-1}B(\delta) \tag{2.51}$$

where the pencil $|sE(\delta) - A(\delta)|$ is assumed to be regular for all $\delta \in \Pi$.

In the following, a general method is presented to determine a pair $(M, \Delta)$ such that

$$G(s, \delta) = \mathcal{F}_u(M, \Delta)$$

with

$$\begin{aligned} M &= \left[\begin{array}{c|c} A_M & B_M \\ \hline C_M & D_M \end{array}\right], \\ \Delta &= \mathrm{diag}(I_n/s, I_{r_0}, \delta_1 I_{r_1}, \ldots, \delta_k I_{r_k}). \end{aligned} \tag{2.52}$$

Note, in this LFR, the integration operator $1/s$ is also included in $\Delta$.

The LFR realization for parametric descriptor systems has been addressed in [90, 76] for the particular case when all system matrices depend polynomially on the components of the parameter vector $\delta$. Moreover, in [76] it was assumed, that $E(\delta)$ is invertible. In what follows, it is shown that a generalized LFR for $G(s, \delta)$ can be constructed in the most general case of rational parametric matrices, and without assuming the invertibility of $E(\delta)$.

For the efficient realization of an LFR for $G(s, \delta)$, two cases are distinguished: (1) $E(\delta)$ general (possibly non-invertible); (2) $E(\delta)$ invertible.

### 2.4.1 $E(\delta)$ general

The LFR for $G(s, \delta)$ can be built using the following steps:

1. Use the object-oriented LFR realization procedure of section 2.3.3 to obtain gener-

alized LFRs for each system matrix of (2.50), that is, realize

$$A(\delta) = \mathcal{F}_u\left(\left[\begin{array}{c|c} A_A & B_A \\ \hline C_A & D_A \end{array}\right], \Delta_A\right),$$
$$B(\delta) = \mathcal{F}_u\left(\left[\begin{array}{c|c} A_B & B_B \\ \hline C_B & D_B \end{array}\right], \Delta_B\right),$$
$$C(\delta) = \mathcal{F}_u\left(\left[\begin{array}{c|c} A_C & B_C \\ \hline C_C & D_C \end{array}\right], \Delta_C\right),$$
$$D(\delta) = \mathcal{F}_u\left(\left[\begin{array}{c|c} A_D & B_D \\ \hline C_D & D_D \end{array}\right], \Delta_D\right),$$
$$E(\delta) = \mathcal{F}_u\left(\left[\begin{array}{c|c} A_E & B_E \\ \hline C_E & D_E \end{array}\right], \Delta_E\right).$$

2. Then $G(s,\delta) = \mathcal{F}_u(M,\Delta)$ with

$$M = \left[\begin{array}{ccccccc|c} 0 & -D_A & -C_A & C_B & 0 & 0 & 0 & D_B \\ I_n & D_E + I_n & 0 & 0 & 0 & 0 & C_E & 0 \\ 0 & B_A & A_A & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_B & 0 & 0 & 0 & B_B \\ 0 & -B_C & 0 & 0 & A_C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_D & 0 & B_D \\ 0 & B_E & 0 & 0 & 0 & 0 & A_E & 0 \\ \hline 0 & -D_C & 0 & 0 & C_C & C_D & 0 & D_D \end{array}\right] \tag{2.53}$$
$$\Delta = \mathrm{diag}(I_n/s, I_n, \Delta_A, \Delta_B, \Delta_C, \Delta_D, \Delta_E).$$

3. Employ Lemma 2.4 to reorder $(M,\Delta)$ such that $\Delta$ is of the form as given in (2.52).

### 2.4.2 $E(\delta)$ invertible

In case of invertible $E(\delta)$ we can derive the following procedure:

1. Use the object-oriented LFR realization procedure of section 2.3.3 to obtain a generalized LFR, such that

$$\left[\begin{array}{cc} \overline{N}(\delta) & \overline{D}(\delta) \end{array}\right] = \mathcal{F}_u\left(\left[\begin{array}{c|cc} \overline{A} & \overline{B}_N & \overline{B}_D \\ \hline \overline{C} & \overline{D}_N & \overline{D}_D \end{array}\right], \overline{\Delta}\right),$$

with

$$\overline{N}(\delta) = \left[\begin{array}{cc} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{array}\right], \overline{D}(\delta) = \left[\begin{array}{cc} E(\delta) & 0 \\ 0 & I_p \end{array}\right]$$

2. Apply Corollary 2.3 to obtain

$$\begin{aligned}\overline{D}^{-1}(\delta)\overline{N}(\delta) &= \left[\begin{array}{cc} E^{-1}(\delta)A(\delta) & E^{-1}(\delta)B(\delta) \\ C(\delta) & D(\delta) \end{array}\right] \\ &= \mathcal{F}_u\left(\left[\begin{array}{cc|c} \overline{D}_D + I_l & \overline{C} & \overline{D}_N \\ \overline{B}_D & \overline{A} & \overline{B}_N \\ \hline -I_l & 0 & 0 \end{array}\right], \left[\begin{array}{cc} I_l & 0 \\ 0 & \overline{\Delta} \end{array}\right]\right) \\ &= \mathcal{F}_u\left(\left[\begin{array}{c|cc} A' & B_1' & B_2' \\ \hline C_1' & 0_n & 0 \\ C_2' & 0 & 0_p \end{array}\right], \left[\begin{array}{cc} I_l & 0 \\ 0 & \overline{\Delta} \end{array}\right]\right).\end{aligned}$$

   with $l = n + p$.

3. Then $G(s, \delta) = \mathcal{F}_u(M, \Delta)$ with

$$M = \left[\begin{array}{ccc} 0_n & C_1' & 0 \\ B_1' & A' & B_2' \\ \hline 0 & C_2' & 0_p \end{array}\right], \Delta = \left[\begin{array}{ccc} I_n/s & 0 & 0 \\ 0 & I_l & 0 \\ 0 & 0 & \overline{\Delta} \end{array}\right].$$

4. Employ Lemma 2.4 to reorder $(M, \Delta)$ such that $\Delta$ is of the form as given in (2.52).

The main advantage of the second LFR realization procedure is that the symbolic preprocessing techniques described in chapter 3 can be applied to the concatenated symbolic matrix $\left[\begin{array}{cc} \overline{N}(\delta) & \overline{D}(\delta) \end{array}\right]$ (see step 1), which contains all matrices of the system. Hence, the symbolic structure and common symbolic expressions of system matrices can be exploited in the preprocessing and it is expected that the resulting LFR is of lower order than an LFR, which is realized using the more general procedure of section 2.4.1, where each system matrix is realized separately.

# 3 Symbolic techniques for low order LFR modelling

As already mentioned in chapter 1, for the application of LFT-based robust control techniques it is of paramount importance to obtain LFRs of *least possible orders*. Obviously this aim is not a trivial task and the introduction of the generalized LFR in chapter 2 is one means to reduce the LFR order. In this chapter it will be shown, that symbolic preprocessing is a powerful complementary tool to achieve low orders of the LFRs. For a given parametric matrix $G(\delta)$, the symbolic processing allows to find equivalent representations of this matrix, which automatically lead to lower order LFRs, when employed in conjunction with the object-oriented LFR realization approach described in chapter 2. The object-oriented LFR realization is very flexible and can easily be automated. However, a blindfold application of this procedure may yield LFRs of larger order than the least possible one.

**Example 3.1.** Consider the standard LFR $\mathcal{F}_u(M, \Delta)$ for the sum $G(\delta) = \delta_1 + \delta_2$, which is obtained using (i) of Lemma 2.1 as

$$M = \left[\begin{array}{cc|c} 0 & 0 & 1 \\ 0 & 0 & 1 \\ \hline 1 & 1 & 0 \end{array}\right], \Delta = \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix}. \tag{3.1}$$

This LFR has a least order of two. However, applying the above construction to the expression $G(\delta) = \delta_1 + \delta_1$, one may realize it as an LFR of order two with $M$ given in (3.1) and $\Delta = \mathrm{diag}(\delta_1, \delta_1)$. Obviously, a first order LFR for $G(\delta) = \delta_1 + \delta_1$ is possible starting from an equivalent expression

$$G(\delta) = 2\delta_1 = \mathcal{F}_u\left(\left[\begin{array}{c|c} 0 & 2 \\ \hline 1 & 0 \end{array}\right], \delta_1\right).$$

From this very simple example it can be seen that the trivial symbolic simplification of $G(\delta) = \delta_1 + \delta_1$ to $G(\delta) = 2\delta_1$ allows to reduce the LFR order by one, when the object-oriented LFR realization technique is directly applied. Therefore it is clear that the resulting order of the generated LFR depends crucially on the way the expressions underlying the LFR realization are evaluated and the role of symbolic preprocessing is to find simpler evaluation schemes of rational expressions and matrices which finally lead to LFRs of lower order.

## 3.1 Limitation of numerical order reduction

In chapter 1 special order reduction techniques [29, 83] have been mentioned as an additional means to reduce the order of an LFR. These are typically applied as a final (or

postprocessing) step of LFR modelling and are based on the application of similarity transformations (see Lemma 2.2) to identify non-minimal parts of the LFR, which are then removed (thus reducing the order of the LFR).

Consider again example 3.1 with $M$ given in (3.1) and $\Delta = \mathrm{diag}(\delta_1, \delta_1)$. A similarity transformation applied to $M$ with

$$Q = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \; Z = 1/2 \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix},$$

yields $\mathcal{F}_u(M, \Delta) = \mathcal{F}_u(\widetilde{M}, \Delta)$ with

$$\widetilde{M} = \left[\begin{array}{cc|c} 0 & 0 & 0 \\ 0 & 0 & 2 \\ \hline 0 & 1 & 0 \end{array}\right]$$

and it is clear that $(\widetilde{M}, \Delta)$ can be replaced by

$$\widetilde{M}_r = \left[\begin{array}{c|c} 0 & 2 \\ \hline 1 & 0 \end{array}\right], \; \Delta_r = \delta_1.$$

Hence, for the simple example 3.1, symbolic simplification and order reduction both yield a least order LFR. However, in the multidimensional case (more than one uncertain parameter included in $G(\delta)$), there exists a basic limitation of the order reduction techniques arising from the association of all $\delta_i$ with non-commuting operators, that is, generally $\delta_i \delta_j \neq \delta_j \delta_i$ is assumed.

**Example 3.2.** Consider the LFRs $\mathcal{F}_u(M_1, \Delta) = \delta_1 \delta_2$ and $\mathcal{F}_u(M_2, \Delta) = \delta_2 \delta_1$ with

$$M_1 = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \end{array}\right], \; M_2 = \left[\begin{array}{cc|c} 0 & 0 & 1 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \end{array}\right], \; \Delta = \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix}.$$

The LFR of $\mathcal{F}_u(M_3, \Delta_3) = \mathcal{F}_u(M_1, \Delta_1) - \mathcal{F}_u(M_2, \Delta_2)$ constructed by subtracting the LFRs $(M_1 \Delta)$ and $(M_2, \Delta)$, is given by

$$M_3 = \left[\begin{array}{cccc|c} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 \end{array}\right], \; \Delta_3 = \begin{bmatrix} \delta_1 & 0 & 0 & 0 \\ 0 & \delta_1 & 0 & 0 \\ 0 & 0 & \delta_2 & 0 \\ 0 & 0 & 0 & \delta_2 \end{bmatrix}.$$

Although $\mathcal{F}_u(M_3, \Delta_3) = \delta_1 \delta_2 - \delta_2 \delta_1 \equiv 0$, an order reduction of the LFR $(M_3, \Delta_3)$ can not be achieved by using a block-diagonal (condition $Z\Delta = \Delta Z$ in Lemma 2.2) similarity transformation based order reduction methods, because there exists no similarity transformation to transform $(M_1, \Delta)$ into $(M_2, \Delta)$.

**Remark 3.1.** In general, symbolic preprocessing and order reduction can be seen as complementary tools, as the symbolic techniques described in the following are very powerful but not perfect, i.e. in general they do not allow to obtain LFRs of minimal order. Therefore in most applications (see for example chapter 5) it is usually possible to further reduce the order of an LFR by a certain amount using order reduction techniques in a postprocessing step.

**Remark 3.2.** Order reduction techniques for standard LFRs as described in [77, 29], can be directly applied to the special form of the generalized LFR (see section 2.3.5), where the constant block $I_{r_0}$ in $\Delta$ is simply considered as an additional system dimension.

## 3.2 Definitions

For the very simple examples above minimal order LFRs trivially result using elementary symbolic simplifications. For more complicated parametric expressions or in the case of parametric matrices, several ad-hoc or systematic symbolic pre-processing techniques [64, 24, 83, 43, 44] ranging from simple polynomial factorizations to more complex matrix decomposition algorithms are available. In several practical examples the symbolic preprocessing appears to be the most effective step in generating low order LFRs.

Let $\delta = (\delta_1, \ldots, \delta_k)$ be the already defined parameter vector and denote by $\delta^{-1} = (\delta_1^{-1}, \ldots, \delta_k^{-1})$ the vector of reciprocal variables. Three classes of matrices depending on $\delta$ are considered, for which symbolic transformation techniques are discussed: $\mathbb{R}[\delta]^{m\times n}$ - the set of $m \times n$ matrices with multivariate polynomial entries; $\mathbb{R}(\delta)^{m\times n}$ - the set of $m \times n$ matrices with multivariate rational entries; and $\mathbb{R}[\delta, \delta^{-1}]^{m\times n}$ - the set of $m \times n$ matrices with multivariate Laurent polynomial entries. This last case is explicitly considered since many aircraft and automotive related parametric models are described in terms of such matrices (see chapters 5 and 6).

A multivariate Laurent polynomial $g(\delta)$ has the expanded form

$$g(\delta) = \sum_{r=1}^{l} c_r \delta_1^{n_{r,1}} \delta_2^{n_{r,2}} \ldots \delta_k^{n_{r,k}}, \tag{3.2}$$

where $c_r$ are real coefficients and $n_{r,1}, \ldots, n_{r,k} \in \mathbb{Z}$ for $r = 1, \ldots, l$ are integer exponents. We can associate to this polynomial the order of the LFR which results when applying the object-oriented LFR realization approach as described in section 2.3.3 to the polynomial in the above expanded form. This order is given by

$$\operatorname{ord}(g(\delta)) = \sum_{r=1}^{l} \sum_{s=1}^{k} |n_{r,s}|$$

where we assumed that negative and positive powers of indeterminates contribute in the same way to the order. This assumption is valid when employing the generalized inversion formula of Corollary 2.2 to realize the reciprocal variables. If $g(\delta)$ is a general multivariate rational function of the form

$$g(\delta) = \frac{a(\delta)}{b(\delta)}$$

where $a(\delta)$ and $b(\delta)$ are polynomials in expanded forms, the associated order is given by

$$\text{ord}(g(\delta)) = \text{ord}(a(\delta)) + \text{ord}(b(\delta))$$

We can also associate to an $m \times n$ rational matrix $G(\delta)$ with elements $g_{ij}(\delta)$ the total order

$$\text{ord}(G(\delta)) = \sum_{i=1}^{m} \sum_{j=1}^{n} \text{ord}(g_{ij}(\delta)) \tag{3.3}$$

which corresponds to realize $G(\delta)$ element-wise using row and column concatenations via the object-oriented LFR realization approach.

The role of symbolic pre-processing in building low order LFRs of a given rational matrix $G(\delta)$ is to find equivalent representations of individual matrix elements, entire rows/columns or even the whole matrix which lead to LFRs of lower order than given by (3.3). In the following several transformation techniques, which can be used for this purpose are presented.

## 3.3 Single element conversions

Several conversions can be performed on single rational functions which can be useful to obtain equivalent representations which lead automatically to reduced order LFRs via an object oriented realization. These conversions can be performed either iteratively with respect to selected single variables or can be performed simultaneously for several variables in a specified order. Using such conversions, it may be possible to determine in each case a least achievable order of the corresponding LFRs over all permutations of the variables. However, performing such exhaustive searches leads generally to combinatorial problems with exponential complexity. Therefore, unless the number of variables is small (say below 10), exhaustive searches are impracticable. In what follows some of possible conversions are illustrated by examples.

### 3.3.1 Horner form

The conversion of a multivariate polynomial to a nested Horner form is useful for an efficient numerical evaluation of polynomials [53]. For a given polynomial the Horner form may be employed for the following operations with minimal computational cost:

- evaluation of polynomials
- division by a linear factor (e.g., $(\delta_1 - 1)$)
- calculation of derivatives
- reordering of powers of a linear factor

The basic idea in all these computations is to avoid the time consuming calculation of powers of variables by appropriate factorizations. In addition to the reduced computation time, rounding errors are reduced and large intermediate results are avoided [62].

**Example 3.3.** Let $g(\delta)$ be a given polynomial in one variable as

$$g(\delta) = a_0 + a_1\delta_1 + \ldots + a_n\delta_1^n, \tag{3.4}$$

with $a_i \in \mathbb{R}$. The Horner form of $g(\delta)$ is given by

$$g(\delta) = (\ldots(a_n\delta_1 + a_{n-1})\delta_1 + \ldots + a_1)\delta_1 + a_0. \tag{3.5}$$

Assuming that $a_i \neq 0$, the computational effort for the evaluation of $g(\delta)$ is reduced from $2n-1$ multiplications and $n$ additions for evaluating (3.4) to $n$ multiplications and $n$ additions for evaluating (3.5).

In addition to the numerical benefits, the Horner form is also very useful to generate low order LFRs by applying it to the numerator and denominator polynomials of a rational function as proposed in [83]. For the polynomial given in (3.4) the resulting order of the LFR will be $\mathrm{ord}(g(\delta)) = n(n+1)/2$ (see (3.3)), whereas the order of the LFR for the Horner form (3.5) will be $n$, which is minimal. Hence, for a one-dimensional (univariate) polynomial, the Horner form applied as symbolic preprocessing step allows to obtain a minimal order LFR by applying the object-oriented LFR realization approach.

For multivariate polynomials with $k$ parameters there exists no unique conversion to nested Horner form and an exhaustive search for the least order of the corresponding LFTs involves $k!$ conversions. This approach is effective especially when a few variables have significantly larger powers than the rest of variables. Thus, in the case of many variables, exhaustive searches are meaningful only for the few variables with the highest powers. The conversion to Horner form can be easily extended to multivariate Laurent polynomials as well as can be generalized to multivariate polynomial matrices [60].

**Example 3.4.** Consider the polynomial

$$g(\delta) = \delta_1^3 + \delta_1^2 + \delta_1^2\delta_2 + \delta_1 + \delta_2.$$

A realization of this polynomial without any preprocessing would lead to an LFR of order 10. As the polynomial depends on two parameters the nested Horner form is determined recursively with respect to each parameter. Depending on the ordering of the two parameters, two different nested Horner forms are obtained:

$$g(\delta) = (1 + (1 + \delta_1)\delta_1)\delta_1 + (\delta_1^2 + 1)\delta_2 \tag{3.6}$$

$$g(\delta) = \delta_2 + (1 + (\delta_2 + 1 + \delta_1)\delta_1)\delta_1 \tag{3.7}$$

The form in (3.6) is obtained by starting with a Horner factorization for $\delta_1$ and the resulting LFR will have order 6, whereas (3.7) is obtained for the opposite ordering and the resulting LFR will have order 5.

The above example shows that in principle all possible Horner forms must be calculated to obtain the LFR with the lowest order. However, already in the case of this simple multidimensional polynomial the nested Horner form does not allow to find the least order LFR, which is of order 4. Therefore more sophisticated factorization techniques as described in section 3.5 may be applied.

### 3.3.2 Partial fraction decomposition

This conversion allows to represent a rational function in an additively decomposed partial fraction form, where the individual terms have usually much simpler forms.

Consider the univariate rational function $g(\delta_1) = n(\delta_1)/d(\delta_1)$. The partial fraction form is obtained as follows [62]:

1. Factorize $d(\delta_1)$ as
$$d(\delta_1) = (\delta_1 - a_1)^{p_1} \dots (\delta_1 - a_m)^{p_m} c,$$
where $c \in \mathbb{R}$ is a real constant, $a_i \in \mathbb{C}$ are zeros of $d(\delta_1)$ with multiplicities $p_i$.

2. Then $g(\delta_1)$ can be represented as
$$g(\delta_1) = p(\delta_1) + \sum_{i=1}^{m} \sum_{j=1}^{p_i} \frac{A_{ij}}{(\delta_1 - a_i)^j},$$
with $p(\delta_1)$ as a polynomial in $\delta_1$. The $A_{ij}$ are determined by comparison of coefficients.

For multivariate rational functions an iterative realization procedure can be in principle devised by performing the above procedure with respect to a selected order of variables. In each step, the basic procedure is performed on all partial fractions computed at a previous step. For a multivariate rational function with $k$ parameters an exhaustive search for the least order of corresponding LFRs involves $k!$ conversions. The main difficulty of employing this conversion is the need to symbolically compute the roots of multivariate polynomials.

**Example 3.5.** Consider the rational function

$$g(\delta) = \frac{2\delta_1^2 - 7\delta_1 - 3\delta_2 + \delta_2\delta_1^2 + \delta_2^2\delta_1 - \delta_2^2 + 3}{\delta_1^3 + \delta_2\delta_1^2 - 4\delta_1^2 - 4\delta_1\delta_2 + 3\delta_1 + 3\delta_2} \tag{3.8}$$

The partial fraction decomposition leading to the least order LFR is given by

$$g(\delta) = \frac{1}{\delta_1 - 1} + \frac{\delta_2}{\delta_1 - 3} + \frac{1}{\delta_1 + \delta_2}$$

This decomposition results for both orderings of variables and the resulting LFR has order 5 instead of the expected order of 24 for a direct LFR realization of (3.8). Using nested Horner forms of the numerator and denominator polynomials leads to an LFR with order 11.

### 3.3.3 Continued-fraction form

The conversion to continued-fraction form is useful for an efficient numerical evaluation of rational expressions [53] and can also be applied as symbolic preprocessing for rational expressions.

For a univariate rational expression $g(\delta_1)$ the continued fraction form has the following structure

$$g(\delta_1) = p(\delta_1) + \cfrac{1}{c_1\delta_1 + d_1 + \cfrac{1}{c_2\delta_1 + d_2 + \dots}},$$

with $p(\delta_1)$ as polynomial expressions in $\delta_1$ and $c_j, d_j \in \mathbb{R}$.

For a multivariate rational function $g(\delta)$, this conversion is usually performed for a selected parameter $\delta_i$ and the resulting coefficients $c_j, d_j$ depend generally on the rest of the parameters. Although nested representations involving the representation of coefficients in continued-fraction form are in principle possible to be computed, this computation is however not straightforward and can be frequently replaced by conversions to Horner form. The main advantage of this conversion is that it can be performed for arbitrary rational functions. In particular, for any univariate rational function, this conversion allows to obtain the least order LFR.

**Example 3.6.** The continued-fraction form of

$$g(\delta) = \frac{8\delta_1^2 - 8\delta_1 + 2\delta_3 + 2\delta_1\delta_2 - \delta_2 + 2}{4\delta_1^2 - 4\delta_1 + \delta_3 + 1} \tag{3.9}$$

with respect to $\delta_1$ is given by

$$g(\delta) = 2 + \cfrac{\delta_2}{2\left(\delta_1 - \cfrac{1}{2} + \cfrac{\delta_3}{4\left(\delta_1 - \cfrac{1}{2}\right)}\right)},$$

and allows to obtain an LFR of least order 4 instead of expected order 11. Note, that the conversion of (3.9) to partial fraction form allows to obtain an LFR of order 6 and using the nested Horner form for the numerator and denominator polynomial in (3.9) yields an LFR of order 8.

## 3.4 Matrix conversions

### 3.4.1 Morton's method

Any affine parameter dependent matrix $G(\delta)$ can be expressed as an affine combination

$$G(\delta) = G_0 + \sum_{i=1}^{k} \delta_i G_i, \tag{3.10}$$

where $G_i \in \mathbb{R}^{p\times m}$, $i = 0, \dots, k$ are constant matrices. Following (3.3) the resulting LFR order for each term $\delta_i G_i$ would be equal to the number of nonzero elements in $G_i$.

Let $G_i = L_i R_i$ be full rank factorizations of $G_i$, where $L_i \in \mathbb{R}^{p \times r_i}$ and $R_i \in \mathbb{R}^{r_i \times m}$. The method proposed in [64] rewrites $G(\delta)$ as

$$G(\delta) = G_0 + \sum_{i=1}^{k} L_i \delta_i I_{r_i} R_i \tag{3.11}$$

and the resulting LFR can be directly realized as

$$M = \left[\begin{array}{ccc|c} 0 & \dots & 0 & R_i \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & R_k \\ \hline L_1 & \dots & L_k & G_0 \end{array}\right], \Delta = \operatorname{diag}(\delta_1 I_{r_1}, \dots, \delta_k I_{r_k}).$$

The main advantage of this method in obtaining low order LFRs is that it exploits the fact that frequently the constant matrices $G_i$ have non-full ranks $r_i$. Moreover, instead of a maximum multiplicity of $pm$ for $\delta_i$ in $\Delta$ (if there are $pm$ non-zero elements), only a multiplicity of $r_i \leq \max(p, m)$ results. Note that for affine parameter dependent matrices, an object-oriented LFR realization of $G(\delta)$ in the form (3.11) yields a minimal order LFT, which is generally not the case for a direct object-oriented LFR realization of (3.10).

A straightforward extension of this method is considered in [21], where $G(\delta) \in \mathbb{R}(\delta)^{p \times m}$ is a rational matrix and the coefficients in (3.11) are not the $\delta_i$ but rational expressions $c_i(\delta)$. Note, that in this case the resulting LFR is generally not of minimal order, as common factors or dependencies between the different $c_i(\delta)$ are not considered.

**Example 3.7.** Consider

$$G(\delta) = \begin{bmatrix} \delta_1 & \delta_1 \\ 2\delta_1 & 2\delta_1 \end{bmatrix}. \tag{3.12}$$

A direct object-oriented LFR realization of (3.12) yields an LFR with order 4, whereas a first order LFR is obtained for $G(\delta)$ in the following form

$$G(\delta) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \delta_1 \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

### 3.4.2 Enhanced tree decomposition

An efficient technique applicable to multivariate polynomial matrices is the *tree-decomposition* (TD) based approach proposed in [24]. This method exploits the structure of a polynomial matrix to break it down into sums and products of "simple" terms and factors for which low order LFRs can be easily constructed. The TD approach involves the following elementary operations:

**Direct sum decomposition:** Recall that $(s_1, \dots, s_l)$ is a partition of the finite set $s = (1, \dots, k) \subset \mathbb{N}$, if $\bigcup_{i=1}^{l} s_i = s$ and $s_i \cap s_j = \emptyset, \forall i \neq j, 1 \leq i, j \leq l$. $\delta_s$ denotes a parameter vector including all parameters $\delta_i$ with index $i \in s$. $G(\delta)$ has a direct-sum decomposition, if a nontrivial partition $(s_1, \dots, s_l)$ of the parameter index set $s$ exists such that

$$G(\delta) = G(0) + G_1(\delta_{s_1}) + \dots + G_l(\delta_{s_l}). \tag{3.13}$$

Note that the direct sum decomposition reduces the original k-dimensional problem into $l$ independent problems of dimension less than or equal to $k$.

**Affine factorization:** Let $\delta_m$ be a common factor of the $i$th row (respectively column) of $G(\delta)$. Then $G(\delta)$ admits the factorized representation

$$G(\delta) = Q_i(\delta_m)R(\delta)(= \overline{R}(\delta)\overline{Q}_i(\delta_m)),$$

where $Q_i(\delta_m)$ (respectively $\overline{Q}_i(\delta_m)$) is an identity matrix with its $i$th diagonal element substituted by $\delta_m$. The LFR order of $Q_i$ is one and this factorization allows to reduce the overall LFR order of $G(\delta)$ by at least $y-1$, where $y$ is the number of nonzero elements in the $i$th row (column) of $G(\delta)$.

**Weighted sum decomposition:** If none of the above operations can be applied to $G(\delta)$, the TD approach splits $G(\delta)$ as $G(\delta) = G_1(\delta) + G_2(\delta)$, where $G_1(\delta)$ contains all entries depending on one specific parameter $\delta_i$ and $G_2(\delta)$ does not depend on $\delta_i$. The purpose of this operation is to prepare in the next step an affine factorization with respect to $\delta_i$ of the term $G_1(\delta)$, which may potentially lead to order reduction. In particular, one wants to find the parameter $\delta_i$ for which the sum decomposition $G(\delta) = G_1(\delta) + G_2(\delta)$ maximizes the order reduction obtained by the affine factorization of $G_1(\delta)$. This sum decomposition is called weighted-sum decomposition.

**Algorithm 1** TD($G$)

```
1:  T = 0
2:  𝔾 ← Direct-Sum(G)
3:  for S ∈ 𝔾 do
4:    if S not simple then
5:      if S has affine row/column factor then
6:        [Q, R] ← Affine-Factor(S)
7:        T = T+Realize-LFR(TD(Q)*TD(R))
8:      else
9:        [S1, S2] ← Weighted-Sum(S)
10:       [Q, R] ← Affine-Factor(S1)
11:       T = T+Realize-LFR(TD(Q)*TD(R)+TD(S2))
12:     end if
13:   else
14:     T = T+Realize-LFR(S)
15:   end if
16: end for
17: Output= T
```

Algorithm 1 shows a pseudo-code for the TD algorithm, where $\mathbb{G}$ denotes the set of matrices including the additive terms obtained by a direct-sum decomposition (3.13), Realize-LFR($G$) describes the application of the object-oriented LFR realization procedure to the parametric matrix $G$ and "simple" in line 4 of the algorithm means that the object-oriented LFR realization procedure will yield a minimal order LFR for $S$.

The TD approach can be employed to construct LFRs of general rational matrices represented in polynomial fractional forms. It is well-known that any rational matrix $G(\delta) \in \mathbb{R}(\delta)^{p\times m}$ can be expressed as a right or left factorization $G(\delta) = N(\delta)D^{-1}(\delta)$ or $G(\delta) = \widetilde{D}^{-1}(\delta)\widetilde{N}(\delta)$, respectively, where $N(\delta)$, $D(\delta)$, $\widetilde{N}(\delta)$, $\widetilde{D}(\delta)$ are polynomial matrices. Then, from the LFRs of the compound polynomial matrices $[\,N^T(\delta)\ D^T(\delta)\,]^T$ or $[\,\widetilde{N}(\delta)\ \widetilde{D}(\delta)\,]$ one can easily determine the LFRs of $G(\delta)$ by direct formulas (see Lemmas 2.6 and 2.7).

The *enhanced tree decomposition* (ETD) method is an extension of the TD method to the more general case of multivariate Laurent polynomial matrices $G(\delta, \delta^{-1}) \in \mathbb{R}[\delta, \delta^{-1}]$. The enhanced method formally substitutes each reciprocal variable $\delta_i^{-1}$ in the Laurent polynomial matrix $G(\delta, \delta^{-1})$ by a new variable, say $\widetilde{\delta}_i$, and applies the standard TD method to the resulting polynomial matrix $G(\delta, \widetilde{\delta})$. Furthermore, Morton's technique (see section 3.4.1) is integral part of the ETD and is applied in all cases where affine combinations of the form $\sum c_i(\delta)G_i$ arise as intermediate results during the decomposition. In practical examples (see chapters 5 and 6) it could be observed that this occurs very often especially after the direct-sum decomposition steps. In addition to the resulting lower orders of LFRs, the integration of Morton's approach leads usually to significant time savings. In the case of the RCAM example presented in chapter 5, a reduction of about 20% of the LFR realization time has been achieved.

The TD approach as proposed in [24] requires an initial expansion of products into sums of monomials for the multivariate polynomial entries $g_{ij}(\delta, \delta^{-1})$ of $G(\delta, \delta^{-1})$. However, expanding a polynomial of the form $(\delta_1 + \delta_2)^5(\delta_3 + \delta_4)^5$ may yield an LFR of order up to 360 instead of an LFR with minimal order 20, which can be obtained by a direct LFR realization of the polynomial in factorized form. The implementation of the ETD in [47] tries to avoid such expansions by performing direct sum decompositions and factorizations first on the level of whole expressions and not only on the level of monomials of a polynomial in expanded form as proposed in [24]. An example of a highly complex parametric model of a fighter aircraft can be found in [32], where it is shown that ignoring the existing structure of polynomial expressions and expanding the multivariate model equations increases substantially the order of the resulting LFR. Algorithm 2 shows a pseudo-code for the ETD algorithm, where Sub-Reciproc($G$) and Resub-Reciproc($G$) denote the substitution and re-substitution of reciprocal variables $\delta^{-1}$, respectively and SVD($\widetilde{S}$) denotes the singular value decomposition of a numeric matrix $\widetilde{S}$.

**Example 3.8.** To illustrate the importance of the integration of Morton's method into the TD approach, consider again $G(\delta)$ of example 3.7, with

$$G(\delta) = \begin{bmatrix} \delta_1 & \delta_1 \\ 2\delta_1 & 2\delta_1 \end{bmatrix}.$$

After two affine factorizations the TD approach will yield

$$G(\delta) = \begin{bmatrix} \delta_1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \delta_1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix},$$

with a resulting LFR order of two, whereas in example 3.7 an LFR with order one was obtained.

**Algorithm 2** ETD($G(\delta, \delta^{-1})$)

$T = 0$
$\widetilde{G}(\delta, \widetilde{\delta}) \leftarrow$ Sub-Reciproc($G(\delta, \delta^{-1})$)
$\mathbb{G} \leftarrow$ Direct-Sum($\widetilde{G}(\delta, \widetilde{\delta})$)
**for** $S(\delta, \widetilde{\delta}) \in \mathbb{G}$ **do**
  **if** $S(\delta, \widetilde{\delta})$ not simple **then**
    **if** $S(\delta, \widetilde{\delta}) = c(\delta, \widetilde{\delta})\widetilde{S}$ **then**
      $[L, I_r, R] \leftarrow$ SVD($\widetilde{S}$)
      $T = T$+Realize-LFR($L$*ETD($c(\delta, \widetilde{\delta})$)$*I_r * R$)
    **else**
      **if** $S$ has affine factor **then**
        $[Q(\delta, \widetilde{\delta}), R(\delta, \widetilde{\delta})] \leftarrow$ Affine-Factor($S(\delta, \widetilde{\delta})$)
        $T = T$+Realize-LFR(ETD($Q(\delta, \widetilde{\delta})$)*ETD($R(\delta, \widetilde{\delta})$))
      **else**
        $[S_1(\delta, \widetilde{\delta}), S_2(\delta, \widetilde{\delta})] \leftarrow$ Weighted-Sum($S(\delta, \widetilde{\delta})$)
        $[Q(\delta, \widetilde{\delta}), R(\delta, \widetilde{\delta})] \leftarrow$ Affine-Factor($S_1(\delta, \widetilde{\delta})$)
        $T = T$+Realize-LFR(ETD($Q(\delta, \widetilde{\delta})$)*ETD($R(\delta, \widetilde{\delta})$)+ETD($S_2(\delta, \widetilde{\delta})$))
      **end if**
    **end if**
  **else**
    $T = T$+Realize-LFR($S(\delta, \widetilde{\delta})$)
  **end if**
**end for**
Output $\leftarrow$ Resub-Reciproc($T$)

## 3.5 Variable splitting factorization

### 3.5.1 Scalar case

For multivariate Laurent polynomials $g(\delta, \delta^{-1})$, a *variable splitting* (VS) technique can be employed to express such a polynomial in a factored form, where the factors contain disjoint subsets of $\delta$ and $\delta^{-1}$, respectively. It is easy to show that any Laurent polynomial can be expressed as a product

$$g(\delta, \delta^{-1}) = v(\delta_{s_1}, \delta_{s_2}^{-1})^T u(\delta_{s_3}, \delta_{s_4}^{-1}) \tag{3.14}$$

where $v(\delta_{s_1}, \delta_{s_2}^{-1})$ and $u(\delta_{s_3}, \delta_{s_4}^{-1})$ are vectors depending on the parameter vectors $\delta_{s_1}, \delta_{s_2}$ and $\delta_{s_3}, \delta_{s_4}$, respectively. The parameter index sets $s, s_1, \ldots, s_4$ are chosen such that $s = (1, \ldots, k)$, $s = s_1 \cup s_3 = s_2 \cup s_4$, $s_1 \cap s_3 = s_2 \cap s_4 = \emptyset$ and $\delta_s$ denotes a parameter vector including all parameters $\delta_i$ with $i \in s$.

Typically, one chooses one of the factors, say $v(\delta_{s_1}, \delta_{s_2}^{-1})$, to have only entries expressed by multivariate Laurent monomials. The VS factorization allows to transform the initial realization problem into two realization problems, but each with fewer variables. This technique is beneficial for cases, where scalar techniques as transformation to Horner form do not allow a splitting of variables.

**Example 3.9.** Consider

$$g(\delta) = \delta_1 + \delta_1\delta_2 + \delta_2$$

Using transformations to Horner form one obtains either $g(\delta) = \delta_1(1 + \delta_2) + \delta_2$ or $g(\delta) = \delta_1 + (\delta_1 + 1)\delta_2$. Both representations allow to reduce the resulting LFR order from 4 to 3.

By choosing $\delta_{s_1} = \delta_1$ and $\delta_{s_3} = \delta_2$, we obtain the VS factorization as

$$g(\delta) = \begin{bmatrix} \delta_1 & 1 \end{bmatrix} \begin{bmatrix} 1 + \delta_2 \\ \delta_2 \end{bmatrix}$$

which allows an additional decomposition such that

$$g(\delta) = \begin{bmatrix} \delta_1 & 1 \end{bmatrix} \left( \begin{bmatrix} 1 \\ 1 \end{bmatrix} \delta_2 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)$$

yielding an LFR of minimal order 2.

The nice result of example 3.9 can be generalized in the following lemma.

**Lemma 3.1.** Consider the Laurent polynomial $g(\delta, \delta^{-1})$ depending on two variables $\delta_1, \delta_2$, that is, $\delta = (\delta_1, \delta_2)$. A VS factorization of the form

$$g(\delta, \delta^{-1}) = v(\delta_1, \delta_1^{-1})^T u(\delta_2, \delta_2^{-1})$$

allows to realize minimal order LFRs $(M_v, \Delta_v)$, $(M_u, \Delta_u)$ with $\mathcal{F}_u(M_v, \Delta_v) = v(\delta_1, \delta_1^{-1})$, $\mathcal{F}_u(M_u, \Delta_u) = u(\delta_2, \delta_2^{-1})$ and the LFR $(M_g, \Delta_g)$ obtained using (ii) of Lemma 2.1 is a minimal order LFR for $g(\delta, \delta^{-1})$.

**Proof.** A Laurent polynomial $g(\delta, \delta^{-1})$ of the form (3.2) can be VS factorized as

$$g(\delta, \delta^{-1}) = v(\delta_1, \delta_1^{-1})^T u(\delta_2, \delta_2^{-1}) = \begin{bmatrix} c_1\delta_1^{n_{1,1}} & \dots & c_l\delta_1^{n_{l,1}} \end{bmatrix} \begin{bmatrix} \delta_2^{n_{1,2}} \\ \vdots \\ \delta_2^{n_{l,2}} \end{bmatrix}.$$

As negative and positive exponentials both contribute to the order of the LFR, the minimal LFR orders $\mathrm{ord}_{v,\min}$, $\mathrm{ord}_{u,\min}$ for $v(\delta_1, \delta_1^{-1})$, $u(\delta_2, \delta_2^{-1})$, respectively are given by

$$\mathrm{ord}_{v,\min} = \max_i(n_{i,1}, 0) + \max_i(0, -n_{i,1}) \tag{3.15}$$

$$\mathrm{ord}_{u,\min} = \max_i(n_{i,2}, 0) + \max_i(0, -n_{i,2}), \tag{3.16}$$

and the minimal order of an LFR for $g(\delta, \delta^{-1})$ is given by $\mathrm{ord}_{g,\min} = \mathrm{ord}_{v,\min} + \mathrm{ord}_{u,\min}$.

Without loss of generality it is assumed that the entries of $u(\delta_2, \delta_2^{-1})$ are ordered, such that $n_{1,2} < \ldots < n_{r,2} = 0 < \ldots < n_{l,2}$ and $u(\delta_2, \delta_2^{-1})$ can be represented as

$$\begin{aligned} u(\delta_2, \delta_2^{-1}) = \begin{bmatrix} \delta_2^{n_{1,2}} \\ \vdots \\ \delta_2^{n_{r-1,2}} \\ \delta_2^{n_{r,2}} \\ \delta_2^{n_{r+1,2}} \\ \vdots \\ \delta_2^{n_{l,2}} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \delta_2^{-1} \left( \dots + \delta_2^{-1} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \right) \dots \\ + \delta_2 \left( \dots + \delta_2 \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} \right) \dots \end{aligned} \tag{3.17}$$

Using the object-oriented LFR realization method proposed in section 2.3.2, it is obvious that a minimal LFR can be obtained for $u(\delta_2, \delta_2^{-1})$ given in the form (3.17). A minimal order LFR can be found similarly for $v(\delta_1, \delta_1^{-1})$. Applying (ii) of Lemma 2.1 yields an LFR for $g(\delta, \delta^{-1})$ with order $\mathrm{ord}_{g,\min}$.

■

### 3.5.2 Vector case

The VS factorization can be simply extended to parametric vectors of the form $g_{\text{vec}} = \left[\begin{array}{ccc} g_1(\delta,\delta^{-1}) & \dots & g_m(\delta,\delta^{-1}) \end{array}\right] = V^T U$ as

$$g_{\text{vec}} = \left[\begin{array}{c} v_1(\delta_{s_1,1},\delta_{s_2,1}^{-1}) \\ \vdots \\ v_m(\delta_{s_1,m},\delta_{s_2,m}^{-1}) \end{array}\right]^T \left[\begin{array}{ccc} u_1(\delta_{s_3,1},\delta_{s_4,1}^{-1}) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & u_m(\delta_{s_3,m},\delta_{s_4,m}^{-1}) \end{array}\right], \tag{3.18}$$

where $v_i(\delta_{s_1,i},\delta_{s_2,i}^{-1})$ and $u_i(\delta_{s_3,i},\delta_{s_4,i}^{-1})$, $i = 1,\dots,m$, are parametric vectors.

A limitation of the simple vector extension of the VS factorization is that the factor $U$ in (3.18) does not allow to exploit any dependencies and common factors of the $u_i(\delta_{s_3,i},\delta_{s_4,i}^{-1})$ for the reduction of the resulting LFR order. This may be simply improved by applying the following *condensation algorithm*, which condenses the factors $V$ and $U$ of (3.18) in cases where elements of $V$ only differ by a constant factor $c \in \mathbb{R}$. Therefore, we introduce a matrix $P_{ij,c}$, which is an identity matrix of appropriate dimension, where the element $p_{ij}, i \neq j$ is substituted by a constant $c \in \mathbb{R}$ and the $j$th row is removed. A left multiplication of $U$ with $P_{ij,c}$ yields $\widetilde{U}$, where $c$ times the $j$th row of $U$ is added to the $i$th row of $U$ and the $j$th row of $U$ is removed. As an example, consider $U = \text{diag}(u_1,u_2,u_3)$. Left multiplying $U$ with $P_{12,4}$ yields

$$P_{12,4}U = \left[\begin{array}{ccc} 1 & 4 & 0 \\ 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc} u_1 & 0 & 0 \\ 0 & u_2 & 0 \\ 0 & 0 & u_3 \end{array}\right] = \left[\begin{array}{ccc} u_1 & 4u_2 & 0 \\ 0 & 0 & u_3 \end{array}\right].$$

**Algorithm 3** Condensation_Vec$(V,U)$

```
L=length(V)
if L > 1 then
  for i = 1 to L − 1 do
    for j = (i + 1) to L do
      if V(j) = cV(i) then
        Remove V(j) from V
        U = P_{ij,c} U
        L = L − 1
      end if
    end for
  end for
end if
```

After the application of algorithm 3, the condensed version of $U$ may now have more than one non-zero entry in a row. If these entries have common factors, a further reduction of the resulting LFR order may be achieved.

**Example 3.10.** Consider the VS factorization of the form (3.18) for the parametric vector $g_{\text{vec}}$ given by

$$g_{\text{vec}} = \begin{bmatrix} 2\delta_1\delta_2 + 2\delta_2 & \delta_1\delta_2 + \delta_1\delta_3 + \delta_2 \end{bmatrix} = \begin{bmatrix} 2\delta_1 \\ 2 \\ \delta_1 \\ 1 \end{bmatrix}^T \begin{bmatrix} \delta_2 & 0 \\ \delta_2 & 0 \\ 0 & \delta_2 + \delta_3 \\ 0 & \delta_2 \end{bmatrix} \tag{3.19}$$

where a application of the ETD yields

$$g_{\text{vec}} = \left( \delta_1 \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T + \begin{bmatrix} 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}^T \right) \left( \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_2 & 0 \\ 0 & \delta_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \delta_3 \\ 0 & 0 \end{bmatrix} \right),$$

allowing to obtain an LFR of order 4 instead of order 8 for a direct realization of $g_{\text{vec}}$ without symbolic preprocessing. However, applying algorithm 3 after generating the VS factorization of the form (3.18) yields

$$g_{\text{vec}} = \begin{bmatrix} \delta_1 & 1 \end{bmatrix} \begin{bmatrix} 2\delta_2 & \delta_2 + \delta_3 \\ 2\delta_2 & \delta_2 \end{bmatrix}$$

and further application of the ETD yields

$$g_{\text{vec}} = \begin{bmatrix} \delta_1 & 1 \end{bmatrix} \left( \begin{bmatrix} 1 \\ 1 \end{bmatrix} \delta_2 \begin{bmatrix} 2 & 1 \end{bmatrix} + \begin{bmatrix} 0 & \delta_3 \\ 0 & 0 \end{bmatrix} \right),$$

allowing to obtain an LFR of minimal order 3.

### 3.5.3 Matrix case

The VS factorization for vectors can be simply extended to parametric matrices as

$$G = \begin{bmatrix} g_{\text{vec},1} \\ \vdots \\ g_{\text{vec},p} \end{bmatrix} = \overline{V}\,\overline{U} = \begin{bmatrix} V_1^T & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & V_p^T \end{bmatrix} \begin{bmatrix} U_1 \\ \vdots \\ U_p \end{bmatrix}. \tag{3.20}$$

The matrix $\overline{V}$ has only one non-zero element in each column. In analogy the vector case of the VS factorization, a condensation of the matrices $\overline{V}$ and $\overline{U}$ in (3.20) may be possible. The condensed version of matrix $\overline{V}$ may have more than one non-zero entry in a column, which can be further exploited to reduce the resulting LFR order. The condensation of $\overline{V}$ and $\overline{U}$ is performed using algorithm 4, where $\overline{P}_{ji,c}$ is an identity matrix of appropriate dimension, where the element $\overline{p}_{ji}, i \neq j$ is substituted by a constant $c \in \mathbb{R}$ and the $j$th column is removed. The $i$th row of $\overline{U}$ is denoted as $\overline{U}(i)$.

**Algorithm 4** Condensation_Mat($\overline{V}, \overline{U}$)

```
1: L=rowlength(Ū)
2: if L > 1 then
3:   for i = 1 to L − 1 do
4:     for j = (i + 1) to L do
5:       if Ū(j) = cŪ(i) then
6:         Remove row Ū(j) from Ū
7:         V̄ = V̄ P̄_{ij,c}
8:         L = L − 1
9:       end if
10:    end for
11:  end for
12: end if
```

## 3.6 Lower-bound for LFR order

As already mentioned, in multidimensional-system theory there exists no solution for the generation of a minimal order LFR for a multivariate parametric matrix. However, in order to quantify the complexity of an LFR, which is obtained after application of symbolic preprocessing, object-oriented LFR realization and exact numerical order reduction, it is important to have at least a lower bound for the achievable minimal LFR order.

A simple procedure to determine a lower bound for the LFR order of a polynomial parametric matrix $G(\delta)$ is to determine for each parameter $\delta_i$ the maximum power $m_{\delta_i}$ over all matrix entries $g_{i,j}(\delta)$. A lower bound is then given by $\sum_{i=1}^{k} m_{\delta_i}$. As an example, a lower bound of $s$ is obtained for the parametric vector $G(\delta) = \begin{bmatrix} \delta_1^s & \delta_1^{s-1} & \dots & \delta_1 \end{bmatrix}$, where the maximum power of $\delta_1$ over all vector entries is $s$. In this case the lower bound is exact and describes the minimal LFR order for this parametric vector. However, the same lower bound is obtained for the parametric matrix $G(\delta) = \mathrm{diag}(\delta_1^s, \delta_1^{s-1}, \dots, \delta_1)$, where the minimal LFR order is $s(s+1)/2$. This shows that for parametric matrices, this method may result in a very bad estimate for the minimal LFR order, which comes from the fact, that the structural information for the occurrence of the parametric expressions in the matrix is not considered.

To overcome this problem the following procedure to calculate a more accurate lower bound for the minimal LFR order is proposed:

1. Set counter $i = 1$.
2. Substitute all parameters in $G(\delta)$, except $\delta_i$, with random values resulting in the one-parametric matrix $G_i(\delta_i)$.
3. Construct a minimal order LFR $(M_i, \Delta_i = \delta_i I_{m_i})$ for $G_i(\delta_i)$.
4. If $i < k$ then increment $i$ and go to step 3, otherwise got to step 5.
5. The lower bound is given by $m = \sum_{i=1}^{k} m_i$.

The proposed procedure can be easily implemented and for practical examples it yields quite good estimates (see chapters 5 and 6). For the above mentioned matrix $G(\delta) = \mathrm{diag}(\delta_1^s, \delta_1^{s-1}, \ldots, \delta_1)$, the lower bound will now be exactly the minimal LFR order. The following example shows, that in some cases a gap between the lower bound calculated with proposed procedure and the exact minimal LFR order can not be avoided and to the best of the authors knowledge there exists no method to exactly calculate the minimal LFR order.

**Example 3.11.** Consider the parametric vector $G(\delta) = \begin{bmatrix} \delta_1\delta_2 & \delta_1 + \delta_2 \end{bmatrix}$. In this case the procedure yields a lower bound of $m = 2$. However, there will exists no LFR of order less than 3.

# 4 Enhanced LFR-toolbox for MATLAB

The LFR-toolbox is a MATLAB toolbox for the realization of LFRs for uncertain system models. With this toolbox, LFRs can be directly obtained from symbolic expressions or via object-oriented manipulation of LFR objects.

The version 1 of the LFR-toolbox has been implemented by the author of [57] and generates standard LFRs for systems with structured (parametric) or unstructured uncertainties of real or complex type. The generation of low order LFRs is supported in various ways. Special functions for symbolic preprocessing techniques as Mortons method [64] for affine uncertainty representations and the tree decomposition [24] for polynomial matrices are provided. Furthermore numerical multidimensional order reduction and approximation methods [29, 77] for LFRs are available. These algorithms rely on standard minimal realization tools available in the Control Toolbox of MATLAB.

The present version 2 of the LFR-toolbox [46, 45, 47] includes major enhancements, which are mainly focused to improve the capabilities for low order LFR modelling. With the support of the generalized LFR [42], described in section 2.3.5 it is now possible to realize arbitrary rational expressions as LFRs. Furthermore, the new LFR object definition is more transparent, user friendly and supports additional types of uncertainties to be directly compatible to other MATLAB toolboxes like the $\mu$-Analysis and Synthesis toolbox [12], the LMI toolbox [37] and the Robust Control toolbox [11]. Significant enhancements of the computational efficiency and of numerical accuracy have been achieved by employing efficient and numerically robust Fortran implementations of order reduction tools via *mex*-function interfaces. The new enhancements in conjunction with improved symbolical preprocessing lead generally to a faster generation of LFRs with significantly lower orders.

Figure 4.1 shows the contributions of DLR (Deutsches Zentrum für Luft-und Raumfahrt) and ONERA (Office nationale d'Études et de Recherches Aérospatiales) to version 2 of the LFR-toolbox, which was developed within the common research project HA-FUN (Handling of Flight Uncertainties). The contributions of the author including a new implementation of the underlying MATLAB LFR object, the related object-oriented manipulation functions, symbolic preprocessing routines and numerical order reduction routines are briefly described in the following sections. From ONERA side, interfaces for graphical robust stability/performance analysis and interfaces to use LFR objects within Simulink are provided. In addition, functions for the synthesis of automatically gain-scheduled controllers based on eigenstructure assignment were developed.

For a list of all functions see appendix C.

## 4.1 Object definition

The core function `lfr` to create an LFR object is called inside almost all functions of the toolbox. An LFR object `L` can be created with the command

```
L = lfr(A,B,C,D,blk);
```

where the first four input arguments specify the LFR system matrices $A, B, C, D$ (submatrices of $M$) and the fifth argument `blk` describes the block-diagonal structure of $\Delta$. The argument `blk` is a structure array with two fields, `names` and `desc`, containing, respectively, the names associated to the diagonal blocks of $\Delta$ and the corresponding uncertainty type description. The five input arguments can be recovered from the object `L` as the fields `L.a`, `L.b`, `L.c`, `L.d`, and `L.blk`, respectively.

As an example, the fields `names` and `desc` of the structure description argument of an LFR object with $\Delta = \mathrm{diag}(\delta_1 I_2, \delta_2)$ can be specified as

```
blk.names = {'d1','d2'};
blk.desc = [ 2  1     % row-dimension of blocks
             2  1     % column-dimension of blocks
             1  1     % real(1) / complex(0) blocks
             1  1     % scalar(1) / full(0) blocks
             1  1     % linear(1) / nonlinear(0) blocks
             1  1     % time-inv.(1) / time-var.(0) blocks
             1  1     % min/max(1)/sector(2)/freq.(>2) bound
             2  2     % min/max(2)/sector(1)/freq.(>2) bound
            -1 -1     % minimum value of bounds
             1  1 ] ; % maximum value of bounds
```

where `blk.names` is a cell-array of two strings containing the names `'d1'` and `'d2'` given to the two diagonal blocks of $\Delta$, and the values in each column of the real array `blk.desc` specifies the corresponding information describing each diagonal block (see below).

Enhanced LFR-toolbox

| DLR | ONERA |
|---|---|
| • new, user-friendly Matlab LFR-object supporting generalized LFRs | • LFT-based controller synthesis |
| • efficient and reliable numerical algorithms for order reduction | • interfaces for graphical stability and performacne analysis |
| • symbolic preprocessing and model manipulations in Maple with interfaces to Matlab | • interfaces for Simulink |

Figure 4.1: Contributions to Enhanced LFR-toolbox

Each block in $\Delta$ is uniquely identified by its name, which makes the manipulation of LFR objects flexible and transparent. For example, additional uncertainties can be easily added to an LFR object and the already defined block names can be modified (e.g., by using the function `set`). The special names `'1/s'` and `'1/z'` are reserved for the integrator block $I/s$ (continuous-time systems) and the delay block $I/z$ (discrete-time systems), respectively. These blocks can be included in $\Delta$ to represent standard linear time-invariant systems (continuous- or discrete-time) as LFR objects. Furthermore the special name `'1'` is reserved for a constant identity matrix block $I$ in $\Delta$. This block plays a major role in representing arbitrary rational parametric expressions as LFRs. An internal LFR object reordering (function `reorderlfr`) is performed after each LFR object manipulation where the constant block (if exists) is placed in the first block diagonal position of $\Delta$, the integrator/delay block (if exists) is placed in the second block diagonal position followed by all the uncertainty blocks in a lexicographic order.

For each name in the field `names` there exists a corresponding column in the field `desc`, which describes the row/column dimensions and properties of this block. The LFR object supports real or complex structured uncertainty (or dynamic) blocks and real or complex full unstructured uncertainty blocks. These blocks can have the properties linear/nonlinear and time-invariant/time-varying (in the case of nonlinear uncertainties the property time-invariant means memoryless). Furthermore, the field `desc` includes bound information for each uncertainty block, which can be described by min/max-values, a sector bound (for nonlinear uncertainties) or a SISO frequency dependent bound.

For MATLAB versions 6.1/6.5, conversions to LFR objects from LTI-objects of the Control Toolbox, PCK-system representations of the $\mu$-Synthesis Toolbox as well as constant matrices, are automatically performed via the core function `lfr`. For MATLAB versions 7.0/7.1, conversions from `umat`, `ureal`, `ucomplex` and `uss`-objects of the Robust Control Toolbox to LFR objects are supported. Furthermore, one may use the functions `lfr2mu`, `lfr2mubnd`, `lfr2mussv`, `lfr2mustab` and `lfr2lmip` to generate the required data structures for robust stability/performance analysis and controller synthesis using the $\mu$-Analysis and Synthesis toolbox or the LMI-toolbox under MATLAB 6.1/6.5 or the function `lfr2rob` for conversions to the objects supported by the Robust Control toolbox under MATLAB versions 7.0/7.1.

## 4.2 Symbolic preprocessing

All the methods for decomposition of multivariate rational functions and matrices described in chapter 3 are supported by the function `sym2lfr` of the toolbox. The function is called with several options.

Besides the Horner form, partial fraction form, continued fraction form and the ETD the function also allows to choose code generation techniques for optimized evaluation of polynomial/rational functions as symbolic preprocessing [83]. Therefore the Maple function `optimize` from the `codegen` package is employed. A *tryhard* option can be chosen to check all possible permutations for the Horner, partial fraction and continued fraction forms and the LFR with the lowest resulting order is provided as output.

For the VS factorization the user can choose between a separate factorization of each

matrix element or a column/row wise application as described in section 3.5.3 (for row-wise factorization). The resulting two factors (see (3.20)) are then further processed with the ETD. The condensation of the factors as described in sections 3.5.2 and 3.5.3 is automatically performed. The sets $\delta_{s,i}$ for the definition of the VS factorization can be determined automatically, manually or one may choose a *tryhard* option, where all possible combinations are calculated and the LFR with the lowest resulting order is provided as output.

When the ETD is chosen, a standard feature of the function `sym2lfr` is to split a rational matrix $G(\delta)$ into a pure rational part $G_1(\delta)$ and a Laurent polynomial part $G_2(\delta, \delta^{-1})$, such that $G(\delta) = G_1(\delta) + G_2(\delta, \delta^{-1})$. The matrix $G_2(\delta, \delta^{-1})$ can be directly processed, whereas for $G_1(\delta)$, a left (or right) fractional representation of the form $G_1(\delta) = D^{-1}(\delta)N(\delta)$ (or $G_1(\delta) = \overline{N}(\delta)\overline{D}^{-1}(\delta)$) is calculated and the ETD is applied to the concatenated polynomial matrix $\left[\begin{array}{cc} N(\delta) & D(\delta) \end{array}\right]$ (or $\left[\begin{array}{cc} \overline{N}(\delta)^T & \overline{D}(\delta)^T \end{array}\right]^T$). Finally, Lemma 2.6 (or Lemma 2.7) are used to obtain an LFR for $G_1(\delta)$. It is possible to avoid the splitting of $G(\delta)$ and to calculate a left (or right) fractional representation for the whole matrix. However, the calculation of common denominators, which is necessary to obtain the fractional representation, may result in more complex symbolic expressions and this may increase the resulting LFR order.

To increase the efficiency of the symbolic calculations, many of the core functions are directly implemented as Maple functions and executed within the efficient Maple kernel via the Extended Symbolic Toolbox of MATLAB.

## 4.3 Numerical order reduction

Efficient and numerically reliable tools for order reduction of LFRs are of primary importance to ease the usability of such models. To achieve efficiency of computation, numerical robustness and a high accuracy of results, the toolbox relies on Fortran based robust implementations of algorithms for basic computations related to order reduction. A language like Fortran allows to easily exploit all structural features of a computational problem with low additional computational effort and minimum memory usage. Fortran routines can be easily executed within the user friendly environment MATLAB via external functions, the so called *mex*-functions. Several *mex*-functions based on powerful Fortran routines from the LAPACK-based [5] public domain control library SLICOT [15, 81] form the order reduction computational kernel of the LFR-toolbox.

The LFR-toolbox provides several order reduction tools for exact or approximative reduction of order. The *exact* 1-d order reduction technique [77] can be performed using the function `minlfr1` which is based on the efficient ($O(n^3)$ complexity, with $n$ as the order of a state-space system) SLICOT-based *mex*-function `ssminr`[1] for the calculation of minimal realizations. Note that a pure MATLAB-based implementation using the MATLAB Control Toolbox function `minreal` would have a $O(n^4)$ worst-case complexity. For a comparison of computation times see chapter 5.

The *approximative* 1-d order reduction [83] can be performed using `redlfr1`, which is

[1] The functions `balsys`, `ssminr`, `sysred`, `sscof` and partly `minlfr` have been implemented by A. Varga

based on the collection of model reduction tools available in SLICOT [81], covering the balanced truncation, singular perturbation approximation and Hankel-norm approximation approaches. All these methods are implemented in a single *mex*-function `sysred`[1] which is called by `redlfr1` to cyclically reduce 1-d systems (assimilated to discrete-time). With an appropriate scaling of the $A$ matrix of the LFRs (necessary to ensure stability in discrete-time sense), this function can be also employed to perform *exact* order reduction.

The function `minlfr` can be used for n-d order reduction [29]. In version 2 of the LFR Toolbox this function has been completely re-implemented to improve efficiency. The calculation of the n-d controllability/observability staircase forms relies on the $O(n^3)$ complexity SLICOT-based *mex*-function `sscof`[1] to compute controllability/observability staircase forms using orthogonal transformations. Note that a pure MATLAB-based implementation using the MATLAB Control Toolbox function `ctrbf` would have a $O(n^4)$ worst-case complexity. For a comparison of computation times see chapter 5.

The SLICOT-based *mex*-function `balsys` is systematically called in all order reduction functions to perform a system scaling of the LFRs as a preliminary operation within the order reduction routines. As the LFRs resulting from the object-oriented realization approach can have matrices with a wide range of values, this operation is essential before computing numerically sensitive controllability staircase forms.

The order reduction functions can be applied manually at any stage of the LFR realization or can be executed automatically after each object-oriented LFR manipulation (e.g., multiplication, addition, etc.). To set global options (e.g., to perform or not automatic order reduction), the function `lfropt` can be used. This function basically defines a set of global variables to control the order reduction and to set the associated tolerances.

# 5 Robust stability analysis for the RCAM


Figure 5.1: Airbus A300-600ST Beluga

In the GARTEUR (Group for Aeronautical Research and Technology in Europe) Action Group 8 on Robust Flight Control [58] it was intended to to demonstrate to European aircraft manufacturers, that significant improvements in the control design process can be achieved by the application of modern robust control techniques. It was not the aim to produce an optimal control law, but to show that modern robust control theory can be applied to realistic problems and to show the limitations of these techniques.

One of the benchmark problems was the design of an autopilot for the final segments of a landing approach for a fictitious transporter aircraft (closely related to the Airbus Beluga, see figure 5.1), which is referred to as RCAM (Research Civil Aircraft Model). The control laws had to be robust with respect to variation of the speed, weight, vertical and horizontal position of center of gravity, time delays, nonlinearities and engine-failures. Furthermore disturbance decoupling had to be guaranteed to track a predefined flight path within certain tolerances.

The RCAM is given as a six degree of freedom nonlinear model, including nonlinearities of actuators (saturations) and a model of disturbances as proposed by Aérospatiale (now part of Airbus). The model, used for stability analysis has 5 inputs, 12 states and 15 measured outputs, which are described in the following tables.

Table 5.1: RCAM input definition

| Input | Description | Unit |
|---|---|---|
| $u(1)$ | aileron deflection | rad |
| $u(2)$ | tailplane deflection | rad |
| $u(3)$ | rudder deflection | rad |
| $u(4)$ | throttle position of engine 1 | rad |
| $u(5)$ | throttle position of engine 2 | rad |

Table 5.2: RCAM state definition

| Input | Description | Unit |
|---|---|---|
| $x(1)$ | roll rate (in $F_B$) | rad/s |
| $x(2)$ | pitch rate (in $F_B$) | rad/s |
| $x(3)$ | yaw rate (in $F_B$) | rad/s |
| $x(4)$ | roll angle (Euler angle) | rad |
| $x(5)$ | pitch angle (Euler angle) | rad |
| $x(6)$ | heading angle (Euler angle) | rad |
| $x(7)$ | $x$ component of inertial velocity in $F_B$ | m/s |
| $x(8)$ | $y$ component of inertial velocity in $F_B$ | m/s |
| $x(9)$ | $z$ component of inertial velocity in $F_B$ | m/s |
| $x(10)$ | $x$ position of aircraft CoG in $F_E$ | m |
| $x(11)$ | $y$ position of aircraft CoG in $F_E$ | m |
| $x(12)$ | $z$ position of aircraft CoG in $F_E$ | m |

The notations $F_E, F_B, F_V$ denote the earth-fixed, body-fixed and vehicle-carried reference frames, respectively. For more details see [58].

The uncertain parameter vector $\delta$ of the RCAM is defined as $\delta = (m, V, X_{cg}, Z_{cg})$, where $m$ is the mass of the aircraft, $V$ is the airspeed and the horizontal and vertical position of the center of gravity are given by $X_{cg}$ and $Z_{cg}$, respectively. The corresponding ranges of variation are summarized in Table 5.4.

The nominal values of the parameters are assumed to be centered within their variation range, that is, $V_{\text{nom}} = 80\,\text{m/s}$, $m_{\text{nom}} = 125000\,\text{kg}$, $X_{cg_{\text{nom}}} = 1.525\,\text{m}$, $Z_{cg_{\text{nom}}} = 0.695\,\text{m}$.

In [84] an automated procedure for the generation of LFRs for the RCAM is presented. Starting from a nonlinear, parametric description of the form (2.1), trimming and symbolic linearization is performed to obtain an explicit, linear, rational parametric state space system of the form (2.7). To increase the validity of the linear model, all entries in the linear system matrices, that explicitly depend on the trim solutions $x_t(\delta)$ and $u_t(\delta)$ of states and inputs, respectively, are further corrected by polynomial functions. Textbook information was used to get hints about suitable parametric dependencies of the correction terms. With this approach the validity of the linear system description was increased over the whole range of flight conditions and parameter variations. The corresponding linear, rational parametric system matrices can be found in appendix D and represents

Table 5.3: RCAM output definition

| Output | Description | Unit |
|---|---|---|
| $y(1)$ | $x(2)$ | rad/s |
| $y(2)$ | horizontal load factor in $F_B$ | - |
| $y(3)$ | vertical load factor in $F_B$ | - |
| $y(4)$ | $z$ component of inertial velocity in $F_V$ | m/s |
| $y(5)$ | $x(12)$ | m |
| $y(6)$ | air speed | m/s |
| $y(7)$ | total inertial velocity | m/s |
| $y(8)$ | angle of sideslip | rad |
| $y(9)$ | $x(1)$ | rad/s |
| $y(10)$ | $x(3)$ | rad/s |
| $y(11)$ | $x(4)$ | rad |
| $y(12)$ | $x$ component of inertial velocity in $F_V$ | m/s |
| $y(13)$ | $y$ component of inertial velocity in $F_V$ | m/s |
| $y(14)$ | $x(11)$ | m |
| $y(15)$ | inertial track angle | rad |

Table 5.4: Parameter ranges of variation

| Parameter | Range | Unit |
|---|---|---|
| $m$ | $[100000, 150000]$ | kg |
| $V$ | $[70, 90]$ | m/s |
| $X_{cg}$ | $[1, 2.05]$ | m |
| $Z_{cg}$ | $[0, 1.39]$ | m |

one of the most complicated parametric models available in literature.

In [58] the PUM (Parametric Uncertainty Modelling) Toolbox for MATLAB [54] was employed to develop LFRs for the RCAM. The PUM-Toolbox supports object-oriented LFR realization and 1-d repeated order reduction [77]. Using this tool, two LFRs with the following block structures $\Delta = \text{diag}(mI_{r_1}, VI_{r_2}, X_{cg}I_{r_3}, Z_{cg}I_{r_4})$ and total order $r$ were realized:

Table 5.5: Orders of LFRs realized in GARTEUR AG 08

| Model | $\{r_1, r_2, r_3, r_4\}$ | r |
|---|---|---|
| I | $\{17, 0, 15, 3\}$ | 35 |
| II | $\{47, 109, 30, 7\}$ | 193 |

Model II was realized after application of the object-oriented LFR realization procedure followed by 1-dimensional, exact, numerical order reduction, which results in an LFR of order 193. For robust stability analysis using $\mu$-analysis the total order of model II

was too large, which lead to numerical problems using the available $\mu$-analysis software. Therefore, in [58] it was decided to perform the stability analysis for RCAM only for fixed nominal speed $V = V_{\text{nom}} = 80\,\text{m/s}$. Substituting the uncertain parameter $V$ with its nominal value in the parametric system matrices allowed to realize model I, which was of order 35. However, the results from stability analysis were overly optimistic, as the variation in speed was not considered. This could be deduced as in parallel to the $\mu$-analysis, an optimization based worst-case search considering also variations in speed was applied. For some controllers, that were developed in [58], the $\mu$-analysis indicated robust stability, whereas the worst-case search detected parameter combinations that resulted in unstable closed-loop systems.

In the following section, the methods for low order LFR realization as described in the previous chapters will be applied to realize an almost least order LFR for the RCAM. The recent version 2 of the LFR-toolbox (see chapter 4 or [47]) will be employed to generate all the following LFRs. The final LFR will include all uncertainties of the RCAM and it will be of reasonable order for robust stability analysis using $\mu$. The results of the $\mu$-analysis using the new low order LFR will be comparable to the worst case-search results, thus showing that methods for low order LFR realization are of paramount importance for successful application of $\mu$-analysis, which then can be seen as a fast complement to the time-consuming optimization-based worst-case search.

## 5.1 LFR model realization for the element $\mathbf{a_{29}}$

As a starting point, the effectiveness of the proposed methods for low order LFR realization will be demonstrated by comparing different techniques for realizing an LFR for the most complicated element $a_{29}$ of the parametric system matrix $A(\delta)$. The expression of $a_{29}$ can be put into the form $a_{29} = 0.061601\frac{\tilde{a}_{29}}{C_w V}$, where

$$
\begin{aligned}
C_w =& \frac{mg}{\frac{1}{2}\rho V^2 S},\\
\tilde{a}_{29} =& 1.6726 X_{cg} C_w^2 Z_{cg} - 0.17230 X_{cg}^2 C_w - 3.9324 X_{cg} C_w Z_{cg}\\
&- 0.28903 X_{cg}^2 C_w^2 Z_{cg} - 0.070972 X_{cg}^2 Z_{cg} + 0.29652 X_{cg}^2 C_w Z_{cg}\\
&+ 4.9667 X_{cg} C_w - 2.7036 X_{cg} C_w^2 + 0.58292 C_w^2 - 0.25564 X_{cg}^2\\
&- 1.3439 C_w + 100.13 X_{cg} - 14.251 Z_{cg} - 1.9116 C_w^2 Z_{cg}\\
&+ 1.1243 X_{cg} Z_{cg} + 24.656 C_w Z_{cg} + 0.45703 X_{cg}^2 C_w^2 - 46.850,
\end{aligned}
$$

and $S = 260\text{m}^2$ (wing planform area), $g = 9.81\text{m/s}^2$ and $\rho = 1.225\text{kg/m}^3$ (air density). Note, that the expression of $a_{29}$ is "singular" in the parameters $m$ and $V$ and therefore symbolic normalization is obligatory for generation techniques relying on standard LFRs.

Performing symbolic normalization and expansion of $a_{29}$, the direct application of the object-oriented LFR realization procedure yields an LFR of order 405. In [83] the same procedure was applied followed by a splitting of $a_{29}$ into a numerator and denominator polynomial. The polynomials were realized individually and after division of the resulting LFRs the final LFR was of order 293. This order reduction (from 405 to 293) resulted from factorizations that were implicitly performed in Maple 5 by calculating the numerator

and denominator polynomials. It was tried to reconstruct the LFR with order 293 of [83], but with the more recent version 8 of Maple, the same symbolic steps combined with the object-oriented LFR realization yield an LFR of order 193.

When using the generalized LFR (including a constant block in $\Delta$), the preliminary normalization can be avoided. The generated LFR for the expanded expression of $a_{29}$ has a total order $r$ of 69. This illustrates clearly that a preliminary normalization has often the effect to increase substantially (almost seven times in this example) the order of the generated LFR. Details about the models are summarized in Table 5.6.

Table 5.6: Initial LFR orders for element $a_{29}$

| Model | $\{r_1, r_2, r_3, r_4\}$ | r |
|---|---|---|
| III | $\{84, 240, 54, 27\}$ | 405 |
| IV | $\{49, 136, 72, 36\}$ | 293 |
| V | $\{31, 81, 54, 27\}$ | 193 |
| VI | $\{12, 30, 18, 9\}$ | 69 |

### 5.1.1 Enhanced numerical order reduction

To illustrate the enhancements of the numerical order reduction capabilities of version 2 of the LFR toolbox (see section 4.3), numerical 1-d and n-d order reductions are performed on Model V, using pure *m*-function based implementations (M) and *mex*-function based implementations (MEX) of the order reduction tools. In Table 5.7 the computational times resulted on a PC with a 1.2 GHz AMD ATHLON processor running MATLAB 6.5 under Windows NT are given.

Table 5.7: Order reduction results for Model V

| Method | Time [s] | $\{r_1, r_2, r_3, r_4\}$ | $r$ |
|---|---|---|---|
| 1-d (M) | 9.61 | $\{5, 28, 2, 9\}$ | 44 |
| 1-d (MEX) | 0.1 | $\{5, 7, 2, 4\}$ | 18 |
| n-d (M) | 0.54 | $\{5, 7, 2, 3\}$ | 17 |
| n-d (MEX) | 0.13 | $\{5, 7, 2, 3\}$ | 17 |

A significant reduction of computational times for both the 1-d reduction (almost 100 times faster!) and the n-d reduction (more than four times faster) can be observed. Note also that for this example, the 1-d reduction performed using the *mex*-file based implementation led to a much smaller order than the pure *m*-file based implementation.

### 5.1.2 Comparison of low order LFR realization techniques

Without any symbolic preprocessing an LFR of order 69 (Model VI in Table 5.6) could be generated using the object-oriented LFR realization procedure based on the generalized

LFR. In Table 5.8, several results involving symbolic preprocessing as a preliminary step before the object-oriented LFR realization are presented, where for each specific symbolic method the resulting total orders without and with additional exact numerical n-d order reduction [29] are given in the successive columns. For the partial-fraction form(Parfrac), continued-fraction form (Confrac), Horner form and the VS factorization, all possibilities (parameter permutations or combinations) are calculated (tryhard-option of `sym2dlfr` function, see chapter 4) and the LFR with the lowest order is taken.

The total orders $r$ without further exact numerical order reduction are ranging from 48 to 11, however it is very interesting to see that sometimes a symbolic method that achieves a better initial LFR order does not allow to obtain a better result after the exact numerical order reduction. As an example the ETD allows to directly realize an LFR of order 23, which can be further reduced to order 14, whereas the continued fraction form only yields an LFR of order 34 but this LFR can also be reduced to order 14. This shows that the symbolic preprocessing methods should always be evaluated in combination with the exact n-d numerical order reduction. It is also very interesting that the ETD technique, which is specially suited for Laurent polynomial expressions, clearly outperforms the standard Tree Decomposition (TD), which is applied to a polynomially factorized representation of $a_{29}$ as proposed in [24].

Table 5.8: Reduced LFR model orders for element $a_{29}$

| Preprocessing | $r$ | $\{r_1, r_2, r_3, r_4\}$ | $r$(red) | $\{r_1, r_2, r_3, r_4\}$(red) |
|---|---|---|---|---|
| Parfrac | 48 | $\{9, 23, 7, 9\}$ | 24 | $\{5, 9, 5, 5\}$ |
| Confrac | 34 | $\{2, 5, 18, 9\}$ | 14 | $\{2, 5, 4, 3\}$ |
| Optimize | 33 | $\{4, 10, 18, 1\}$ | 16 | $\{4, 9, 2, 1\}$ |
| TD | 31 | $\{5, 13, 6, 7\}$ | 21 | $\{3, 11, 4, 3\}$ |
| Horner | 26 | $\{4, 7, 6, 9\}$ | 13 | $\{4, 4, 4, 1\}$ |
| ETD | 23 | $\{3, 6, 8, 6\}$ | 14 | $\{3, 6, 3, 2\}$ |
| VS+ETD | 11 | $\{2, 4, 3, 2\}$ | 11 | $\{2, 4, 3, 2\}$ |

The best result is obtained by the combined VS and ETD approach. The VS factorization of $a_{29}(\delta) = v(\delta_{s_1}, \delta_{s_2}^{-1})^T u(\delta_{s_3}, \delta_{s_4}^{-1})$ with $\delta_{s_1} = \delta_{s_2} = \{m, V\}$ and $\delta_{s_3} = \delta_{s_4} = \{X_{cg}, Z_{cg}\}$ yields

$$v(\delta_{s_1}, \delta_{s_2}^{-1}) = \begin{bmatrix} \frac{V}{m} \\ \frac{m}{V^3} \\ \frac{1}{V} \end{bmatrix}, \quad u(\delta_{s_3}, \delta_{s_4}^{-1}) = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

with

$$\begin{aligned} u_1 &= -46.849 + 100.133X_{cg} - 14.2516Z_{cg} \\ &\quad -0.2556X_{cg}^2 - 0.0710X_{cg}^2Z_{cg} + 1.1243X_{cg}Z_{cg} \\ u_2 &= 0.0022 - 0.0103X_{cg} - 0.0073Z_{cg} \\ &\quad +0.0017X_{cg}^2 - 0.0011X_{cg}^2Z_{cg} + 0.0063X_{cg}Z_{cg} \\ u_3 &= -0.0828 + 0.3060X_{cg} + 1.5189Z_{cg} \\ &\quad -0.0106X_{cg}^2 + 0.0183X_{cg}^2Z_{cg} - 0.2422X_{cg}Z_{cg} \end{aligned}$$

and the application of the ETD yields LFRs of order 6 for $v(\delta_{s_1}, \delta_{s_2}^{-1})$ and order 5 for

$u(\delta_{s_3}, \delta_{s_4}^{-1})$, resulting in an LFR of order 11 for $a_{29}(\delta)$. Remarkably, the order 11, obtained exclusively by symbolic pre-processing is smaller than "least" orders about 15 achieved by combining various symbolic and numerical order reduction tools in [83], starting from initial realizations of orders up to 405. The order 11 LFR is also very close to the lower bound for the LFR order for $a_{29}$, which is 9 ($\{r_1, r_2, r_3, r_4\} = \{2, 4, 2, 1\}$). We conjecture that 11 is already the minimal order.

## 5.2 LFR realization for the full RCAM

The parametric state space matrices $A(\delta), B(\delta), C(\delta), D(\delta)$ of the RCAM are given in appendix D and have only elements as Laurent polynomials in the indeterminates. Several LFRs for the concatenated matrix

$$S(\delta) = \begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix}$$

are computed and the results are presented in Table 5.9. Without symbolic preprocessing an LFR with order 400 ($\{108, 201, 69, 22\}$) is obtained using the object-oriented LFR realization approach based on the generalized LFR.

Table 5.9: LFR orders for RCAM

| Preprocessing | $r$ | $\{r_1, r_2, r_3, r_4\}$ | $r$(red) | $\{r_1, r_2, r_3, r_4\}$(red) |
|---|---|---|---|---|
| Optimize | 370 | $\{98, 181, 69, 22\}$ | 170 | $\{41, 94, 25, 10\}$ |
| Parfrac | 312 | $\{87, 164, 39, 22\}$ | 110 | $\{36, 44, 17, 13\}$ |
| Confrac | 260 | $\{64, 105, 69, 22\}$ | 103 | $\{25, 48, 23, 7\}$ |
| Horner | 253 | $\{67, 112, 52, 22\}$ | 106 | $\{29, 45, 27, 5\}$ |
| TD | 137 | $\{35, 61, 28, 13\}$ | 107 | $\{35, 50, 17, 5\}$ |
| ETD | 109 | $\{27, 45, 26, 11\}$ | 91 | $\{24, 38, 21, 8\}$ |
| VS+ETD | 66 | $\{16, 30, 15, 5\}$ | 65 | $\{16, 30, 14, 5\}$ |

The partial-fraction, continued-fraction and Horner forms are based on single elements of $S(\delta)$ and do not consider any dependencies between the matrix elements. However the following numerical order reduction also yields quite good order reduction results. The "Optimize" method, although suited for matrices, yields the worst results.

Again, the best result is obtained by employing the combined VS and ETD approach in conjunction with the "try-hard" option. The resulting LFR of $S(\delta)$ is of order 66 and it is possible to exactly reduce this LFR to order 65, which is very close to the theoretical least order bound of 56, calculated with the procedure described in section 3.6. In this specific case, the VS factorization has been applied to the rows of $S(\delta)$ using the variable splitting $\delta_{s_1} = \delta_{s_2} = \{X_{cg}, Z_{cg}\}$ and $\delta_{s_3} = \delta_{s_4} = \{m, V\}$.

For the RCAM, a "try-hard" search using the combined VS plus ETD approach requires 120 distinct applications. With an average of approximately 180s for each decomposition the whole approach takes about 6 hours on a Intel Xeon 2.8 GHz running MATLAB 7.1 on a Linux computer. As the 120 decompositions can be calculated independently a

parallelization of the procedure is straightforward. At the DLR Institute of Robotics and Mechatronics a Linux-Cluster with 30 PCs is available, which allowed to perform the whole calculation within less than 15 minutes.

## 5.3 Accuracy of low order LFRs

For the very complex parametric model given by the RCAM, the combined VS and ETD approach allows to obtain an LFR, where a further exact numerical order reduction is almost unnecessary as the order of 66 can only be reduced to 65. For the $a_{29}$ element, no additional numerical reduction is possible. These are remarkable results as the symbolic preprocessing can be applied without any loss of accuracy (floating-point numbers are represented exactly in rational form), whereas numerical order reduction is always based on tolerance dependent rank decisions. Therefore it is always beneficial for the accuracy of the resulting LFR, if no numerical order reduction must be applied. To illustrate this, the accuracies of the low order LFRs obtained with the combined VS and ETD approach are compared with the accuracies of numerically reduced LFRs obtained with the Robust Control Toolbox 3.0.1 of MATLAB. This toolbox offers two kinds of numerical reduction called "basic" and "full". The "basic" method is based on the 1-d order reduction method of [77] and the MATLAB function `sminreal` is repeatedly called to detect structurally 1-d unobservable/uncontrollable parts. The "full" method is also based on the 1-d order reduction principle, but performs 1-d balancing and truncation, where all system parts with Hankel singular values less than 1e-16 are truncated.

Table 5.10 presents the accuracies of the different LFRs. The accuracy is derived as follows: substitute the symbolic vector $\delta$ in the symbolic matrix $S(\delta)$ with random numerical values $\delta_{\text{rand}}$ and subtract the numerical matrix $S(\delta)|_{\delta=\delta_{\text{rand}}}$ from the related numerical upper-LFT $S'(\delta)|_{\delta=\delta_{\text{rand}}} = \mathcal{F}_u(M,\Delta)|_{\delta=\delta_{\text{rand}}}$ for 200 random parameterizations $\delta_{\text{rand}}$ and the maximum of the 2-norms of the 200 samples is taken as accuracy, i.e.

$$e = \max_i \left( \|S(\delta)|_{\delta=\delta_{\text{rand},i}} - S'(\delta)|_{\delta=\delta_{\text{rand},i}}\|_2 \right),\ i = 1,\ldots,200.$$

The notation "n-d" ("Appr") in Table 5.10 means exact n-d (approximate 1-d) order reduction and using the functions `minlfr` (`redlfr1`) of the LFR Toolbox.

The numerical reduction method "Basic" has almost no effect on the order of the LFRs. The method "Full", which is applied after every step during the object-oriented LFR realization yields a large loss of accuracy with an error $e$ of more than 3 percent. On the other hand, the LFRs obtained after symbolic preprocessing are almost of full accuracy, $e$ =7e-18 for $a_{29}$ and $e$ =6e-14 for $S(\delta)$, which is very important to have reliable LFRs for the application of robust controller synthesis and stability/performance analysis.

It is interesting to see that if one may be confident with the low accuracy obtained by the "Full" method, almost the same accuracy can be obtained by applying numerical approximation to the LFRs obtained after symbolic preprocessing. However, in this case it is possible to further reduce the order to 7 and 48 for $a_{29}$ and $S(\delta)$, respectively.

Table 5.10: Accuracy of LFRs for RCAM

| Symbolic model | Symbolic preprocessing | Numerical reduction | $r$ | $\{r_1, r_2, r_3, r_4\}$ | $e$ |
|---|---|---|---|---|---|
| $a_{29}$ | — | Basic | 61 | {12,30,18,1} | 1e-17 |
| $a_{29}$ | — | Full | 12 | {4,3,4,1} | 4e-4 |
| $a_{29}$ | VS+ETD | — | 11 | {2,4,2,3} | 7e-18 |
| $a_{29}$ | VS+ETD | Appr | 7 | {2,2,2,1} | 1e-4 |
| $S(\delta)$ | — | Basic | 395 | {108,201,69,17} | 6e-14 |
| $S(\delta)$ | — | Full | 94 | {24,42,19,9} | 3.3e-2 |
| $S(\delta)$ | VS+ETD | — | 66 | {16,30,15,5} | 6e-14 |
| $S(\delta)$ | VS+ETD | n-d | 65 | {16,30,14,5} | 6e-14 |
| $S(\delta)$ | VS+ETD | Appr | 48 | {13,22,11,2} | 2.5e-3 |

## 5.4 Improved robust stability analysis using $\mu$

In this section a $\mu$-analysis approach for robust stability analysis is performed. In [58] the RCAM served as a benchmark for the design of a robust controller for a civil aircraft in the approach for landing. Thirteen controllers based on ten different controller design methods were developed by ten companies/universities, see Tables 5.11 and 5.12. The highly nonlinear controller MP-20 was excluded from the $\mu$-analysis, as unpredictable effects with numerical linearization were expected.

Table 5.11: Participating companies/universities

| Acronym | Company/Institute |
|---|---|
| CERT | Centre d'Études et de Recherches de Toulouse |
| CUN | Cranfield University |
| DLR | Deutsches Zentrum für Luft- und Raumfahrt |
| DUT | Delft University of Technology |
| LAAS | Laboratoire d'Analyse et d'Architecture des Systèmes |
| LUT | Loughborough University |
| NLR | National Aerospace Laboratory |
| UCAM | University of Cambridge |
| ULES | University of Leicester |
| UNED | Universidad Nacional de Educación a Distancia |

As already mentioned, due to a lack of suitable symbolic preprocessing techniques and efficient numerical order reduction methods, an LFR for the RCAM of order 193 was derived in [58]. As the order of this LFR was too large for efficient and reliable stability analysis based on $\mu$, it was decided to neglect the variations in airspeed $V$ leading to an LFR of order 35, which was of reasonable size to perform $\mu$-analysis. However, by the comparison with the optimization based worst-case search in [58] it was already obvious, that the results obtained from $\mu$-analysis using the LFR of order 35 (without uncertainties

Table 5.12: RCAM controllers

| Controller | Methodology | Design Team |
|---|---|---|
| MS-11 | $\mu$-Synthesis | DUT |
| MM-12 | Modal multi-model synthesis | CERT |
| CC-13 | Classical control | CUN |
| LY-14 | Lyapunov technique | LAAS |
| FL-15 | Fuzzy logic control | DUT |
| MO-16 | Multi-model multi-objective optimization | DLR |
| EA-18 | Eigenstructure assignment | LUT |
| MS-19 | $\mu$-Synthesis | NLR |
| MP-20 | Model-based predictive control | UCAM |
| HI-21 | $H_\infty$-synthesis | ULES |
| EA-22 | Eigenstructure assignment | UNED |
| HI-24 | $H_\infty$-synthesis | DLR |
| MF-25 | Model following control | DLR |

in $V$) were overly optimistic, that is, for some controllers the value of $\mu$ was clearly less than one but the worst-case search could find uncertain parameter values $\delta \in \Pi$ leading to instability.

Using the low order LFR realization techniques described in this thesis a very accurate LFR of low order 65 could be derived, which considers also uncertainties in $V$. This LFR is of reasonable order to perform $\mu$-analysis.

The overall structure for robust stability analysis is presented in figure 5.2, where $r$ is the reference signal $y$ is the output and $K$ is the controller, which is chosen from the set of linearized controllers in Table 5.12.

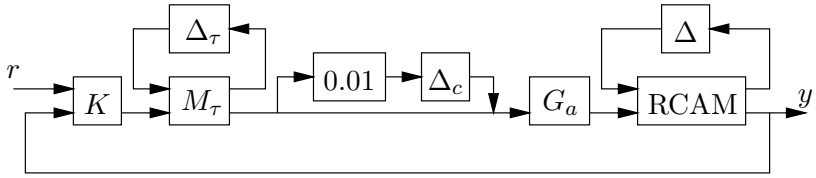


Figure 5.2: Detailed structure for robust stability analysis for RCAM

For each input ($u(1)-u(5)$) of the RCAM an actuator model is defined and the dynamic transfer matrix of the block $G_a$ is given by

$$G_a = \begin{bmatrix} \frac{6.667}{s+6.667} & 0 & 0 & 0 & 0 \\ 0 & \frac{6.667}{s+6.667} & 0 & 0 & 0 \\ 0 & 0 & \frac{3.333}{s+3.333} & 0 & 0 \\ 0 & 0 & 0 & \frac{0.6667}{s+0.6667} & 0 \\ 0 & 0 & 0 & 0 & \frac{0.6667}{s+0.6667} \end{bmatrix}.$$

Furthermore an uncertain time delay $\tau \in [0.05,\ 0.1]$s is assumed at each actuator input. The delay is approximated with a first order Padé-filter as

$$e^{-\tau s} \approx \frac{2 - \tau s}{2 + \tau s}$$

and is reasonably accurate up to a frequency of $\pm 10$rad/s. Since $\tau$ is identical for each actuator input, the uncertainty matrix $\Delta_\tau$ of the LFR $(M_\tau, \Delta_\tau)$ is given by $\Delta_\tau = \tau I_5$. In addition a small extra complex perturbation is added. As it is assumed that $\|\Delta_c\|_\infty = \|\mathrm{diag}(\delta_{c_1}, \delta_{c_2}, \delta_{c_3}, \delta_{c_4}, \delta_{c_5})\|_\infty \leq 1$, the perturbation $0.01\Delta_c$ is only 1% and may for example account for unmodelled dynamics or gain and phase variations at the input of the actuators. However, the main motivation for the introduction of these small complex uncertainties is that the computation of the lower $\mu$-bounds becomes tractable [11]. The final interconnection structure for analysis is shown in figure 5.3, where

$$\Delta_T = \mathrm{diag}(mI_{16}, VI_{30}, X_{cg}I_{14}, Z_{cg}I_5, \tau I_5, \delta_{c_1}, \delta_{c_2}, \delta_{c_3}, \delta_{c_4}, \delta_{c_5}),$$

with $m, V, X_{cg}, Z_{cg}, \tau \in \mathbb{R}$ and $\delta_{c_1}, \delta_{c_2}, \delta_{c_3}, \delta_{c_4}, \delta_{c_5} \in \mathbb{C}$.

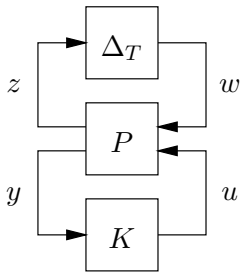


Figure 5.3: Compact structure for robust stability analysis for RCAM

The results of the $\mu$-analysis for controlled RCAM, using the very accurate, low order LFR, are summarized in Table 5.13. In figures 5.4 and 5.5 the upper and lower $\mu$-bounds for the interesting frequency range from $10^{-1}$ rad/s to $10^{1.5}$ rad/s are presented. In this interval, 100 equidistant grid points are chosen over a logarithmic frequency scale. The Robust Control Toolbox of MATLAB [11] is used to perform the $\mu$-analysis.

For comparison the $\mu$-analysis and the optimization-based worst-case search results from [58] are also included. The worst-case search repeatedly calculates the eigenvalues of the linearized closed-loop system within an optimization based parameter search directed to determine the minimum damping $\zeta_{\text{worst}}$ over all admissible parameter values. Each calculation of the least damping involves the trimming of the nonlinear open-loop system and the linearization of the closed-loop system.

In every case the $\mu$-analysis from [58] led to lower values for the maximum of the $\mu$-upper-bound. These more optimistic results come from the fact, that the variation in airspeed $V$ was not considered. Even more important is that the $\mu$-analysis results from [58] indicate robust closed-loop stability for the controllers MS-11, EA-18, HI-24,

Table 5.13: Robust stability analysis results for RCAM

| Controller | Dynamic order of $K$ | $\mu$-upper bound | $\mu$-upper bound from [58] | $\zeta_{\text{worst}}$ | $\zeta_{\text{skew}\mu}$ |
|---|---|---|---|---|---|
| MM-12 | 9 | 0.50 | 0.36 | 0.26 | 0.25 |
| MO-16 | 12 | 0.59 | 0.35 | 0.21 | 0.19 |
| LY-14 | 39 | 0.74 | 0.57 | 0.08 | 0.055 |
| EA-22 | 9 | 0.82 | 0.39 | 0.15 | 0.12 |
| CC-13 | 11 | 0.85 | 0.51 | 0.04 | 0.035 |
| FL-15 | 5 | 0.98 | 0.44 | – | – |
| MF-25 | 36 | 0.99 | 0.65 | 0.05 | – |
| HI-24 | 26 | 0.99 | 0.94 | -0.03 | – |
| MS-11 | 62 | 1.15 | 0.49 | -0.06 | – |
| EA-18 | 5 | 1.26 | 0.83 | -0.04 | – |
| MS-19 | 35 | 1.94 | 1.36 | -0.13 | – |
| HI-21 | 36 | 2.28 | 1.53 | -0.18 | – |

which was proven to be wrong from the worst-case search. The new $\mu$-analysis results, involving the accurate, low order LFR including variations in $V$, mainly coincide with the worst-case search results. The closed-loop system with controller MM-12 yields the lowest $\mu$-upper-bound value and also the best damping value. The controllers MS-11 and EA-18 are clearly indicated to not fulfill the robust closed-loop stability test. For the closed-loop system with controller HI-24 a $\mu$-upper-bound of 0.99 is obtained, which is at the robust stability boundary and practically indicates an unstable system as already shown from the worst-case search.

The last column in Table 5.13 represents results obtained with the Skew-$\mu$ toolbox for MATLAB [36]. Contrarily to the $\mu$-Analysis and Synthesis toolbox, where the $\mu$ upper bound, if less than one, guarantees that the eigenvalues of the controlled RCAM are robustly located in the left half plane of the complex plane, the Skew-$\mu$ toolbox allows to study the robust location of the eigenvalues within a truncated sector, which is defined by a maximum allowable real part and a minimum damping. The last column in Table 5.13 represents the minimum damping value $\zeta_{\text{skew}\mu}$, which was increased until a skew-$\mu$ upper bound value of one was obtained. For the controllers MM-12, MO-16, LY-14, EA-22 and CC-13, the values for $\zeta_{\text{skew}\mu}$ are slightly below the values $\zeta_{\text{worst}}$, which exactly meets the expectations as the skew-$\mu$ results should serve as a guaranteed bound for the worst damping.

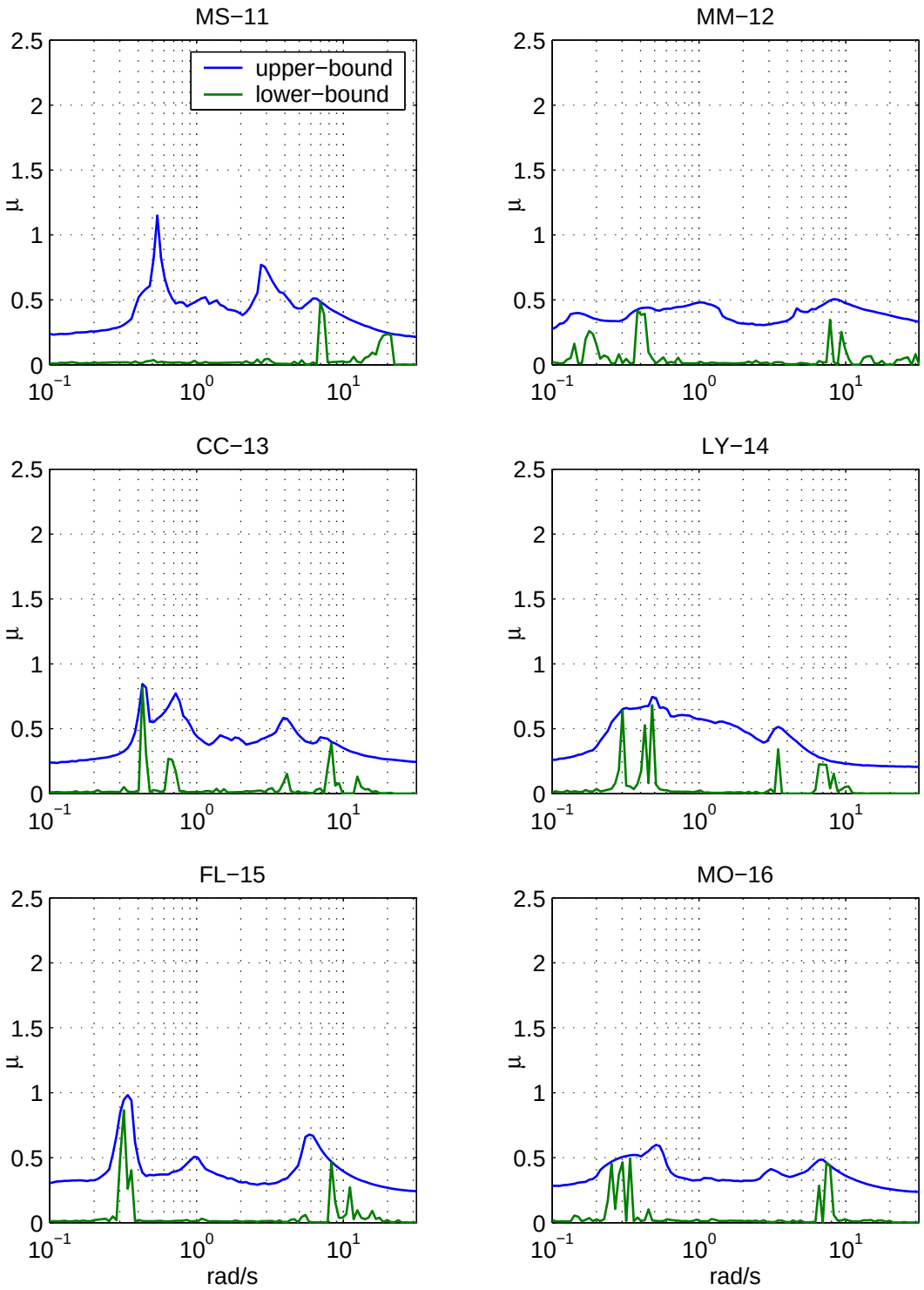


Figure 5.4: $\mu$-analysis results for RCAM (1)

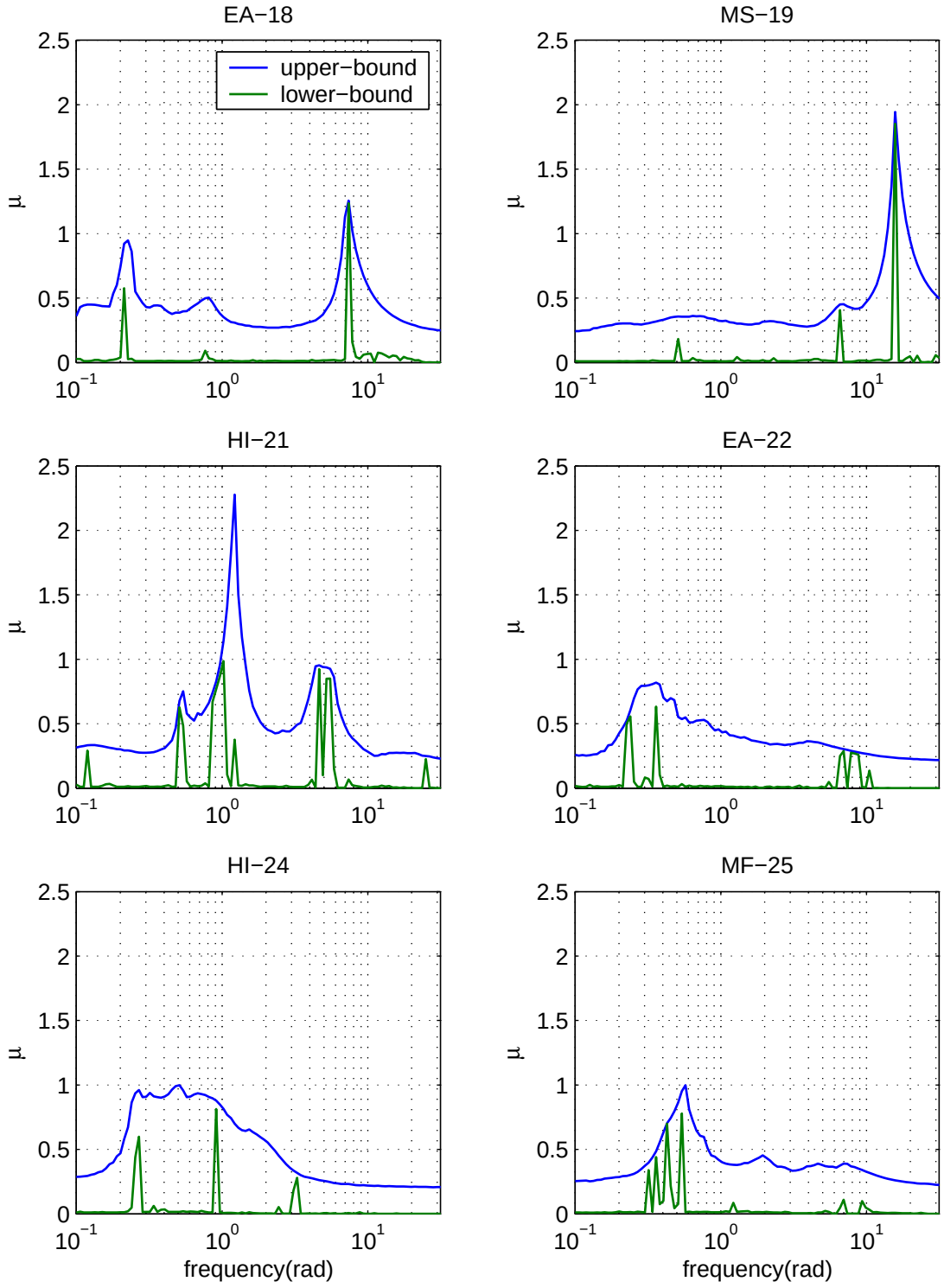


Figure 5.5: $\mu$-analysis results for RCAM (2)

# 6 Robust vehicle steering control design

In this chapter two approaches for the design of a robust steering controller to improve the yaw dynamics of a passenger car are presented [40]. The controller helps to avoid dangerous yaw motions of an automobile, which may result from unsymmetrical perturbations like side wind, unilateral loss of tire pressure or unilaterally icy road ($\mu$-split braking). In such situations the driver may react to slow, allowing large yaw motions, which are then overcompensated by large steering angles and braking leading to instability and accidents. A controller can react very fast to yaw disturbances and may almost completely compensate the perturbations. Hence, the driver is not scared by large yaw dynamics, which avoids panic reactions and high workload. This is a topic of active research for many years.

One approach is to use individual wheel braking [79, 80], another method is to command additional steering angles [2, 3, 20] for compensation of the disturbing yaw moment. The last approach is followed in this chapter and a steer-by-wire actuator is assumed, where the commands to the steering actuator from the steering controller are added to the steering commands of the driver commanded by the steering wheel. A main objective is that the controller must be robust with respect to large variations in longitudinal speed, road adhesion, mass, inertia and unstructured uncertainty accounting for unmodelled dynamics. Furthermore, the control action should not be uncomfortable to the driver and passengers.

Two different controllers are designed. One approach is performed using $\mu$-synthesis, yielding a controller which is robust to variations in longitudinal speed and road adhesion. From a robust stability point of view, this approach is comparable to the results presented in [3, 20], where in principle robust stability is only guaranteed for uncertain but *constant* parameters. In reality, the control action to reduce yaw motion due to unsymmetrical perturbations is usually required during manoeuvres as $\mu$-split braking, where the speed $v$ of the vehicle is by no means constant. Therefore a second gain-scheduling control design is performed, that guarantees robust stability in case of bounded variation rates of $v$ using Linear Parameter Varying (LPV) control design techniques. In all cases it is assumed that the speed $v$ and the yaw rate $r$ can be measured and are available for control. For some basic results about the used LPV approach see appendix B.

Major improvements compared to the results presented in [3, 20] are that parametric uncertainties in the mass and inertia of the vehicle model are explicitly considered as structured uncertainties, which allows to improve the closed-loop performance. The method presented in [3, 20] usually only allows to consider 2 uncertain parameters as structured uncertainty and any additional uncertainty is considered as unstructured. Furthermore the controller synthesis techniques employed in this thesis allow to achieve

robust performance for a considerably larger uncertainty region than presented in [3, 20].

## 6.1 Single-Track model and steering actuator

The vehicle model, which is used for control design is the classical linearized single-track model [1]. A more complex vehicle model consisting of the linearized single-track model with roll-augmentation is also presented in [1], and control laws for rollover avoidance are designed. In this thesis the single-track model with roll augmentation is only used to demonstrate the low-order LFR realization techniques and the simpler single-track model is used for both LFR realization and control design.

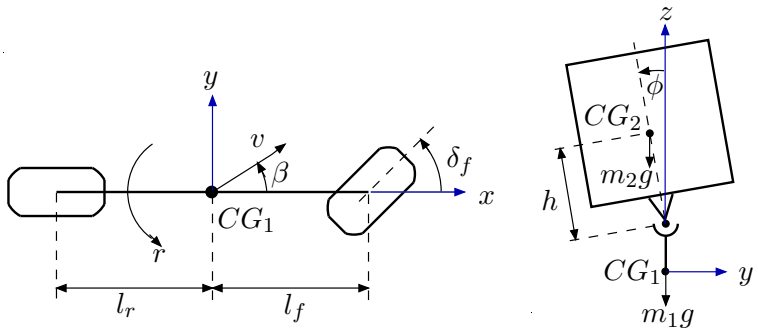


Figure 6.1: Single-Track model with roll augmentation

The linearized equations of motion are given in descriptor form (2.50) with

$$
\begin{aligned}
E(\delta) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & mv & 0 & -hm_2 \\ 0 & 0 & J_z & 0 \\ 0 & -hm_2v & 0 & J_{x,2}+h^2m_2 \end{bmatrix} \\
A(\delta) &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & -(c_f+c_r)\mu & -(c_fl_f-c_rl_r)\frac{\mu}{v}-mv & 0 \\ 0 & -(c_fl_f-c_rl_r)\mu & -(c_fl_f^2+c_rl_r^2)\frac{\mu}{v} & 0 \\ -c_\phi+m_2gh & 0 & hm_2v & -d_{\dot\phi} \end{bmatrix} \\
B(\delta) &= \begin{bmatrix} 0 & c_f\mu & c_fl_f\mu & 0 \end{bmatrix}^T \\
C &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix},\ D = 0,
\end{aligned}
\tag{6.1}
$$

where $m = m_1 + m_2$. The system state vector and the input vector are given by $x = \begin{bmatrix} \phi & \beta & r & \dot\phi \end{bmatrix}^T$ and $u = \delta_f$, respectively.

The steering actuator model [1] is shown in figure 6.2. The parameters of this position controlled electric motor model are $L_a = 0$ (electric time constant neglected), $R_a = 5\,\Omega$, $J_a = 0.004053\,\mathrm{kg\,m^2}$, $k_f = 0.01625$, $K_a = 22.22$, $k_{me} = k_{em} = 0.9\,\mathrm{Nm/A}$.

Note, in the following sections the name $\mu$ is used for the road adhesion parameter but also to denote the structured singular value as described in appendix A. However,

Table 6.1: Vehicle Data

| Parameter (or Variable) | Unit | Description |
|---|---|---|
| $c_f$ | [kN/rad] | front cornering stiffness |
| $c_r$ | [kN/rad] | rear cornering stiffness |
| $c_\phi$ | [kN m/rad] | roll stiffness of passive suspension |
| $d_{\dot\phi}$ | [kN m s/rad] | roll damping of passive suspension |
| $g$ | [m/s$^2$] | acceleration due to gravity |
| $h$ | [m] | nominal height of $CG_2$ over roll axis |
| $J_{x,2}$ | [kg m$^2$] | roll moment of inertia, sprung mass |
| $J_z$ | [kg m$^2$] | overall yaw moment of inertia |
| $l_f$ | [m] | distance of front axle to $CG_1$ |
| $l_r$ | [m] | distance of rear axle to $CG_1$ |
| $m_1$ | [kg] | mass of chassis |
| $m_2$ | [kg] | sprung mass |
| $\mu$ | | road adhesion |
| $v$ | [m/s] | magnitude of velocity vector |
| $r$ | [rad/s] | yaw rate |
| $\beta$ | [rad] | chassis side-slip angle at $CG_1$ |
| $\delta_f$ | [rad] | front wheel steering angle |
| $\phi$ | [rad] | roll angle of sprung mass |

for each occurrence of $\mu$ the meaning should be clear from the context. Furthermore, SI-units are assumed for all model parameters and for simplicity, the units are omitted.

## 6.2 LFR realization for roll-augmented single track model

For the single-track model with roll augmentation the parameters $m_1$, $m_2$, $h$, $v$, $\mu$, $J_{x,2}$ and $J_z$ are considered as uncertain.

The vehicle model (6.1) is given in descriptor form as,

$$\begin{bmatrix} 1 & 0 \\ 0 & M(\delta) \end{bmatrix} \dot{\overline{x}} = \begin{bmatrix} 0 & 1 \\ -K(\delta) & -D(\delta) \end{bmatrix} \overline{x} + \begin{bmatrix} 0 \\ S(\delta) \end{bmatrix} u$$
$$y = C\overline{x}$$

which is the direct representation in state space for a second order physical system model described as

$$M(\delta)\ddot{x} + D(\delta)\dot{x} + K(\delta)x = S(\delta)u.$$

A standard state-space description may be derived as

$$\begin{aligned} \dot{\overline{x}} &= \begin{bmatrix} 0 & 1 \\ -M^{-1}(\delta)K(\delta) & -M^{-1}(\delta)D(\delta) \end{bmatrix} \overline{x} + \begin{bmatrix} 0 \\ M^{-1}(\delta)S(\delta) \end{bmatrix} u \\ y &= C\overline{x}, \end{aligned} \tag{6.2}$$

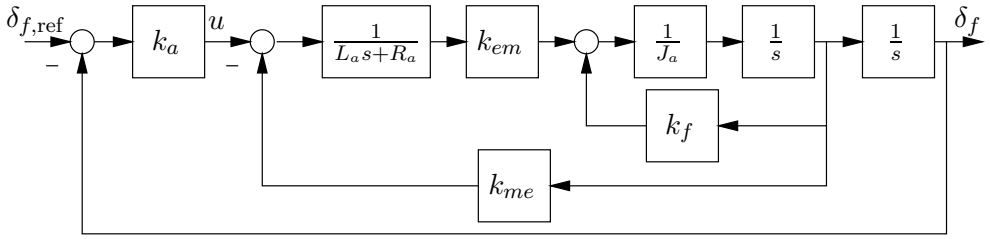


Figure 6.2: Steering actuator model

where the symbolic inversion of $M(\delta)$ may yield very complex symbolic expressions in the entries of the system matrices. It will be shown in the following that this arbitrarily introduced complexity may yield LFRs of higher order compared to LFRs, that are directly realized for the descriptor system representation of the vehicle model. Therefore it can be seen as a good practice to avoid symbolic inversions before the LFR realization and to employ the direct LFR realization methods for descriptor systems as described in section 2.4.

In table 6.2 the resulting orders $r$ of the LFRs, with

$$\Delta = \mathrm{diag}(J_{x,2}I_{r_1}, J_z I_{r_2}, hI_{r_3}, m_1 I_{r_4}, m_2 I_{r_5}, \mu I_{r_6}, vI_{r_7}),$$

which are realized from the standard state-space representation (6.2) of the vehicle model are presented. All the models are realized without preliminary symbolic normalization of (6.2) using the generalized LFR during the object-oriented LFR realization. Without symbolic preprocessing an LFR of order 147 is derived, which can be reduced to order 61 using exact numerical order reduction. For this example, symbolic preprocessing techniques as transformation to Horner form, partial fraction decomposition (Parfrac), continued fraction decomposition (Confrac) and the Maple routine Optimize allow to reduce the initial order from 147 to about 120. However, the resulting numerically reduced LFRs are at least of order 99, which is larger than the order 61 obtained without symbolic techniques. It seems that in these cases the symbolic preprocessing introduces constructs that are disadvantageous in terms of the commutativity problem as described in section 3.1. The combined VS+ETD approach allows to directly derive an LFR of order 24, which cannot be further reduced without loss of accuracy. The ETD approach, which directly yields an LFR of order 27, which can be further exactly reduced to order 23.

In table 6.3 the order of the LFRs realized from the descriptor system representation (6.1) of the vehicle model are presented. In this case the Horner, Optimize, Parfrac and Confrac techniques do not allow to further reduce the initial order of 29 and in all cases an LFR of order 14 is obtained after numerical order reduction. Only the ETD and the combined VS+ETD techniques allow to directly derive an LFR of minimal order 12 without any further numerical order reduction. To show, that 12 is the minimal achievable LFR order for the vehicle model one may simply calculate the lower bound for the LFR order as described in section 3.6, which is 12.

In the case of a preliminary symbolic normalization and expansion of the rational entries of the system matrices in standard state-space form (6.2) a direct object-oriented LFR

Table 6.2: LFR orders for standard state-space vehicle model

| Preprocessing | $r$ | $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7\}$ | $r$(red) | $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7\}$ |
|---|---|---|---|---|
| – | 147 | $\{25, 3, 36, 25, 36, 10, 12\}$ | 61 | $\{7, 1, 17, 7, 16, 5, 8\}$ |
| Parfrac | 127 | $\{18, 3, 29, 23, 31, 13, 10\}$ | 115 | $\{12, 2, 29, 20, 30, 12, 10\}$ |
| Horner | 127 | $\{18, 3, 29, 23, 31, 13, 10\}$ | 115 | $\{12, 2, 29, 20, 30, 12, 10\}$ |
| Confrac | 124 | $\{17, 3, 29, 26, 31, 10, 8\}$ | 108 | $\{12, 2, 20, 20, 30, 7, 8\}$ |
| Optimize | 112 | $\{12, 3, 28, 23, 30, 9, 7\}$ | 99 | $\{12, 1, 28, 20, 27, 6, 5\}$ |
| TD | 41 | $\{7, 1, 11, 6, 4, 4, 8\}$ | 35 | $\{5, 1, 10, 4, 4, 3, 8\}$ |
| ETD | 27 | $\{4, 1, 4, 4, 3, 4, 7\}$ | 23 | $\{3, 1, 4, 3, 3, 2, 7\}$ |
| VS+ETD | 24 | $\{3, 1, 4, 4, 3, 2, 7\}$ | 24 | $\{3, 1, 4, 4, 3, 2, 7\}$ |

Table 6.3: LFR orders for descriptor system vehicle model.

| Preprocessing | $r$ | $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7\}$ | $r$(red) | $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7\}$ |
|---|---|---|---|---|
| – | 29 | $\{1, 1, 6, 2, 7, 6, 6\}$ | 14 | $\{1, 1, 3, 1, 2, 2, 4\}$ |
| Parfrac | 29 | $\{1, 1, 6, 2, 7, 6, 6\}$ | 14 | $\{1, 1, 3, 1, 2, 2, 4\}$ |
| Horner | 29 | $\{1, 1, 6, 2, 7, 6, 6\}$ | 14 | $\{1, 1, 3, 1, 2, 2, 4\}$ |
| Confrac | 29 | $\{1, 1, 6, 2, 7, 6, 6\}$ | 14 | $\{1, 1, 3, 1, 2, 2, 4\}$ |
| Optimize | 29 | $\{1, 1, 6, 2, 7, 6, 6\}$ | 14 | $\{1, 1, 3, 1, 2, 2, 4\}$ |
| TD | 17 | $\{1, 1, 3, 2, 4, 2, 4\}$ | 15 | $\{1, 1, 2, 2, 3, 2, 4\}$ |
| ETD | 12 | $\{1, 1, 2, 1, 2, 2, 3\}$ | 12 | $\{1, 1, 2, 1, 2, 2, 3\}$ |
| VS+ETD | 12 | $\{1, 1, 2, 1, 2, 2, 3\}$ | 12 | $\{1, 1, 2, 1, 2, 2, 3\}$ |

realization would yield an LFR of order 4145, which clearly shows the complexity, that may be introduced by symbolic inversion and normalization. In contrast, the symbolic normalization and expansion of the vehicle model in descriptor form (6.1) yields an LFR of order 58.

## 6.3 LFR realization for single track model without roll augmentation

The single track model without roll augmentation, which is used for control design, is obtained from (6.1) by neglecting the rows and columns related to the states $\phi$ and $\dot{\phi}$, yielding a second order system with the states $\beta$ and $r$ and input $u = \delta_f$. The descriptor system matrices for the single track model without roll augmentation are given as

$$\begin{aligned} E(\delta) &= \begin{bmatrix} mv & 0 \\ 0 & J_z \end{bmatrix}, \; A(\delta) = \begin{bmatrix} -(c_f + c_r)\mu & -(c_f l_f - c_r l_r)\frac{\mu}{v} - mv \\ -(c_f l_f - c_r l_r)\mu & -(c_f l_f^2 + c_r l_r^2)\frac{\mu}{v} \end{bmatrix} \\ B(\delta) &= \begin{bmatrix} c_f\mu & c_f l_f \mu \end{bmatrix}^T, \; C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \; D = 0, \end{aligned} \tag{6.3}$$

and

$$\overline{A}(\delta) = E^{-1}(\delta)A(\delta), \; \overline{B}(\delta) = E^{-1}(\delta)B(\delta), \; \overline{C} = C, \; \overline{D} = D \tag{6.4}$$

represent the system matrices of a standard state-space representation. With nominal parameter values $c_f = 8424.3$, $c_r = 9570.7$, $l_f = 1.25$, $l_r = 1.32$, $m_1 = m = 1296$ and $J_z = 1750$, the model may describe a mid-size passenger car. The parameters $m$, $J_z$, $\mu$ and $v$ are considered as uncertain.

In tables 6.4 and 6.5 the resulting orders $r$ of the LFRs, with

$$\Delta = \mathrm{diag}(J_z I_{r_1}, m I_{r_2}, \mu I_{r_3}, v I_{r_4}),$$

which are realized from the standard state-space representation (6.4) and the descriptor system representation (6.3) of the single-track model without roll augmentation are presented.

Table 6.4: LFR orders for standard state-space vehicle model

| Preprocessing | $r$ | $\{r_1, r_2, r_3, r_4\}$ | $r$(red) | $\{r_1, r_2, r_3, r_4\}$ |
|---|---|---|---|---|
| – | 19 | $\{3, 4, 6, 6\}$ | 14 | $\{1, 3, 4, 6\}$ |
| Parfrac | 16 | $\{3, 3, 6, 4\}$ | 9 | $\{1, 1, 4, 3\}$ |
| Horner | 16 | $\{3, 3, 6, 4\}$ | 9 | $\{1, 1, 4, 3\}$ |
| Confrac | 16 | $\{3, 3, 6, 4\}$ | 9 | $\{1, 1, 4, 3\}$ |
| Optimize | 16 | $\{3, 3, 6, 4\}$ | 9 | $\{1, 1, 4, 3\}$ |
| TD | 13 | $\{1, 2, 3, 7\}$ | 8 | $\{1, 1, 3, 3\}$ |
| ETD | 6 | $\{1, 1, 2, 2\}$ | 6 | $\{1, 1, 2, 2\}$ |
| VS+ETD | 6 | $\{1, 1, 2, 2\}$ | 6 | $\{1, 1, 2, 2\}$ |

Table 6.5: LFR orders for descriptor system vehicle model

| Preprocessing | $r$ | $\{r_1, r_2, r_3, r_4\}$ | $r$(red) | $\{r_1, r_2, r_3, r_4\}$ |
|---|---|---|---|---|
| – | 13 | $\{1, 2, 6, 4\}$ | 6 | $\{1, 1, 2, 2\}$ |
| Parfrac | 13 | $\{1, 2, 6, 4\}$ | 6 | $\{1, 1, 2, 2\}$ |
| Horner | 13 | $\{1, 2, 6, 4\}$ | 6 | $\{1, 1, 2, 2\}$ |
| Confrac | 13 | $\{1, 2, 6, 4\}$ | 6 | $\{1, 1, 2, 2\}$ |
| Optimize | 13 | $\{1, 2, 6, 4\}$ | 6 | $\{1, 1, 2, 2\}$ |
| TD | 11 | $\{1, 2, 3, 5\}$ | 7 | $\{1, 1, 2, 3\}$ |
| ETD | 6 | $\{1, 1, 2, 2\}$ | 6 | $\{1, 1, 2, 2\}$ |
| VS+ETD | 6 | $\{1, 1, 2, 2\}$ | 6 | $\{1, 1, 2, 2\}$ |

The ETD and VS+ETD approaches allow to directly obtain an LFR with minimal order 6, which is also obtained for the single-track model in standard state-space form. Again, this clearly shows the superiority of these preprocessing techniques. The LFR of minimal order 6 is used for the following robust vehicle steering control design.

In the case of a preliminary symbolic normalization and expansion of the rational entries of the system matrices in standard state-space form (6.4) a direct object-oriented LFR realization would yield an LFR of order 50. In contrast, the symbolic normalization and

expansion of the vehicle model in descriptor form (6.3) yields an LFR of order 17. The TD approach combined with numerical order reduction yields an LFR of order 7, which is the largest order after numerical order reduction. This results from the complexity, that is introduced by the calculation of a polynomially factorized representation for the rational single-track model, which is obligatory for the application of the TD.

## 6.4 Problem specification

The problem specifications are taken from [20], where uncertain parameters and their corresponding ranges were defined. Furthermore, frequency dependent specifications for the sensitivity function $S$ and complementary sensitivity function $T$ and specifications for the locations of the closed-loop eigenvalues are given therein.

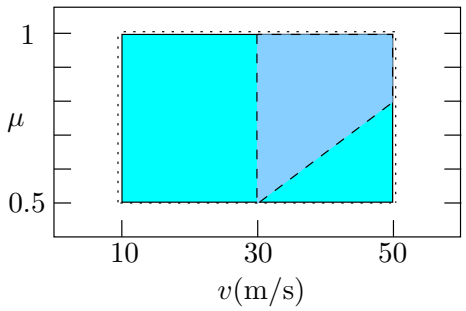


Figure 6.3: Parametric variation range

In figure 6.3 the operating domain of the vehicle in terms of longitudinal speed $v \in [10; 50]$ and road adhesion $\mu \in [0.5; 1]$ is shown. In [20] a controller was designed, which was robust with respect to $\mu$ and scheduled with $v$. In particular, robust performance was achieved within the dashed blue polygon marked in figure 6.3. In this thesis, robust performance is achieved for the whole rectangular uncertainty domain defined by $v \in [10; 50]$ and $\mu \in [0.5; 1]$ (see dotted region in figure 6.3).

Furthermore, in [20] the mass $m$ and the moment of inertia $J_z$ are assumed to be uncertain, with $m \in [1296, 1696]$ and $J_z \in [1750, 2100]$, respectively. As $m$ and $J_z$ do not vary independently, it is assumed that they are linearly related, with

$$J_z = 616 + 0.875m. \tag{6.5}$$

Hence, in the model equations, $J_z$ is substituted according to (6.5) and a minimal order LFR with $\Delta = \mathrm{diag}(mI_2, \mu I_2, vI_2)$ is finally obtained and used for control design.

Note, that the control design method used in [20] usually only allows to consider two parameters ($\mu$ and $v$) as structured uncertainties and therefore the variations in $m$ and $J_z$ were considered as an additional unstructured uncertainty. In this thesis the variations in $m$ and $J_z$ are considered as structured uncertainties, thus reducing conservativeness and improving the performance of the closed-loop system.

### 6.4.1 Controller Structure

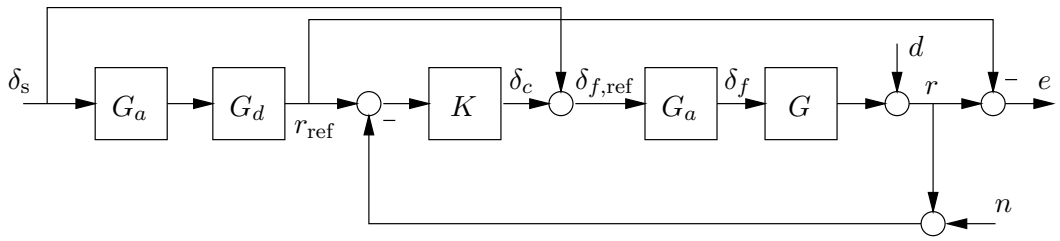


Figure 6.4: Overall controller structure

The overall controller structure is shown in figure 6.4. The driver input at the steering wheel is denoted as $\delta_s$, $G = G(s, m, J_z, \mu, v)$ is the parametric single track model, $G_a$ is the actuator model shown in figure 6.2 and $K$ is the controller. $G_d = G(s, 1496, 1925, 1, v)$ is the single-track model at actual longitudinal speed, with constant $\mu = 1$ (dry road), $m = 1496$, $J_z = 1925$, which is used to generate the desired reference input $r_{\text{ref}} = G_d G_a \delta_s$, that should be tracked by the controlled vehicle, with $e$ as the tracking error. The input $d$ describes disturbance, that should be compensated at a defined low frequency range (see specification for the sensitivity function $S$ in section 6.4.2) and $n$ denotes sensor noise, that should be compensated at high frequencies (see specification for the complementary sensitivity function $T$ in section 6.4.2). The steering wheel input $\delta_s$ is directly forwarded/connected to the steering actuator $G_a$. In case of $G = G_d$ and no disturbance ($n = d = 0$) there should be no additional steering angle command $\delta_c$ from the controller $K$.

### 6.4.2 Mixed sensitivity specifications and synthesis structure

The frequency dependent specifications describing the desired robust performance requirements for the closed-loop system are defined in the following. The controller synthesis is formulated as an $H_\infty$ mixed-sensitivity problem [38] for the tracking interconnection as shown in figure 6.5, where only the feedback part of the interconnection in figure 6.4 is considered. The disturbances $d$ and $n$ are not explicitly included in figure 6.5, however the desired rejection of these disturbances is accounted for by the formulation of the mixed-sensitivity specifications.

The sensitivity function $S$, the complementary sensitivity function $T$ and the control sensitivity function $R$ are defined as

$$S = \frac{e_1}{r_{\text{ref}}} = \frac{1}{1 + GG_aK} = \frac{e}{d} \tag{6.6}$$

$$T = \frac{e_3}{r_{\text{ref}}} = \frac{GG_aK}{1 + GG_aK} = \frac{-e}{n} \tag{6.7}$$

$$R = \frac{e_2}{r_{\text{ref}}} = \frac{K}{1 + GG_aK} = \frac{-\delta_c}{d} = \frac{-\delta_c}{n} = \frac{\delta_c}{r_{\text{ref}}}, \tag{6.8}$$

with $e_1$, $e_2$, $e_3$, $r_{\text{ref}}$ and $d$, $n$, $e$, $\delta_c$, $r_{\text{ref}}$ as shown in figures 6.5 and 6.4, respectively.

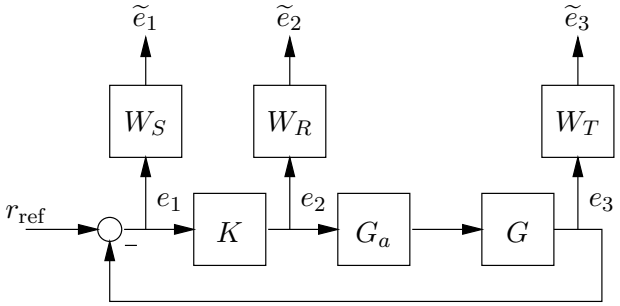


Figure 6.5: Mixed-sensitivity synthesis structure

From equations (6.6)-(6.8) the design objectives for the feedback loop can be deduced. For example, equation (6.6) shows, that the effect of the disturbance $d$ on the error $e$ can be made "small" by making the sensitivity function $S$ small. On the other hand, equation (6.7) shows that a small $T$ reduces the effect of sensor noise $n$ on the error $e$. However, as $S + T = 1$ there is a conflict between disturbance rejection and sensor-noise reduction as in the frequency range where a good disturbance rejection is achieved (i.e., $\|S\|_\infty \ll 1$) the sensor-noise is "passed-through" to the error $e$ (i.e., $\|T\|_\infty \approx 1$). Therefore, a basic assumption is that $n$ is significant in the high-frequency range, whereas disturbance $d$ should be rejected at low frequencies, which allows a trade-off in formulating the specifications for $S$ and $T$.

A further requirement on $T$ is related to robustness to unstructured model uncertainty. Assuming that the perturbed plant model has the form $(1 + \Delta)G$ with $\Delta$ stable and the closed-loop system is nominally stable. Then the perturbed closed-loop system is stable if

$$\det(I + (I + \Delta)GG_aK) = \det(I + GG_aK)\det(I + \Delta T)$$

has no right-half plane zeros. This generally amounts to requiring that $\|\Delta T\|_\infty$ is small or that $\|T\|_\infty$ is small at frequencies where $\Delta$ is significant, which usually is the case at high frequencies.

The function $R$ reflects the sensitivity of the controller to disturbances and noise over the frequency range. A usual requirement is to have high-gains at low frequencies, where disturbances are significant and uncertainties are small, and to avoid actuator saturation by achieving sufficient controller roll-off at frequencies above the actuator/plant bandwidth.

Summarizing the above statements, to achieve good performance it is necessary to have $\|GG_aK\|_\infty \gg 1 \;\rightarrow\; \|S\|_\infty \ll 1$ and $\|K\|_\infty \gg 1$ in the low frequency range. Good robustness and sensor-noise rejection requires to obtain $\|GG_aK\|_\infty \ll 1 \;\rightarrow\; \|T\|_\infty \ll 1$ in the high-frequency range.

The purpose of the mixed-sensitivity approach is to simultaneously fulfill performance specifications for $S$, $T$ and $R$. Therefore it is necessary to make the different specifications comparable and to gather them into a matrix function that is suitable for $H_\infty$ synthesis.

A very convenient way to reflect the different specifications is to employ appropriate frequency dependent weighting functions $W_S$, $W_T$, $W_R$, which are chosen such that the specifications on $S$, $T$ and $R$ are fulfilled if the following inequalities hold:

$$\|W_S S\|_\infty \leq 1, \ \forall\omega \tag{6.9}$$
$$\|W_T T\|_\infty \leq 1, \ \forall\omega \tag{6.10}$$
$$\|W_R R\|_\infty \leq 1, \ \forall\omega. \tag{6.11}$$

For example, the specification for the sensitivity function $S$ can be translated into inequality (6.9) by choosing $W_S$ such that $W_S \gg 1$ at low frequencies and $W_S$ is sufficiently small at high frequencies.

Gathering the weighted sensitivity functions into one vector $E$, with

$$E = \begin{bmatrix} \frac{\tilde{e}_1}{r_{\text{ref}}} \\ \frac{\tilde{e}_2}{r_{\text{ref}}} \\ \frac{\tilde{e}_3}{r_{\text{ref}}} \end{bmatrix} = \begin{bmatrix} W_S S \\ W_R R \\ W_T T \end{bmatrix}$$

allows to formulate the mixed-sensitivity optimization problem as

$$\min_K \|E\|_\infty,$$

where $\|E\|_\infty \leq 1$ means that all specifications for $S$, $T$ and $R$ are fulfilled.

The specifications for $S$ and $T$ are taken from [20], where $W_S$ was chosen as $W_S = (s+12.6)/(1.8s+1.26)$, which requires $\|S\|_\infty \leq 0.1$ at low frequencies. In [20] a controller structure was chosen, which inherently offers infinite controller gain at $s = 0$ and therefore the specification for $S$ at low frequency will always be fulfilled. Choosing the same $W_S$ for the controller structure as shown in figure 6.4, may yield a controller that allows an error $e$, where only 90% of a constant disturbance $d$ are rejected. This may not be considered satisfactory. To improve performance, the weighting $W_S$ was chosen more demanding as

$$W_S = \frac{s+11}{1.8s+0.11},$$

thus requiring $\|e\|_2 \leq 0.01\|d\|_2$ at low frequencies. Due to Bode's sensitivity integral relation [91], the required small norm of $S$ at low frequencies unavoidably results in a norm larger than one at high frequencies, which shall be bounded with $\|S\|_\infty \leq 1.8$. In figure 6.6 the frequency dependent magnitude of the upper bound $W_S^{-1}$ for $S$ is shown.

Robustness with respect to multiplicative unstructured uncertainty is formulated in terms of a performance specifications for the complementary sensitivity function $T$. The controller must be robust to 10% magnitude uncertainty at low frequencies, where the model of the vehicle and actuator is reasonably accurate, and 500% uncertainty at high frequencies, where unmodelled dynamics come into play. A transition frequency of 6 Hz between low and high frequency uncertainty was chosen based on the knowledge of the model's accuracy. This specification is formulated as

$$W_T = 5\frac{s+3.77}{s+188.5}.$$

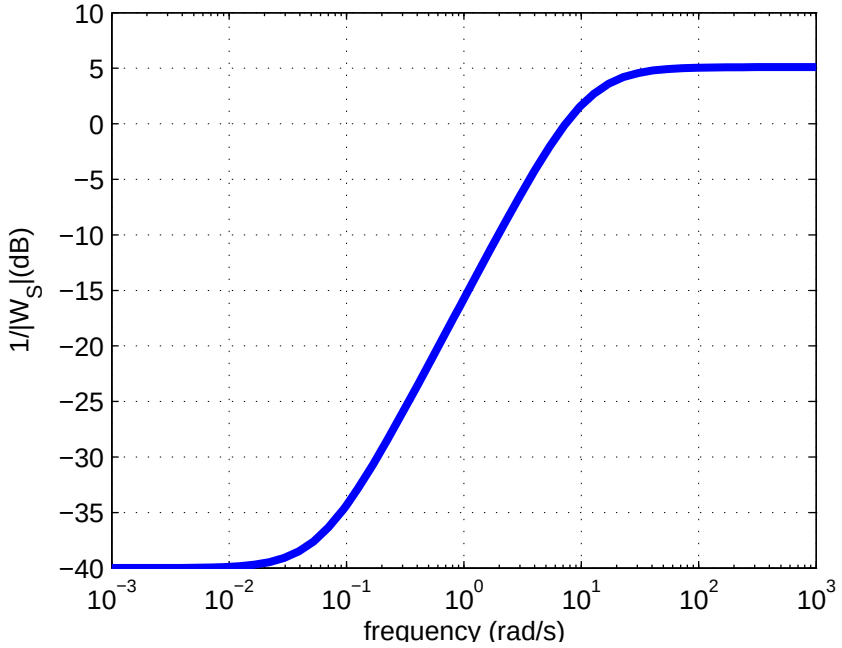


Figure 6.6: Magnitude of frequency dependent upper bound for $S$

Note, that in [20] a second specification on $T$ is formulated to account for an unstructured uncertainty describing variations in $m$ and $J_z$. Here, this is not included as these variations are explicitly considered as structured uncertainties.

Finally, sufficiently small controller gains at high frequencies could be obtained by using

$$W_R = \frac{0.25s + 0.9}{s + 90}.$$

### 6.4.3 Closed-loop eigenvalue specification

In [20] a specification on the location of the closed-loop eigenvalues was formulated, which is defined as a convex region in the complex plane. In particular, the region $\mathcal{D}$, where the closed-loop poles should be located (see blue region in figure 6.9), is defined as the intersection of three convex regions $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ defined as

$$\begin{aligned}
\mathcal{D}_1 &:= \{s \in \mathbb{C} : \mathrm{Re}(s) \leq -2\} \\
\mathcal{D}_2 &:= \left\{ s \in \mathbb{C} : \left| \arctan \frac{\mathrm{Im}(s)}{\mathrm{Re}(s)} \right| \leq \theta = 1.0472 = \arccos(D) \right\} \\
\mathcal{D}_3 &:= \{s \in \mathbb{C} : |s| \leq 20\pi\},
\end{aligned}$$

where $\mathcal{D}_1$ describes a shifted imaginary axis ensuring a limited settling time, $\mathcal{D}_2$ describes a conic sector ensuring a minimum damping $D = 0.5$ and $\mathcal{D}_3$ describes a circle centered at the origin ensuring that the natural frequency of any pole does not exceed 10 Hz.

For the mixed-sensitivity approach followed in this thesis it is in general possible to formulate this pole placement specifications in terms of Linear Matrix Inequality (LMI)

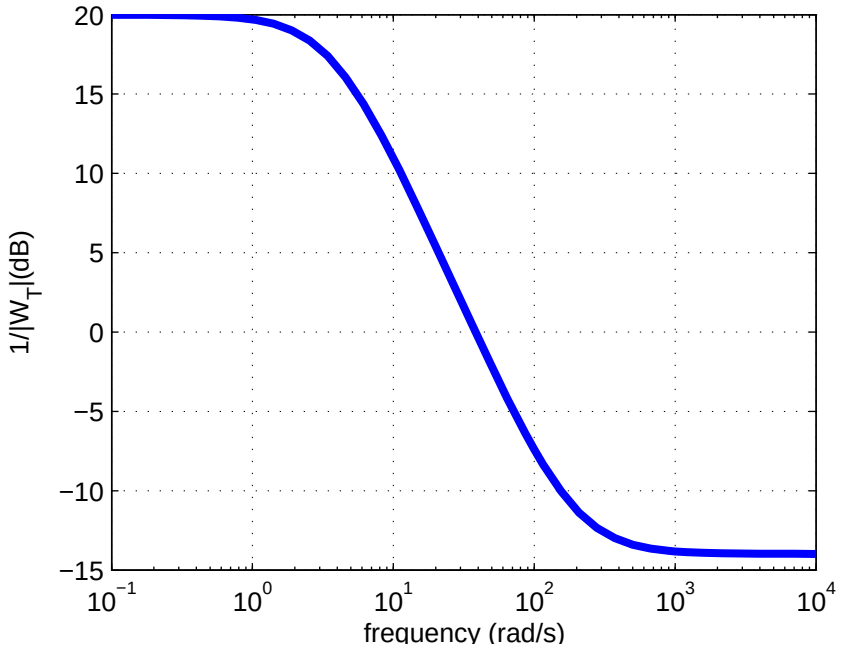


Figure 6.7: Magnitude of frequency dependent upper bounds for $T$

constraints [22] for the $H_\infty$-controller synthesis. However, the eigenvalues of the weighting functions $W_S$, $W_T$ and $W_R$ are also included in the set of closed-loop poles and as these functions are simply multiplied to the outputs $e_1$, $e_2$, $e_3$ of the synthesis structure in figure 6.5, the controller $K$ has no influence on the poles of the weighting functions. Therefore these poles are fixed and cannot be placed into a desired region which limits the choice of $\mathcal{D}$. As an example, the weighting function $W_T$ has a pole at $s = -188.5$ and therefore the pole placement specification defined by the region $\mathcal{D}_3$ cannot be met. The same holds for $W_S$ and region $\mathcal{D}_1$. Besides the weighting function poles, also the poles of the scaling functions $D$ (see (A.7)) are fixed and it may happen that even the pole placement in the region $\mathcal{D}_2$ cannot be demanded. However, such a case was not encountered during the control design. As a result, only the region $\mathcal{D}_2$ is considered as constraint for the controller design in section 6.5.

Although the desired pole-placement in the region $\mathcal{D}$ cannot be a priori included as a design specification, it will be shown that the final frequency weighted controller reduction in section 6.5.2 yields a controller such that the closed-loop poles are almost located in the region $\mathcal{D}$. For the LPV controller design in section 6.6, no a priori pole placement constraints are included, but also in this case the controller reduction step allows to almost fulfill the desired pole-placement specifications.

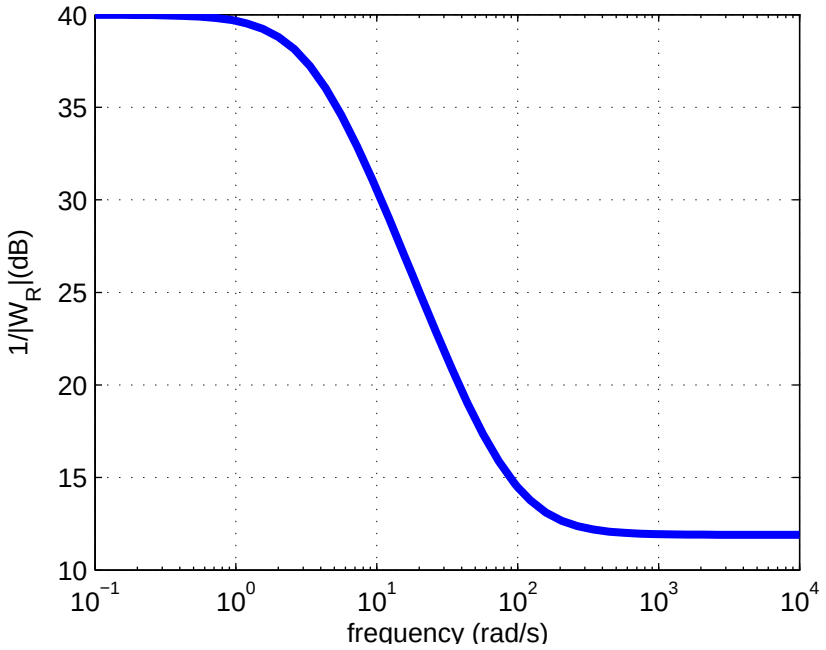


Figure 6.8: Magnitude of frequency dependent upper bound for $R$

## 6.5 $\mu$-Synthesis

### 6.5.1 Synthesis procedure

In this section the $\mu$-synthesis approach as described in section A.4 is applied to design a robust controller for the vehicle. The control design is performed using the $\mu$-Analysis and Synthesis Toolbox Version 3.0.7 [12] and the LMI Control Toolbox Version 1.0.8 [37] running under MATLAB 6.5. The LMI Control Toolbox is used to formulate the desired closed-loop pole placement specification and the function `hinfmix` is employed for LMI-based $H_\infty$ synthesis under pole placement constraints. For efficiency the LMI solver from the Sedumi package [75] is used instead of the MATLAB solver. The already mentioned sixth order LFR model, with $\Delta = \text{diag}(mI_2, \mu I_2, vI_2)$, obtained using the LFR-toolbox and the linear actuator model (see figure 6.2) are used to describe the uncertain plant for control design.

The overall synthesis structure as shown in figure A.5 is finally generated, where $P$ has the particular structure as shown in figure 6.10, with $G = \mathcal{F}_u(M, \Delta)$, $w = \Delta z$, $\Delta = \text{diag}(mI_2, \mu I_2, vI_2)$ and the fictitious performance uncertainty block is given as $\Delta_p \in \mathbb{C}^{1\times 3}$. As $\Delta_p$ is not a square matrix, the $\mu$-synthesis optimization problem (A.7) is slightly reformulated as

$$\min_{K(s)} \inf_{D_L(s), D_R(s)\in H_\infty} \|D_L(s)\mathcal{F}_l(P(s), K(s))D_R(s)\|_\infty, \tag{6.12}$$

where

$$D_L(s) = \text{diag}(D_1(s), D_2(s), D_3(s), I_3), \;\; D_R(s) = \text{diag}(D_1^{-1}(s), D_2^{-1}(s), D_3^{-1}(s), I_1),$$

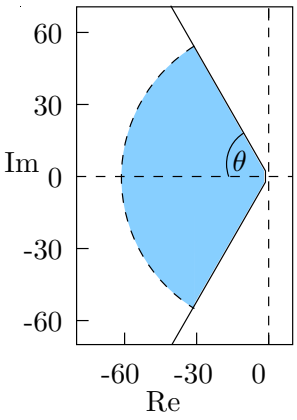


Figure 6.9: Closed-loop eigenvalue specification

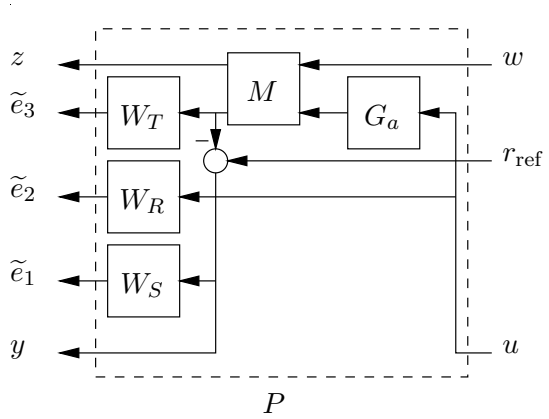


Figure 6.10: Internal structure of P for $\mu$-synthesis

with $D_1, D_2, D_3 \in \mathbb{C}^{2\times 2}$. The resulting value of (6.12) is defined as $\gamma$.

The $D$-$K$-iteration procedure (see section A.4) was initialized with $D_L = I_9$ and $D_R = I_7$. The controller synthesis in step two of the procedure was performed using the `hinfmix` function of the LMI-Toolbox including a constraint to place the closed-loop poles in the region $\mathcal{D}_2$. In step four of the procedure the fitting of the 12 dynamic entries of the scaling matrices $D_L$ and $D_R$ was restricted to third order systems, allowing a maximum order of 36 for $D_L$ and $D_R$. As the controller order is equal to the weighted plant order (order 7) plus twice the order of the scaling matrix, a maximum order of 79 may be obtained for $K$.

After eight iterations a controller of order $n_c = 33$ was obtained, which guarantees robust stability for the full uncertain parameter domain ($\mu$-analysis for robust stability yields $\mu = 0.79$) and a robust performance value of $\gamma = 1$ was achieved. This indicates that all the mixed-sensitivity performance specifications are robustly fulfilled.

### 6.5.2 Frequency-weighted controller reduction

When using standard $H_\infty$ controller design techniques [91], the order of the controller is the same as the generalized plant order. During the $\mu$-synthesis procedure for the vehicle, scaling matrices $D_L$ and $D_R$ of order 13 were generated and together with the weighted-plant order of 7 a controller of order $n_c = 33$ was derived. Obviously the controller is very complex to be finally applied to a fourth order model consisting of the single-track model plus actuator. Furthermore, $H_\infty$ control design tends to generate controllers $K$ with very fast dynamics, which causes implementation problems (one pole of the $\mu$-synthesis controller $K$ has a pole with real part at -180 rad/s).

To overcome these problems, it is common practice to apply model reduction techniques [6] to calculate a controller $K_r$ with order $n_r < n_c$, such that the error $e_m$ defined as

$$e_m = \|K - K_r\|_\infty \tag{6.13}$$

is small. A drawback of this formulation is that the error $e_m$ is considered equally over the whole frequency range, thus possibly preserving also very fast and probably unnecessary dynamics of the controller. More importantly, there is no direct relation between $e_m$ and the robust stability and performance properties of the closed-loop system and it may happen that a small error $e_m$ yields a quite remarkable degradation of the performance value $\gamma$. When calculating $K_r$, it is more natural to preserve stability and performance properties of the closed-loop system instead of finding an almost exact approximation of the original controller $K$. Therefore a more sophisticated approach is to calculate $K_r$ by applying frequency-weighted controller reduction [4, 82] with the goal to keep the closed-loop approximation error $e_c$ small, which is defined as

$$e_c = \|W_o(K - K_r)W_i\|_\infty, \tag{6.14}$$

with

$$W_o = \frac{GG_a}{I + GG_aK}, \quad W_i = \frac{1}{I + GG_aK}.$$

The minimization of $e_c$ in (6.14) allows to directly enforce closed-loop stability and performance.

**Remark 6.1.** The weighting functions $W_o$ and $W_i$ include the plant model $G$, which depends on the parameters $m$, $v$, $\mu$ and the choice of these parameters may influence the calculation of $K_r$. For the single-track model several parameterizations were checked, and the best closed-loop performance (calculated using $\mu$-analysis) was achieved for $W_o$ and $W_i$ with $G$ at $m = 1296$, $v = 10$ and $\mu = 1$.

For the solution of the minimization problems in (6.13) or (6.14), the *balanced truncation approximation* (BTA) [63] and the *singular perturbation approximation* (SPA) [56] are used as basis techniques to calculate $K_r$. For the vehicle controller, the best results were achieved using the SPA method, which probably results from the exact steady-state and good low-frequency approximation performed by this method.

For comparison two reduced order controllers $K_{r_1}$, $K_{r_2}$ are calculated. $K_{r_1}$ is calculated using frequency-weighted model reduction with approximation error $e_c$ as formulated in

(6.14). $K_{r_2}$ is calculated using standard model reduction with $e_m$ as formulated in (6.13). Furthermore the SPA method is used to calculate $K_{r_1}$ and BTA is used for $K_{r_2}$. The *mex*-functions `conred` and `sysred` [85] are used for frequency-weighted controller reduction and standard model reduction, respectively. These functions are based on a collection of order reduction tools available in SLICOT [15].

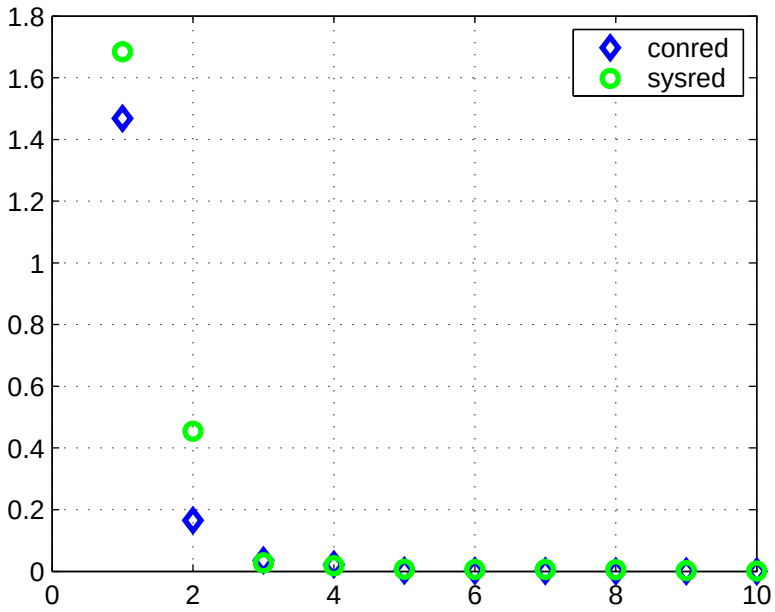


Figure 6.11: Largest 10 Hankel singular values

Table 6.6: Comparison of reduced order controllers

| Controller | Method | Order | Natural frequency (Hz) | $\gamma$ |
|---|---|---|---|---|
| $K_{r_1}$ | conred/SPA | 3 | 14.2 | 1.03 |
| $K_{r_2}$ | sysred/BTA | 3 | 15.8 | 1.09 |

The controller $K$ has one slightly unstable pole at $s = 0.05$ rad/s, which is always preserved in the reduced order controllers. For controller reduction, $K$ is split as $K = K_s + K_u$, where $K_s$ and $K_u$ denote the stable and unstable controller parts, respectively. In figure 6.11 the ten largest Hankel singular values of the stable controller part $K_s$ for frequency-weighted (`conred`) and standard (`sysred`) order reduction are shown. From this values it was decided to use a second order approximation of $K_s$ in both the `conred` and `sysred` case.

In Table 6.6 the properties of the reduced order controllers are presented in terms of the resulting controller order, the maximum natural frequency of the controller poles and the achieved closed-loop performance $\gamma$. Controller $K_{r_1}$ is a slightly better approximation of $K$ at low frequencies and allows the largest deviations in the high frequency range (see figure 6.12), which is anyway outside the bandwidth of the plant.

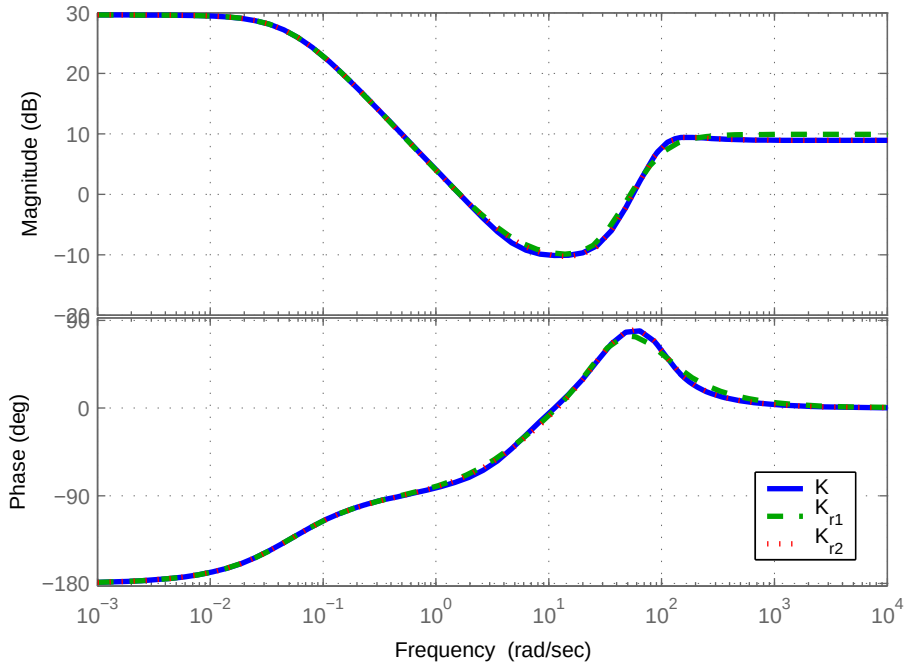


Figure 6.12: Bode plot of controllers

Finally controller $K_{r_1}$ is chosen as there is almost no performance degradation ($\gamma = 1.03$ compared to $\gamma = 1$ for $K$) and the maximum natural frequency of the controller poles is 1.6 Hz smaller than for $K_{r_2}$. The state-space matrices $A$, $B$, $C$, $D$ of the controller $K_{r_1}$ are

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{ccc|c} 0.05 & 0 & 0 & 26.01 \\ 0 & -122.5 & 47.3 & 22.98 \\ 0 & -118.3 & -19.52 & 20.67 \\ \hline 0.06 & -15.31 & 2.09 & 3.125 \end{array}\right].$$

**Remark 6.2.** It is important to note, that a $\mu$-controller was also synthesized for a non-minimal LFR, with uncertainty matrix $\Delta = \mathrm{diag}(mI_2, \mu I_3, vI_3)$ (instead of a minimal order LFR with $\Delta = \mathrm{diag}(mI_2, \mu I_2, vI_2)$). The resulting controller yields a degradation of the closed-loop performance index $\gamma$ of more than 15%. This may be caused by the considerably more involved generation of the scaling matrices $D_R(s) = \mathrm{diag}(D_1^{-1}(s), D_2^{-1}(s), D_3^{-1}(s), I_1)$, $D_L(s) = \mathrm{diag}(D_1(s), D_2(s), D_3(s), I_3)$ with

$$D_R(s) = \left[\begin{array}{ccccccccc} * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & * & * & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & * & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & * & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & * & * & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * & * & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * & * & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

where for each of the 22 frequency varying entries (marked by $*$), a low-order transfer function must be generated to approximately fit a set of frequency grid-points for each of these functions. Hence the transfer function scaling matrices $D_R(s)$ and $D_L(s)$ are only an approximation to the optimal scalings at each frequency point. Compared to only 12 frequency varying entries in $D_R(s)$ and $D_L(s)$ for a minimal order LFR, the accumulated approximation error for $D_R(s)$ and $D_L(s)$ may increase for the non-minimal LFR. Furthermore, the controller order, which is equal to order of the generalized plant plus the order of the scaling matrices $D_R(s)$ and $D_L(s)$ may dramatically increase. This clearly shows the importance of generating least order LFRs.

### 6.5.3 Frequency domain results

In this section, frequency domain results obtained with $K_{r_1}$ are presented. Beginning with the pole specifications, the closed-loop transfer function (without feed-forward controller)

$$T = \frac{r}{r_{\text{ref}}} = \frac{GG_aK_{r_1}}{1 + GG_aK_{r_1}}$$

is considered and figure 6.13 shows the closed-loop poles for the parametric single-track model at the vertices of the uncertain parameter domain. An exhaustive gridding approach has shown that the single-track model at the vertices of the parameter domain yields the worst-case closed-loop pole locations with respect to the specifications as defined in section 6.4.3. Therefore only the vertices are considered in figure 6.13. The damping specification (minimum damping of 0.5), which was explicitly included as a constraint for the controller design, is fulfilled for all closed-loop poles. The worst-case, with a minimum damping of 0.618, is given for $T(s, m, \mu, v)|_{m=1296,\mu=0.5,v=50}$. The specification on the natural frequency (maximum natural frequency of 10 Hz), which could not be a-priori included as a constraint for the controller design, is usually violated for the transfer functions $T$ at $\mu = 0.5$, with a maximum value for the natural frequency of 14.1 Hz. However, this violation is acceptable and still may allow a real-time implementation of the controller $K_{r_1}$. Finally, the third specification (maximum value of -2 rad/s for the real part of the closed loop poles), which also could not be a-priori included as a constraint for the controller design, is violated for the transfer functions $T(s, m, \mu, v)|_{m=1296,\mu=0.5,v=50}$ and $T(s, m, \mu, v)|_{m=1696,\mu=0.5,v=50}$, with maximum real parts at -0.99 rad/s and -1.3 rad/s, respectively. This specification was included to guarantee a minimum settling-time for the closed-loop system. However, the simulation results in section 6.7 show very good settling-time properties for these cases.

In figures 6.14, 6.15 and 6.16 the sensitivity, complementary sensitivity and controller sensitivity functions together with their upper-bound specifications $1/|W_S|$, $1/|W_T|$ and $1/|W_R|$ are shown. The Bode diagrams of these figures present the sensitivity functions for the single-track models at the vertices of the considered uncertain parameter domain, which also include the worst-cases. It can be seen, that all the sensitivity specifications are fulfilled for the considered uncertain parameter domain. Therefore, the robust performance $\mu$-analysis result of $\gamma = 1.03$ for the closed-loop system with $K_{r_1}$ results from the slight conservative upper-bound calculation of the $\mu$-analysis.

Considering robust stability only, the $\mu$-analysis results shown in figure 6.17 yield a maximum upper-bound of 0.79 for the structured singular value $\mu$, which indicates enough stability margin for the uncertain parameter domain under consideration.

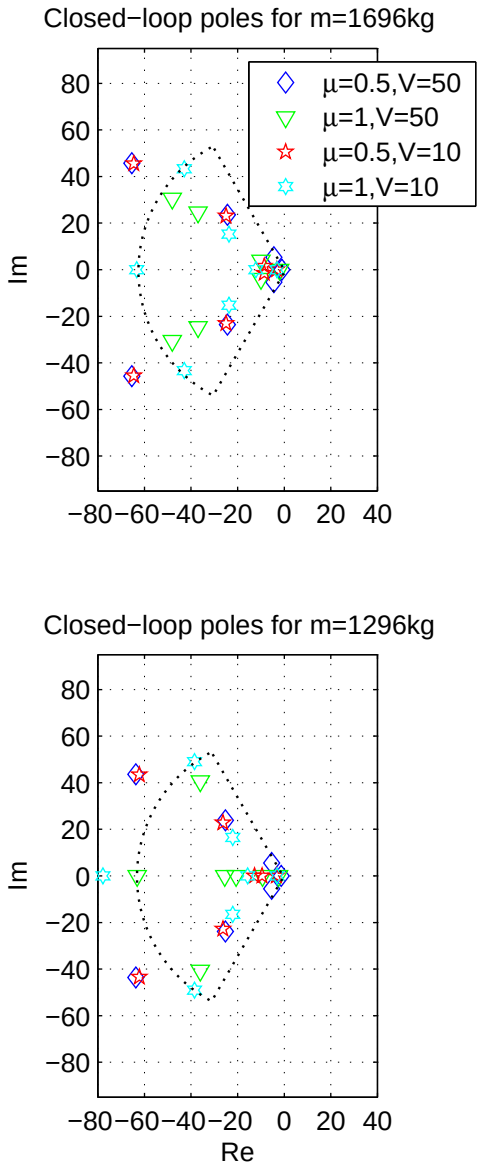


Figure 6.13: Closed-loop poles

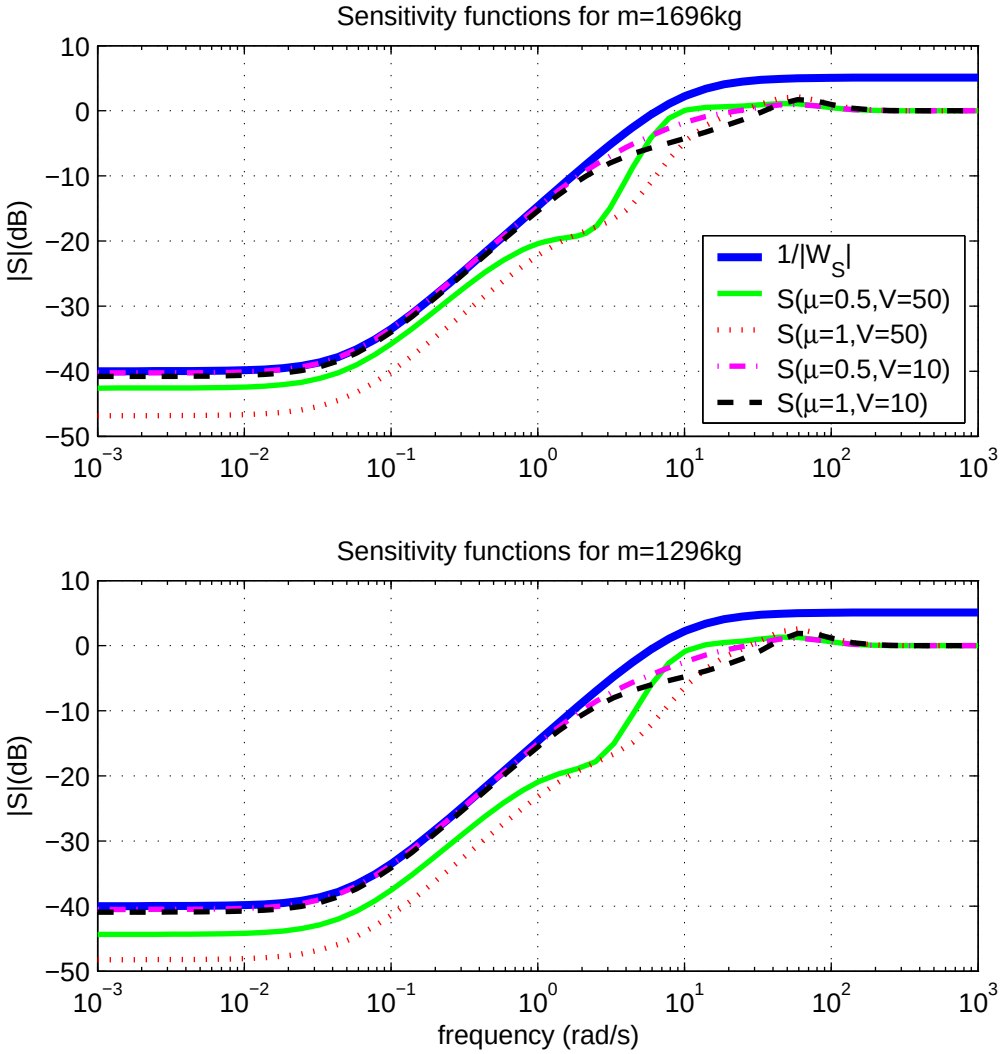


Figure 6.14: Bode diagram of sensitivity functions

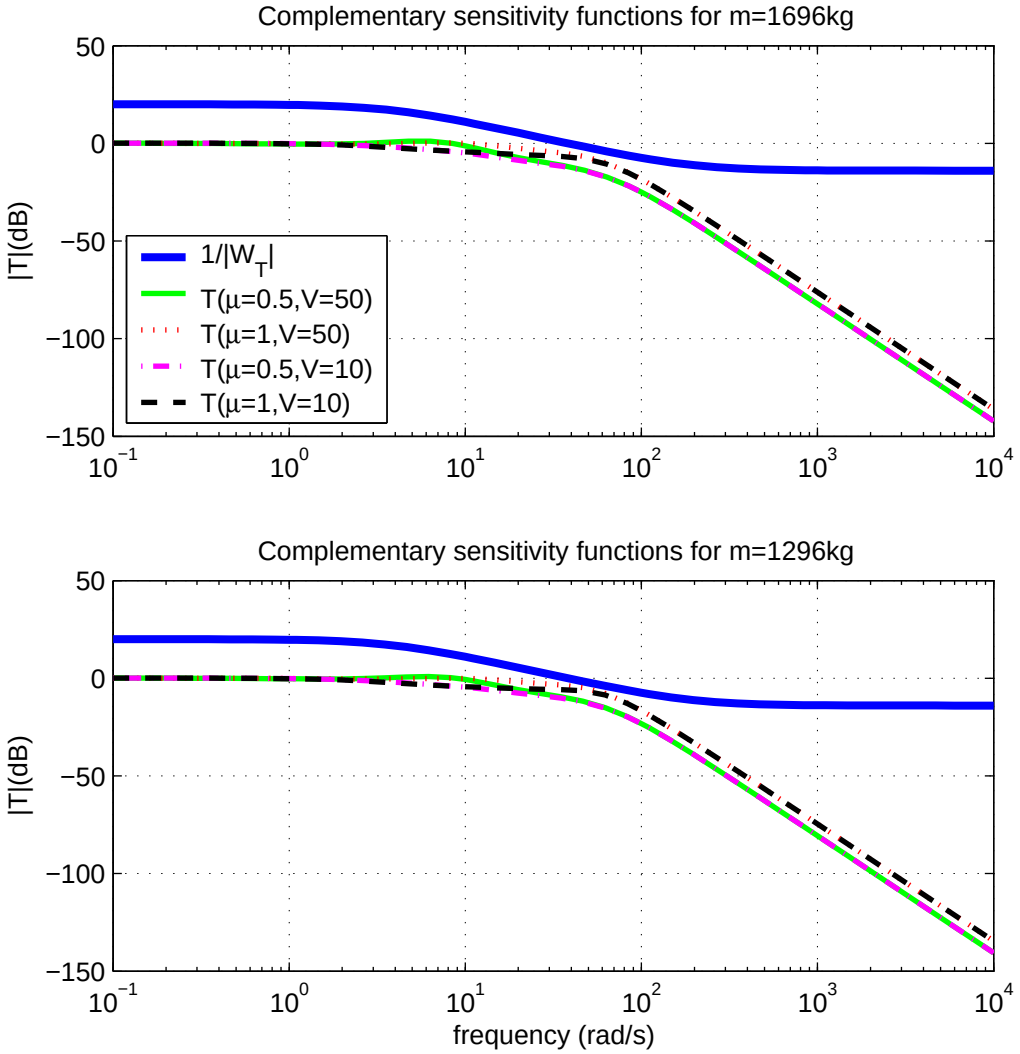


Figure 6.15: Bode diagram of complementary sensitivity functions

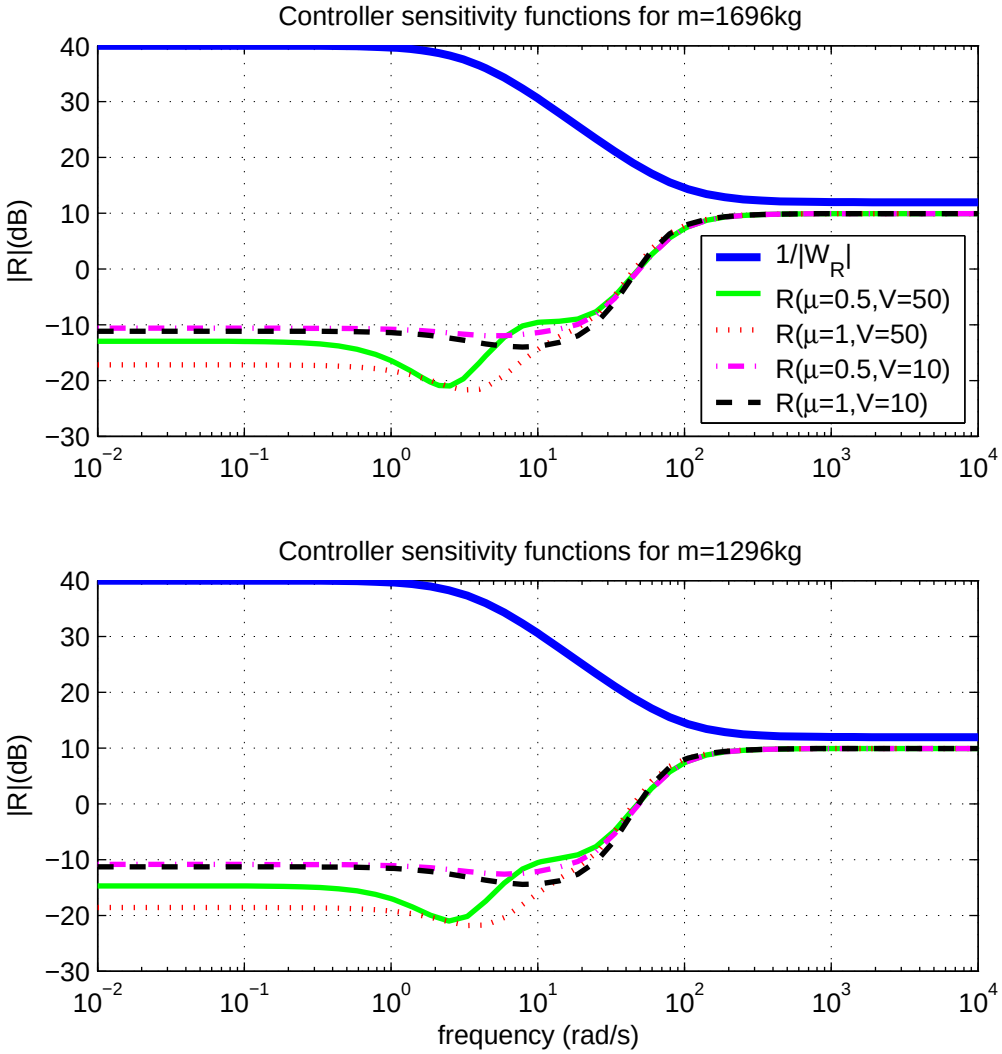


Figure 6.16: Bode diagram of controller sensitivity functions

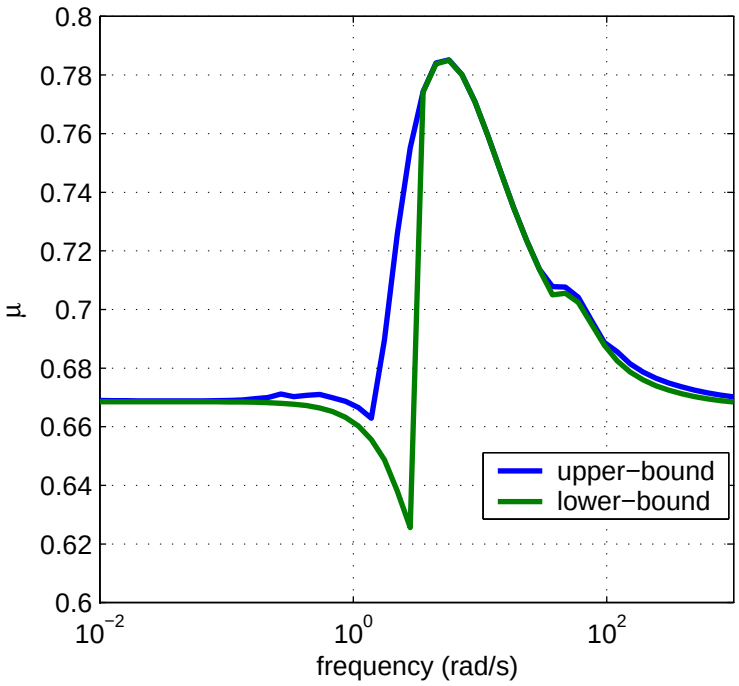


Figure 6.17: Robust stability $\mu$-analysis results

## 6.6 LPV-control design

In this section Linear Parameter Varying (LPV) control design methods as proposed in [88, 87] are applied to the uncertain single-track model. In contrast to the method in [20] and the $\mu$-synthesis approach of the previous section, where robust stability and performance is only guaranteed for model parameters that are uncertain but *constant*, the LPV approach allows bounded variation rates of parameters that are measurable and on-line available for control. For the single-track model, the uncertain parameters $v$, $\mu$, $m$ and $J_z$ are explicitly considered. However, as it may be hard to measure the road adhesion $\mu$, mass $m$ and moment of inertia $J_z$ on-line, only the longitudinal vehicle speed $v$ is assumed to be online measurable for control, whereas $\mu$, $m$ and $J_z$ are considered to be uncertain but constant. Providing robust stability and performance despite variations in $v$ is a very important property of the controller, as dangerous yaw disturbances are caused for example by $\mu$-split braking, where high variation rates of $v$ may occur.

In principle, one may derive an LPV controller with no restrictions on the variation rate of $v$. However, guaranteeing robust performance for arbitrary parameter trajectories $v(t)$ may be overly conservative (or even not possible) and may lead to poor performance of the closed-loop system [10]. To improve performance, the LPV control design technique proposed in [88, 87] is employed, where the variation rate of $v$ is assumed to be bounded. This corresponds to the real physics, where the longitudinal speed $v$ does not change arbitrarily fast. A drawback of this method is that the control design requires to solve an infinite-dimensional convex feasibility problem, which can only be solved approximately by searching for feasibility over a finite dimensional subspace. Therefore a finite gridding of the single-track model with respect to $v$ is performed and only this finite set of models is considered for control design, which if successful, yields a controller with guaranteed properties.

The resulting controller also consists of a finite set of grid-point controllers and the state-space matrices of the grid-point controllers are linearly interpolated based on the nearest grid-point plant model. This is in contrast to traditional gain-scheduling design processes, where several equilibrium point controller designs are performed, followed by the design of a scheduling procedure to interpolate between the point designs. For some background on the employed LPV control design procedure see appendix B.

### 6.6.1 Synthesis structure

For the LPV controller synthesis, only the longitudinal speed $v$ is assumed to be measurable and on-line available for control, whereas $\mu$, $m$ and $J_z$ are supposed to be uncertain. Hence, the resulting controller is scheduled with $v$ and must be robust to uncertainties in $\mu$, $m$ and $J_z$.

The LPV control design approach requires a gridding of $G(s, v, m, J_z, \mu)$ over the measurable, varying parameter space $v \in [10, 50]$. Some trials have shown, that a rather coarse, equally-spaced 3-point gridding, with $v = v_i \in \{10, 30, 50\}$ is sufficient for the LPV controller synthesis. Therefore three models $\overline{G}_i(s, m, J_z, \mu)$, $i = 1, \ldots, 3$, with $\overline{G}_i(s, m, J_z, \mu) = G(s, v, m, J_z, \mu)|_{v=v_i}$ are calculated, where $v$ is substituted with the corresponding numerical grid-point value $v_i$. For each $\overline{G}_i(s, m, J_z, \mu)$, $J_z$ is substi-

tuted according to (6.5), yielding $G_i(s, m, \mu)$ and LFRs $(M_i, \Delta)$ of minimal order 4, with $\Delta = \text{diag}(mI_2, \mu I_2)$ are generated such that $G_i(s, m, \mu) = \mathcal{F}_u(M_i, \Delta)$. Finally, weighted mixed-sensitivity synthesis models $P_i$, $i = 1, \ldots, 3$, with internal structure as shown in figure 6.18 are calculated. As in the $\mu$-synthesis approach, a scaling matrix

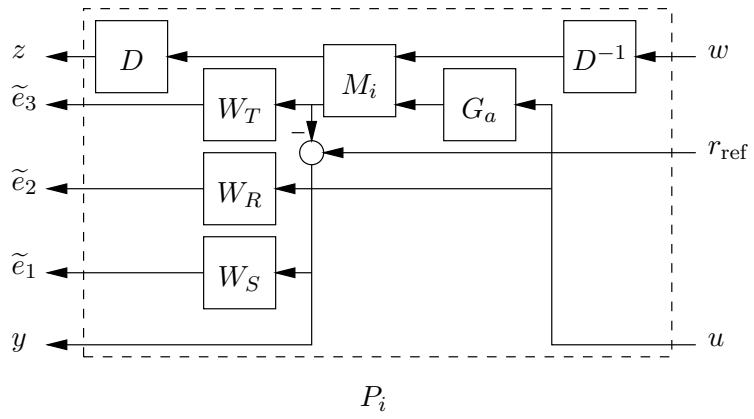


Figure 6.18: Structure of synthesis models for LPV-control design

$D = \text{diag}(D_1, D_2)$ with $D_1, D_2 \in \mathbb{C}^{2\times 2}$ is employed to account for the structured uncertainty matrix $\Delta = \text{diag}(mI_2, \mu I_2)$. In contrast to the iterative update of $D$ during the $\mu$-synthesis approach ($D$-$K$-iteration with synthesis model $P$ as shown in figure 6.10), the scaling matrix $D$ is now a fixed integral part of the $P_i$ (see figure 6.18). The iterative update of $D$ is not considered as the related $\mu$-analysis procedure may not be straightforwardly applied for the closed-loop system controlled by the scheduled LPV controller, which consists of linearly interpolated grid-point controllers. As a result, the structured uncertainty matrix $\Delta$ may not be optimally considered, which may reduce the performance of the closed-loop system. However the results are very satisfactorily.

The stable, real, rational, proper, minimum-phase scaling matrix $D = D(s)$ for the LPV controller synthesis, is obtained by taking only the first two diagonal blocks of the scaling matrix $D_L(s)$, which was generated during the $\mu$-synthesis approach in section 6.5. The dynamic order of $D(s)$ is five.

### 6.6.2 Linear point designs

To determine the best achievable performance for the LPV-controlled system, three independent $H_\infty$ controllers, one for each plant $P_i$, using the interconnection of figure 6.18, are synthesized. As for these designs, no dependence or variation of $v$ is considered, the resulting performance levels $\gamma$ should serve as a lower bound for the gain-scheduled LPV-controlled single-track model. The resulting values of $\gamma$ are shown in Table 6.7, where the best performance is achieved for low speed, which may have the physical interpretation that achieving robust performance for road adhesion values $\mu$ down to 0.5 is more challenging for high-speed cases.

Table 6.7: Performance of $H_\infty$ point designs

| Model | $v$ | $\gamma$ |
|---|---|---|
| $P_1$ | 10 | 0.85 |
| $P_2$ | 30 | 0.90 |
| $P_3$ | 50 | 0.93 |

To equalize the objectives across operating points, the performance input $r_{\text{ref}}$ of the $P_i$ (see interconnection in figure 6.18) is normalized with the inverse of the achieved $H_\infty$ norm $\gamma$. With this scaling each $H_\infty$ point design will achieve a $\gamma$ of 1. Hence, the performance $\gamma$ obtained with the gain-scheduled LPV controller can be easily compared to the point designs.

### 6.6.3 LPV design

The main motivation for synthesizing an LPV controller was to guarantee robust performance in spite of bounded variation rates of the longitudinal speed $v$. To achieve good performance, the admissible variation rate should be as small as possible. From physical considerations the variation rate $\dot{v}$ of $v$ is assumed to be bounded by $\dot{v} \in [-10, 3]$, which may correspond to a maximum deceleration of -10 m/s$^2$ at $\mu = 1$ and a quite high acceleration of 3 m/s$^2$ (corresponding to an acceleration from 0-100 km/h in 9.2 s).

In appendix B it is briefly shown, that the performance analysis and synthesis for the rate-bounded LPV controller is based on parameter dependent scalings $X(\delta)$. For computational tractability - approximating an infinite dimensional problem with a finite-dimensional problem - it is necessary to define basis functions $f_i(\delta)$ to describe the $X(\delta) = \sum_{i=1}^{N} f_i(\delta) X_i$ matrices. Here, three basis functions were employed: the constant $f_1 = 1$, $f_2 = v$ and $f_3 = v^2$. Hence, the scalings $X(\delta)$ have the form

$$X(\delta) = X_0 + vX_1 + v^2 X_2.$$

In [87] there exists no recommendation how to choose these basis functions and only the parametric dependence of the plant may give some hints. For example, if the plant model $G(s, \delta)$ includes expressions as $\sin(\delta_1)$ one may also include a basis function $f_i = \sin(\delta_1)$ to describe $X(\delta)$.

The controller synthesis requires the solution of a large set of LMIs and is computationally quite demanding (200 s on a Pentium 4, 1.7 GHz with Windows XP and MATLAB 6.5). The proper coding of all the LMIs is cumbersome and error-prone. Therefore an already developed, not publicly available LPV-synthesis toolbox from Gary Balas was used to derive the gain-scheduled controller. The synthesis consists of a two step procedure, where in the second step a further convex optimization step is performed to limit the high-frequency controller poles. Finally, a gain-scheduled controller is derived, which can be simulated using additional MATLAB `mex`-functions. Due to the complexity of the toolbox and a lack of documentation it was not straightforward to add additional LMI constraints for the controller synthesis. Therefore especially the closed-loop pole specifications were not explicitly included and it can be seen, that for the resulting controller

the minimum damping requirement is slightly violated. However, in the time simulations (see section 6.7) there is no evidence of undamped behavior.

Finally, a controller consisting of three grid-point controllers, each of them of order 17, is synthesized. Consider the concatenated matrices

$$S_i = \left[\begin{array}{c|c} A_i & B_i \\ \hline C_i & D_i \end{array}\right], \; i = 1, \ldots, 3,$$

with $A_i \in \mathbb{R}^{17\times17}$, $B_i \in \mathbb{R}^{17\times1}$, $C_i \in \mathbb{R}^{1\times17}$, $D_i \in \mathbb{R}$ as the state-space matrices of the three grid-point controllers for $v_i \in \{10, 30, 50\}$. The state-space matrices $A(v)$, $B(v)$, $C(v)$, $D(v)$ of the gain-scheduled controller are given as

$$\left[\begin{array}{c|c} A(v) & B(v) \\ \hline C(v) & D(v) \end{array}\right] = \begin{cases} \left(\dfrac{30-v}{20}\right) S_1 + \left(\dfrac{v-10}{20}\right) S_2, & \forall\, v \in [10, 30] \\ \left(\dfrac{50-v}{20}\right) S_2 + \left(\dfrac{v-30}{20}\right) S_3, & \forall\, v \in [30, 50] \end{cases}$$

It is remarkable, that the resulting $H_\infty$ performance $\gamma$ for the gain-scheduled LPV controller is 1.02, which indicates that the rate-bounded LPV controller almost exactly recovers the performance of the $H_\infty$ point designs. Hence, by using the gain-scheduled LPV controller, the variations in $v$ do not have any adverse effect on the achievable performance.

### 6.6.4 Frequency weighted controller reduction

Frequency weighted controller reduction techniques are applied to each of the three grid-point controllers. All three controllers can be reduced to order 4 without loss of robust performance. The maximum natural frequency of the controller poles is 15.6 Hz and the minimum damping is 0.48. As in section 6.5.2, the same stability and performance enforcing weighting functions $W_o$ and $W_i$ were employed to describe the reduction error $e_c$ and the SPA-method was used to calculate the reduced order controllers. To allow linear interpolation of the reduced order controllers, the state-space matrices of the reduced order controllers are transformed to block-diagonal form using well-conditioned non-orthogonal similarity transformations. The concatenated representations $S_{1,r}$, $S_{2,r}$,

$S_{3,r}$ of the state-space matrices of the reduced order grid-point controllers are

$$S_{1,r} = \left[\begin{array}{rrrr|r} -46.92 & 185.44 & 0 & 0 & 35.34 \\ -40.15 & -46.92 & 0 & 0 & -9.16 \\ 0 & 0 & -0.06 & 0 & 6.30 \\ 0 & 0 & 0 & -4.03 & -1.81 \\ \hline 0.76 & 10.36 & 0.95 & 0.83 & 2.45 \end{array}\right]$$

$$S_{2,r} = \left[\begin{array}{rrrr|r} -58.54 & 165.74 & 0 & 0 & 37.69 \\ -26.91 & -58.54 & 0 & 0 & -12.04 \\ 0 & 0 & -0.06 & 0 & 6.37 \\ 0 & 0 & 0 & -2.78 & -2.92 \\ \hline -2.63 & 13.41 & 0.99 & 1.23 & 3.48 \end{array}\right]$$

$$S_{3,r} = \left[\begin{array}{rrrr|r} -64.34 & 169.85 & 0 & 0 & 41.25 \\ -22.56 & -64.34 & 0 & 0 & -13.58 \\ 0 & 0 & -0.06 & 0 & 7.57 \\ 0 & 0 & 0 & -1.69 & -4.36 \\ \hline -3.73 & 14.55 & 1.05 & 1.25 & 3.95 \end{array}\right].$$

### 6.6.5 Frequency domain results

In figure 6.19 the closed-loop poles for the vertices of the parameter domain for $m$, $v$, $\mu$ are shown. As no pole-placement constraints were explicitly included for control design, the resulting closed-loop poles slightly violate the minimum-damping requirement, where a minimum damping of 0.43 is obtained for $\mu = 1$, $v = 10$ and $m = 1296$. A maximum natural frequency of 15.1 Hz is reached for $\mu = 0.5$, $v = 10$ and $m = 1696$, and for $\mu = 0.5$, $v = 50$, $m = 1696$ a closed-loop pole is located at $-1.15$ rad/s, which is larger than the desired maximum real-part of $-2$ rad/s.

The specifications on the sensitivity, complementary sensitivity and controller sensitivity functions are fulfilled for the vertices of the parameter domain as shown in figures 6.20, 6.21 and 6.22.

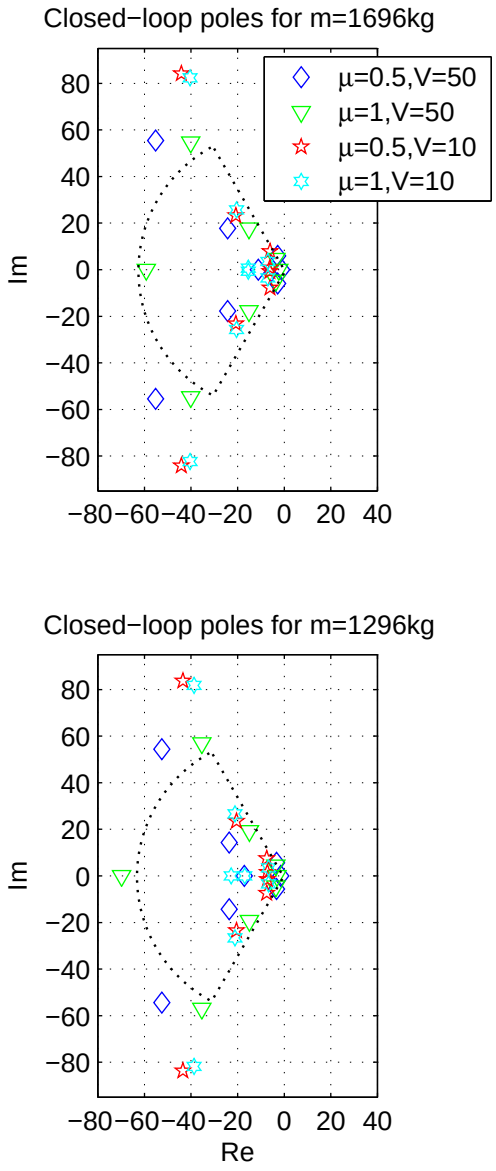


Figure 6.19: Closed-loop poles

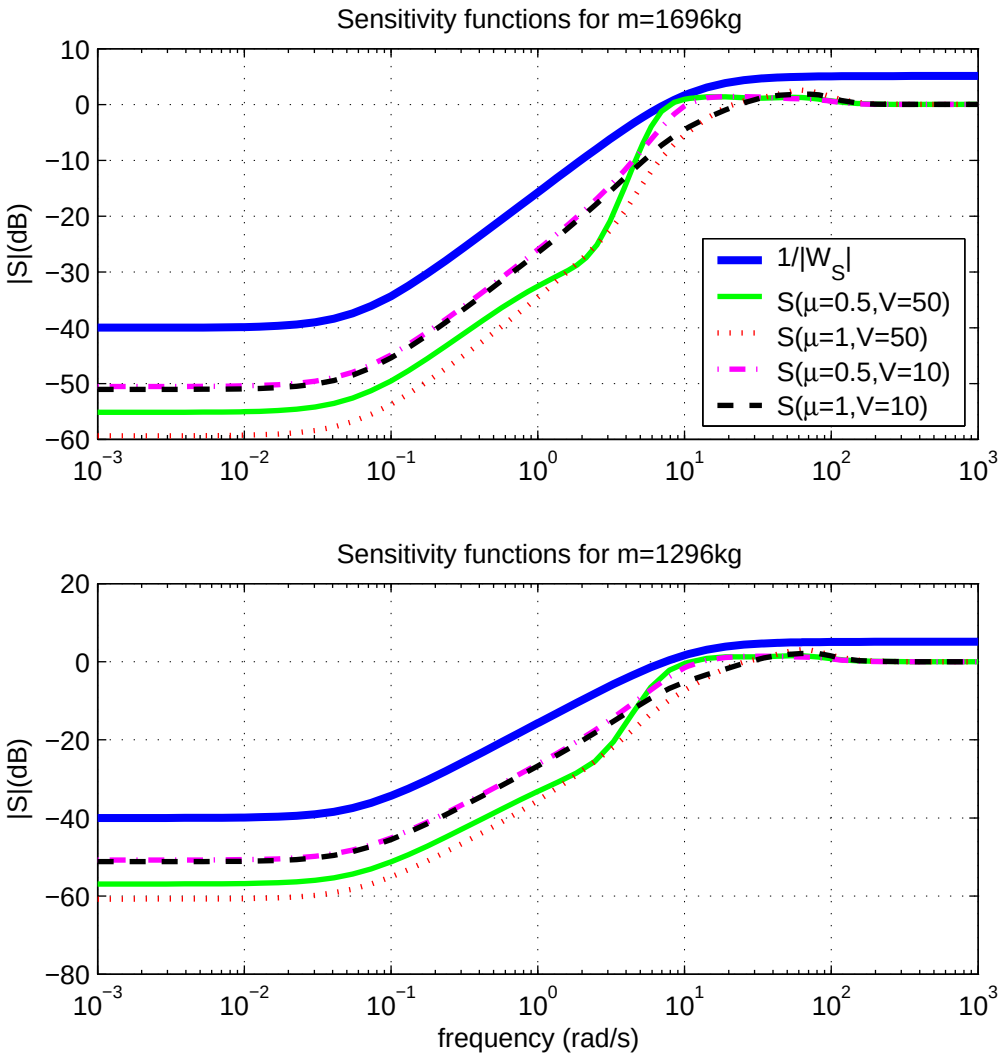


Figure 6.20: Bode diagram of sensitivity functions

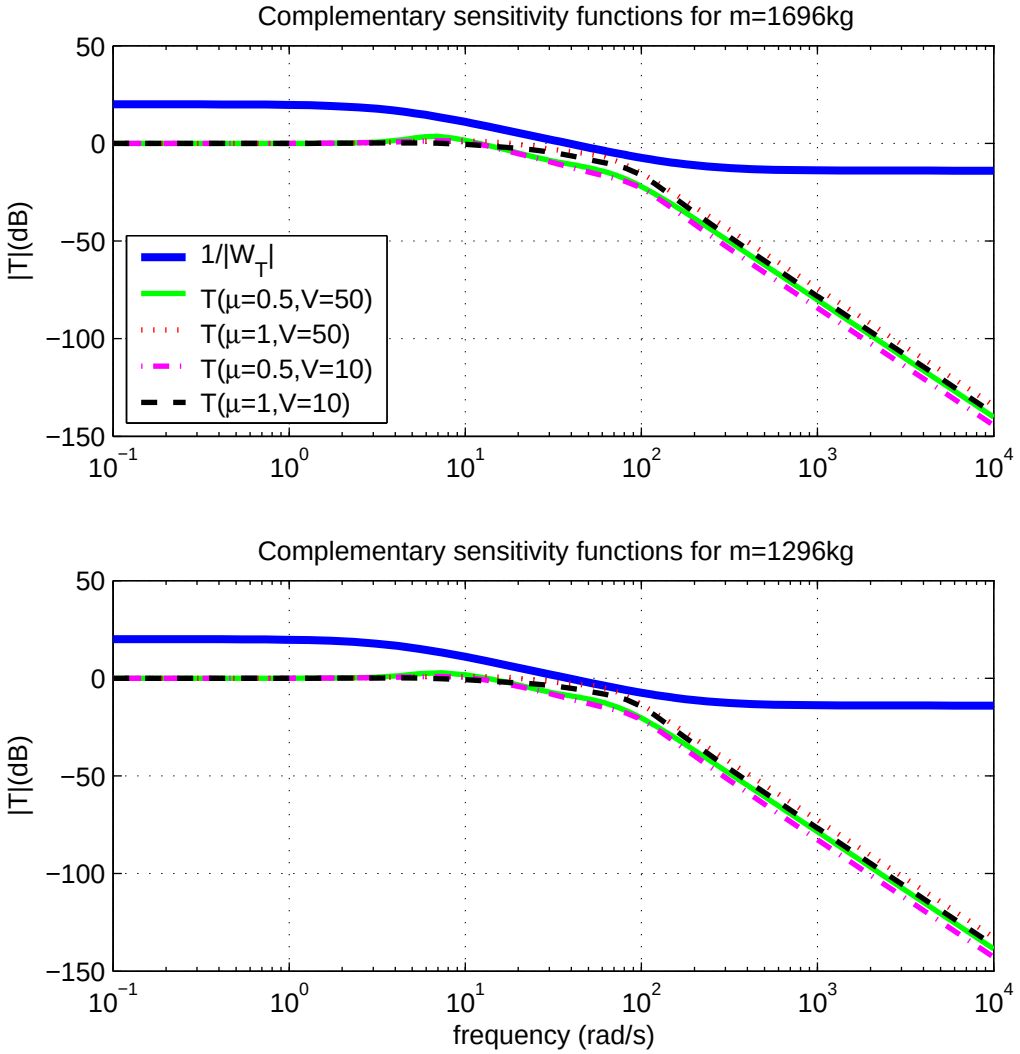


Figure 6.21: Bode diagram of complementary sensitivity functions

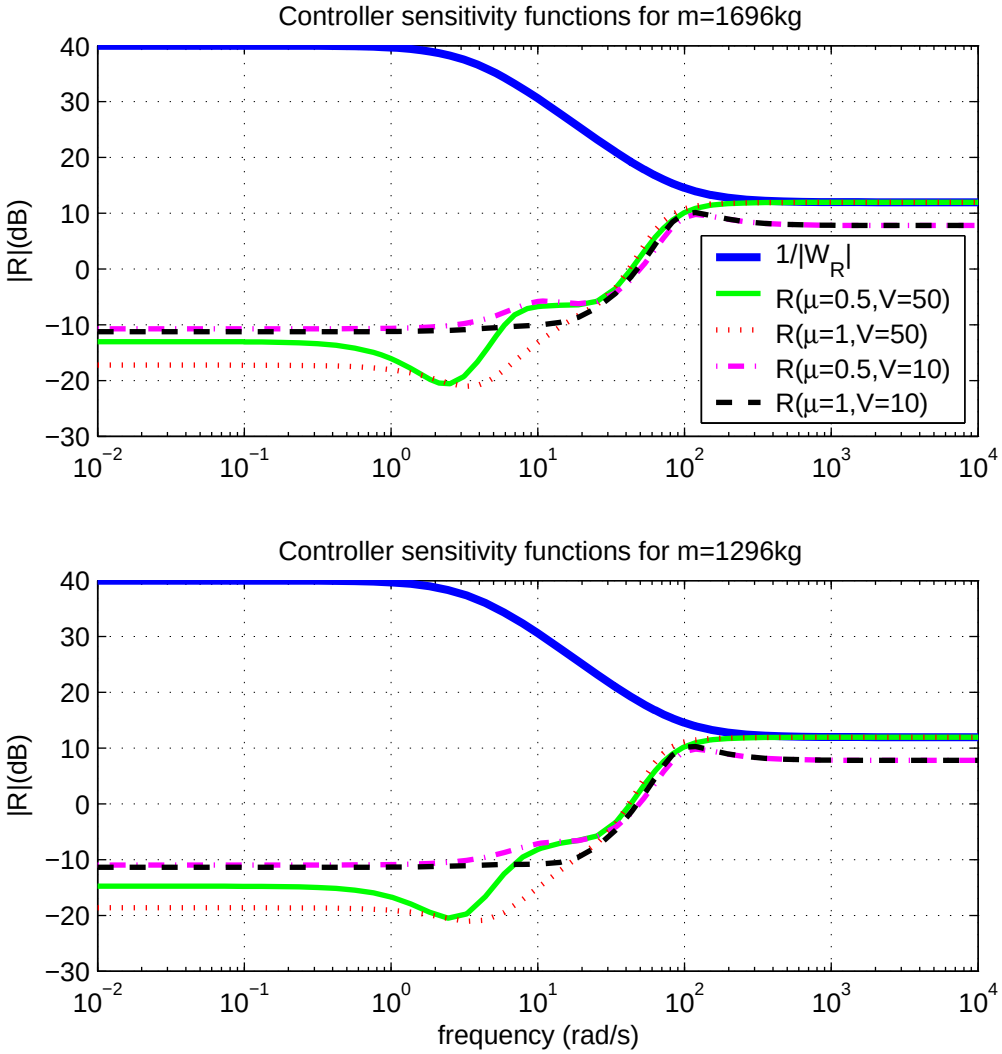


Figure 6.22: Bode diagram of controller sensitivity functions

## 6.7 Simulation results

In this section, the simulation results for the closed-loop system as shown in figure 6.4 are presented, where $K = K(s)$ for the $\mu$-synthesis controller and $K = K(s, v)$ for the LPV- controller. The linear single-track model is taken as plant model $G$ and $G_a$ is given by the actuator model as shown in figure 6.2. As variations in the mass $m$ have no qualitative effect on the simulation results, all simulations are performed for nominal mass $m = 1496$.

Figures 6.23-6.29 represent the reaction of the controlled vehicle to a yaw disturbance torque $M_z$ step input, which may qualitatively describe a $\mu$-split braking manoeuvre. For simulation, the matrices $B(\delta)$ and $D$ of the single-track model as given in (6.1) are substituted by

$$\overline{B} = \begin{bmatrix} 0 & 0 \\ c_f\mu & 0 \\ c_f l_f \mu & J_z \\ 0 & 0 \end{bmatrix}, \overline{D} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

and the new input vector is given by $\overline{u} = \begin{bmatrix} \delta_f & M_z \end{bmatrix}^T$. For comparison the results obtained in [20] are also included in the plots and denoted as PS (*Parameter-Space* design). The uncontrolled vehicle is denoted as OL (Open-Loop).

Figures 6.30-6.33 describe the tracking behavior of the closed-loop system, where a steering wheel step response is shown. It is shown how the reference $r_{\text{ref}}$ (see figure 6.4) can be tracked for $\mu = 0.5$. No comparison with [20] is given for these results as a different reference model was used therein.

In detail, figures 6.23 and 6.24 show the low velocity ($v = 10$) simulations results for a step input of a yaw disturbance torque at $\mu = 0.5$ and $\mu = 1$, respectively. For $\mu = 0.5$ the best result is achieved using the LPV controller, where the maximum yaw rate $r$ is 11% smaller than for the $\mu$-controller. After 0.3 s the LPV controller achieves zero yaw rate and the following negative yaw rate may further help to correct the vehicle motion. For $\mu = 1$ the maximum $r$ is almost the same for all controllers. The LPV controller shows the best convergence properties and the PS controller shows undamped behavior, which may be uncomfortable to the driver.

For the high velocity cases shown in figures 6.25 and 6.26 the LPV controller and the $\mu$-controller clearly outperform the PS controller. For $\mu = 0.5$, the PS controller allows a maximum $r$, which is 19% and 33% larger compared to the $\mu$-controller and LPV controller, respectively. This case is probably the most demanding, as it describes a manoeuvre at high speed on wet road. For $\mu = 1$, the PS-controller allows a maximum $r$, which is 24% and 30% larger compared to the $\mu$-controller and LPV controller, respectively.

As during manoeuvres like $\mu$-split braking a yaw disturbance torque is usually combined with a variation of the velocity $v$, a maybe more realistic scenario is presented in figures 6.27, 6.28, 6.29, where a repeated yaw disturbance step torque $M_z$ is combined with an acceleration of $\dot{v} = 3$ or a full-braking with $\dot{v} = -10$. In all cases, the LPV controller, which is explicitly designed to guarantee robust performance despite bounded variation rates of $v$, clearly provides the best disturbance rejection. For yaw disturbance torques

at velocities $v > 15$, the $\mu$-controller generally achieves better disturbance rejection than the PS controller.

The tracking results for the $\mu$-synthesis and LPV controller are shown in figures 6.30-6.33, where $r_{\text{ref}}$ denotes the yaw rate of the reference model and OL denotes the yaw rate of the vehicle without control. Both controllers show very good tracking performance. In general the maximum tracking error $e$ is about 10% smaller for the LPV controlled vehicle, however the $\mu$-controller achieves a better damping of the error, which results from the larger damping value of the closed-loop poles (see figure 6.13 and 6.19).

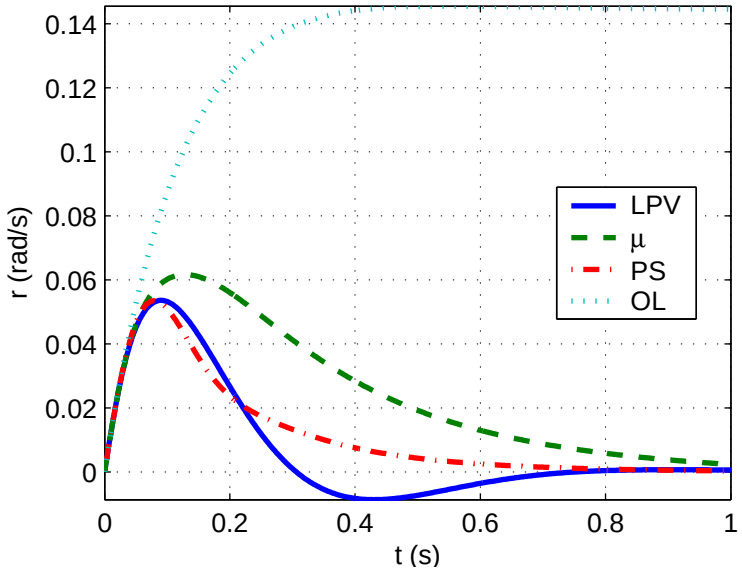


Figure 6.23: Yaw disturbance torque at $v = 10$, $\mu = 0.5$

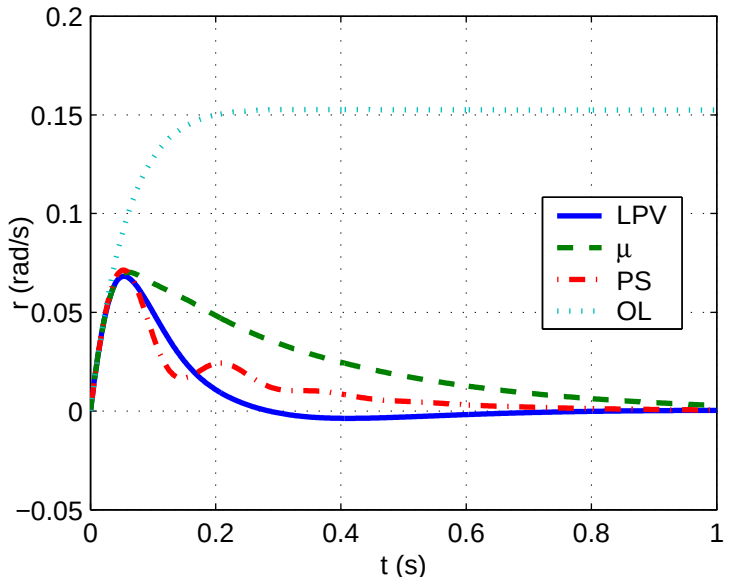


Figure 6.24: Yaw disturbance torque at $v = 10$, $\mu = 1$

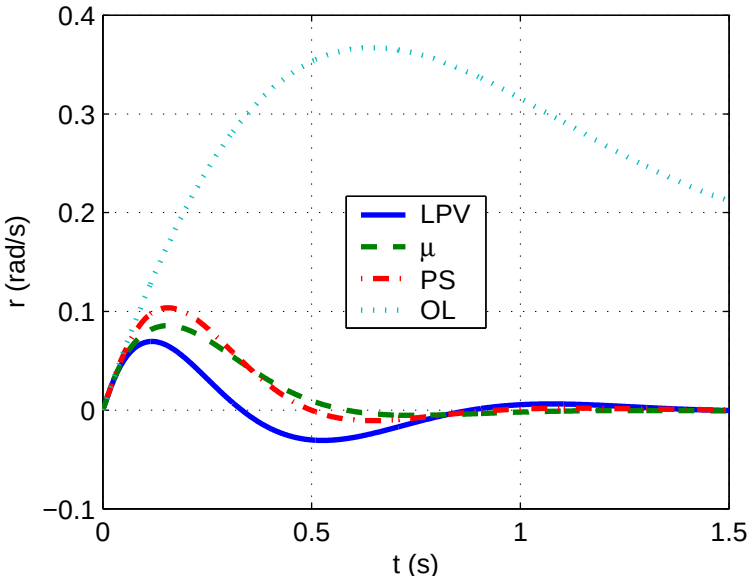


Figure 6.25: Yaw disturbance torque at $v = 50$, $\mu = 0.5$

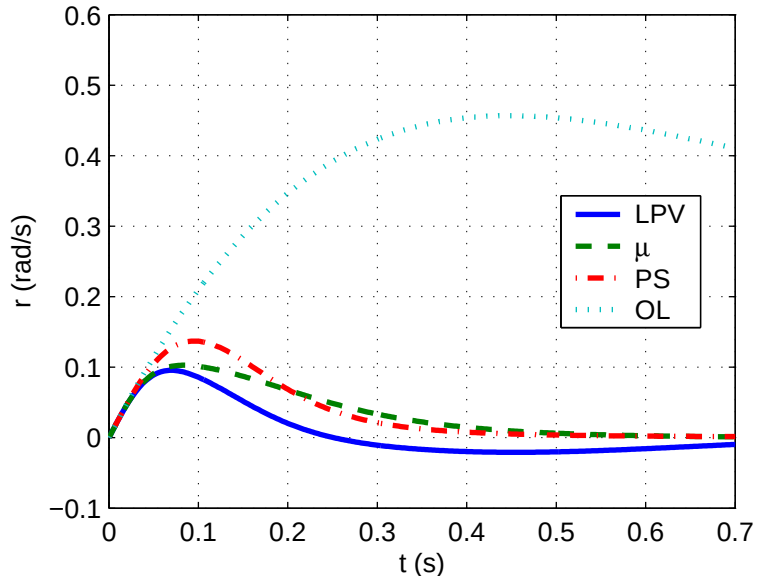


Figure 6.26: Yaw disturbance torque at $v = 50$, $\mu = 1$

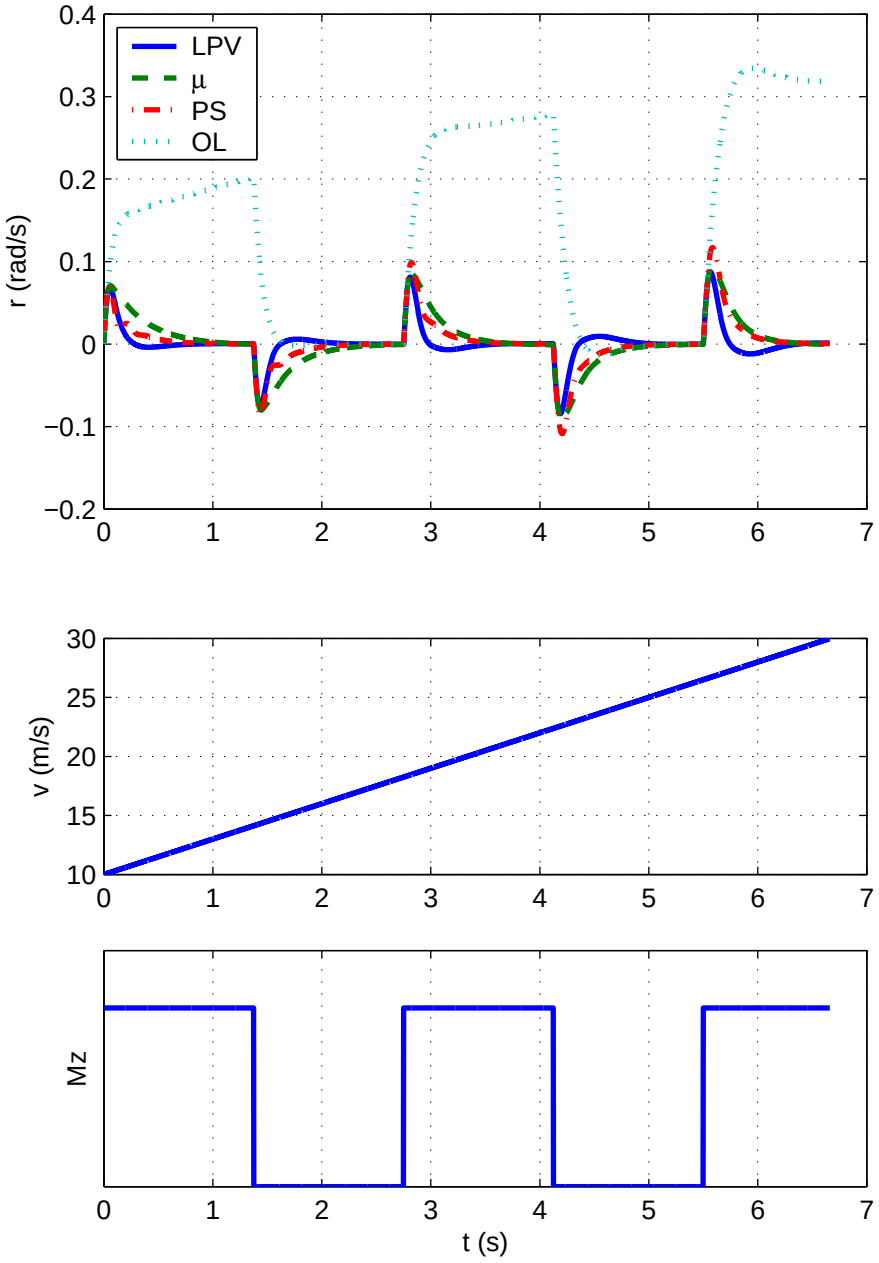


Figure 6.27: Repeated yaw disturbance torque at $v(0) = 10$, $\dot{v} = 3$, $\mu = 1$

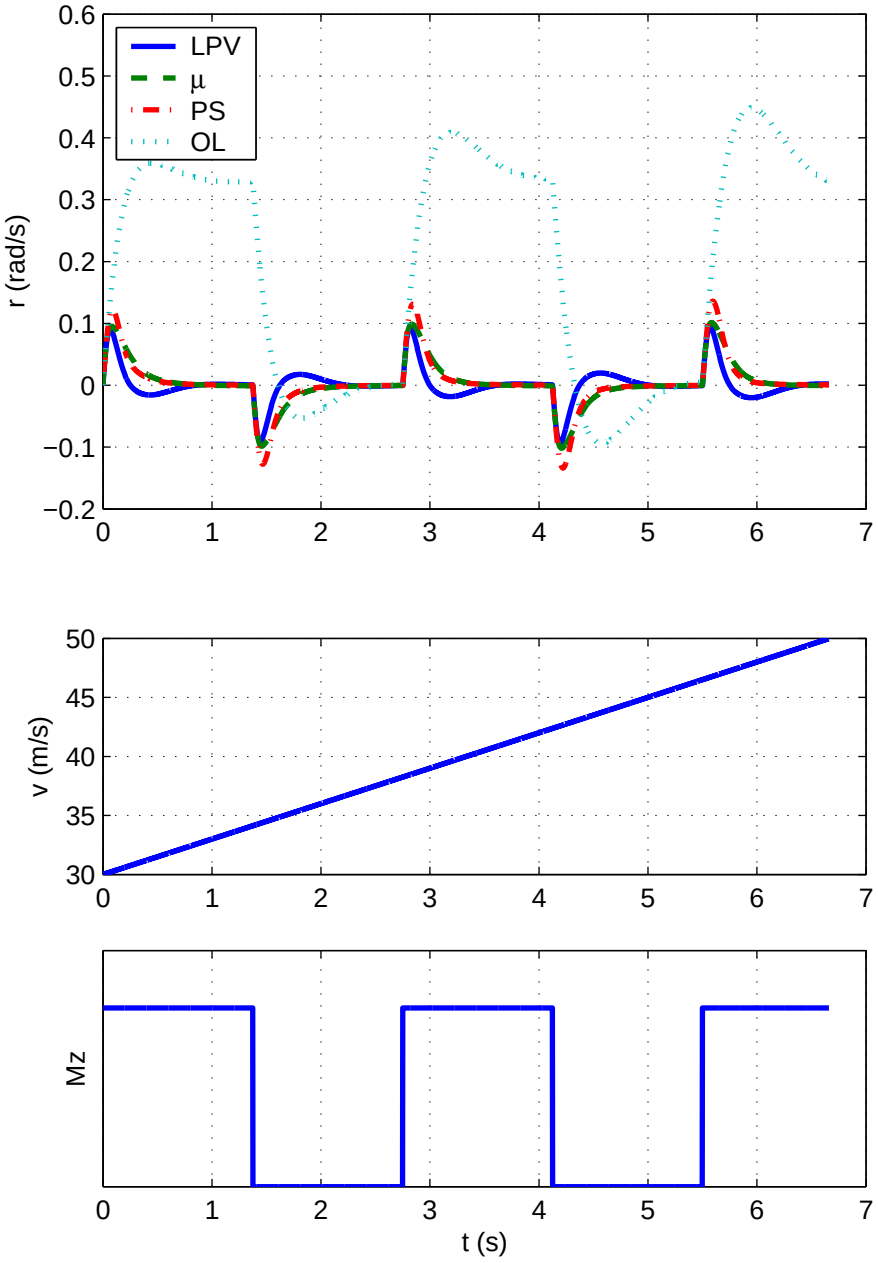


Figure 6.28: Repeated yaw disturbance torque at $v(0) = 30$, $\dot{v} = 3$, $\mu = 1$

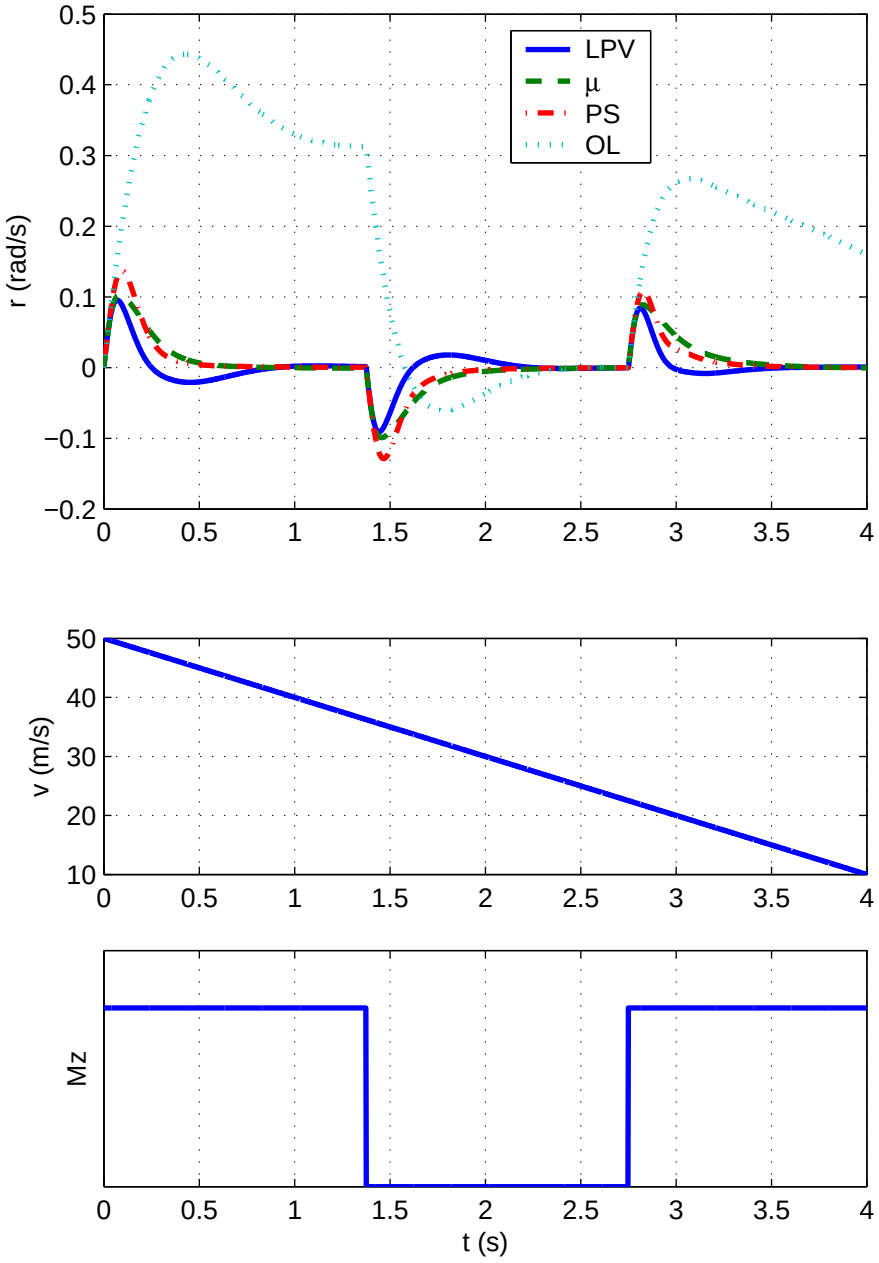


Figure 6.29: Repeated yaw disturbance torque at $v(0) = 50$, $\dot{v} = -10$, $\mu = 1$

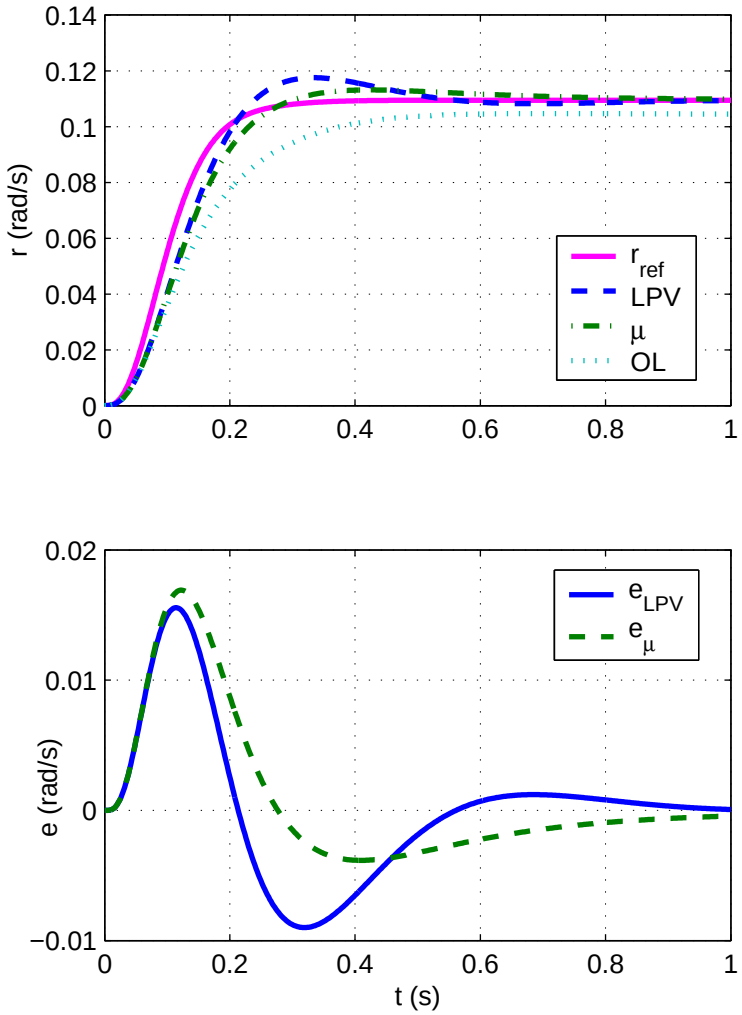


Figure 6.30: Steering wheel step response at $v = 10$, $\mu = 0.5$

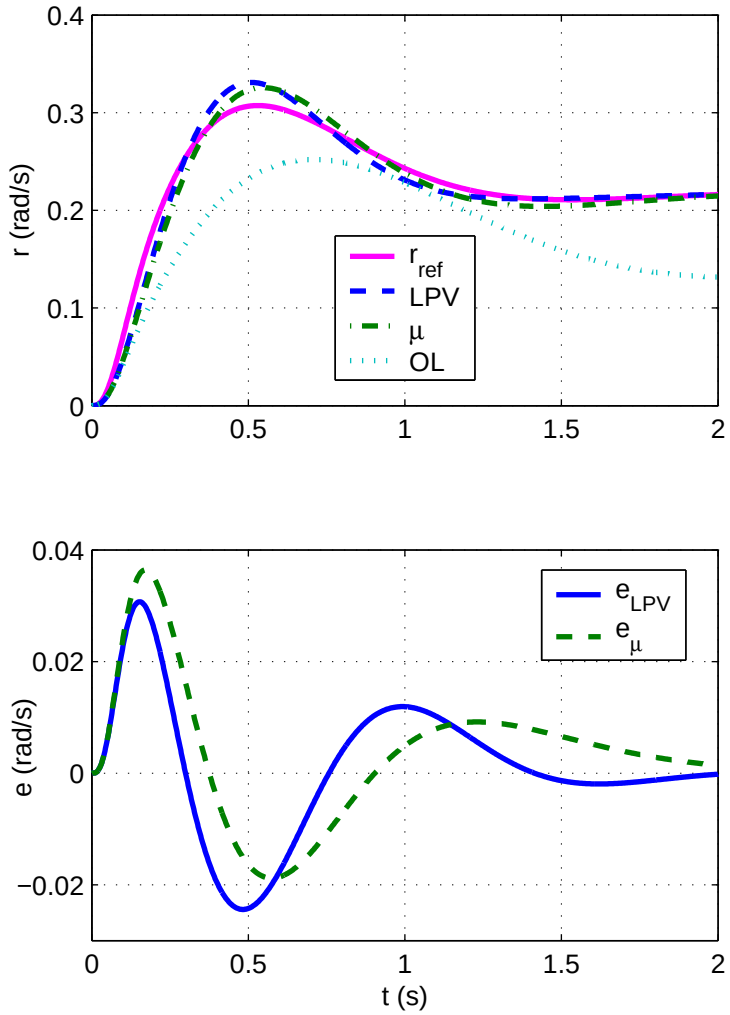


Figure 6.31: Steering wheel step response at $v = 50$, $\mu = 0.5$

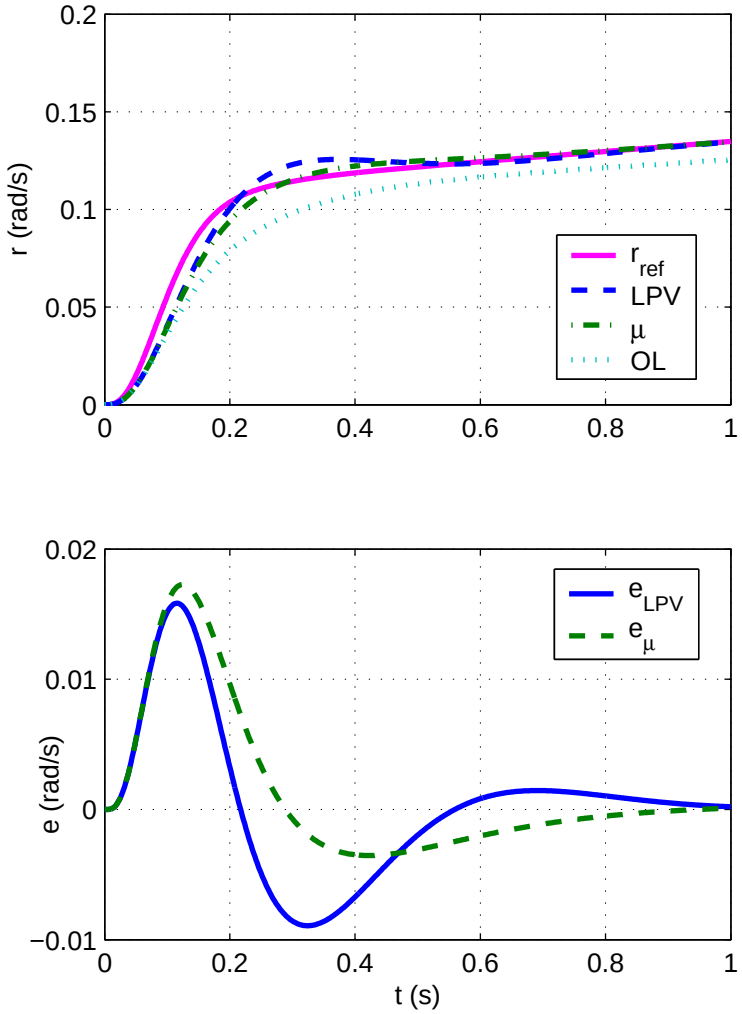


Figure 6.32: Steering wheel step response at $v(0) = 10$, $\dot{v} = 3$, $\mu = 0.5$

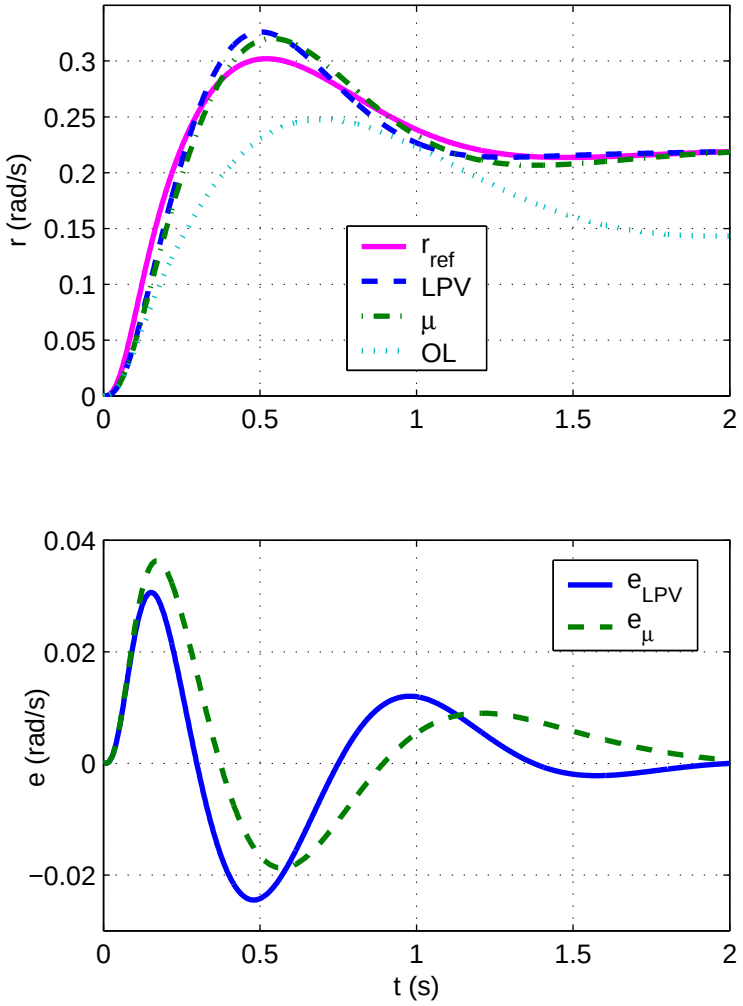


Figure 6.33: Steering wheel step response at $v(0) = 50$, $\dot{v} = -5$, $\mu = 0.5$

## 6.8 Summary

Enhanced symbolic preprocessing techniques and object-oriented LFR realization based on the newly developed generalized LFR were employed to obtain a minimal order LFR for the parametric single-track model, where parametric uncertainties with respect to mass $m$, moment of inertia $J_z$, road adhesion $\mu$ and longitudinal velocity $v$ were assumed. Based on this LFR, two controllers, a $\mu$-synthesis controller and a gain-scheduled LPV controller were designed. The mixed sensitivity specifications for the control design were taken from [20], where a controller was synthesized based on the Parameter Space (PS) method. The PS method is usually restricted to handle only up to two parameters (in our case $\mu$ and $v$) as structured uncertainties. Therefore in [20], additional uncertainties with respect to $m$ and $J_z$ were conservatively approximated as an unstructured uncertainty. For the controllers developed in this thesis, all parametric uncertainties are explicitly considered as structured uncertainties, which resulted in a better closed-loop performance.

The developed controllers are synthesized based on $H_\infty$ control design methods. A general criticism about these methods is the high complexity/dynamic order of the resulting controllers, which is equal to the order of the weighted plant. For the considered vehicle, the orders of the $\mu$-synthesis controller and the LPV controller were 33 and 17, respectively. Furthermore, the employed MATLAB implementations for $H_\infty$ control design tend to generate controllers with very fast dynamics. However, at least for the vehicle control design, all these problems could be simply alleviated by applying frequency-weighted controller reduction techniques, reducing the order of the controllers to 3 for the $\mu$-synthesis controller and 4 for the LPV controller (compared to 3 for the PS controller). The maximum natural frequency of the closed-loop poles was limited to about 15 Hz, which clearly allows real-time implementation of the controllers. All this could be achieved without any degradation of closed-loop stability and performance.

The $\mu$-controller and the LPV controller fulfill the mixed-sensitivity specifications for a considerably larger uncertainty region than the PS controller. From the time simulations, the LPV controller shows the best performance for all road, mass and velocity conditions. At low speed, the time-responses with the PS controller show better performance than the responses with the $\mu$-controller, but at velocities larger than about 15 m/s the $\mu$-controller shows better performance than the PS controller. Especially for the most demanding case, which is a manoeuvre at high velocity ($v$=50 m/s) on wet road ($\mu = 0.5$), the LPV controller and the $\mu$-controller clearly outperform the PS controller.

The gain-scheduled LPV controller guarantees robust performance also for bounded variation rates of $v$, which also results in better performance for such cases. All controllers are designed to reject yaw disturbances, which in real-life situations (e.g., $\mu$-split braking) are usually combined with bounded variations rates of $v$. This motivates the usage of the LPV controller, as at least from theory, the $\mu$-controller and the PS controller do not guarantee robust stability for non-constant parametric uncertainties. The stability-guaranteeing scheduling policy for the LPV controller is directly obtained from the control design and no heuristic switching strategy must be designed, which may affect robust stability.

It is important to note, that the LPV controller and the PS controller need a measurement of the longitudinal vehicle velocity $v$ for the inner control loop. This is not the

case for the $\mu$-controller, which therefore needs less sensor information and allows a more fault-tolerant/robust integration into the vehicle.

# 7 Summary and Future Directions

This thesis deals with new methods and tools for the realization of low order Linear Fractional Transformation (LFT) based representations, which are a standard form to represent uncertainties, nonlinearities and varying parameters in modern robust control theory [92]. The transformation of a parametric system into a Linear Fractional Representation (LFR) is not unique and is equivalent to a multidimensional system realization [18], for which a theory for the generation of representations with minimal order does not exist. The pure application of simple ad-hoc methods [77] for LFR realization tends to generate LFRs with large orders, which may limit/forbid the application of numerically demanding modern robust control techniques. To realize LFRs of almost minimal complexity, a three step procedure is employed in this thesis consisting of (i) symbolic preprocessing of parametric systems using improved and newly developed decomposition techniques, (ii) object-oriented LFR realization based on a newly developed generalized LFR, (iii) numerical multidimensional order reduction based on newly implemented numerically reliable and efficient routines.

Symbolic preprocessing has the role to exploit structural information in the parametric dependence of the system model and to find equivalent representations of the system model, which are specially suited to generate low order LFRs in conjunction with the object-oriented LFR realization step. Furthermore, symbolic methods perform without any loss of accuracy, which is not the case for tolerance dependent numerical order reduction techniques. In this thesis newly developed methods as the Variable Splitting (VS) factorization or the Enhanced Tree Decomposition (ETD) [44, 43] are presented and an overview about existing techniques is given. All techniques are compared by realizing LFRs for a vehicle model (chapter 6) and one of the most complicated parametric aircraft models (chapter 5) available in literature. A combination of the VS and ETD techniques allows to directly realize a minimal order LFR (order 6) for the vehicle model and a nearly minimal order LFR (order 66) for the aircraft model. Especially for the aircraft model, the new methods clearly outperform existing techniques, which directly yield LFRs with orders of at least 137. The impressive order reduction results obtained for rather complex uncertain models show that symbolic preprocessing is the most important step in obtaining low-order LFRs.

The object-oriented LFR realization [77] is the most flexible and efficient way to transform a rational parametric system model into an LFR. However, employing standard LFRs during this step generally requires to perform a preliminary symbolic normalization of the system parameters, which may increase the order of the resulting LFR. A newly developed generalized LFR [42, 41] is presented in this thesis, which allows to directly realize LFRs for rational parametric system models and to perform the normalization of parameters after the object-oriented LFR realization step. Therefore any unnecessary increase of the LFR order is avoided.

Numerical multidimensional order reduction [77, 29] is applied as the last step of the LFR realization procedure and may allow to remove non-minimal system parts, resulting from the preceding steps. For the very complicated aircraft example in chapter 5, numerical order reduction allowed a further marginal reduction of the order from 66 to 65. This shows that the newly developed symbolic preprocessing methods almost tap the full potential in terms of achievable order reduction, while keeping the full accuracy of the original system model. However, for other examples, numerical order reduction techniques may considerably reduce the order and therefore these methods are employed as a complementary third step in the overall low order LFR realization procedure. In addition, numerical order reduction may be used to further reduce the order of an LFR by calculating sufficiently accurate numerical approximations.

The newly developed Enhanced LFR-toolbox for MATLAB [47] (chapter 4) fully supports all symbolic preprocessing methods as described in chapter 3. Most of the functions are directly implemented in Maple and efficiently executed via the Maple kernel of the Extended Symbolic Toolbox of MATLAB. The toolbox also employs the generalized LFR during the object-oriented LFR realization and provides numerically reliable and efficient implementations for multidimensional numerical order reduction. The underlying system manipulations are performed by calling efficient Fortran routines via `mex`-function interfaces, allowing to reduce the computation times up to a factor of 100 compared to standard MATLAB tools. Hence, the LFR-toolbox provides a professional tool to fully automate the proposed three-step low-order LFR realization procedure. The LFR-toolbox fully supports conversions between LFR-objects and objects from the Robust Control toolbox, which also allows to employ the efficient order reduction routines from the LFR-toolbox as a complementary tool to the standard MATLAB order reduction routines.

The capabilities of the toolbox are demonstrated by realizing an almost minimal order LFR for the complex parametric aircraft model called RCAM (chapter 5). This LFR allows the application of $\mu$-analysis to analyze robust stability for the whole flight envelope, which was not possible with earlier generated LFRs of higher complexity. The $\mu$-analysis results fully agree with results obtained via an optimization-based worst-case search. Hence, $\mu$-analysis based on low order and accurate LFRs represents a fast complementary tool to asses robust stability.

A minimal order LFR could be realized for a uncertain parametric model representing the lateral dynamics of a vehicle. Parametric uncertainties/parameter variations are considered for the road adhesion, the longitudinal velocity, mass and moment of inertia of the vehicle. Based on this LFR, two robust vehicle steering controllers were synthesized to improve the yaw dynamics of the vehicle and to increase safety during dangerous situations as $\mu$-split braking. The controller specifications were taken from [20]. Both designs robustly fulfill the specifications on the sensitivity, complementary sensitivity and controller sensitivity functions for a considerably larger uncertainty region than the Parameter Space (PS) controller synthesized in [20]. Without loss in robust stability and performance, the high initial orders of the controllers could be remarkably reduced to at least an order of 4 by using frequency weighted controller reduction techniques. One design was performed using $\mu$-synthesis, guaranteeing robust stability and performance for constant uncertain parameters. As realistic dangerous driving situations as $\mu$-split braking are always combined with relatively high variations rates of the longitudinal

vehicle velocity, the second controller design is based on Linear Parameter Varying (LPV) techniques, guaranteeing robust stability and performance also for bounded variation rates of the vehicle velocity. The resulting LPV controller is scheduled with the vehicle velocity, which therefore requires an online measurement of the vehicle velocity. In the time-simulations, the LPV controller shows the best closed-loop performance results for all mass, road and velocity conditions and in most cases the $\mu$-controller outperforms the PS controller from [20]. A reason for this is that the two LFR-based control designs performed in this thesis allow to non-conservatively consider all parametric uncertainties as structured uncertainties, which is usually limited to 2 parameters for the PS method. Compared to the PS controller and the LPV controller, the $\mu$-synthesis controller does not require a measurement of the vehicle velocity for the inner control loop and therefore allows a more robust/fault-tolerant integration into the vehicle.

Future directions of this research may be concentrated on the improvements of symbolic preprocessing techniques. Actually, to achieve the best results in terms of low order LFR realization, most of the symbolic preprocessing methods provide a "try-hard" option to check all possible combinations/permutations of parameters. This may be very time-consuming for complex models with many parameters (e.g., more than 10 parameters) and therefore it will be important to develop a theory or guidelines, which allow to directly find the best parameter combination/permutation.

For the symbolic decomposition of vectors and matrices, the VS factorization algorithm includes heuristic "condensation" algorithms that increase the order reduction capabilities. Further improvements may be obtained by developing more sophisticated techniques.

# A Structured Singular Value ($\mu$) Framework

In this chapter a general framework for robust stability/performance analysis and robust controller synthesis based on the structured singular value $\mu$ is briefly introduced. For a detailed discussion see [92, 91]. The general interconnection structure for any linear interconnection of input, outputs, disturbances, model perturbations and controllers is illustrated in figure A.1, where $e$ is an error vector, $v$ are external inputs, $y$ is the measurement output to be used for control, $u$ is the controller output, $z$ and $w$ are the input and output to the uncertainty block $\Delta$, $P$ is the interconnection system and $K$ is the controller.

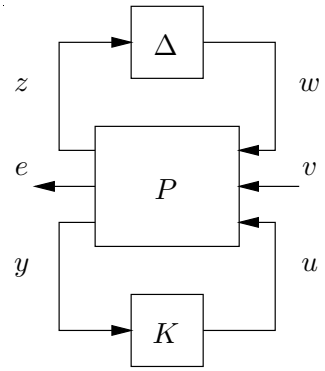


Figure A.1: General Interconnection Structure

Uncertainties can be modelled either as exogenous disturbances (included in $v$) and as model perturbations (included in $\Delta$). While external disturbances can not destabilize the plant, this is not the case for uncertainties represented as model perturbations.

As an example the interconnection system $P$ for the uncertain controlled plant of figure 1.1 is given by

$$\begin{bmatrix} z \\ e \\ y \end{bmatrix} = P \begin{bmatrix} w \\ v \\ u \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & I \\ I & I & 0 & -I & G \\ -I & -I & -I & I & -G \end{bmatrix} \begin{bmatrix} w \\ d \\ n \\ r \\ u \end{bmatrix},$$

with $v = \begin{bmatrix} d^T & n^T & r^T \end{bmatrix}^T$ as the external input vector, $w = (\Delta G)z$ and $u = Ky$.

The interconnection system $P$ is assumed to be a finite dimensional LTI (linear time invariant) system of the form

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) & P_{13}(s) \\ P_{21}(s) & P_{22}(s) & P_{23}(s) \\ P_{31}(s) & P_{32}(s) & P_{33}(s) \end{bmatrix}. \tag{A.1}$$

Closing the feedback loops for $K$ and $\Delta$ yields

$$e = \mathcal{F}_u(\mathcal{F}_l(P, K), \Delta)v = \mathcal{F}_l(\mathcal{F}_u(P, \Delta), K)v.$$

## A.1 Definitions

The following notations and terminologies will be used in the following:

**Definition A.1.** Consider the general interconnection structure in figure A.1, where the nominal closed loop system is given for $\Delta = 0$. Then, the closed-loop system is said to have

- **Nominal Stability (NS):** if the nominal closed-loop system is stable
- **Robust Stability (RS):** if the closed-loop system is stable for all possible closed-loop systems described by the uncertainty structure
- **Nominal Performance (NP):** if the performance requirements are met for the nominal closed-loop system
- **Robust Performance (RP):** if the performance requirements are met for all possible closed-loop systems described by the uncertainty structure

**Definition A.2. Analytic functions:** Let $S \subset \mathbb{C}$ be an open set, and let $f(s)$ be a complex-valued function defined on $S$, that is, $f(s) : S \to \mathbb{C}$. Then, $f(s)$ is said to be *analytic* at a point $z_0 \in S$ if it is differentiable at $z_0$ and also at each point in some neighborhood of $z_0$.

**Definition A.3. $\mathbf{H}_\infty$ Space:** The Hardy space $H_\infty$ describes all complex-valued functions with complex argument that are analytic and bounded in the open right-half plane. The $H_\infty$ norm is defined as

$$\|F\|_\infty := \sup_{Re(s)>0} \overline{\sigma}[F(s)] = \sup_{w\in\mathbb{R}} \overline{\sigma}[F(j\omega)],$$

where $\overline{\sigma}$ denotes the maximum singular value. The real rational subspace of $H_\infty$ is denoted by $\mathcal{R}H_\infty$, which describes all proper and real rational stable transfer functions.

**Definition A.4. Well-posed:** A feedback system is said to be well-posed if all closed-loop transfer matrices are well-defined and proper. Well-defined means that the inverse in the expression of the transfer matrix exists (i.e., $(I - M_{11}\Delta)^{-1}$ in (2.11) must exist for all $\Delta$ is a defined set).

## A.2 Small Gain robust stability test

In this section a stability test for a nominally stable system under unstructured uncertainty is considered. For robust stability analysis it is assumed that the controller $K(s)$ is already designed and is integral part of the plant. Therefore, the general interconnection structure of figure A.1 is reduced to the already presented LFT structure in figure A.2, where

$$M(s) = \mathcal{F}_l(P(s), K(s)) = \begin{bmatrix} M_{11}(s) & M_{12}(s) \\ M_{21}(s) & M_{22}(s) \end{bmatrix}, \tag{A.2}$$

with

$$\begin{aligned} M_{11}(s) &= P_{11}(s) + P_{13}(s)K(s)(I - P_{33}(s)K(s))^{-1}P_{31}(s) \\ M_{12}(s) &= P_{12}(s) + P_{13}(s)K(s)(I - P_{33}(s)K(s))^{-1}P_{32}(s) \\ M_{21}(s) &= P_{21}(s) + P_{23}(s)K(s)(I - P_{33}(s)K(s))^{-1}P_{31}(s) \\ M_{22}(s) &= P_{22}(s) + P_{23}(s)K(s)(I - P_{33}(s)K(s))^{-1}P_{32}(s) \end{aligned}$$

and the transfer function between $v$ and $e$ is given as

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

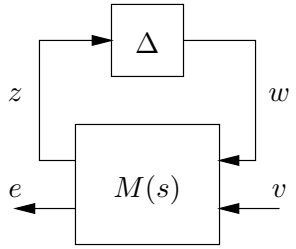


Figure A.2: Robust stability analysis structure

It is assumed that $\Delta(s) \in \mathcal{R}H_\infty$ and the controller $K(s)$ internally stabilizes the nominal plant, that is, $M(s) \in \mathcal{R}H_\infty$, and the nominal transfer function is given as $\mathcal{F}_u(M, 0) = M_{22}(s)$. Therefore, the transfer matrix (A.3) may only become unstable if the inverse $(I - M_{11}(s)\Delta)^{-1}$ becomes unstable, which is summarized in the following theorem.

**Theorem A.1. (Small Gain Theorem)** Suppose $M \in \mathcal{R}H_\infty$ and let $\gamma > 0$. Then the interconnected system shown in figure A.2 is well-posed and internally stable for all $\Delta(s) \in \mathcal{R}H_\infty$ with

(i) $\|\Delta\|_\infty \leq 1/\gamma$ if and only if $\|M_{11}(s)\|_\infty < \gamma$

(ii) $\|\Delta\|_\infty < 1/\gamma$ if and only if $\|M_{11}(s)\|_\infty \leq \gamma$

Hence robust stability of the LFR $(M(s), \Delta(s))$ requires that

$$\|M_{11}(s)\Delta(s)\|_\infty = \sup_{w\in\mathbb{R}} \overline{\sigma}[M_{11}(j\omega)\Delta(j\omega)] < 1.$$

This means that no pole will travel from the left half- plane into the right half- plane due to variations in $\Delta(s)$. The above theorem is necessary and sufficient only in the case of unstructured uncertainty $\Delta(s)$. In the case of parametric uncertainty, where $\Delta(s)$ has block-diagonal structure, the small gain test is only a sufficient stability criterion and may be arbitrarily conservative. This comes from the fact, that in the above theorem the norms of $M_{11}(s)$ and $\Delta(s)$ are calculated separately which does not consider any structure in the product of these matrices.

A generalization of the small gain test for systems with structured uncertainty is presented in the next section.

## A.3 $\mu$-Analysis

As already mentioned in chapter 1, a plant model may consist of several components (actuators, sensors, etc.) and each component may admit an individual uncertainty description, which can be of unstructured ($\Delta_j \in \mathbb{C}^{m_j\times m_j}$) or structured (parametric diagonal matrices $\Delta_i = \delta_i I_{r_i}$, $\delta_i \in \mathbb{C}$) type. Building the overall plant model, which consists of the component models and their related uncertainty descriptions, usually yields a system with structured uncertainty, i.e. unstructured uncertainty at component level becomes structured uncertainty at system level. Therefore, if the overall system model contains at least two components that admit an uncertainty description, one obtains a structured uncertainty for the overall system and the conservativeness of the related analysis or controller synthesis results may be reduced by explicitly considering this structure.

### A.3.1 Definition of $\mu$

Let the set $\mathbf{\Delta}$ of uncertainty matrices $\Delta$ be defined by

$$\mathbf{\Delta} = \{\mathrm{diag}(\delta_1 I_{r_1}, \ldots, \delta_k I_{r_k}, \Delta_1, \ldots, \Delta_f) : \delta_i \in \mathbb{C},\ \Delta_j \in \mathbb{C}^{m_j\times m_j}\} \subset \mathbb{C}^{n\times n}$$

and the norm-bounded subset $\mathbf{B\Delta}$ of $\mathbf{\Delta}$ is defined with

$$\mathbf{B\Delta} = \{\Delta \in \mathbf{\Delta} : \overline{\sigma}(\Delta) \le 1\}.$$

Note, to simplify notation, the full blocks $\Delta_j$ are assumed to be square.

**Definition A.5.** For $M_{11} \in \mathbb{C}^{n\times n}$, $\mu_{\mathbf{\Delta}}(M_{11})$ is defined as

$$\mu_{\mathbf{\Delta}}(M_{11}) := \frac{1}{\min\{\overline{\sigma}(\Delta) : \Delta \in \mathbf{\Delta},\ \det(I - M_{11}\Delta) = 0\}}$$

unless no $\Delta \in \mathbf{\Delta}$ makes $(I - M_{11}\Delta)$ singular, in which case $\mu_{\mathbf{\Delta}}(M_{11}) := 0$.

In words $\mu_{\mathbf{\Delta}}(M_{11})$ is the reciprocal of the smallest $\overline{\sigma}[\Delta]$ that can be found in the set $\mathbf{\Delta}$, that makes the matrix $(I - M_{11}\Delta)$ singular. If no such $\Delta$ exists, $\mu_{\mathbf{\Delta}}(M_{11})$ is taken to be zero.

The main difficulty with $\mu_{\mathbf{\Delta}}(M_{11})$ is, that $\mu_{\mathbf{\Delta}}(M_{11})$ can not be calculated directly. However, one can calculate tight bounds, which are described in the next section.

### A.3.2 Bounds on $\mu$

In section A.2 the Small Gain Theorem was presented as a necessary and sufficient condition for well-posedness (non-singularity) of the inverse $(I - M_{11}\Delta)^{-1}$, if $\Delta \in \mathbb{C}^{n\times n}$. Hence for $\Delta \in \mathbb{C}^{n\times n}$, the following holds

$$\mu_{\mathbf{\Delta}}(M_{11}) = \|M_{11}\|_\infty = \overline{\sigma}(M_{11}).$$

However, for $\Delta \in \mathbf{\Delta}$, where $\Delta$ is a block-diagonal matrix, the small-gain test is only a sufficient condition, such that

$$\mu_{\mathbf{\Delta}}(M_{11}) \leq \overline{\sigma}(M_{11})$$

and $\overline{\sigma}(M_{11})$ can be taken as an upper bound for $\mu_{\mathbf{\Delta}}(M_{11})$.

Furthermore, it can be shown [91] that a lower bound for $\mu_{\mathbf{\Delta}}(M_{11})$ is given by the spectral radius $\rho(M_{11})$ and therefore

$$\rho(M_{11}) \leq \mu_{\mathbf{\Delta}}(M_{11}) \leq \overline{\sigma}(M_{11}).$$

By simple examples [91], it can be shown that the gap between these bounds can be arbitrarily large. Tighter bounds can be obtained by applying transformations on $M_{11}$ that do not change $\mu_{\mathbf{\Delta}}(M_{11})$, but reduce the gap between $\rho(M_{11})$ and $\overline{\sigma}(M_{11})$.

To refine the upper bound, let $\mathcal{D}$ be a subset of $\mathbb{C}^{n\times n}$ with

$$\mathcal{D} = \{\mathrm{diag}(D_{r_1}, \ldots, D_{r_k}, d_1 I_{m_1}, \ldots, d_f I_{m_f}) : \; D_{r_i} = D_{r_i}^* > 0, \; d_j \in \mathbb{R}_+\}, \tag{A.4}$$

such that for $D \in \mathcal{D}$ and $\Delta \in \mathbf{\Delta}$ the commutativity property $D\Delta = \Delta D$ holds. Since $D^{-1}D = I$ one may simple introduce the scaling matrices into the LFT-structure (left diagram in figure A.3) and due to commutativity of $D$ and $\Delta$ it follows (right diagram of figure A.3) that

$$\mu_{\mathbf{\Delta}}(M_{11}) = \mu_{\mathbf{\Delta}}(DM_{11}D^{-1}).$$

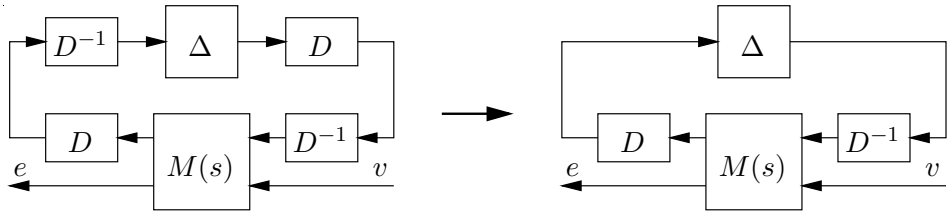


Figure A.3: Scaling for upper bound

More important is the fact, that the scaling $D$ can be used to improve the upper bound, that is,

$$\mu_{\mathbf{\Delta}}(M_{11}) \leq \inf_{D\in\mathcal{D}} \overline{\sigma}(DM_{11}D^{-1}) \leq \overline{\sigma}(M_{11}),$$

and the calculation of $D$ can be reformulated as a convex optimization problem, where a global minimum can be found. However, the upper bound is not always equal to $\mu_{\mathbf{\Delta}}(M_{11})$ [91], but it gives a tight estimate in many practical examples.

**Example A.1.** Consider

$$M_{11} = \begin{bmatrix} 0.5 & 100 \\ 0.001 & 0.1 \end{bmatrix}, \ \Delta = \mathrm{diag}(\delta_1, \delta_2), \ \delta_i \in \mathbb{C},$$

where $\overline{\sigma}(M_{11}) \approx 100$. Hence, from the small gain test, one may follow, that $(I - M_{11}\Delta)^{-1}$ is well-defined for $\|\Delta\|_\infty = \max_{i=1,2}(|\delta_i|) < 0.01$. The conservativeness of this result can be seen by calculating the lower and upper bounds for $\mu$ and the optimal $D$-scaling using the Robust Control Toolbox of MATLAB [11]. Both the lower and the upper bound are 0.655, thus $\mu$ has the same value. The optimal scaling is given by $D = \mathrm{diag}(1, 316.23)$ yielding

$$\begin{aligned} \mu_{\mathbf{\Delta}}(M_{11}) = \overline{\sigma}(DM_{11}D^{-1}) &= \overline{\sigma}\left(\begin{bmatrix} 1 & 0 \\ 0 & 316.23 \end{bmatrix}\begin{bmatrix} 0.5 & 100 \\ 0.001 & 0.1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 316.23 \end{bmatrix}^{-1}\right) \\ &= \overline{\sigma}\left(\begin{bmatrix} 0.5 & 0.3162 \\ 0.3162 & 0.01 \end{bmatrix}\right) = 0.655. \end{aligned}$$

With $\mu_{\mathbf{\Delta}}(M_{11}) = 0.655$, it follows that the inverse $(I - M_{11}\Delta)^{-1}$ is guaranteed to be well-defined for $|\delta_i| < 1/0.655 = 1.527$, $i = 1, 2$. Compared to the conservative result $|\delta_i| < 0.01$, $i = 1, 2$ from the small gain test, this clearly shows the importance of considering structure of $\Delta$ in the analysis.

For a refinement of the lower bound, consider the set $\mathcal{U} \subset \mathbb{C}^{n\times n}$ of scaling matrices with

$$\mathcal{U} = \{U \in \mathbf{\Delta} : UU^* = I_n\}$$

where $\overline{\sigma}(U\Delta) = \overline{\sigma}(\Delta U) = \overline{\sigma}(\Delta)$ and $\mu_{\mathbf{\Delta}}(M_{11}U) = \mu_{\mathbf{\Delta}}(UM_{11}) = \mu_{\mathbf{\Delta}}(M_{11})$. Then a lower bound is given by

$$\max_{U\in\mathcal{U}} \rho(UM_{11}) \leq \mu_{\mathbf{\Delta}}(M_{11}). \tag{A.5}$$

In [34] it was shown that (A.5) is even an equality, however the optimization problem posed by the lower bound has multiple local maxima and it is generally hard to find the global one. The power algorithm [66] implemented in the Robust Control Toolbox of MATLAB yields good results in many practical examples but is not guaranteed to converge to $\mu_{\mathbf{\Delta}}(M_{11})$.

**Remark A.1.** In this section, the matrix $\Delta$ was considered to be complex. In the case of mixed real/complex uncertainties a less conservative upper bound can be obtained by using so called D-G scalings. For more details the reader is referred to [11].

### A.3.3 Robust stability

In the previous sections $\mu$ was introduced to asses well-definedness of LFRs consisting of constant matrices. For uncertain dynamical systems one has $M = M(j\omega)$ and the following theorem is a generalization of the Small Gain Theorem for systems with structured uncertainties.

**Theorem A.2.** Suppose $M(s) \in \mathcal{R}H_\infty$ and $\Delta \in \mathbf{B\Delta}$. Then the interconnected system shown in figure A.2 is well-posed and internally stable if and only if

$$\sup_{\omega \in \mathbb{R}} \mu_{\mathbf{B\Delta}}(M_{11}(j\omega)) < 1.$$

Hence the frequency domain $\mu$ test requires to calculate the described lower and upper bounds at each frequency $\omega$. In practice a limited number of grid points along the frequency axis is chosen and the bounds are calculated at these points. However, the grid must be dense enough in order not to miss thin peaks in $\mu_{\mathbf{B\Delta}}(M_{11}(j\omega))$ sometimes caused by real uncertainties ($\delta_i \in \mathbb{R}$) [71].

**Remark A.2.** In general, a test for robust stability of a parametric dynamical system, requires to calculate the eigenvalues of the system for all possible parameter combinations. Therefore, very dense gridding or extensive Monte-Carlo tests must be performed for each parameter, which may take a very long time. Furthermore, there would be no guarantee that a worst-case (concerning stability) can be found due to the non-convexity of the problem. A $\mu$-upper bound less than one may guarantee robust stability a gridding is performed only with respect to one variable, the frequency $\omega$.

There exists an alternative robust stability test for LFRs based on the Popov criterion [39, 50], which even avoids the hazards of a frequency sweep. This method is also applicable is cases where the $\delta_i$ are sector-bounded nonlinearities. In case of linear uncertainties, this test may yield more conservative results than the $\mu$-test.

### A.3.4 Robust performance

In $H_\infty$ control, performance is usually characterized in terms of the $H_\infty$ norm of the transfer function from disturbance $v \in \mathbb{R}^q$ to the error $e \in \mathbb{R}^p$, that is $\|\mathcal{F}_u(M, \Delta)\|_\infty$ (see (A.3)). Without loss of generality it is assumed that $\|\mathcal{F}_u(M, \Delta)\|_\infty < 1$ means that all performance requirements are fulfilled. This can always be achieved by multiplying the input $v$ and the output $e$ with appropriate weighting/scaling matrices.

Nominal performance (NS) ($\Delta = 0$) is achieved if $\|M_{22}(s)\|_\infty < 1$, which shows that robust stability ($\mu$-test for $M_{11}(s)$) and nominal performance are tested with different submatrices of $M$.

The main objective in control design will be to achieve the required performance also in the presence of uncertainty, that is, to obtain robust performance. The robust performance test can be reformulated as a robust stability test by associating a complex unstructured uncertainty $\Delta_p \in \mathbb{C}^{q \times p}$ with the performance norm (see figure A.4), that is, $v = \Delta_p e$.

With the following definition of the augmented block structure

$$\mathbf{\Delta_p} := \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_p \end{bmatrix} : \Delta \in \mathbf{B\Delta}, \Delta_p \in \mathbb{C}^{q \times p} \right\},$$

the robust performance theorem can be formally stated:

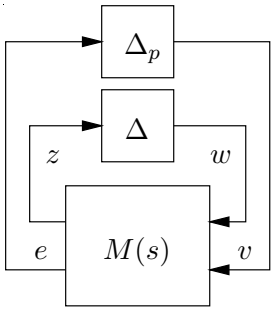


Figure A.4: Interconnection structure for robust performance test

**Theorem A.3.** Suppose $M(s) \in \mathcal{R}H_\infty$ and $\Delta \in \mathbf{B\Delta}$. Then $\mathcal{F}_u(M, \Delta)$ is well-posed, internally stable and $\|\mathcal{F}_u(M, \Delta)\|_\infty < 1$ if and only if

$$\sup_{\omega \in \mathbb{R}} \mu_{\mathbf{\Delta_p}}(M(j\omega)) < 1.$$

## A.4 $\mu$-Synthesis

In analogy to the robust stability tests using the Small Gain Theorem or the $\mu$-analysis, the $\mu$-synthesis can be seen as an attempt to generalize the $H_\infty$ control design methodology to systems with structured uncertainty.

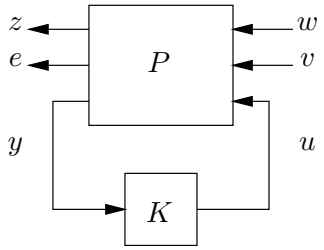


Figure A.5: Interconnection structure for controller synthesis

The robust $H_\infty$ synthesis problem involves finding a controller $K$ satisfying the performance requirements for the uncertain system. This is formulated as an optimization problem: find a stabilizing controller $K$, which minimizes the $H_\infty$ norm of the transfer function from the input vector $\overline{w} = \begin{bmatrix} w^T & v^T \end{bmatrix}^T$ to the output vector $\overline{z} = \begin{bmatrix} z^T & e^T \end{bmatrix}^T$, that is

$$\min_K \|\mathcal{F}_l(P(s), K(s))\|_\infty, \tag{A.6}$$

with $\mathcal{F}_l(P(s), K(s))$ as in (A.2). However, the formulation in (A.6) may be overly conservative, as already in the case where $\Delta$ consists of a single unstructured uncertainty,

that is, $\Delta \in \mathbb{C}^{n\times n}$, the $H_\infty$ optimization does not consider the block diagonal structure of the set $\mathbf{\Delta_p}$.

To address this limitation, the $\mu$-synthesis approach combines the $H_\infty$ optimization with the upper bound calculation of the $\mu$-analysis by considering the optimization problem

$$\min_{K} \inf_{D,D^{-1}\in H_\infty} \|D\mathcal{F}_l(P(s),K(s))D^{-1}\|_\infty. \tag{A.7}$$

As this two-parameter optimization problem is in general not solvable by convex, finite dimensional methods, the idea of $\mu$-synthesis is to split the problem into two simpler problems, which can be solved. This is done by iteratively solving for $D$ and $K$, the so-called $D$-$K$ iteration. When $D$ is fixed, $K$ is determined by standard $H_\infty$ control design. For a fixed stabilizing controller $K$, pointwise, constant solutions $D_\omega$ for $D$ can be obtained for a frequency grid by convex optimization (upper bound calculation in $\mu$-analysis) and the frequency varying $D$ is generated by calculating a single real, rational, proper, minimum phase approximation for all the $D_\omega$.

Given plant model $P(s)$ (see (A.1)) with related uncertainty matrix $\Delta \in \mathbf{B\Delta}$, the $\mu$-synthesis procedure can be summarized as follows:

1. **Initialization:** Set the counter $k = 1$, $m_0 = \infty$ and initialize the scaling, e.g., $D_k(s) = I$.

2. **$\mathbf{H_\infty}$-synthesis:** For output $\overline{z}$ and input $\overline{w}$, find a stabilizing controller $K_k(s)$, which minimizes $\|D_k(s)\mathcal{F}_l(P(s),K_k(s))D_k^{-1}(s)\|_\infty$.

3. **$\mu$-analysis:** Calculate the upper bound for $\mu_{\mathbf{\Delta_p}}(M(j\omega))$ pointwise for a predefined, "sufficiently dense" frequency grid, yielding a constant $D_{\omega,k+1}$ for each frequency point. If the maximum $m_k$ of these upper bounds across frequency is less than 1, than the robustness and performance requirements are satisfied and the final controller is given by $K_k$. Furthermore, if $m_k$ is equal or greater than $m_{k-1}$, then stop the iteration and the final controller is given by $K_k$ or $K_{k-1}$, respectively.

4. **Generate $\mathbf{D_{k+1}}$:** Generate a real, rational, proper, minimum-phase, transfer matrix $D_{k+1}(s)$, which approximates the set of optimal, constant $D_{\omega,k+1}$ at each frequency point. In general the approximation may result in a high order $D_{k+1}(s)$, which in turn may increase the order of the controller $K$ in step 2, as twice the number of states of the $D$-scale is added to the plant model (multiplication of the plant with $D_k(s)$ and $D_k^{-1}(s)$) and the resulting $H_\infty$-controller is usually of the same order as the scaled plant. Therefore, in practice the order of the $D$-scale is a-priori limited to a reasonable size.

5. Increment $k$ and go to step 2.

**Remark A.3.** Although the optimization problems to find $D$ and $K$ individually are convex, the problem is not jointly convex in both $D$ and $K$, and therefore the iteration method may converge towards a local minimum instead of the global minimum. One way to possibly avoid this problem is to repeat the iteration process for different initial estimates of $D$. Furthermore, the low order, approximate solution for $D$ may introduce

some errors and the results are restricted to the frequency range under consideration. However, in many practical examples the $\mu$-synthesis was applied with great success, and the resulting controller may be close to the global optimum. For structured uncertainty, the $\mu$-synthesis allows to reduce conservativeness compared to the standard $H_\infty$ synthesis, which may be interpreted as an upper bound for the $\mu$-synthesis.

# B Background on LPV control design

In this appendix some basic results for a quadratic Lyapunov approach to guarantee closed-loop stability and performance for parameter dependent systems are briefly summarized, which may help to understand the idea of the LPV control design procedure used in Chapter 6. For a detailed discussion see [87].

## B.1 Stability and performance of LPV systems

Consider the linear time-invariant system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx. \end{aligned} \tag{B.1}$$

**Theorem B.1.** [19](Exponential stability) System (B.1) is exponentially stable if and only if there exists a matrix $X = X^T > 0$ such that

$$A^T X + XA < 0.$$

**Theorem B.2.** [19](Bounded Real Lemma) System (B.1) is exponentially stable and

$$\|C(sI - A)^{-1}B\|_\infty < 1$$

if and only if there exists a matrix $X = X^T > 0$ such that

$$A^T X + XA + XBB^T X + C^T C < 0. \tag{B.2}$$

Using the Schur complement formula [19], the inequality (B.2) can also be written as

$$\begin{bmatrix} A^T X + XA & XB & C^T \\ B^T X & -I & 0 \\ C & 0 & -I \end{bmatrix} < 0,$$

which is denoted as a linear matrix inequality (LMI) in the variable $X$ and represents a convex constraint on $X$. The feasibility of this LMI and a solution for $X$ can be determined by solving a convex optimization problem. The same results can be applied for parameter-dependent systems.

**Lemma B.1.** [87] Consider parameter dependent state-space matrices $A(\delta)$, $B(\delta)$, $C(\delta)$ and suppose there exists a matrix $X = X^T$ such that

$$\begin{bmatrix} A^T(\delta)X + XA(\delta) & XB(\delta) & C^T(\delta) \\ B^T(\delta)X & -I & 0 \\ C(\delta) & 0 & -I \end{bmatrix} < 0,$$

for all $\delta \in \Pi$. Then for every parameter trajectory $\delta(t) \in \Pi$, the system is exponentially stable and $\|C(\delta)(sI - A(\delta))^{-1}B(\delta)\|_\infty < 1$.

The preceding lemma may be used for analysis of closed-loop parameter dependent systems and it is also a starting point for the synthesis of parameter dependent controllers for systems without rate bounded parametric variations. For the single-track model considered in chapter 6, the parameter $v$ does not change arbitrarily fast and therefore a controller derived assuming a plant without rate bounded parametric variations may be overly conservative. In [87] a generalization of the preceding lemma is presented, which allows to exploit a priori known bounds on the parameter variation rate:

**Lemma B.2.** [87] Suppose that $v > 0$. If there exists a continuously differentiable function $X(\delta)$ with $X(\delta) > 0$ and the two inequalities

$$\begin{bmatrix} v\frac{\mathrm{d}X(\delta)}{\mathrm{d}\delta} + A^T(\delta)X(\delta) + X(\delta)A(\delta) & X(\delta)B(\delta) & C^T(\delta) \\ B^T(\delta)X(\delta) & -I & 0 \\ C(\delta) & 0 & -I \end{bmatrix} < 0 \tag{B.3}$$

$$\begin{bmatrix} -v\frac{\mathrm{d}X(\delta)}{\mathrm{d}\delta} + A^T(\delta)X(\delta) + X(\delta)A(\delta) & X(\delta)B(\delta) & C^T(\delta) \\ B^T(\delta)X(\delta) & -I & 0 \\ C(\delta) & 0 & -I \end{bmatrix} < 0 \tag{B.4}$$

are fulfilled for all $\delta \in \Pi$, then the system is exponentially stable for any trajectory $\delta(t) \in \Pi$, $|\dot{\delta}(t)| \leq v$ and $\|C(\delta)(sI - A(\delta))^{-1}B(\delta)\|_\infty < 1$.

Note, that now $X$ is itself a matrix function of $\delta$, i.e. an unknown element of a function space, so that the inequalities (B.3) and (B.4) are infinite-dimensional LMIs. One way to approximately solve this convex optimization problem is to grid the plant model over the varying parameter space (see [88]). Furthermore, the structure of $X(\delta)$ must be fixed. This can be done by choosing suitable continuously differentiable basis functions $f_i(\delta)$, such that $X(\delta) = \sum_{i=1}^{N} f_i(\delta)X_i$, where the $f_i(\delta)$ may be chosen based on the parameter dependence of the plant itself.

## B.2 Robust LPV control design

Based on Lemma B.2 a procedure is proposed in [87] how to derive a parameter dependent gain-scheduled controller $K(s, \delta)$. This requires that all parameters $\delta$ can be measured and used for control. In general the parameter vector $\delta$ may be represented as $\delta = (\delta_m, \delta_u)$, where $\delta_m$ is a vector of measurable parameters and $\delta_u$ is a vector of uncertain parameters. Hence, only the parameters included in $\delta_m$ can be used for gain-scheduling, i.e., $K = K(s, \delta_m)$. To achieve robustness with respect to the uncertain parameter vector $\delta_u$, a $D$-$K$ iteration can be performed, which is sketched in the following procedure for a given plant model $G(s, \delta) \in \mathbb{C}^{p\times m}$.

1. Generate an LFR $(M(\delta_m), \Delta(s, \delta_u))$ such that $G(s, \delta) = \mathcal{F}_u(M(\delta_m), \Delta(s, \delta_u))$, with

$$\Delta(s, \delta_u) = \begin{bmatrix} I_n/s & 0 \\ 0 & \Delta_u(\delta_u) \end{bmatrix}, \ \Delta_u(\delta_u) \in \mathbb{R}^{r\times r},$$

   and initialize the scaling matrix, e.g. $D(s) = I_r$.

2. Partition $M(\delta_m)$ as

$$M(\delta_m) = \left[\begin{array}{c|c} M_{11}(\delta_m) & M_{12}(\delta_m) \\ \hline M_{21}(\delta_m) & M_{22}(\delta_m) \end{array}\right] = \left[\begin{array}{cc|c} A(\delta_m) & B_1(\delta_m) & B_2(\delta_m) \\ C_1(\delta_m) & D_{11}(\delta_m) & D_{12}(\delta_m) \\ \hline C_2(\delta_m) & D_{21}(\delta_m) & D_{22}(\delta_m) \end{array}\right],$$

such that the equation based representation of the LFR $(M(\delta_m), \Delta(s, \delta_u))$ is given as

$$\begin{aligned} \dot{x} &= A(\delta_m)x + B_1(\delta_m)w + B_2(\delta_m)u \\ z &= C_1(\delta_m)x + D_{11}(\delta_m)w + D_{12}(\delta_m)u \\ y &= C_2(\delta_m)x + D_{21}(\delta_m)w + D_{22}(\delta_m)u \\ x &= I_n/s\dot{x} \\ w &= \Delta_u z. \end{aligned}$$

3. Apply the methods proposed in [87] to synthesize a gain-scheduled controller $K(s, \delta_m) \in \mathbb{C}^{k\times l}$, which solves the optimization problem

$$\min_{K(s,\delta_m)} \left\| \left[\begin{array}{cc} D(s) & 0 \\ 0 & I_{p-l} \end{array}\right] \mathcal{F}_l\left(\overline{G}(s, \delta_m), K(s, \delta_m)\right) \left[\begin{array}{cc} D^{-1}(s) & 0 \\ 0 & I_{m-k} \end{array}\right] \right\|_\infty,$$

with

$$\overline{G}(s, \delta_m) = \mathcal{F}_u\left(\left[\begin{array}{c|c} A(\delta_m) & B(\delta_m) \\ \hline C(\delta_m) & D(\delta_m) \end{array}\right], I_n/s\right),$$

$$B(\delta_m) = \left[\begin{array}{cc} B_1(\delta_m) & B_2(\delta_m) \end{array}\right],$$

$$C(\delta_m) = \left[\begin{array}{c} C_1(\delta_m) \\ C_2(\delta_m) \end{array}\right],\ D(\delta_m) = \left[\begin{array}{cc} D_{11}(\delta_m) & D_{12}(\delta_m) \\ D_{21}(\delta_m) & D_{22}(\delta_m) \end{array}\right].$$

4. Generate a stable, real, rational, proper, minimum-phase transfer matrix $D(s) \in \mathbb{C}^{r\times r}$, with $D(s)\Delta_u = \Delta_u D(s)$, which solves the optimization problem

$$\gamma = \inf_{D(s)} \left\| \left[\begin{array}{cc} D(s) & 0 \\ 0 & I_{p-l} \end{array}\right] \mathcal{F}_l\left(\overline{G}(s, \delta_m), K(s, \delta_m)\right) \left[\begin{array}{cc} D^{-1}(s) & 0 \\ 0 & I_{m-k} \end{array}\right] \right\|_\infty.$$

Stop the procedure if $\gamma < 1$, otherwise go to step 3.

# C LFR-toolbox

In this appendix a list of the basic functions of the LFR-toolbox version 2 are presented. Additional modules for Simulink interfaces, gain-scheduling design and interfaces to an ONERA-toolbox for skew-$\mu$ analysis are not yet finished and listed.

**Overloaded functions for LFR objects:**

| | | |
|---|---|---|
| display | - | show contents of lfr-objects |
| size | - | show size information of lfr-objects |
| isempty | - | check if lfr-object is empty |
| get | - | get fields of an lfr-object |
| set | - | change block-names, block-bounds or object-fields |
| plus | - | addition of lfr-objects |
| minus | - | subtraction of lfr-objects |
| uminus | - | sign change of lfr-object |
| mtimes | - | product of lfr-objects |
| mpower | - | repeated product of lfr-objects |
| mrdivide | - | division of lfr-objects |
| inv | - | inversion of lfr-objects |
| horzcat | - | horizontal (column) concatenation of lfr-objects |
| vertcat | - | vertical (row) concatenation of lfr-objects |
| append | - | block-diagonal concatenation |
| blkdiag | - | block-diagonal concatenation |
| transp | - | transposition of lfr-objects |
| subsref | - | subscripted reference for lfr-objects |
| subsasgn | - | subscripted assignment for lfr-objects |
| feedback | - | feedback interconnection of lfr-objects |
| dcgain | - | steady-state gain |
| eval | - | evaluation from values in workspace |
| diff | - | differentiation of lfr-objects |
| eig | - | nominal eigenvalues of dynamic lfr-object |
| null | - | null-space of lfr-object |
| real | - | real part of lfr-object |
| imag | - | imaginary part of lfr-object |

**LFR object generation:**

| | | |
|---|---|---|
| lfr | - | core function for creation of lfr-object |
| lfrs | - | short-cut for realization of elementary real or complex lfr-objects |
| rlfr | - | generate random lfr-objects |
| bnds2lfr | - | generate lfr-object from min/max-bounds of a matrix |
| sym2lfr | - | generate lfr-object from symbolic expression |
| data2lfr | - | generate lfr-object from least mean-squares interpolation |
| gmorton | - | generalized Morton realization |

**Conversions:**

| | | |
|---|---|---|
| lfr | - | conversion of various MATLAB objects to lfr-objects |
| abcd2lfr | - | conversion from state-space to input-output form |
| lfr2abcd | - | conversion from input-output to state-space form |
| lf2lfr | - | realize lfr-object from left fractional factorization |
| rf2lfr | - | realize lfr-object from right fractional factorization |
| lfr2rob | - | convert lfr-object to related Robust Control toolbox object |

**Symbolic preprocessing:**

| | | |
|---|---|---|
| sym2lfr | - | preprocessing and lfr-object realization of symbolic expressions |
| symtreed | - | standard tree decomposition |
| etd | - | enhanced tree decomposition |
| vs_etd | - | VS factorization followed by a call of etd for each factor |
| loadmprocs | - | load Maple routines (necessary for etd) |

**Numerical order reduction:**

| | | |
|---|---|---|
| minlfr1 | - | repeated 1-d order reduction |
| minlfr | - | order reduction based on n-d Kalman decomposition |
| redlfr1 | - | repeated 1-d approximation |
| balsys | - | mex-function for 1-d balancing of lfr-objects |
| ssminr | - | mex-function for 1-d minimal realization |
| sscof | - | mex-function for calculation of controllability staircase form |
| sysred | - | mex-function for 1-d approximation |

**Manipulation of uncertainty block:**

| | | |
|---|---|---|
| uplft | - | closes (partially) the M-$\Delta$ loop |
| normalizelfr | - | normalize parametric uncertainty blocks |
| starplfr | - | star product |

**Distances:**

| | | |
|---|---|---|
| distlfr | - | calculate a lower bound for the distance between two lfr-objects |
| udistlfr | - | calculate an upper bound for the distance between two lfr-objects |

**$\mu$-analysis:**

| | | |
|---|---|---|
| lfr2mustab | - | from lfr-object to input arguments for mustab |
| lfr2mubnd | - | from lfr-object to input arguments for mubnd |
| lfr2mu | - | from lfr-object to input arguments for mu |
| lfr2mussv | - | from lfr-object to input arguments for mussv |
| ns_rad | - | calculate non-singularity radius |
| wp_rad | - | calculate well-posedness radius |
| min_max | - | calculate min/max-values of a 1 by 1 real lfr-object |

**Miscellaneous:**

| | | |
|---|---|---|
| aeblkchk | - | check regularity of lfr-object |
| remalgequ | - | remove non-dynamic modes of constant $\Delta$-block |
| lfrdata | - | get all data of an lfr-object |
| plotlfr | - | plot gridding for SISO-entries of an lfr-object |
| upper_lft_sym | - | calculate upper-lft symbolically |
| lfrtol | - | set global tolerances for numerical order reduction |

For some of the functions the Control toolbox (mainly for numerical order reduction), the Symbolic toolbox (symbolic preprocessing), the Extended Symbolic toolbox (for enhanced tree decomposition), the $\mu$-analysis and synthesis toolbox and the LMI toolbox (both for $\mu$-analysis) are required.

# D RCAM linear parametric system matrices

$$A(\delta) = \begin{bmatrix}
-117.05\,\frac{1}{C_w\ V_A} & 0 & 50.807\,\frac{1}{C_w\ V_A} & 0 & 0 & 0 \\
0 & \frac{0.70528\ Z_{cg}-96.507+24.879\ X_{cg}}{C_w\ V_A} & 0 & 0 & 0 & 0 \\
4.8192\,\frac{1}{C_w\ V_A} & 0 & -48.116\,\frac{1}{C_w\ V_A} & 0 & 0 & 0 \\
1.0 & 0 & \alpha & 0 & 0 & 0 \\
0 & 1.0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.0004 & 0 & 0 & 0 \\
0 & \frac{-1.9860\ \tilde{b}_{72}-1.0\ {V_A}^2\alpha\ C_w}{C_w\ V_A} & 0 & 0 & -9.8061 & 0 \\
V_A\,\alpha & 0 & -V_A & 9.8061 & 0 & 0 \\
0 & \frac{-241.25+0.0040000\ C_w\ V_A+{V_A}^2\ C_w}{C_w\ V_A} & 0 & 0 & -9.8100\,\alpha & 0 \\
0 & 0 & 0 & 0 & 0.000043244 & 0 \\
0 & 0 & 0 & -V_A\,\alpha & 0 & V_A \\
0 & 0 & 0 & 0 & -V_A & 0
\end{bmatrix}$$

$$\begin{bmatrix}
0 & \frac{-2.2278-0.054189\ X_{cg}+2.5880\ Z_{cg}}{C_w\ V_A} & 0 & 0 & 0 & 0 \\
0.061601\,\frac{\tilde{a}_{27}}{C_w\ V_A} & 0 & 0.061601\,\frac{\tilde{a}_{29}}{C_w\ V_A} & 0 & 0 & 0 \\
0 & 0.061601\,\frac{\tilde{a}_{38}}{C_w\ V_A} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-0.061601\,\frac{\tilde{a}_{77}}{C_w\ V_A} & 0 & -0.061601\,\frac{\tilde{a}_{79}}{C_w\ V_A} & 0 & 0 & 0 \\
0 & -15.697\,\frac{1}{C_w\ V_A} & 0 & 0 & 0 & 0 \\
-0.061601\,\frac{\tilde{a}_{97}}{C_w\ V_A} & 0 & -0.061601\,\frac{\tilde{a}_{99}}{C_w\ V_A} & 0 & 0 & 0 \\
1 & 0 & \alpha & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
-\alpha & 0 & 1 & 0 & 0 & 0
\end{bmatrix}$$

$B(\delta) =$

$$\begin{bmatrix}
\frac{-0.97053}{C_w} & 0 & \frac{0.33355+0.00813\,X_{cg}-0.38821\,Z_{cg}}{C_w} & 301.18\,\frac{1}{C_w\,{V_A}^2} & -301.18\,\frac{1}{C_w\,{V_A}^2} \\
0 & \frac{0.0219\,Z_{cg}-2.9935+0.7717\,X_{cg}}{C_w} & 0 & \frac{2152.8+7478.4\,Z_{cg}}{C_w\,{V_A}^2} & \frac{2152.8+7478.4\,Z_{cg}}{C_w\,{V_A}^2} \\
\frac{-0.02032}{C_w} & 0 & \frac{-0.41990+0.15568\,X_{cg}-0.00813\,Z_{cg}}{C_w} & 5768.4\,\frac{1}{C_w\,{V_A}^2} & -5768.4\,\frac{1}{C_w\,{V_A}^2} \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & -0.00077\,\frac{\tilde{b}_{72}\,V_A}{C_w} & 0 & \frac{72517}{C_w\,{V_A}^2} & \frac{72517}{C_w\,{V_A}^2} \\
0 & 0 & 2.3545\,\frac{1}{C_w} & 0 & 0 \\
0 & -7.48317\,\frac{1}{C_w} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

with $C_w = \frac{mg}{\frac{1}{2}\rho V_A^2 S}$ and

$$\begin{aligned}
\tilde{a}_{27} &= 2.1451\,X_{cg}\,{C_w}^2 Z_{cg} + 0.058556\,X_{cg}\,C_w\,Z_{cg} - 20.291\,X_{cg}\,C_w + 1.1425\,X_{cg}\,{C_w}^2 \\
&\quad -0.90635\,{C_w}^2 - 9.5334 + 9.2389\,C_w + 18.030\,X_{cg} - 5.7399\,Z_{cg} - 5.6075\,{C_w}^2 Z_{cg} \\
&\quad -0.97164\,X_{cg}\,Z_{cg} + 5.7418\,C_w\,Z_{cg} \\
\tilde{a}_{29} &= 1.6726\,X_{cg}\,{C_w}^2 Z_{cg} - 0.17230\,{X_{cg}}^2 C_w - 3.9324\,X_{cg}\,C_w\,Z_{cg} - 0.28903\,{X_{cg}}^2 {C_w}^2 Z_{cg} - 46.850 \\
&\quad -0.070972\,{X_{cg}}^2 Z_{cg} + 0.29652\,{X_{cg}}^2 C_w\,Z_{cg} + 4.9667\,X_{cg}\,C_w - 2.7036\,X_{cg}\,{C_w}^2 + 0.58292\,{C_w}^2 \\
&\quad -0.25564\,{X_{cg}}^2 - 1.3439\,C_w + 100.13\,X_{cg} - 14.251\,Z_{cg} - 1.9116\,{C_w}^2 Z_{cg} + 1.1243\,X_{cg}\,Z_{cg} \\
&\quad +24.656\,C_w\,Z_{cg} + 0.45703\,{X_{cg}}^2 {C_w}^2 \\
\tilde{a}_{38} &= 0.096425\,{X_{cg}}^2 C_w - 0.086069\,{X_{cg}}^2 + 1.6082\,X_{cg}\,C_w - 16.591\,X_{cg} - 7.0577\,C_w + 18.418 \\
\tilde{a}_{77} &= 1.5667\,{C_w}^2 - 16.241\,C_w + 65.449 \\
\tilde{a}_{79} &= -201.39\,C_w + 121.84 \\
a_{97} &= 144.91\,C_w + 171.66 \\
\tilde{a}_{99} &= 24.355\,{C_w}^2 + 6.0937\,C_w + 962.75 \\
\alpha &= -0.041088\,X_{cg}\,C_w - 0.0053886\,X_{cg} + 0.17559\,C_w - 0.16287 \\
\tilde{b}_{72} &= 4.9092\,X_{cg}\,C_w + 0.73956\,X_{cg} - 21.270\,C_w + 19.721
\end{aligned}$$

$$
C(\delta) = \begin{bmatrix}
0 & 1.0 & 0 & 0 & 0 & 0 \\
0 & -0.20245 \frac{\tilde{b}_{72}}{C_w\ V_A} & 0 & 0 & 0.00040155 & 0 \\
0 & \frac{-24.593+0.00040000\ C_w\ V_A}{C_w\ V_A} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -V_A & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
1.0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.000043244 & 0 \\
0 & 0 & 0 & -V_A\,\alpha & 0 & V_A \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\alpha & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
-0.0062794 \frac{\tilde{a}_{77}}{C_w\ V_A} & 0 & -0.0062794 \frac{\tilde{a}_{79}}{C_w\ V_A} & 0 & 0 & 0 \\
-0.0062794 \frac{\tilde{a}_{97}}{C_w\ V_A} & 0 & -0.0062794 \frac{\tilde{a}_{99}}{C_w\ V_A} & 0 & 0 & 0 \\
-\alpha & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.0 \\
1 & 0 & \alpha & 0 & 0 & 0 \\
1 & 0 & \alpha & 0 & 0 & 0 \\
0 & V_A^{-1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & \alpha & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.0 & 0 \\
0 & V_A^{-1} & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$$D(\delta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -0.0000785\frac{V_A \tilde{b}_{72}}{C_w} & 0 & \frac{7392.15}{C_w\, V_A^2} & \frac{7392.15}{C_w\, {V_A}^2} \\ 0 & -\frac{0.76281}{C_w} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Bibliography


[1] J. Ackermann, P. Blue, T. Bünte, L. Güvenc, D. Kaesbauer, M. Kordt, M. Muhler, and D. Odenthal, *Robust Control. The Parameter Space Approach*, 2nd ed., ser. Communications and Control Engineering. Springer, October 2002.

[2] J. Ackermann, T. Bünte, W. Sienel, H. Jeebe, and K. Naab, "Driving safety by robust steering control," in *Proc. of the International Symposium on Advanced Vehicle Control and Safety*, Aachen, Germany, 1996.

[3] B. Aksun-Güvenç, T. Bünte, D. Odenthal, and L. Güvenç, "Robust two degree-of-freedom vehicle steering controller design," *IEEE Tranactions on Control Systems Technology*, vol. 12, no. 4, pp. 627–636, 2004.

[4] B. Anderson and Y. Liu, "Controller reduction: Concepts and approaches," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 802–812, 1989.

[5] E. Anderson, Z. Bai, J. Bishop, J. Demmel, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK User's Guide, Second Edition*, SIAM, Philadelphia, France, 1995.

[6] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, ser. Advanced in Design and Control, R. C. Smith, Ed. SIAM, 2005.

[7] P. Apkarian and P. Gahinet, "A convex characterization of gain-scheduled $H_\infty$ controllers," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 853–864, 1995.

[8] K. Åström, P. Albertos, M. Blanke, A. Isidori, W. Schaufelberger, and R. Sanz, *Control of Complex Systems*. London: Springer, 2001.

[9] G. Balas, "Robust control of flexible structures: Theory and experiments," Ph.D. dissertation, California Institute of Technology, 1990.

[10] G. Balas, "Linear, parameter-varying control and its application to a turbofan engine," *Int. Journal of Robust and Nonlinear Control*, vol. 12, no. 9, pp. 763–796, 2002.

[11] G. Balas, R. Chiang, A. Packard, and M. Safonov, *Robust Control Toolbox 3.0.1*, The MathWorks, Natick, MA, USA, March 2005.

[12] G. Balas, J. Doyle, K. Glover, A. Packard, and R. Smith, *$\mu$-Analysis and Synthesis Toolbox*, The MathWorks, Natick, MA, USA.

[13] C. Belcastro, "Parametric uncertainty modelling: an overview," in *Proc. American Control Conference, Philadelphia, PA*, 1998.

[14] C. M. Belcastro, "Uncertainty modeling of real parameter variations for robust control applications," Ph.D. dissertation, University of Drexel, US, December 1994.

[15] P. Benner, V. Mehrmann, V. Sima, S. V. Huffel, and A. Varga, *SLICOT - A Subroutine Library in Systems and Control Theory*, ser. In B. N. Datta, Ed., Applied and Computational Control, Signals and Circuits. Birkhäuser, 1998.

[16] H. W. Bode, "Relations between attenuation and phase in feedback amplifier design," *Bell System Technical Journal*, vol. 19, pp. 421–454, 1940.

[17] H. W. Bode, *Network Analysis and Feedback Amplifier Design.* New York: Van Nostrand, 1945.

[18] N. K. Bose, *Applied Multidimensional Systems Theory.* Van Nostrand Reinhold Company, 1982.

[19] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory.* Siam, 1994.

[20] T. Bünte, D. Odenthal, B. Aksun-Güvenç, and L. Güvenç, "Robust vehicle steering control design based on the disturbance observer," *IFAC Annual Reviews in Control*, vol. 26, pp. 139–149, 2002.

[21] Y. Cheng and B. De Moor, "A multidimensional realization algorithm for parametric uncertainty modeling problems and multiparameter margin problems," *International Journal of Control*, vol. 60, pp. 3022–3023, 1994.

[22] M. Chilali and P. Gahinet, "$H_\infty$ design with pole placement constraints: an LMI approach," in *Proc. of Conference on Decision and Control*, 1994, pp. 553–558.

[23] J. Cockburn, "Discussion on: "generalized lft-based representation of parametric uncertain models"," *European Journal of Control*, vol. 10, no. 4, pp. 338–340, 2004.

[24] J. Cockburn and B. Morton, "Linear fractional representation of uncertain systems," *Automatica*, vol. 33, no. 7, pp. 1263–1271, 1997.

[25] L. Dai, *Singular Control Systems*, ser. Lecture Notes in Control and Information Sciences. Heidelberg: Springer-Verlag, Berlin, 1989, no. 118.

[26] A. Damen and S. Weiland, "Lecture notes: Robust control," Measurement and Control Group, Eindhoven University of Technology, The Netherlands, 2002.

[27] R. D'Andrea, "$H_\infty$ optimal interconnections," *Systems and Control Letters*, vol. 32, pp. 313–322, 1997.

[28] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Trans. Automat. Contr.*, vol. 48, no. 9, 2003.

[29] R. D'Andrea and S. Khatri, "Kalman decomposition of linear fractional transformation representations and minimality," in *Proc. of the American Control Conference*, Albuquerque, New Mexico, 1997, pp. 3557–3561.

[30] R. D'Andrea and F. Paganini, "Interconnection of uncertain behavioral systems for robust control," in *Proc. of the 32nd Conference on Decision and Control*, San Antonio, Texas, 1993, pp. 3642–3647.

[31] M. Dettori and C. W. Scherer, "Control design for a compact disc player with multiple norm specifications," *Control Systems Technology*, vol. 10, pp. 635–645, 2002.

[32] C. Döll and A. Knauf, "Comparison in terms of size of several LFT models for the lateral motion of a generic fighter aircraft," in *Proc. AIAA Guidance Navigation and Control Conference*, 2005.

[33] J. C. Doyle, "Guaranteed margins for LQG regulators," *IEEE Trans. Automat. Contr.*, vol. 23, pp. 756–757, 1978.

[34] J. C. Doyle, "Analysis of feedback systems with structured uncertainties," *IEE-D*, vol. 129, pp. 242–250, 1982.

[35] J. C. Doyle, K. Glover, P. Kkhargonekar, and B. Francis, "State-space solutions to standard $H_2$ and $H_\infty$ control problems," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 831–847, 1989.

[36] G. Ferreres and J. Biannic, "A skew mu toolbox (SMT) for robustness analysis," in *Proc. International Symposium on Computer Aided Control Systems Design (CACSD)*, Taipei, Taiwan, 2004.

[37] P. Gahinet, A. Nemirovski, A. Laub, and M. Chilali, *LMI Control Toolbox*, The MathWorks, Natick, MA, USA.

[38] M. Green and D. Limebeer, *Linear Robust Control*, ser. Information and System Sciences, T. Kailath, Ed. Englewood Cliffs, New Jersey: Prentice Hall, 1995.

[39] W. Haddad and D. Berstein, "Parameter-dependent lyapunov functions, constant real parameter uncertainty, and the Popov citerion in robust analysis and synthesis," in *Proc. Conference on Decision and Control*, 1991, pp. 2274–2279 and 2632–2633.

[40] S. Hecker, "Robust $H_\infty$ based vehicle steering control design," in *Proc. Conference on Control Applications*, Munich, Germany, 2006.

[41] S. Hecker and A. Varga, "Generalized LFT-based representation of parametric uncertain models," in *Proc. European Control Conference ECC*, Cambridge, UK, 2003.

[42] S. Hecker and A. Varga, "Generalized LFT-based representation of parametric uncertain models," *European Journal of Control*, vol. 10, no. 4, pp. 326–337, 2004.

[43] S. Hecker and A. Varga, "Symbolic techniques for low order LFT-modelling," in *IFAC World Congress*, Prag, Czech Republic, 2005.

[44] S. Hecker and A. Varga, "Symbolic manipulation techniques for low order LFT-based parametric uncertainty modelling," *International Journal of Control*, vol. 79, no. 11, pp. 1485–1494, 2006.

[45] S. Hecker, A. Varga, and J. Magni, "Enhanced LFR-toolbox for Matlab," in *Proc. International Symposium on Computer Aided Control Systems Design (CACSD)*, Taipei, Taiwan, 2004.

[46] S. Hecker, A. Varga, and J. Magni, "Enhanced LFR-toolbox for Matlab and LFT-based gain scheduling," in *Proc. 6th ONERA-DLR Aerospace Symposium (ODAS)*, Berlin, Germany, 2004, pp. 80–89.

[47] S. Hecker, A. Varga, and J. Magni, "Enhanced LFR-Toolbox for Matlab," *Aerospace Science and Technology*, vol. 9, pp. 173–180, 2005.

[48] I. M. Horowitz, *Synthesis of Feedback Systems*. New York: Academic Press, 1963.

[49] I. M. Horowitz, "Quantitative feedback design theory (QFT)," in *QFT Publications*, Boulder, Colorado, 1993.

[50] J. How and S. Hall, "Connection between the Popov stability citerion and bounds for real parameter uncertainty," in *Proc. American Control Conference*, 1993, pp. 1084–1089.

[51] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Trans ASME (J. Basic Engineering)*, vol. 83, pp. 95–108, 1961.

[52] R. E. Kalman, Y. Ho, and K. S. Narendra, *Controllability of Linear Dynamical Systems*, ser. Contributions to Differential Equations. New York: John Wiley & Sons, Inc., 1963.

[53] D. E. Knuth, *The Art of Computer Programming - Seminumerical Algorithms*, 2nd ed., ser. Computer Science and Information Processing. Addison Wesley, 1981.

[54] P. Lambrechts and J. Terlouw, *A Matlab Toolbox for Parametric Uncertainty Modeling*, Philips Research Eindhoven and NLR Amsterdam, 1992.

[55] D. Leith and W. Leithead, "Survey of gain-scheduling analysis and design," *International Journal of Control*, vol. 73, no. 11, pp. 1001–1025, 2000.

[56] Y. Liu and B. D. O. Anderson, "Singular perturbation approximation of balanced systems," *International Journal of Control*, vol. 50, pp. 1379–1405, 1989.

[57] J. F. Magni, "Presentation of the linear fractional representation toolbox (LFRT)," in *Proc. CACSD'2002 Symposium, Glasgow, Scotland*, 2002.

[58] J. Magni, S. Bennani, and J. Terlouw, *Robust Flight Control - A Design Challenge*, ser. Lecture Notes in Control and Information Sciences. Springer, 1997, no. 224.

[59] A. Marcos and G. Balas, "Development of linear-parameter-varying models for aircraft," *Journal of Guidance, Control and Dynamics*, vol. 27, no. 3, 2004.

[60] A. Marcos, D. Bates, and I. Postlethwaite, "A multivariate polynomial matrix order-reduction algorithm for linear fractional transformation modelling," in *Proc. IFAC World Congress*, Prague, Czech Republic, 2005.

[61] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Trans. Automat. Contr.*, vol. 42, no. 6, pp. 819–830, 1997.

[62] K. Meyberg and P. Vachenauer, *Höhere Mathematik 1*, 6th ed. Berlin, Heidelberg: Springer, 2003.

[63] B. C. Moore, "Principal component analysis in linear system: Controllability, observability and model reduction," *IEEE Trans. Automat. Contr.*, vol. 26, pp. 17–32, 1981.

[64] B. Morton, "New applications of $\mu$ to real parameter variation problems," in *Proc. Conference on Decision and Control*, Fort Lauderdale, Florida, 1985, pp. 233–238.

[65] A. Packard, "Gain scheduling via linear fractional transformations," *System and Control Letters*, vol. 22, pp. 79–92, 1994.

[66] A. Packard, M. K. H. Fan, and J. C. Doyle, "A power method for the structured singular value," in *Proc. Conference on Decision and Control*, 1988.

[67] P. Pellanda, P. Apkarian, and H. D. Tuan, "Missile autopilot design via multi-channel LFT/LPV control method," *Int. Journal of Robust and Nonlinear Control*, vol. 12, pp. 1–20, 2002.

[68] W. Rugh and F. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, pp. 1401–1425, 2000.

[69] A. Saberi, B. M. Chen, and P. Sannuti, *Loop Transfer Recovery: Analysis and Design*. Springer, London, 1993.

[70] M. G. Safonov and M. Athans, "Gain and phase margins for for multiloop LQG regulators," *IEEE Trans. Automat. Contr.*, vol. 22, pp. 173–179, 1977.

[71] C. W. Scherer, "Lecture notes for course on theory of robust control," Dutch Institute of Systems and Control, Delft, The Netherlands, 2001.

[72] C. W. Scherer, "LPV control and full block multipliers," *Automatica*, vol. 37, pp. 361–375, 2001.

[73] C. W. Scherer and S. Weiland, "Lecture notes DISC course on linear matrix inequalities in control," Dutch Institute of Systems and Control, Delft, The Netherlands, 1999.

[74] G. Scorletti and L. E. Ghaoui, "Improved LMI conditions for gain scheduling and related control problems," *Internatinal Journal of Robust and Nonlinear Control*, vol. 8, pp. 845–877, 1998.

[75] J. Sturm, *Sedumi 1.05R5 a Matlab Toolbox for Optimization over Symmetric Cones*, Department of Econometrics, Tilburg University, The Netherlands, 1998-2003.

[76] T. Sugie and M. Kawanishi, "$\mu$-analysis/synthesis based on exact expression of physical parameter variations," in *Proc. European Control Conference*, Rome, Italy, 1995, pp. 159–164.

[77] J. Terlouw, P. Lambrechts, S. Bennani, and M. Steinbuch, "Parametric uncertainty modeling using LFTs," in *Proc. American Control Conference*, San Francisco, CA, 1993, pp. 267–272.

[78] M. van de Wal, G. van Baars, F. Sperling, and O. Bosgra, "Multivariable $H_\infty/\mu$ feedback control design for high-precision wafer stage motion," *Control Engineering Practice*, vol. 10, pp. 739–755, 2002.

[79] A. T. van Zanten, "Bosch ESP systems: 5 years of experience," 2000, SAE Paper no. 2000-01-1633.

[80] A. T. van Zanten, R. Erhardt, and G. Pfaff, "VDC, the vehicle dynamics control system of Bosch," 1995, SAE Paper no. 950-759.

[81] A. Varga, "Model reduction software in the SLICOT library," in *Applied and Computational Control, Signals and Circuits*, ser. The Kluwer International Series in Engineering and Computer Science, B. N. Datta, Ed. Kluwer Academic Publishers, Boston, 2001, vol. 629, pp. 239–282.

[82] A. Varga and B. D. O. Anderson, "Accuracy-enhancing methods for balancing-related frequency-weighted model and controller reduction," *Automatica*, vol. 39, pp. 919–927, 2003.

[83] A. Varga and G. Looye, "Symbolic and numerical software tools for LFT-based low order uncertainty modeling," in *Proc. CACSD'99 Symposium, Kohala Coast, Hawaii*, 1999.

[84] A. Varga, G. Looye, D. Moormann, and G. Grübel, "Automated generation of LFT-based parametric uncertainty descriptions from generic aircraft models," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 4, pp. 249–274, 1998.

[85] A. Varga, *Controller reduction using accuracy-enhancing methods.* Springer Verlag, 07 2005, vol. 45, pp. 225 – 260.

[86] J. C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Trans. Automat. Contr.*, vol. 36, no. 3, pp. 259–294, 1991.

[87] F. Wu, "Control of linear parameter varying systems," Ph.D. dissertation, University of California, Berkley, 1995.

[88] F. Wu, X. Yang, A. Packard, and G. Becker, "Induced $L_2$-norm control for LPV system with bounded parameter variation rates," in *Proc. American Control Conference*, 1995.

[89] G. Zames, "Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverse," *IEEE Trans. Automat. Contr.*, vol. 26, no. 2, pp. 301–320, 1981.

[90] E. Zerz, "Linear fractional representations of polynomially parametrized descriptor systems," in *Proc. 3rd Portuguese Conference on Automatic Control (Controlo 98)*, 1998.

[91] K. Zhou and J. C. Doyle, *Essentials of Robust Control*. Prentice Hall, 1998.

[92] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall, 1996.