Robust
Fault-Tolerant Control

Robust
Fault-Tolerant Control

Stoyan Kanev

Ph.D. Thesis • University of Twente • The Netherlands

Thesis committee:

Prof. dr. ir. A. Bliek (voorzitter)	University of Twente
Prof. dr. ir. M. Verhaegen (promotor)	University of Twente
Prof. Dr. R. Tempo	Politecnico di Torino
Prof. M. Kinnaert	Université Libre de Bruxelles
Prof. dr. A. Stoorvogel	Delft University of Technology
Prof. dr. ir. J. van Amerongen	University of Twente
Prof. dr. ir. B. Roffel	University of Twente

The work presented in this thesis has been sponsored by the Dutch Technology Foundation (STW) under project number DEL. 4506.

Publisher: FEBO-DRUK
Javastraat 123, 7512 ZE Enschede, the Netherlands
`www.febodruk.nl`

© 2004 by Stoyan Kanev
All rights reserved. Published 2004.

Cover design: V. Kaneva and S. Kanev

ISBN 90-9017903-8

ROBUST FAULT-TOLERANT CONTROL

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof.dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 12 maart 2004 om 13.15 uur

door

Stoyan Kamenov Kanev

geboren op 26 juni 1975

te Sofia

Dit proefschrift is goedgekeurd door de promotor Prof. dr. ir. M. Verhaegen

“Perhaps we’ll all find the answers
somewhere in time…”,

– *Savatage*, Somewhere in Time, 1991

Contents

Acknowledgements **xiii**

1 Introduction **1**
1.1 Why Fault-Tolerant Control? 1
1.2 Fault Classification 3
1.3 Modelling Faults 4
1.3.1 Multiplicative faults 5
1.3.2 Additive faults 7
1.3.3 Component faults 9
1.4 Main Components in a FTCS 9
1.5 The State-of-the-Art in FTC 12
1.5.1 Passive Methods for FTC 12
1.5.2 Active Methods for FTC 13
1.6 Scope of the thesis 21
1.7 Outline of the thesis 23
1.8 Organization of the thesis 30
1.9 Contributions 31

2 Probabilistic Approach to Passive State-Feedback FTC **33**
2.1 Introduction 34
2.2 Preliminaries 35
2.2.1 Notation and Problem Formulation 35
2.2.2 The Subgradient Iteration Algorithm 41
2.3 The Ellipsoid Algorithm: Feasibility 45
2.4 Finding an Initial Ellipsoid $E^{(0)}$ 49
2.4.1 Procedure for General LMI Problems 50
2.4.2 The Constrained Robust Least-Squares Problem 54
2.5 The Ellipsoid Algorithm: Optimization 58
2.6 Experimental part 59
2.6.1 Comparison with SIA 61
2.6.2 Passive FTC Design 62
2.7 Conclusions 62

3 BMI Approach to Passive Robust Output-Feedback FTC 65
3.1 Introduction 66
3.2 Preliminaries and Problem Formulation 69
    3.2.1 Notation 69
    3.2.2 Output-Feedback Passive FTC 70
    3.2.3 $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Norm Computation for Uncertain Systems . . 71
    3.2.4 Problem Formulation 74
3.3 Locally Optimal Robust Controller Design 75
3.4 Initial Robust Multiobjective Controller Design 79
    3.4.1 Step 1: Robust Multiobjective State-Feedback Design . . . 80
    3.4.2 Step 2: Robust Multiobjective Output-Feedback Design . . 81
3.5 Summary of the Approach 85
3.6 Illustrative Examples 86
    3.6.1 Locally Optimal Robust Multiobjective Controller Design . 86
    3.6.2 A Comparison Between Some Local BMI Approaches . . . 91
3.7 Conclusions 92

4 LPV Approach to Robust Active FTC 95
4.1 Introduction 96
4.2 Deterministic Method for Multiplicative Sensor and Actuator Faults 97
    4.2.1 Problem Formulation 97
    4.2.2 LPV Controllers Design 100
    4.2.3 Controller Reconfiguration Strategy 108
4.3 Probabilistic Method for Component Faults 110
    4.3.1 Problem Formulation 111
    4.3.2 State-Feedback Case 114
    4.3.3 Output-Feedback Case 115
    4.3.4 The Probabilistic Approach to the LPV Design 116
4.4 Illustrative Examples 118
    4.4.1 Deterministic Approach to LPV-based FTC 119
    4.4.2 Probabilistic Approach to LPV-based FTC 120
4.5 Conclusion 123

5 Robust Output-Feedback MPC Method for Active FTC 125
5.1 Introduction 126
5.2 Notation and Problem Formulation 127
5.3 Integrated State Prediction and Trajectory Tracking Control 129
    5.3.1 The Kalman filter over a finite-time horizon 129
    5.3.2 Combination of the Kalman filter and MPC 130
5.4 Computation of the covariance matrix $P_{k+1|k}$ 134
5.5 A Case Study 139
5.6 Conclusion 143

6 Brushless DC Motor Experimental Setup 145
6.1 Introduction 146
6.2 Model of a Brushless DC Motor 147
6.3 Problem Formulation 151
    6.3.1 Fault Detection and Diagnosis Problem 151

6.3.2 Robust Active Controller Reconfiguration Problem 152
6.4 Combined FDD and robust active FTC 153
6.4.1 Algorithm for FDD 153
6.4.2 Algorithm for Controller Reconfiguration 158
6.5 Experimental results 161
6.6 Conclusions 165

7 Multiple-Model Approach to Fault-Tolerant Control 167
7.1 Introduction 168
7.2 The model set 170
7.2.1 Hybrid dynamic model 170
7.2.2 Nonlinear system 171
7.2.3 The model set design 172
7.3 The IMM estimator for systems with offset 172
7.4 The MM-based GPC 173
7.4.1 The GPC for systems with offset 174
7.4.2 The combination of the GPC with the IMM estimator . . . 179
7.5 Simulation results 182
7.5.1 Experiment with the SRM 182
7.5.2 Experiment with the inverted pendulum on a cart 185
7.6 Conclusion 189

8 Conclusions and Recommendations 191
8.1 Conclusions 191
8.2 Recommendations 194

Summary 197

Samenvatting 203

Notation 209

List of Abbreviations 213

References 215

Curriculum Vitae 237

Acknowledgements

The work presented in this thesis have been carried out over a period of four years under the supervision of Prof. Michel Verhaegen. It was his excellent supervision with many informal discussions and brainstorming sessions that have thought me how to do research. Thank you, Michel, for steering me towards all these interesting and challenging open questions out there, and for your ideas and suggestions that helped me in the process of looking for the answers to these questions. I have really learn a lot from you.

The research during my Ph.D. course has been sponsored by the Dutch Technology Foundation (STW) under project number DEL.4506 *"Neuro-Fuzzy Modelling in Model Based Fault-Detection, Fault Isolation and Controller Reconfiguration"*. I was involved in the second work-package dealing with the controller reconfiguration part. The financial support of STW is much appreciated. I thank all the user's committee members of the project, and especially Dr. Jan Frans Bos (Dutch Space) and Dr. Jan Breeman (NLR) for the valuable feedback they were always giving me during the user's committee meetings, as well as for providing me with the linear model of ERA.

Then I would like to thank Prof. Carsten Scherer and Dr. Bart De Schutter who, as co-authors of some of my papers, have a direct contribution to some important results in this dissertation. Thank you very much for the many technical discussions that we had and for providing me with useful feedback.

Additionally, I want to thank all the members of my Ph.D. defence committee: Prof. Michel Kinnaert (Université Libre de Bruxelles), Prof. Roberto Tempo (Politecnico di Torino), Prof. Job van Amerongen (University of Twente), Prof. Brian Roffel (University of Twente), Prof. Anton Stoorvogel (TU-Eindhoven and TU-Delft). Special thanks to Prof. Roberto Tempo, with whom we have had fruitful communications on many occasions, and to Prob. Michel Kinnaert for the many interesting discussions during conferences and workshops. Special thanks also to Prof. Anton Stoorvogel for the very thorough review of the draft version my thesis, for providing me with many constructive comments and for giving very interesting interpretations on some points of the thesis.

The longest part of the period as a Ph.D. student, the first three years, I have spent at the Systems and Control Engineering group, University of Twente. I therefore want to thank all of my ex-colleagues there: Bas Benschop, Niek Bergboer, Stijn Donders, Rufus Fraanje, Karel Hinnen, Dr. Ichiro Jikuja, Gerard Nijsse, Dr. Hiroshi Oku, Dr. Vincent Verdult. Thank you all guys, I was really nice working with you. I would also like to thank to all the members of the former Control

Laboratory group (TU-Delft) and the new Delft Center for Systems and Control (TU-Delft), where I spent the last year of my Ph.D. I especially want to thank Karel Hinnen, with whom we shared the last year (and still do) the same room at TU-Delft, for revising the chapter "*Samenvatting*".

I am very thankful for all the support that was given me by my family, Marusia, Kamen and Martin. You have always given me the right assistance at the right time; *you* made this thesis happen. Thank you, I will always there for you.

And last, but most, I thank my wife Ina for all the love, support and for painting my life in color. *"Jij bent de zon en de maan voor mij."*

1 Introduction

1.1 Why Fault-Tolerant Control?

Nowadays, control systems are everywhere in our life. They are all around us, often remaining invisible for the eye of most of us. They are in our kitchens, in our DVD-players and computers. They are driving the elevators, we have them in our cars, ships, aircraft and spacecraft. Control systems are present in every industry, they are used to control chemical reactors, distillation columns, and nuclear power plants. They are constantly and inexhaustibly working, making our life more comfortable and more pleasant... Until the system fails.

Faults in technological systems are events that happen rarely, often at unexpected moments of time. In Isermann and Ballé (1997) the following definition for a fault is made:

> **fault** is an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition.

Faults are difficult to foresee and prevent. Their further development into overall system failures may lead to consequences that take different forms and scales, ranging from having to spend another €50 for a new coffee machine to enormous economical and human losses in safety-critical systems. There are numerous examples of such dramatic incidents as a result of failures in safety-critical systems. Several such examples are

1. the explosion at the nuclear power plant at Chernobyl, Ukraine, on 26 April 1986 (Mahmoud et al. 2003). About 30 people were killed immediately, while another 15,000 were killed and 50,000 left handicapped in the emergency clean-up after the accident. It is estimated that five million people were exposed to radiation in Ukraine, Belarus and Russia (BBC World 2001).

2. the crash of the AMERICAN AIRLINES flight 191, a McDonnell-Douglas DC-10 aircraft, at Chicago O'Hare International Airport on 25 May 1979. 271 persons on board and 2 on the ground were killed when the aircraft crashed into an open field (NTSB 1979; Patton 1997).

3. the explosion of the Ariane 5 rocket on 4 June 1996, where the reason was a fault in the Internal Reference Unit that has the task to provide the control system with altitude and trajectory information. As a result, incorrect altitude information was delivered to the control unit (Mahmoud et al. 2003).

4. the crash of the Boeing 747-200F freighter on 4 October 1992. Shortly after takeoff from Schiphol Amsterdam International Airport multiple engine separations on the right wing occurred leading to different severe damages. Fifteen minutes later the aircraft crashed into an eleven-floor building (Maciejowski and Jones 2003).

The question that immediately arises is *"Could something have been done to prevent these disasters?"*. While in most situations the occurrences of faults in the systems cannot be prevented, subsequent analysis often reveals that *the consequences of faults could be avoided* or, at least, that their severity (in terms of economic losses, casualties, ets.) could be minimized. If faults could timely be detected and diagnosed in many cases it is possible to subsequently reconfigure the control system so that it can safely continue its operation (possibly with degraded performance) until the time comes when it can be switched off for maintenance. In order to minimize the chances for catastrophic events as those summarized above, safety-critical systems must possess the properties of increased reliability and safety.

A way to achieve that is by means of fault-tolerant control system (FTCS) design. An FTCS could have been designed to lead to a safe shutdown of the Shernobyl reactor way before it exploded (Mahmoud et al. 2003). Subsequent studies after the McDonnell-Douglas DC-10 crash showed that the crash could have been avoided (Patton 1997). At the last minutes of the Ariane 5 crash the normal altitude information has been replaced by some diagnostic information that the control system was not designed to understand (Mahmoud et al. 2003). Finally, in the case with the Boeing freighter, simulation studies (Maciejowski and Jones 2003) have subsequently revealed that it was possible to reconfigure the controller so that the aircraft could be landed safely. Fortunately, such positive outcomes are not only possible in theory and simulations, but can also happen in practise:

1. A McDonnell-Douglas DC-10 aircraft executing flight 232 of UNITED AIRLINES from Denver to Minneapolis experienced a disastrous failure in the hydraulic lines that left the plane without any control surfaces at 37,000 ft. The captains then improvised a control strategy that used only the throttles of the two wing engines and managed to successfully crash-land the plane in Sioux City, Iowa, saving the lives of 184 out of the 296 passengers on board (Jones 2002; Maciejowski and Jones 2003).

2. In the DELTA AIRLINES flight 1080 an elevator became jammed at 19 degrees up. The pilot has not been given any indication of what has actually occurred and still was able to reconfigure the remaining lateral control elements and to land the aircraft safely (Patton 1997).

All these examples clearly motivate the need for increased fault-tolerance in order to improve to the maximum possible extent the safety, reliability and avail-

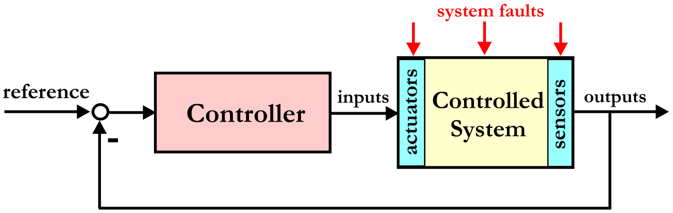


Figure 1.1: According to their location, faults are classified into sensor, actuator and component faults.

ability of the safety-critical modern controlled systems that have constantly increasing complexity. The examples above also explain the large amount of research in the field of fault detection, diagnosis and fault-tolerant control. An overview of this research will be provided in section 1.5 of this chapter. The purpose of this thesis is to develop methods for achieving increased fault-tolerance by means of fault-tolerant control.

1.2 Fault Classification

Fault are events that can take place in different parts of the controlled system. In the FTCS literature faults are classified according to their location of occurrence in the system as (see Figure 1.1)

actuator faults: they represent *partial* or *total* (complete) loss of control action. An example of a completely lost actuator is a "stuck" actuator that produces no (controllable) actuation regardless of the input applied to it. Total actuator fault can occur, for instance, as a result of a breakage, cut or burned wiring, shortcuts, or the presence of outer body in the actuator. Partially failed actuator produces only a part of the normal (i.e. under nominal operating conditions) actuation. It can result from, e.g., from hydraulic or pneumatic leakage, increased resistance or fall in the supply voltage. Duplicating the actuators in the system in order to achieve increased fault-tolerance is often not an option due to their high prices and large sizes.

sensor faults: these faults represent incorrect reading from the sensors that the system is equipped with. Sensor faults can also be subdivided into *partial* and *total.* Total sensor faults produce information that is not related to value of the measured physical parameter. They can be due to broken wires, lost contact with the surface, etc. Partial sensor faults produce reading that is related to the measured signal in such a way that useful information could still be retrieved. This can, for instance, be a gain reduction so that a scaled version of the signal is measured, a biased measurement resulting in a (usually constant) offset in the reading, or increased noise. Due to their smaller sizes sensors can be duplicated in the system to increase the fault tolerance. For instance, by using three sensors to measure the

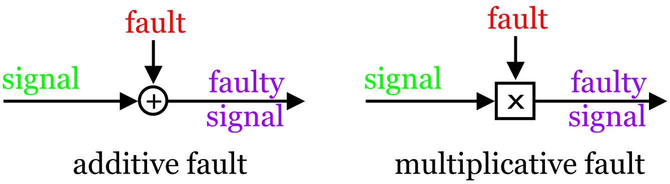


Figure 1.2: According to their representation, faults are divided into additive and multiplicative.

same variable one may consider it reliable enough to compare the readings from the sensors to detect faults in (one and only one) of them. The so-called "majority voting" method can then be used to pinpoint the faulty sensor. This approach usually implies significant increase in the related costs.

component faults: these are faults in the components of the plant itself, i.e. all faults that cannot be categorized as sensor or actuator faults will be referred to as component faults. These faults represent changes in the the physical parameters of the systems, e.g. mass, aerodynamic coefficients, damping constant, etc., that are often due to structural damages. They often result in a change in the dynamical behavior of the controlled system. Due to their diversity, component faults cover a very wide class of (unanticipated) situations, and as such are the most difficult ones to deal with.

The approaches developed in this thesis deal with sensor, actuator and/or component faults.

Further, with respect to the way faults are modelled, they are classified as *additive* and *multiplicative*, as depicted on Figure 1.2. Additive faults are suitable to represent component faults in the system, while sensor and actuator faults are in practice most often multiplicative by nature.

Faults are also classified according to their time characteristics (see Figure 1.3) as *abrupt*, *incipient* and *intermittent*. Abrupt faults occur instantaneously often as a result of a hardware damage. Usually they are very severe as they affect the performance and/or the stability of the controlled system, and as such require prompt reaction by the FTCS. Incipient faults represent slow in time parametric changes, often as a result of aging. They are more difficult to detect due to their slow time characteristics, but are also less severe. Finally, intermittent faults are faults that appear and disappear repeatedly, for instance due to partially damaged wiring.

1.3 Modelling Faults

As already mentioned in Section 1.2, according to the way of representation faults are divided into additive and multiplicative. In this section we further concentrate on the mathematical representation of these faults and will provide a

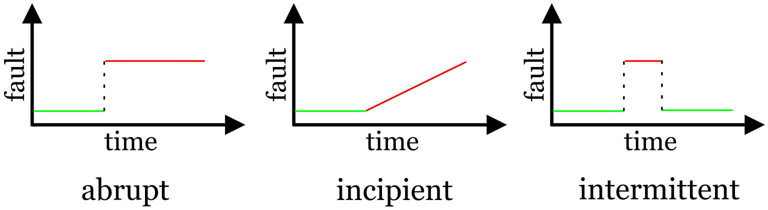


Figure 1.3: With respect to their time characteristics faults can be abrupt, incipient and intermittent.

discussion on when and why the one representation is more appropriate than the other.

Throughout this thesis the state-space representation of dynamical systems is used, so that the relation from the system inputs $u \in \mathbb{R}^m$ to the measured outputs $y \in \mathbb{R}^p$ is written in the form

$$S_{nom} : \left\{ \begin{array}{rcl} x_{k+1} & = & Ax_k + Bu_k \\ y_k & = & Cx_k + Du_k, \end{array} \right. \tag{1.1}$$

where $x \in \mathbb{R}^n$ denotes the state of the system.

1.3.1 Multiplicative faults

Multiplicative modelling is mostly used to represent sensor and actuator faults.

Actuator faults represent malfunctioning of the actuators of the system, for example as a result of hydraulic leakages, broken wires, stuck control surfaces in an aircraft, etc. Such faults can be modelled as an abrupt change of the nominal control action from u_k to

$$u_k^f = u_k + (I - \Sigma_A)(\bar{u} - u_k), \tag{1.2}$$

where $\bar{u} \in \mathbb{R}^m$ is a (not necessarily constant) vector that cannot be manipulated, and where

$$\Sigma_A = \textbf{diag}\{[\ \sigma_1^a, \quad \sigma_2^a, \quad \ldots, \quad \sigma_m^a\ ]\},\ \sigma_i^a \in \mathbb{R}.$$

In this way $\sigma_i^a = 0$ represents a total fault (or, in other words, a complete failure) of i-th actuator of the system so that the control action coming from this i-th actuator becomes equal to the i-th element of the uncontrollable offset vector $\bar{u}$, i.e. $u_k^f(i) = \bar{u}(i)$. On the other hand, $\sigma_i^a = 1$ implies that the i-th actuator operates normally ($u_k^f(i) = u(i)$). The quantities σ_i^a, $i = 1, 2, \ldots, m$ can also take values in between 0 and 1, making it in this way possible to represent partial actuator faults. Substituting the nominal control action u_k in equation (1.1) with the faulty u_k^f results in the following state-space model

$$S_{mult,af} : \left\{ \begin{array}{rcl} x_{k+1} & = & Ax_k + B\Sigma_A u_k + B(I - \Sigma_A)\bar{u} \\ y_k & = & Cx_k + D\Sigma_A u_k + D(I - \Sigma_A)\bar{u}. \end{array} \right. \tag{1.3}$$

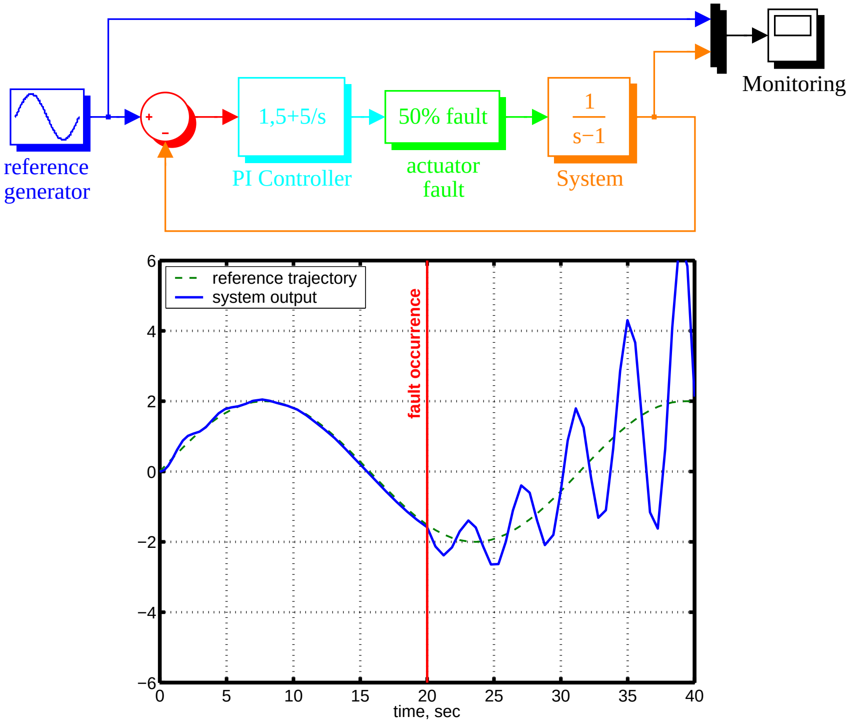


Figure 1.4: After a multiplicative fault the system may become unstable if no reconfiguration takes place.

Models in the form (1.3) are referred to as multiplicative fault models and have been widely used in the literature on FTC (see, e.g., Tao et al. (2001); Noura et al. (2000); Bošković and Mehra (2003)).

It needs to be noted here that while such multiplicative actuator faults do not directly affect the dynamics of the controlled system itself, they can significantly affect the dynamics of the closed-loop system, and may even affect the controllability of the system. Figure 1.4 presents a simple example with a partial 50% actuator fault that results in instability of the closed-loop system. In the example of Figure 1.4 a system with transfer function $S(s) = 1/(s-1)$ is controlled by a PI controller with transfer function $C(s) = 1.5 + \frac{5}{s}$, so that a sinusoidal reference signal is tracked in under normal operating conditions (i.e. during the first 20 seconds from the simulation). At time instant $t = 20$ sec, a 50% loss of control effectiveness is introduced and as a result the closed-loop system stability is lost. This example makes it clear that even “seemingly simple” faults may significantly degrade the performance and can even destabilize the system.

Similarly, sensor faults occurring in the system (1.1) represent incorrect reading from the sensors, so that as a result the real output of the system y_k^{real} differs from the variable being measured. Multiplicative sensor faults can be modelled in the following way

$$y_k^f = y_k + (I - \Sigma_S)(\bar{y} - y_k), \tag{1.4}$$

where $\bar{y} \in \mathbb{R}^p$ is an offset vector, and

$$\Sigma_S = \mathbf{diag}\{[\ \sigma_1^s, \quad \ldots, \quad \sigma_p^s\ ]\},\ \sigma_i^s \in \mathbb{R},$$

so that $\sigma_j^s = 0$ represents a total fault of j-th sensor, and $\sigma_j^s = 1$ models the normal mode of operation of the j-th sensor. Partial faults are then modelled by taking $\sigma_j^s \in (0,\ 1)$. Substitution of the nominal measurement y_k in (1.1) with its faulty counterpart y_k^f results in the following state-space model that represents multiplicative sensor faults

$$S_{mult,sf} : \left\{ \begin{array}{rcl} x_{k+1} & = & Ax_k + Bu_k \\ y_k & = & \Sigma_S C x_k + \Sigma_S D u_k + (I - \Sigma_S)\bar{y}. \end{array} \right. \tag{1.5}$$

In this way, combinations of multiplicative sensor and actuator faults are represented in the following way

$$S_{mult} : \left\{ \begin{array}{rcl} x_{k+1} & = & Ax_k + B\Sigma_A u_k + b(\Sigma_A, \bar{u}) \\ y_k & = & \Sigma_S C x_k + \Sigma_S D \Sigma_A u_k + d(\Sigma_A, \Sigma_S, \bar{u}, \bar{y}), \end{array} \right. \tag{1.6}$$

with

$$\begin{array}{l} b(\Sigma_A, \bar{u}) = B(I - \Sigma_A)\bar{u}, \\ d(\Sigma_A, \Sigma_S, \bar{u}, \bar{y}) = \Sigma_S D(I - \Sigma_A)\bar{u} + (I - \Sigma_S)\bar{y}. \end{array}$$

The multiplicative model is thus a “natural” way to model a wide variety of sensor and actuator faults, but cannot be used to represent more general component faults. This fault model representation is most often used in the design of the controller reconfiguration scheme of an active FTCS as for controller redesign one usually needs the state-space matrices of the faulty system. For that reason, the methods developed in this thesis are mostly based on the multiplicative sensor and actuator fault representation, as well as on the more general component fault representation discussed in section 1.3.3. It is further assumed throughout this thesis that the *faulty system remains at least stabilizable*[1]. It this assumption does not hold then no stabilizing controller reconfiguration is possible so that other measures for safe shutdown of the system would have to be taken when possible. Such measures, however, fall outside of the focus of this thesis.

1.3.2 Additive faults

The additive faults representation is more general than the multiplicative one. A state-space model with additive faults has the form

$$S_{add} : \left\{ \begin{array}{rcl} x_{k+1} & = & Ax_k + Bu_k + Ff_k \\ y_k & = & Cx_k + Du_k + Ef_k, \end{array} \right. \tag{1.7}$$

---

[1] We note that this condition is weaker than a controllability condition. It makes sure that there exists control action that results in stable closed-loop system. Additionally, in the case when the state is not directly available for measurement a similar detectability condition is assumed for the same reason.

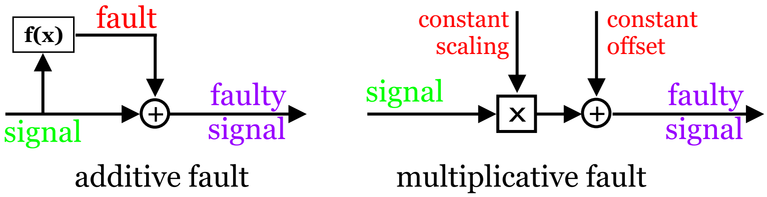


Figure 1.5: Using additive fault representation to model total sensor (actuator) faults results in a fault signal that depends on y_k (u_k). This is not the case with the multiplicative model where the fault magnitude and the offset are independent on the signals in the state-space model.

where $f_k \in \mathbb{R}^{n_f}$ is a signal describing the faults. This representation may, in principle, be used to model a wide class of faults, including sensor, actuator, and component faults. Using model (1.7), however, often results in the signal f_k becoming related to one or more of the signals u_k, y_k and x_k. For instance, if one would use this additive fault representation to model a total fault in all actuators (set $\Sigma_A = 0$ and $\bar{u} = 0$ in equation (1.2) on page 5) then in order to make model (1.7) equivalent to model (1.3) one needs to take a signal f_k such that $\left[\begin{smallmatrix} F \\ E \end{smallmatrix}\right] f_k = -\left[\begin{smallmatrix} B \\ D \end{smallmatrix}\right] \Sigma_A u_k$ holds, making f_k dependent on u_k. Clearly, the fault signal being a function of the control action is not desirable for controller design. On the other hand, f_k is independent on u_k when multiplicative representation is utilized. Figure 1.5 illustrates this.

Another disadvantage of the additive model when used to represent sensor and actuator faults is that, in terms of input-output relationships, these two faults become difficult to distinguish. Indeed, suppose that the model

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + f_k^a \\ y_k &= Cx_k + Du_k + f_k^s, \end{aligned}$$

is used to represent faults in the sensors and actuators. By writing the corresponding transfer function

$$y(z) = (C(zI - A)^{-1}B + D)u_k + C(zI - A)^{-1} f_k^a + f_k^s,$$

it becomes indeed clear, that the effect of an actuator fault on the output of the system can be modelled not only by the signal f_k^a, but also by f_k^s.

An advantage is, as already mentioned, that the additive representation can be used to model a more general class of faults than the multiplicative one. In addition to that, it is more suitable for the design of FDD schemes because the faults are represented by one *signal* rather than by changes in the state-space matrices of the system as is the case with the multiplicative representation. For that reason the majority of FDD methods are focused on additive faults (Gertler 2000; Basseville 1998; Kinnaert 2003; Frank et al. 2000).

1.3.3 Component faults

The class of component faults was defined in Section 1.2 as the most general as it includes faults that may bring changes in practically any element of the system. It was defined as the class of all faults that cannot be classified as sensor or actuator faults. Component fault may introduce changes in each matrix of the state-space representation of the system due to the fact they may all depend on the same physical parameter that undergoes a change. Component faults are often modelled in the form of a linear parameter-varying (LPV) system

$$\begin{aligned} x_{k+1} &= A(f)x_k + B(f)u_k \\ y_k &= C(f)x_k + D(f)u_k, \end{aligned} \tag{1.8}$$

where $f \in \mathbb{R}^{n_f}$ is a parameter vector representing the component faults. Obviously, this model might also be used for modelling sensor and actuation faults, in addition. Due to the fact the the matrices may depend in a general nonlinear way on the fault signal f_k this model is less suitable for fault detection and diagnosis. Later in this thesis we will present an algorithm for on-line fault-tolerant control (FTC) for the general model (1.8) when the fault signal f is only known to lie in some uncertainty interval with time-varying size.

In the next section we continue the discussion with the structure and main components of a FTCS.

1.4 Main Components in a FTCS

FTCS are generally divided into two classes: *passive* and *active*. Passive FTCS are based on robust controller design techniques and aim at synthesizing one (robust) controller that makes the closed-loop system insensitive to certain faults. This approach requires no online detection of the faults, and is therefore computationally more attractive. Its applicability, however, is very restricted due to its serious disadvantages:

- In order to achieve such robustness to faults, usually a very restricted subset of the possible faults can be considered; often only faults that have a “small effect” on the behavior of the system can be treated in this way.

- Achieving increased robustness to certain faults is only possible at the expense of decreased nominal performance. Since faults are effects that happen very rarely it is not reasonable to significantly degrade the fault-free performance of the system only to achieve some insensitivity to a restricted class of faults.

As opposed by the passive methods, the *active* approach to the design of FTCS is based on controller redesign, or selection/mixing of predesigned controllers. This technique usually requires a fault detection and diagnosis (FDD) scheme that has the task to detect and localize the faults that eventually occur in the system. The structure of an active FDD-based FTCS is presented on Figure 1.6. The FDD part uses input-output measurement from the system to detect

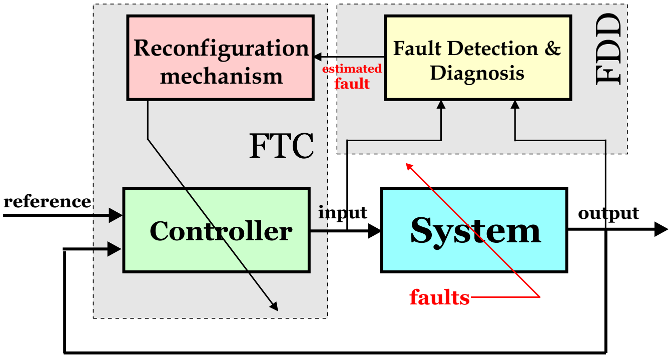


Figure 1.6: Main components of an active FTCS.

and localize the faults. The estimated faults are subsequently passed to a reconfiguration mechanism (RM) that changes the parameters and/or the structure of the controller in order to achieve an acceptable post-fault system performance.

Depending on the way the post-fault controller is formed, the active FTC methods are further subdivided into *projection-based* methods and *on-line redesign* methods. The projection based methods rely on the controller selection from a set of off-line predesigned controllers. Usually each controller from the set is designed for a particular fault situation and is switched on by the RM whenever the corresponding fault pattern has been diagnosed by the FDD scheme. In this way only a restricted, finite class of faults can be treated. The on-line redesign methods involve on-line computation of the controller parameters, referred to as *reconfigurable control*, or recalculation of both the structure and the parameters of the controller, called *restructurable control*. Comparing the achievable post-fault system performances, the on-line redesign method is superior to the passive method and the off-line projection-based method. However, it is computationally the most expensive method as it often boils down to on-line optimization.

There are a number of important issues when designing active FTCS. Probably the most significant one is the *integration between the FDD part and the FTC* part. The majority of approaches in the literature are focused on one of these two parts by either considering the absence of the other or assuming that it is perfect. To be more specific, many FDD algorithms do not consider the closed-loop operation of the system on the one hand, and many FTC methods assume the availability of perfect fault estimates from the FDD scheme on the other hand. The interconnection of such methods is clearly infeasible and there can be no guarantees that a satisfactory post-fault performance, or even stability, can be maintained by such a scheme. It is therefore very important that the designs of the FDD and FTC, when carried out separately, are each performed bearing in mind the presence and imperfection of the other. For making the interconnection possible, one should first investigate what information from the FDD is needed by the FTC, as well as what information can actually be provided by the FDD scheme. Imprecise information from the FDD that is incorrectly interpreted by the FTC scheme might lead to a complete loss of stability of the system.

An usual situation in practise is that after the occurrence of a fault in the system there is initially not enough information in terms of input/output measurements from the system that is available to make it possible for the FDD scheme to diagnose the fault. For this reason, only after some time elapses and more information becomes available the FDD scheme can detect that a fault has occurred, and even more time to localize the fault and its magnitude. As a result, the information that is provided to the FTC part is initially more imprecise (i.e. with larger uncertainty), and it gets more and more accurate (with less uncertainty) as more data becomes available from the system. The FTC scheme should be able to deal with such situations. Therefore, the FTC should necessarily be capable of dealing with *uncertainty in the FDD* information/estimates, and should perform satisfactorily (guaranteeing at least the stability) during the *transition period* that the FDD scheme needs to diagnose the fault(s).

Very often the dynamics of real physical systems cannot be represented accurately enough by linear dynamical models so that *nonlinear models* have to be used. This necessitates the development of techniques for FTCS design that can explicitly deal with nonlinearities in the mathematical representation of the system. Nonlinearities are, in fact, very often encountered in the representations of complex safety-critical controlled systems like aircraft and spacecraft. For instance, it is usual that the lateral and longitudinal dynamics of an aircraft are decoupled so that they have no effect on each other. This significantly simplifies the model of the aircraft and makes it possible to design the corresponding controllers independently. This decoupling condition can approximately be achieved for a healthy aircraft, but certain faults can easily destroy it, so that the two controllers could not be considered separately.

An important issue in the design of FTCS is that even for a fixed operating region, where a nonlinear system eventually allows approximation by a linear model, it is very difficult to obtain an accurate linear representation, either due to the fact that the physical parameters in the nonlinear model are not exactly known or because they vary with time. Even the nonlinear model is often derived after some simplifying assumptions, so that it only approximates the behavior of the system. Even more, this uncertainty is further increased due to the linearization that basically consists in truncating second and higher order terms in the Taylor series expansion of the nonlinear function. As a result only a representation with *uncertainty* is available. It is important that the FTCS is designed to be robust to such uncertainties in the model of the controlled system.

Another very important issue is that every real-life controlled system has *control action saturation*, i.e. the input signal cannot get higher than a certain value. In the design phase of a control system usually the effect of the saturation is taken care of by making sure that the control action will not get overly active and will remain inside the saturation limits under normal operating conditions. Faults, however, can have the effect that the control action stays at the saturation limit. For instance, when a partial 50% loss of effectiveness in an actuator has been diagnosed, a standard and easy way to accommodate the fault is to re-scale the control action by two so that the resulting actuation approximates the fault-free actuation. As a result the control action becomes twice as big and may go to the saturation limits. Clearly, in such situations one should not try to completely accommodate the fault but one should be willing to accept to accept certain performance degradation imposed by the saturation. In other words, a trade-off between achievable performance and available actuator capability might need to be made after the occurrence of a fault. This situation is often referred to as *graceful performance degradation* Jiang and Zhang (2002).

1.5 The State-of-the-Art in FTC

In this section an overview of the existing work in the area of FTC is given, an area that is gaining more and more attention lately. For all classes of methods a short discussion is included with its advantages, drawbacks, and relations to the methods presented in this thesis. Some overview books and papers in the field of FTC are (Astrom et al. 2001; Blanke et al. 2003; Hajiyev and Caliskan 2003; Zhang and Jiang 2003; Patton 1997; Blanke et al. 2000; Rauch 1995; Stengel 1991; Blanke et al. 2001, 1997; Blanke 1996; van Schrik 2002; Huzmezan and Maciejowski 1997; Liaw and Y.Liang 2002).

1.5.1 Passive Methods for FTC

As explained in the previous section, the passive methods aim at achieving insensitivity to certain faults by means of making the system robust with respect to them. When applied for dealing with component faults (see model (1.8) on page 9) these methods usually assume that the state-space matrices of the system depend on the fault signal f in some specific way, e.g. affinely Wu (1997b); Stoustrup et al. (1997), in the form of a linear fractional transformation (LFT) Niemann and Stoustrup (2002); Chen et al. (1998a), etc. To overcome this restriction, in Chapter 2 of this thesis an algorithm is proposed that does not impose any assumption on the way the system matrices depend on the fault signal, i.e. f can enter the state-space matrices in a general way as long as they remain bounded.

In addition to that, in passive FTC methods often fault-tolerance is achieved by means of representing certain faults as uncertainty in the system so that a robust controller can be designed. By doing this, however, the *structure* of this uncertainty is often neglected in order to arrive at a convex (usually $\mathcal{H}_\infty$) optimization problem Chen and Patton (2001); Niemann and Stoustrup (2003). To reduce the resulting conservatism, a nonconvex optimization approach is proposed in Chapter 3 of this thesis that has guaranteed convergence to a local optimum of the cost function ($\mathcal{H}_2$ and $\mathcal{H}_\infty$ cost functions are considered).

A summary of some approaches to passive FTC is provided below.

Reliable control:

This passive controller approach aims at making the closed-loop system reliable so that it pertains stability/performance in the cases of some specific anticipated faults. The goal is to search for a controller that optimizes the so called *worst-fault performance* (usually in terms of LQR or $\mathcal{H}_\infty$ design) for all possible anticipated faults (usually a set of sensor or actuator outages). The approach assumes that complete failures may occur only in a predefined subset of the set of sensors and actuators of the system. For an overview of reliable control methods consult (Veillette 1992, 1995; Hsieh 2002; Yang et al. 1999; Zhao and Jiang 1998; Niemann and Stoustrup 2002; Liang et al. 2000; Liao et al. 2002; Seo and Kim 1996; Chang 2000; Cho and Bien 1989; Suyama 2002; Suyama and Zhang 1997; Ge et al. 1996; Suyama and Zhang 1997; Ferreira 2002).

Robust control:

This is another class of passive approaches that aims at the design of one robust controller that meets not only the design specifications under normal operating conditions, but also achieves some performance in cases of some faults. These approaches are usually based on *quantitative feedback theory* (Keating et al. 1997; Niksefat and Sepehri 2002) or *robust $\mathcal{H}_\infty$ controller design* (Zhou and Ren 2001; Zhou 2000; Chen and Patton 2001; Niemann and Stoustrup 2003; Stoustrup et al. 1997; Stoustrup and Niemann 2001; Tyler and Morari 1994; Murad et al. 1996; Chen et al. 1998a,b; Demetriou 2001b; Hamada et al. 1996; Joshi 1997; Maghami et al. 1998; Suzuki and Tomizuka 1999; Wu 1997b, 1993).

1.5.2 Active Methods for FTC

Due to their improved performance and their ability to deal with a wider class of faults, the active methods for FTC have gained much more attention in the literature than the passive ones. A bibliographical overview is presented below.

Pseudo Inverse:

The *pseudo-inverse method* (PIM) (Gao and Antsaklis 1991) is one of the most cited active methods to FTC due to its computational simplicity and its ability to handle a very large class of system faults. The basic version of the PIM considers a nominal linear system

$$\left\{\begin{array}{rcl} x_{k+1} & = & Ax_k + Bu \\ y_k & = & Cx_k, \end{array}\right. \tag{1.9}$$

with linear state-feedback control law $u_k = Fx_k$, under the assumption that the state vector is available for measurement. The method allows for a very general post-fault system representation

$$\left\{\begin{array}{rcl} x^f_{k+1} & = & A_f x^f_k + B_f u^R_k \\ y^f_k & = & C_f x^f_k, \end{array}\right. \tag{1.10}$$

where the new, reconfigured control law is taken with the same structure, i.e. $u^R_k = F_R x^f_k$. The goal is then to find the new state-feedback gain matrix F_R in such a way that the "distance" (defined below) between the A-matrices of the nominal and the post-fault closed-loop systems is minimized, i.e.

$$\text{PIM} : \left\{ \begin{array}{rcl} F_R & = & \arg\min_{F_R} \|(A + BF) - (A_f + B_f F_R)\|_F \\ & = & B_f^{\dagger}(A + BF - A_f), \end{array} \right. \qquad (1.11)$$

where $B_f^{\dagger}$ is the pseudo-inverse of the matrix B_f. The advantages of this approach are that it is very suitable for on-line implementation due to its simplicity, and moreover, that it allows for changes in all state-space matrices of the system as a consequence of the faults. A very strong disadvantage is, however, that the optimal control law computed by equation (1.11) does not always stabilize the closed-loop system. Simple examples that confirm this fact can easily be generated, see e.g. Gao and Antsaklis (1991). To circumvent this problem, the *modified pseudo-inverse method* was developed in Gao and Antsaklis (1991) that basically solves the same problem under the additional constraint that the resulting closed-loop system remains stable. This, however, results in a constraint optimization problem that increases the computational burden. Similar approach is also discussed in (Rauch 1994; Liu 1996), where the reconfigured control action u_k^R is directly computed from the nominal control u_k as $u_k^R = B_f^{\dagger} B u_k$. Other modifications of this approach were proposed considering additive faults on the state equation and additive term on the control action to compensate for them (Theilliol et al. 1998; Noura et al. 2000, 1999), static output-feedback is considered in (Konstantopoulos and Antsaklis 1999, 1995), and matching of the frequency responses of the nominal and the post-fault closed-loop systems is considered in Yang and Blanke (2000a). Another disadvantage of the approach is that it deals with the state-feedback case, and that it is, in general, not applicable to sensor faults as well as to problems with model and/or FDD uncertainty. An extension of this method that deals with both sensor and actuator faults is proposed in Kanev and Verhaegen (2000a) where a bank of reconfigurable LQG controllers has been developed. The stability is enforced through LMI optimization.

Eigenstructure assignment:

The eigenstructure assignment (EsA) method (Liu and Patton 1998; Seron et al. 1996) to controller reconfiguration is a more intuitive approach than the PIM as it aims at matching the eigenstructures (i.e. the eigenvalues and the eigenvectors) of the A-matrices of the nominal and the faulty closed-loop systems. The main idea is to exactly assign some of the most dominant eigenvalues while at the same time minimizing the 2-norm of the difference between the corresponding eigenvectors. The procedure has been developed both under constant state-feedback (Zhang and Jiang 1999a, 2000) and output-feedback (Konstantopoulos and Antsaklis 1996a,b; Belkharraz and Sobel 2000). More specifically, in the state-feedback case, if λ_i, $i = 1, 2, \ldots, n$ are the eigenvalues of the A-matrix of the nominal closed-loop system formed as the interconnection of (1.9) with the constant state-feedback control action $u_k = Fx_k$, and if v_i are their corresponding eigenvectors, the EsA method computes the state-feedback gain F_R for the faulty model (1.10) as the solution to the following problem (Zhang and Jiang 1999a)

$$\text{EsA}: \begin{cases} \text{Find} & F_R \\ \text{such that} & (A_f + B_f F_R) v_i^f = \lambda_i v_i^f,\ i = 1, \ldots, n, \\ \text{and} & v_i^f = \arg\min_{v_i^f} \|v_i - v_i^f\|_{W_i}^2, \end{cases} \tag{1.12}$$

where $\|v_i - v_i^f\|_{W_i}^2 = (v_i - v_i^f)^T W_i (v_i - v_i^f)$. In other words, the new gain F_R needs to be such that the poles of the resulting closed-loop system coincide with the poles of the nominal closed-loop system and, in addition, the eigenvectors of the closed-loop A-matrices are as close as possible. As both the eigenvectors and the eigenvalues determine the shape of the time response of the closed-loop system, this method can be thought of as trying to preserve the nominal closed-loop system time-response after the occurrence of faults. Thus, the objective of the EsA method seems more “natural” than that of the PIM and, moreover, the stability is guaranteed. The computational burden of the approach is not high since analytic expression for the solution to (1.12) is available (Zhang and Jiang 1999a), i.e. no on-line optimization is necessary. Disadvantage is that model and FDD uncertainties cannot be easily incorporated in the optimization problem, and that only static controllers are considered.

Multiple Model:

The multiple model (MM) method is another active approach to FTC that belongs rather to the class of projection based methods than to the on-line redesign methods. It is based on a finite set of linear models M_i, $i = 1, 2, \ldots, N$ that describe the system in different operating conditions, i.e. in the presence of different faults in the system. For each such local model M_i a controller C_i is designed (off-line). The key in the design is to develop a an on-line procedure that determines the global control action through a (probabilistically) weighted combination of the different control actions can be taken (Athans et al. 1977; Maybeck and Stevens 1991; Griffin and Maybeck 1997; Zhang and Jiang 2001, 1999b; Theilliol et al. 2003; Demetriou 2001a). The control action mixing is sometimes called blending (Griffin and Maybeck 1997). The mixing is usually based on a bank of Kalman filters, where each Kalman filter is designed for one of the local models M_i. On the basis of the residuals of the Kalman filters the probabilities $\mu_i \geq 0$ of each model to be in effect are computed that subsequently act as weights in the computation of the control action

$$u(k) = \sum_{i=1}^{N} \mu_i(k) u_i(k),\ \sum_{i=1}^{N} \mu_i = 1, \tag{1.13}$$

where $u_i(k)$ is the control action produced by the controller designed for the i-th local model.

The multiple model method is a very attractive tool for modelling and control of nonlinear systems. However, these approaches usually only consider a finite number of *anticipated* faults only and proceed by building one local model for each anticipated fault. In this way, at each time instant only one model, say model M_i, is assumed to be in effect, so that its corresponding weight μ_i is approximately equal to one and all other weights μ_j, $j \neq i$ are close to zero. In such cases at each time instant one local controller is "active", namely the one corresponding to the model M_i that is in effect. The disadvantage here is that if the current model is not in the predesigned model set and is instead formed by some convex combination of the local models in the model set (representing, for instance, *unanticipated* faults) then, in general, the control action (1.13) is not the optimal one for this model. A simple example is provided in Chapter 7 of this thesis that illustrates that forming the global control action as in (1.13) can in such cases even lead to instability of the closed-loop system. In order to avoid that when dealing with unanticipated faults, the approach proposed in Chapter 7 considers a bank of predictive controllers and forms the global control action in an optimal way (in terms of minimizing a cost function), so that the optimal control action for the current model is used at each time instant instead of (1.13). Another disadvantage of the MM approaches is that model uncertainties, as well as uncertainties in the weights $\mu_i(k)$, can not be considered.

Controller switching:

This method practically represents the class of projection-based methods to active FTC. Similarly to the MM method, its starting point is a set of local linear models that represent the system at some pre-defined (anticipated) fault situations. A controller is then designed for each model so that it can be switched on when the corresponding model best matches the current dynamical behavior of the system. The difference with the MM method is that no mixing of control actions is performed here, but only switching, i.e. only one controller is active at each time instant. In Bošković et al. (1999); Bošković and Mehra (1998); Gopinathan et al. (1998); Lemos et al. (1999); Musgrave et al. (1997) the outputs of the local models are compared to the measured system output, and on the basis of the so-formed residuals decisions are taken about which model best describes the current mode of operation of the system. Recently, an interesting approach was proposed by Yamé and Kinnaert (2003) where the switching is performed based on closed-loop performance monitoring. In Ge and Lin (1996); Mahmoud et al. (2000a) more attention is payed on the design of the bank of controllers in an integrated manner via coupled Ricatti equations by assuming the fault process and the FDD process to be first order Markov processes with given transition probability matrices. There are also many other approaches based on controller switching (Maki et al. 2001; Rato and Lemos 1999; Chang et al. 2001; Médar et al. 2002). The problem of reducing the transients during switching has also been recently considered by Kovácsházy et al. (2001).

A drawback of the approaches based on controller switching is that they can only deal with a limited set of anticipated faults. Advantage is that model uncertainty can easily be considered by means of designing the local controllers robust with respect to it.

Integrated FDD & FTC:

There exist a number of papers that do not consider the problems of FTC and FDD separately, but rather combine them in one framework. For instance, many multiple model control approaches can readily be combined with MM-based FDD schemes like, for instance, the Interacting Multiple Model (IMM) estimator (Zhang and Jiang 1999b). Such approaches are considered in Zhang and Jiang (2001); Maybeck and Stevens (1991); Zhang and Jiang (1999b,a). There are, however, many other integrated FDD & FTC methods, e.g. combination of MM-based FDD methods with control redistribution (Maybeck 1999) or with PID controller (Zhou and Frank 1998), combination of adaptive methods for FDD and FTC (Bošković and Mehra 2003), reconfiguration based on adding a scaled residual signal from the FDD scheme to the nominal control action (Jakubek and Jorgl 2000).

These integrated methods, however, do not consider model uncertainty. Moreover, they are usually developed by means of directly interconnecting an FDD scheme with an FTC scheme, paying little or no attention on possible imprecisions in the FDD information. Striving to overcome these drawbacks, in Chapter 4 of this thesis a method is developed that ensures robustness with respect to both model and FDD uncertainties. It is also shown in the same chapter how the performance of this method can further be improved when the FDD scheme provides not only fault estimates, but also the size of the uncertainty in these estimates.

Model Following:

The model following method is another approach to active FTC. Basically, the method considers a reference model of the form

$$\begin{array}{rcl} x_{k+1}^M & = & A_M x_k^M + B_M r_k, \\ y_k^M & = & x_k^M, \end{array}$$

where r_k is a reference trajectory signal. The goal is to compute matrices K_r and K_x such that the feedback interconnection of the open-loop system (1.9) and the state-feedback control action

$$u_k = K_r r_k + K_x x_k$$

matches the reference model. To this end the reference model and closed-loop system are written in the form

$$\begin{array}{rcl} y_{k+1}^M & = & A_M x_k^M + B_M r_k, \\ y_{k+1} & = & (CA + CBK_x)x_k + CBK_r r_k, \end{array}$$

so that *perfect model following* (PMF) can be achieved by selecting

$$\text{PMF:} \left\{ \begin{array}{rcl} K_x & = & (CB)^{-1}(A_M - CA), \\ K_r & = & (CB)^{-1}B_M, \end{array} \right. \tag{1.14}$$

provided that the system is square (i.e. $\dim(y) = \dim(u)$), and that the inverse of the matrix CB exists. When the exact system matrices (A, B) in (1.14) are unknown, they can be substituted by some estimated values $(\hat{A}, \hat{B})$, resulting in *the indirect (explicit) method* (Bodson and Groszkiewicz 1997). The indirect method provides no guarantees for closed-loop stability, and in addition, the matrix $(C\hat{B})$ may not be invertible. In order to avoid the need for estimating the plant parameters, the *direct (implicit) method* to model following can be used that directly estimates the controller gain matrices K_r and K_x by means of an adaptive scheme. Two approaches to direct model following exist, the output error method and the input error method. For more details on that the reader is referred to Bodson and Groszkiewicz (1997); Morse and Ossman (1990); Gao and Antsaklis (1992); Bošković et al. (2000a); Zhang and Jiang (2002). We note here, that the direct model following method is based on adaptation rules and as such is also a candidate for the group of adaptive control methods.

Similar direct model-following ideas were used in an interesting series of recent publications that deal with multiplicative actuator faults, where both the state-feedback (Tao et al. 2001, 2002b, 2000b) and the output-feedback (Tao et al. 2002a, 2000a; Fei et al. 2003) cases have been considered.

The model following methods have the advantage that they usually do not require FDD scheme. A strong drawback is, however, that they are not applicable to sensor faults. In addition to that these methods do not deal with model uncertainty.

Adaptive Control:

Adaptive control methods form a class of methods that is very suitable for active FTC. Due to their ability to automatically adapt to changes in the system parameters, these methods could be called "self-reconfiguable", i.e. they often don't require the blocks "reconfiguration mechanism" and "FDD" in Figure 1.6. This, however is mostly true for component faults and actuator faults, but not for some sensor faults. If one, for instance, makes use of an adaptive control scheme based on output-feedback design to compensate for sensor faults it will make the faulty measurement (rather than the true signal) track a desired reference signal, and this in turn may even lead to instability. Indeed, in a case of a total sensor failure an adaptive controller may try to increase the control action to make the faulty measured signal equal to the desired value that will not be possible due to the complete failure of the sensor. In such cases an FDD scheme is needed to detect the sensor failure, and a reconfiguration mechanism would have to appropriately reconfigure the adaptive controller. We note here that the direct model following approaches and the MM approaches, discussed above, also belong to the class of adaptive control algorithms. Linear parameter-varying control methods to FTC design (Bennani et al. 1999; Ganguli et al. 2002; Shin et al. 2002) are also members of this class. The approaches developed in Chapter 4 of this thesis also belong to the LPV approaches to FTC. The improvement there is that these methods deal with structured parametric and FDD uncertainty, and that they are applicable to a much wider class of faults as the fault signal is allowed to enter the state-space matrices of the system in any way as long as the matrices remain bounded. Other adaptive methods for FTC can be found in (Dionísio et al. 2003; Jiang et al. 2003; Kececi et al. 2003b,a; Ahmed-Zaid et al. 1991; Bošković et al. 2000b; Ikeda and Shin 1998; Kim et al. 2001a; Siwakosit and Hess 2001; Qu et al. 2001). These, however, do not deal with model uncertainty.

Model Predictive Control:

Model predictive control (MPC) is an industrially relevant control strategy that has received a lot of attention lately. Due to the underlying optimization that needs to be executed at each time instant, it is an attractive method mainly for slower processes such as those encountered in the chemical industry (Kothare et al. 1996). This optimization is based on matching (in the vector 2-norm sense) a prediction of the system output to some desired reference trajectory. The latter is assumed to be known in advance. In addition, MPC features the property that it can handle constraints on the inputs and states of the system in an explicit way by incorporating them into the optimization problem.

As discussed in Astrom et al. (2001), the MPC architecture allows fault-tolerance to be embedded in a relatively easy way by: (a) redefining the constraints to represent certain faults (usually actuator faults), (b) changing the internal model, (c) changing the control objectives to reflect limitations due to the faulty mode of operation. In such a way there is practically no additional optimization that needs to be executed on-line as a consequence of a fault being diagnosed, so that this method can be viewed as having an inherent self-reconfiguration property. However, if state-feedback MPC is used in an interconnection with an observer one should also take care to also reconfigure the observer appropriately in order to achieve fault-tolerant state estimation. For an overview of the work on MPC-based FTC the reader is referred to Maciejowski and Jones (2003); Huzmezan and Maciejowski (1999, 1998a,c,b); Kerrigan and Maciejowski (1999) and the references therein.

With its self-reconfiguration capability the MPC is very suitable and attractive for the purposes of achieving fault-tolerance. Most state-space approaches to MPC are, however, derived under the assumption that the state of the system is measured. In such cases the algorithms can readily be extended to deal with model uncertainties as in Kothare et al. (1996). When the state is not measured, if no uncertainty is present in the model an observer can be designed to provide the missing state information. In the model uncertainty case, however, the separation principle is no longer valid, so that the observer and the state-feedback MPC controller cannot be designed separately. For that reason an approach is proposed in Chapter 5 of this thesis that integrates the design of a Kalman filter and a finite-horizon MPC into one optimization, making it in this way possible to include model uncertainties into the problem. Disadvantage here is the increased computational complexity.

Analysis of FTCS:

Recently, there has been quite some interest in the analysis of FTCS (Mahmoud et al. 2003). The stability of FTCS systems has been studied in different publications in a stochastic framework (Mahmoud et al. 2003, 2001, 1999, 2000b,c, 2002;

Ge and Frank 1995). In this formulation a system of the form

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(\eta(t))u(x(t), \Psi(t), t),\\ u(x(t), \Psi(t), t) &= -K(\Psi(t))x(t),\end{aligned}$$

is considered, where $\eta(t)$ represents actuator fault process, and where $\Psi(t)$ represents the FDD process. For the analysis it is assumed that $\eta(t)$ and $\Psi(t)$ are Markov processes with finite state spaces $S = \{1, 2, \ldots, s\}$ and $R = \{1, 2, \ldots, r\}$, respectively. In this way only a finite set if anticipated actuator faults can be considered. It is further assumed that the transition probabilities of the two Markov processes are given. As discussed in (Mahmoud et al. 2003) it is in practice very difficult to obtain these transition probabilities. For such systems the stochastic stability is analyzed in the presence of noise, uncertainties, and input saturations by means of coupled matrix Riccati equations. Markov models were also used for reliability analysis in some recent publications (Wu 2001a,b; Wu and Patton 2003). The reconfigurability property of systems have also been studied and measures for the level of redundancy have been proposed in Wu et al. (2000b,c); Staroswiecki (2002). Some other works on FTCS analysis can be found in Bonivento et al. (2003a); Shin and Belcasrto (2003); Frei et al. (1999); Gehin and Staroswiecki (1999); Staroswiecki et al. (1999); Yang and Hicks (2002); Izadi-Zamanabadi and Staroswiecki (2000).

Online optimization/redesign:

The approaches based on on-line redesign and on-line optimization are computationally more expensive algorithms. The control re-allocation method, for instance, is an on-line optimization approach (Buffington et al. 1999; Burken et al. 1999; Maybeck 1999; Eberhardt and Ward 1999). This is a strategy that is usually applied in aircraft control for providing actuator fault tolerance, where increased hardware redundancy is present in the effectors. The goal is after a failure of an effector to redistribute/re-allocate its actuation over the remaining effectors, which is achieved by means of on-line optimization. Other methods to FTC design based on on-line optimization can be found in Looze et al. (1985); Dardinier-Maron et al. (1999); Tortora et al. (2002); Wu et al. (2000a); Yang and Stoustrup (2000); Yang and Blanke (2000b); Zhang et al. (2002); Marcos et al. (2003). We note here that the method for robust output-feedback MPC discussed in Chapter 5 of this thesis, classified above into the MPC methods, might also be viewed as a member of the class of online optimization methods.

Fault-Tolerant Measurement/State Estimation:

Providing fault-tolerant state estimation is also an important issue when the controller is dependent on the state estimates provided by an observer. In such cases sensor, actuator and component faults result in incorrect state-estimates that are being fed to the controller. This may result in degraded performance and/or instability. Reconstruction of the state of the system from faulty measurements has been considered in Theilliol et al. (2001). For output-feedback controllers, the sensor fault masking method (Wu et al. 2003) is an example of a technique for providing increased fault-tolerance in the measurements by substituting the missing measurements by estimates. Similar idea has been used in Ponsart et al. (2001).

Neuro-Fuzzy:

Methods based on neural networks and fuzzy logic have also received attention by the FTC society. These methods have the advantage that they are applicable to FTC for nonlinear systems that are usually modelled by means of a Takagi-Sugeno fuzzy model as, e.g., in Diao and Passino (2001). The learning capabilities of these methods make it possible to adapt the model and the controller after the occurrence of a fault in the system. For more details on neuro-fuzzy methods for FTC, the interested reader is referred to Fray et al. (2003); Ballé et al. (1998); Chen and Narendra (2001); Diao and Passino (2001, 2002); Lopez-Toribio et al. (1999); Marcu et al. (1999); Schram et al. (1998); Wise et al. (1999); Wu (1997a); Yen (1994); Yen and Ho (2000); Zhang et al. (2002).

Application oriented:

There are also many papers that are focused on a particular application. Some of them are Askari et al. (1999); Battaini and Dyke (1998); Blanke et al. (1998); Bonivento et al. (2003b, 2001b,a); Ho and Yen (2001); Jonckheere and Lohsoonthorn (2000); Kim et al. (2001b); Li et al. (1999); Piug and Quevedo (2001); Mohamed et al. (1997); Podder and Surkar (2001); Schdeier and Frank (1999); Somov et al. (2002); Visinski et al. (1995); Liu et al. (2000); Gaspar et al. (2003). There are, however, may others. For more references see Zhang and Jiang (2003); Astrom et al. (2001).

Benchmark problems have also been proposed for testing and demonstrating the capabilities of different approaches for FDD and FTC. The most popular are *the ship propulsion system* benchmark (Izadi-Zamanabadi and Blanke 1999), *the diesel actuator benchmark* model (Blanke et al. 1995), the *three-tank system benchmark* (Heiming and Lunze 1999; Astrom et al. 2001).

1.6 Scope of the thesis

The overview from the previous section shows that there are numerous publications in the field of FTC. Still there are certain topics that have not yet received the required attention. As argued in the overview papers of Zhang and Jiang (2003) and Patton (1997), one of the most important research topics that still need to be considered in a FTCS design are the following:

(P1) how to deal with *model and FDD uncertainties*, and

(P2) how to deal with *nonlinear systems*.

As discussed in Section 1.4, real physical control *systems are always nonlinear*. Moreover, it is not always the case that a linear model can be built up that sufficiently accurately describes the dynamic behavior of the system in a wide

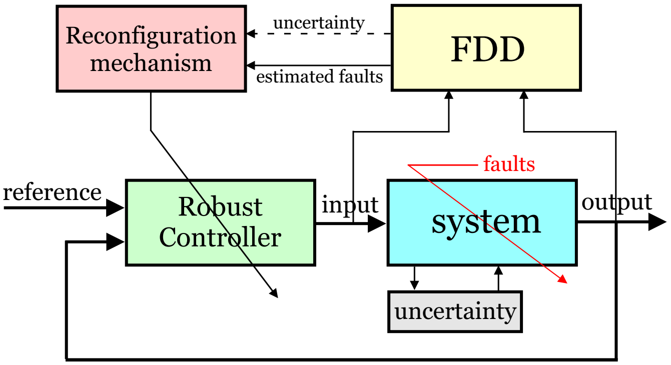


Figure 1.7: A realistic FTCS considers both plant-model mismatch and uncertainty in the FDD.

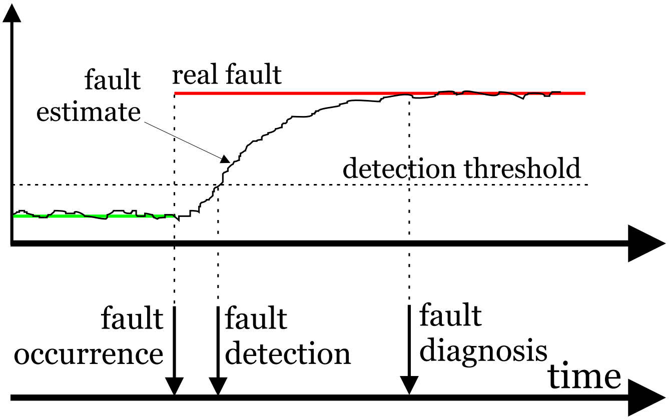


Figure 1.8: Visualization of the delay in the FDD process.

range of operating conditions. Such linear models are usually only valid locally. They are obtained either by means of linearization of the nonlinear model around a given operating point, or by means of data-driven model identification. Even the resulting local linear model, however, could be imprecise due to not exactly known (or time-varying) values of some physical parameters, linearization errors, etc. The resulting mismatch between the model and real system is referred to as *model uncertainty*. Clearly, there is always some discrepancy beween model and system, so that *model uncertainty is always present*! It is therefore important in the development of methods to FTC, when aimed at a general class of systems, that model uncertainty is considered, i.e. that the controller is made robust with respect to these uncertainties (see Figure 1.7).

When faults occur in an active FDD-based FTCS these need to be first detected and diagnosed so that the controller could subsequently be reconfigured.

Since no FDD scheme is perfect, another important aspect needs to be considered by the reconfiguration mechanism, namely how to deal with delays in the detection and diagnosis. Figure 1.8 visualizes a detection process where the true fault signal changes abruptly at the time instant of the fault occurrence. After the fault occurrence it takes some time for a real-life FDD process to detect the fault due to the fact that it needs to collect sufficient input-output data from the process in order to make such a decision. In the visualization on Figure 1.8 it is assumed that the FDD scheme is based on the direct estimation of a fault signal, and that a fault detection flag is triggered when the estimated signal passes a used-specified threshold. In the time interval between the fault occurrence and the fault detection the reconfiguration mechanism is unaware of the fault and, therefore, cannot initialize any reconfiguration of the controller. At the time of the fault detection the reconfiguration mechanism becomes aware of the occurrence of the fault but it has no information about the exact magnitude. Finally, some time after the fault detection comes the fault diagnosis so that a final reconfiguration can take place. The final estimate of the fault, however, can also not be expected to perfectly match the true value of the fault due to measurement noise, model uncertainty, etc. Therefore, it is important that such imperfections in terms of uncertainties and delays in the fault estimates are considered in the design of the FTC.

Additionally, it might be useful for the reconfiguration mechanism to have an idea about the size of the uncertainty in the fault estimates, should the FDD scheme be capable of providing it. In fact the size of the FDD uncertainty is in practice often time-varying. Indeed, as it can be seen from Figure 1.8, immediately after the fault occurrence the fault estimates are rather imprecise due to the lack of sufficient input-output measurement data. Gradually, as more data becomes available from the system the estimates are refined, i.e. they become more accurate and the uncertainty size decreases until it reaches its minimum around the time of the fault diagnosis. This idea is pursued in the results in Chapter 4 in this thesis.

The results presented in this thesis are mainly intended as an attempt to develop methods for FTC design with a clear focus on problems (P1) and (P2) discussed in this section.

1.7 Outline of the thesis

The dynamics of a real-life physical system can be represented in state-space in the following general form

$$S(p_k) : \left\{ \begin{array}{rcl} x_{k+1} & = & f(x_k, u_k, p_k), \\ y_k & = & h(x_k, u_k, p_k), \\ x_0 & = & \hat{x}_0, \end{array} \right. \tag{1.15}$$

where the vector $x_k \in \mathcal{X} \subseteq \mathbb{R}^n$ represents the state of the system $S(p_k)$, $u_k \in \mathcal{U} \subseteq \mathbb{R}^{m+n_\xi}$ represents the inputs to the system, $y_k \in \mathbb{R}^{p+n_z}$ denotes the outputs of the system. At each time instant t the system $S(p_k)$ is parametrized by a (possibly unknown) parameter vector $p_k \in \mathcal{P} \subseteq \mathbb{R}^{n_p}$. The vector p_k may represent uncertain physical parameters in the system or system faults.

Nonlinear models of systems are in general inconvenient to work with due to their complexity and due to the lack of a well-developed theory for analysis and synthesis for general nonlinear models. The usual strategy to deal with them is either by approximating them with more convenient models (e.g. by means of blending of a set of local linear models as in the multi-model and in the Fuzzy control theories) or by assuming certain structure (e.g. bilinear systems, Hammerstein-Wiener systems, linearity in the input, etc.).

In the multiple model approach the state space $\mathcal{X}$ is divided into N representative and disjoint regions $\mathcal{X}_i$, with $\bigcup_{i=1}^{N} \mathcal{X}_i \equiv \mathcal{X}$, and in each region a point $(x^{(i)}, u^{(i)}) \in \mathcal{X}_i \times \mathcal{U}$ is chosen around which the nonlinear system $S(p_k)$ is approximated by a linear model. Under the assumption that $f(\cdot), g(\cdot) \in C^1$, the local linear approximation $M_i(p_k)$ of the system $S(p_k)$ within the open ball neighborhood

$$\mathcal{B}(x^{(i)}, u^{(i)}) \doteq \left\{ (x,u) \in \mathcal{X} \times \mathcal{U} : \left\| \begin{bmatrix} x - x^{(i)} \\ u - u^{(i)} \end{bmatrix} \right\|_2 < \epsilon \right\},$$

is called the p_k-parametrized local linear model

$$M_i(p_k) : \begin{cases} x_{k+1}^{(i)} &= A_i(p_k)x_k^{(i)} + B_i(p_k)u_k + b_i(p_k), \\ y_k^{(i)} &= C_i(p_k)x_k^{(i)} + D_i(p_k)u_k + c_i(p_k), \\ x_0^{(i)} &= \bar{x}_0, \end{cases}$$

with

$$\begin{aligned} &A_i(p_k) \doteq \partial_x f(x^{(i)}, u^{(i)}, p_k), \;\; B_i(p_k) \doteq \partial_u f(x^{(i)}, u^{(i)}, p_k) \\ &C_i(p_k) \doteq \partial_x h(x^{(i)}, u^{(i)}, p_k), \;\; D_i(p_k) \doteq \partial_u h(x^{(i)}, u^{(i)}, p_k) \\ &b_i(p_k) \doteq f(x^{(i)}, u^{(i)}, p_k) - A(p_k)x^{(i)} - B(p_k)u^{(i)} \\ &c_i(p_k) \doteq h(x^{(i)}, u^{(i)}, p_k) - C(p_k)x^{(i)} - D(p_k)u^{(i)}, \end{aligned}$$

where $\partial_x f$, $\partial_u f$, $\partial_x h$, and $\partial_u h$ represent the partial derivaties of the functions $f(\cdot)$ and $h(\cdot)$ with respect to the vectors x and u.

Each local linear model $M_i(p_k)$ describes the behavior of the nonlinear system within one regime $\mathcal{X}_i$. A global approximation can be then formed by interpolating the local models using smooth interpolation functions $\phi_i(x_k, u_k, p_k) > 0$ that depend on the operating point (x_k, u_k) as well as on the parameter vector p_k, i.e.

$$\hat{y}_k = \sum_{i=1}^{N} \mu_k^{(i)} y_k^{(i)}, \text{ with } \mu_k^{(i)} = \frac{\phi_i(x_k, u_k, p_k)}{\sum_{i=1}^{N} \phi_i(x_k, u_k, p_k)}. \qquad (1.16)$$

Such approximations are widely used in the literature (see, for instance, Johansen and Foss (1995)). In fact it is shown in Johansen (1994) that, under certain smoothness properties, the nonlinear system $S(p_k)$ can be approximated to any desired accuracy on a compact subset of the state and input spaces by means of the representation (1.16) for a sufficiently large number of local models.

The multiple model representation (1.16) is both intuitive and attractive, and is very much related to the Takagi-Sugeno fuzzy model, where the weights $\mu_k^{(i)}$ in the linear combination of the local outputs are called degrees of membership.

Suppose that the parameter vector p_k is formed by two vectors, $\delta_k \in \mathbf{\Delta} \subseteq \mathbb{R}^{n_\delta}$ and $f_k \in \mathcal{F} \subseteq \mathbb{R}^{n_f}$, so that

$$p_k = \begin{bmatrix} \delta_k \\ f_k \end{bmatrix}, \tag{1.17}$$

where the vector δ_k is used to represent unknown, time-varying physical parameters of the system, and where the vector f_k represents faults in the system. For consistency in the dimensions it should hold that $n_\delta + n_f = n_p$. While both vectors are unknown, the fault vector f_k is assumed to be estimated by an FDD scheme, and its estimate is denoted here as $\hat{f}_k$. Let $\delta_0 \in \mathbf{\Delta}$ represent the nominal values of the uncertain parameters, and $f_0 \in \mathcal{F}$ represent the fault-free mode of operation.

Let us collect all local models $M_i(p_k)$ into a model set

$$\mathcal{M}(p_k) \doteq \{M_1(p_k), M_2(p_k), \ldots, M_N(p_k)\}, \tag{1.18}$$

and consider only one element of the set $\mathcal{M}(p_k)$ which in view of (1.17) is denoted as $M(\delta, f)$. For simplicity in notations, the time symbol is omitted in $M(\delta, f)$.

This thesis is focused on the following topics with the clear intention to address the problems (P1) and (P2) defined on page 21:

- *passive robust FTC:* design one controller K that achieves some desired performance for the model $M(\delta, f)$ for all possible uncertainties $\delta_k \in \mathbf{\Delta}$ and faults $f_k \in \mathcal{F}$,
- *active robust FTC:* given an estimate $\hat{f}$ of the fault vector f by some FDD scheme, design controller $K(\hat{f})$ that achieves some desired performance for the model $M(\delta, f)$ for all possible uncertainties $\delta_k \in \mathbf{\Delta}$ and faults $f_k \in \mathcal{F}$,
- *active MM-based FTC:* design a controller that achieves some desired performance for the nonlinear system $S(p_k)$ for some fixed $\delta_k = \delta_0 \in \mathbf{\Delta}$ (i.e. in the case of no uncertainty) and for all possible faults $f_k \in \mathcal{F}$.

A natural continuation of this research activity is to combine the MM-based representation of the nonlinear system with the passive and active approaches to FTC in an attempt to deal with nonlinear systems with uncertainty as in the (1.15). This is a topic for future research to be discussed in the concluding chapter of this thesis.

We will next provide some more technical insight in the above-defined objectives. Suppose that a continuous map, that we call the *performance index*, is given

$$J: \mathcal{R}^{n_z \times n_\xi} \mapsto \mathbb{R}^+,$$

such that $J(M) = \infty$ for any $M \notin \mathcal{RH}_\infty$, where $\mathcal{R}^{n_z \times n_\xi}$ denotes the set of rational transfer $n_z \times n_\xi$ matrices, and $\mathcal{RH}_\infty$ denotes the set of stable real rational transfer matrices. Let $M(\delta, f) \in \mathcal{R}^{(p+n_z)\times(m+n_\xi)}$ be partitioned as follows

$$M(\delta, f) = \begin{bmatrix} M_{11}(\delta, f) & M_{12}(\delta, f) \\ M_{21}(\delta, f) & M_{22}(\delta, f) \end{bmatrix},$$

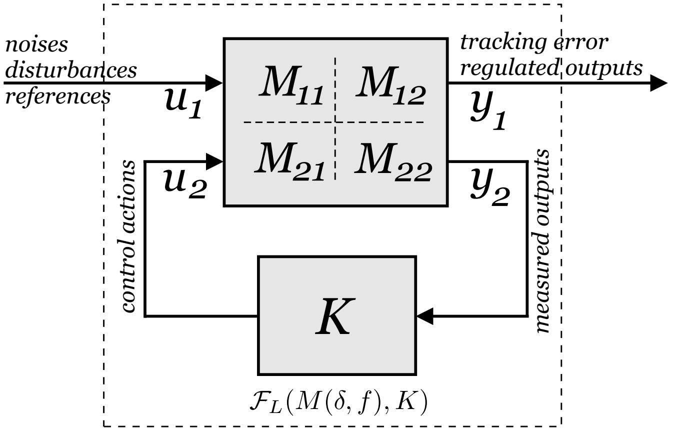


Figure 1.9: Partitioning of the model $M(\delta,f)$ and forming the closed-loop with the controller K.

were, as depicted on Figure 1.9, the subsystem $M_{22}(\delta,f) \in \mathcal{R}^{p\times m}$ gives the relationships between the control actions and the measured output signals, and the subsystem $M_{11}(\delta,f) \in \mathcal{R}^{n_z \times n_\xi}$ describes the relationships between all exogenous inputs (such as noises, disturbances, reference signals) and the regulated (controlled) outputs that are related to the performance of the system (e.g. tracking errors). The feedback interconnection of the model $M(\delta,f)$ with some controller $K \in \mathcal{R}^{m\times p}$ is represented by the lower linear fractional transformation

$$\mathcal{F}_L(M(\delta,f),K) \doteq M_{11}(\delta,f) + M_{12}(\delta,f)K(I - M_{22}(\delta,f)K)^{-1}M_{21}(\delta,f).$$

For a fixed controller K, the performance of the resulting closed-loop is therefore represented by $J(\mathcal{F}_L(M(\delta,f),K))$.

Passive Fault-Tolerant Control

The passive robust FTC problem is then defined as the following optimization problem

Passive FTC:

$$K_P = \arg\min_{K} \sup_{\substack{\delta\in\boldsymbol{\Delta}\\ f\in\mathcal{F}}} J(\mathcal{F}_L(M(\delta,f),K)). \tag{1.19}$$

In this way a controller needs to be found that minimizes the worst-case performance over all possible values for the uncertainty vector δ and the fault vector f. This problem is considered in Chapters 2 and 3 (see Figure 1.10) where methods are developed for robust controller design in the presence of structured uncertainty.

In practise, two main difficulties arise with the optimization problem (1.19), both being related to convexity. In the case when the state vector x_k is directly measured (or, equivalently, when $y_k = x_k$), the optimization problem (1.19) is

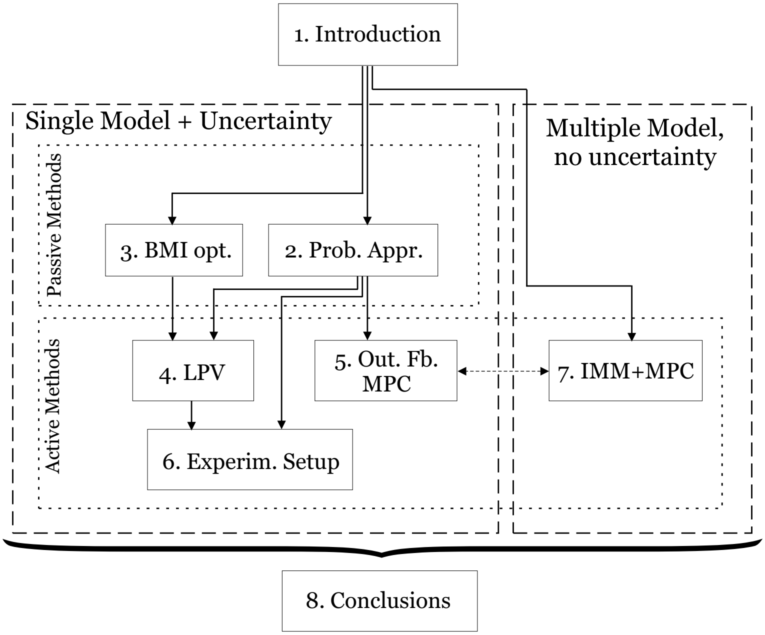


Figure 1.10: Organization of the thesis.

convex in the controller parameters for many standard performance indexes (e.g. $J(\cdot) = \|\cdot\|_2$, $J(\cdot) = \|\cdot\|_\infty$, etc.) provided that the set $\{M(\delta, f) : \delta \in \mathbf{\Delta},\ f \in \mathcal{F}\}$ is a convex polytope. In such cases (1.19) can be represented as a linear matrix inequality (LMI) optimization problem, for which there exist very efficient and computationally fast solvers. If $M(\delta, f)$ is not a convex set, however, the original problem (1.19) is also nonconvex and the LMI solvers cannot be used. A "brute force" way to deal with this problem is to embed the set $M(\delta, f)$ into a convex set. This, however, introduces unnecessary conservatism that for some problems might be unacceptable or undesirable.

In order to deal with such problems a probabilistic design approach is proposed in **Chapter 2** that is basically applicable for any bounded set $M(\delta, f)$, as long as (1.19) can be rewritten as a robust LMI optimization problem (as for most state-feedback controller design problems). This method is basically an iterative algorithm that at each iteration generates a random uncertainty sample for which an ellipsoid is computed with the properties that (a) it contains the solution set (the set of all solutions to the robust LMI problem), (b) it has a smaller volume than the ellipsoid at the previous iteration. The approach is proved to converge to the solution set in a finite number of iterations with probability one.

In the output-feedback case the probabilistic method of Chapter 2 cannot be directly applied because the optimization problem (1.19) cannot be rewritten as a robust LMI optimization problem. The reason for that is that the output-feedback problem in the presence of uncertainty is a bilinear matrix inequality (BMI) problem, and BMI problems are not convex. Actually, such problems have been shown to be NP-hard meaning that they cannot be expected to have polynomial time complexity. A local BMI optimization approach is developed in **Chapter 3** that is guaranteed to converge to a local optimum of the cost function $J(\mathcal{F}_L(M(\delta, f), K))$.

Active Fault-Tolerant Control

Whenever an estimate $\hat{f}$ of the fault vector f is provided by some FDD scheme, and if the imprecision in this estimate is described by and additional uncertainty $\Delta_f \in \mathbf{\Delta}_f$ so that $f = (I + \Delta_f)\hat{f}$, the active robust FTC can be defined as the problem:

$$\begin{aligned}&\text{given } f = (I + \Delta_f)\hat{f}, \text{ evaluate}\\ &\tilde{K}_A(\hat{f}) = \arg\min_{K(\hat{f})} \sup_{\substack{\delta \in \mathbf{\Delta} \\ \Delta_f \in \mathbf{\Delta}_f}} J(\mathcal{F}_L(M(\delta, f), K(\hat{f}))).\end{aligned} \tag{1.20}$$

The resulting controller would, in this way, be scheduled by the fault estimate $\hat{f}$ and will be robust with respect to uncertainties both in the model $M(\delta, f)$ and in the estimate of f. Clearly, the way in which the scheduling parameter $\hat{f}$ enters the controller needs to be assumed before one could proceed with the optimization.

Above, Δ_f represents the FDD uncertainty that, as already discussed, usually increases after the occurrence of a fault, and then subsequently decreases as the FDD scheme refines the estimate based on the availability of more input-output data from the impaired system. As a result the "maximal uncertainty" is only active for some relatively short periods of time compared with the duration of the operation of the system. Therefore, assuming a maximal uncertainty size during the complete operation might be overly conservative since the robust controller practically trades off performance for increased robustness to uncertainties. Hence, it is interesting to allow the controller to be able to deal with time-varying size of the FDD uncertainty. To this end, however, the FDD scheme should be capable of providing not only an estimate of the fault but also an upper bound on the size of the uncertainty in this estimate (see the dashed line in Figure 1.7 on page 22). The size of the FDD uncertainty might, for instance, be represented by a scalar $\gamma_f(k)$ such that $f_k = (I + \gamma_f(k)\bar{\Delta}_f)\hat{f}_k$ with $\|\bar{\Delta}_f\|_2 \leq 1$. In this way the size of the uncertainty set is allowed to vary with time. In fact $\gamma_f(k)$ might be a vector to make it possible to assign different uncertainty sizes on the different entries of the fault vector f_k. Therefore, provided that the FDD scheme produces $(\hat{f}_k, \gamma_f(k))$ at each time instant, the achievable performance in (1.20) may further be improved by computing the controller by solving the following optimization problem

Active FTC:

$$\begin{aligned}&\text{given } f = (I + \gamma_f\bar{\Delta}_f)\hat{f}, \text{ evaluate}\\ &K_A(\hat{f}, \gamma_f) = \arg\min_{K(\hat{f}, \gamma_f)} \sup_{\substack{\delta \in \mathbf{\Delta} \\ \bar{\Delta}_f \in \bar{\mathbf{\Delta}}_f \\ \underline{\gamma_f} \leq \gamma_f \leq \bar{\gamma}_f}} J(\mathcal{F}_L(M(\delta, f), K(\hat{f}, \gamma_f))),\end{aligned} \tag{1.21}$$

where $\bar{\mathbf{\Delta}}_f = \{\Delta \in \mathbf{\Delta}_f : \|\Delta\| \leq 1\}$, and where the vectors $\{\underline{\gamma_f}, \bar{\gamma}_f\}$, assumed known *a-priori*, define a lower and an upper bound on the possible uncertainty sizes. In this way methods can be developed for the design of robust active FTC for one uncertain local model $M(\delta, f)$. The robust active FTC design problem is considered in Chapters 4 and 5. The approach from Chapter 4 is subsequently illustrated on a experimental setup with a brushless DC motor (BDCM) in Chapter 6 (see Figure 1.10 on page 27).

Chapter 4 is focused on the development robust active FTC approaches based on LPV controller design. Two approaches are proposed. The first LPV approach can deal with multiplicative sensor and actuator faults and consists of the online design of a set of parameter-varying robust output-feedback controllers, in which the only scheduling parameter is the size $\gamma_f(k)$ of the FDI uncertainty. A set of such predesigned LPV controllers is built up, each controller corresponding to a suitably defined fault scenario. After a fault has been diagnosed the reconfigured controller is taken as a scaled version of one of the predesigned controllers. Although a finite set of controllers are initially designed, the reconfiguration scheme deals with an arbitrary combination of multiplicative sensor and actuator faults as long as the system remains stabilizable and detectable. This approach is based on LMIs that are derived by neglecting the structure of the uncertainty. In order to circumvent the resulting conservatism, another approach is proposed by making use of the probabilistic method developed in Chapter 2. This second approach to robust output-feedback FTC has the following advantages: (a) it is scheduled by both the fault estimates $\hat{f}_k$ and the size $\gamma_f(k)$ of its uncertainty, (b) it deals with structured uncertainty, (c) it is applicable to not only sensor and actuator faults, but also to component faults. A disadvantage is that this controller is originally developed for the state-feedback case due to the non-convexity of the output-feedback problem. However, the method is further extended by means of a two-step procedure, borrowed from the BMI approach in Chapter 3, that allows to consider the output-feedback case as well.

In **Chapter 5** a finite-horizon output-feedback MPC design approach is presented that is robust with respect to model and FDD uncertainties. The approach consists in a combination of a Kalman filter and a finite-horizon MPC into one min-max (worst-case) optimization problem, that is solved at each iteration by making use of the probabilistic method of Chapter 2. This method has the advantage that it deals with the robust output-feedback problem directly without having to solve BMI optimization problems. A disadvantage is its computational demand and the lack of guaranteed closed-loop stability.

We note here that the LPV controllers are very suitable for online implementation due to the fact that the design is performed completely off-line. This results in limited on-line computations for controller re-configuration after the occurrence of a fault. The LPV approach based on the probabilistic design is tested in **Chapter 6** on a real-life experimental setup consisting of a brushless DC motor. The FTC approach is combined there with an FDD scheme for the detection and estimation of parameter and sensor faults.

Dealing with Nonlinear Systems

The passive and active approaches to FTC discussed above are focused on only one local linear model in the presence of uncertainty in both the model description and in the FDD scheme. The question of how to extend this to deal with the complete multiple model representation of a nonlinear system is much more difficult and is in this thesis only partially addressed. Specifically, in the case of no uncertainty (i.e. $\delta_k = \delta_0 = const$) a method is developed in **Chapter 7** (see Figure 1.10 on page 27) that can be used for control of nonlinear systems represented by multiple local models. The starting point is the construction of a model set $\mathcal{M}$ that contains either local linear approximations of a nonlinear system or models representing faulty modes of operation of a (linear) system. In this way the elements of the model set are time-invariant, which is a special case of the more general representation in (1.18) on page 25. The method is a combination of a multiple model estimator that provides local and global state-estimates as well as estimates of the weights $\hat{\mu}_i$. They are then used to parametrize an MPC. The multiple model estimator consists of a bank of Kalman filters, one for each local model. The Kalman filters are independently designed from the MPC. In the case when uncertainty is present in the system (and, therefore, also in the local models), however, the design of the state observer and the controller can no longer be executed separately due to the fact that the well known separation principle no longer holds. It therefore remains for future research to investigate how to deal with uncertainties in the elements of the model set $\mathcal{M}$.

1.8 Organization of the thesis

This thesis is organized as follows.

Chapters 2 and 3 propose methods for achieving robustness with respect to system faults and model uncertainties by means of passive FTC. The state-feedback case is first considered in a probabilistic design framework in Chapter 2 for a very general class of model uncertainties and faults. In Chapter 3 the output-feedback case is considered by proposing an approach based on nonlinear local BMI optimization for systems with polytopic uncertainty.

Chapters 4–6 present algorithms for active FTC for linear systems in the presence of model and FDD uncertainty. In Chapter 4 approaches are developed for the design of linear parameter-varying FTC that are scheduled by both the fault estimates as well as the sizes of their uncertainties. In Chapter 5 a finite-horizon output-feedback MPC design approach is presented that consists in a combination of a Kalman filter and a finite-horizon MPC into one robust least squares optimization problem, in this way circumventing the need for solving nonconvex BMI problems. The methods from Chapter 4 are tested in Chapter 6 on a real-life experimental setup consisting of a brushless DC motor.

In Chapter 7 an approach is presented that can be used for control of nonlinear systems represented by multiple local models in the case when no uncertainty is present into the model description. Finally, Chapter 8 gives the conclusions and recommendations.

The relations between the chapters are visualized in Figure 1.10 on page 27.

1.9 Contributions

The results presented in this thesis have been published or submitted for publication elsewhere. Each subsequent chapter consist of an adapted version of one or more such publications. In this setup every chapter might be regarded as stand-alone although there is some relationship between the chapters as discussed in Section 1.7. An attempt is made to keep the notation consistent throughout the thesis; still to prevent any confusion, the notation is sometimes explained at the beginning of the chapter. All references are provided at the end of the thesis. A summary of the contributions and their relations with the chapters in this thesis is provided below.

Chapter 2 The probabilistic ellipsoid algorithm for solving robust LMI problems has been published in

> S. Kanev, B. De Schutter and M. Verhaegen,
> An Ellipsoid Algorithm for Probabilistic Robust Controller Design,
> *Systems & Control Letters*, **49(5)**, 2003, pp. 365–375.
>
> S. Kanev, B. De Schutter and M. Verhaegen,
> The Ellipsoid Algorithm for Probabilistic Robust Controller Design,
> *Proceedings of the 41th IEEE Conference on Decision and Control (CDC'02)*,
> Las Vegas, Nevada, USA, 2002.

Chapter 3 This chapter presents the BMI optimization algorithm for robust output-feedback controller design for systems with polytopic uncertainties. It is based on the following publications:

> S. Kanev, C. Scherer, M. Verhaegen and B. De Schutter,
> Robust Output-Feedback Controller Design via Local BMI Optimization,
> *accepted for publication in Automatica*, 2003.
>
> S. Kanev, C. Scherer, M. Verhaegen and B. De Schutter,
> A BMI Optimization Approach to Robust Output-Feedback Control,
> *to appear in Proceedings of the 41th IEEE Conference on Decision and Control (CDC'03)*,
> Maui, Hawaii, USA, 2003.

Chapter 4 The LPV-based approaches to robust active FTC, presented in this chapter, are based on

> S. Kanev and M. Verhaegen,
> Controller Reconfiguration in the Presence of Uncertainty in the FDI,
> *Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003)*, Washington, D.C., USA, 2003.
>
> S. Kanev and M. Verhaegen,
> Combined FDD and Robust Active FTC for a Brushless DC Motor,
> *submitted to Control Engineering Practice*, 2003.

Chapter 5 The approach to robust output-feedback MPC from this chapter can be found in

> S. Kanev and M. Verhaegen,
> Robust Output-Feedback Integral MPC: A Probabilistic Approach,
> *submitted to Automatica*, 2003.

S. Kanev and M. Verhaegen,
Robust Output-Feedback Integral MPC: A Probabilistic Approach,
to appear in Proceedings of the 41th IEEE Conference on Decision and Control (CDC'03),
Maui, Hawaii, USA, 2003.

Chapter 6 This chapter presents experimental results obtained on the BMDC experimental setup. It is based on the following application-oriented paper

S. Kanev and M. Verhaegen,
Combined FDD and Robust Active FTC for a Brushless DC Motor,
submitted to Control Engineering Practise, 2003.

Chapter 7 The multiple model approach based on combination of the IMM estimator and an MPC presented in this chapter has appeared in

S. Kanev and M. Verhaegen,
Controller Reconfiguration for Non-Linear Systems,
Control Engineering Practice, **8(11)**, 2000, pp. 1223–1235.

In addition, the following publications were also written during my period as a Ph.D. student:

S. Kanev and M. Verhaegen,
A Bank of Reconfigurable LQG Controllers for Linear Systems Subjected to Failures,
Proceedings of the 39th IEEE Conference on Decision and Control (CDC'00), Sydney, Australia, 2000.

S. Kanev and M. Verhaegen and G. Nijsse,
A Method for the Design of Fault-Tolerant Systems in Case of Sensor and Actuator Faults,
Proceedings of the 6th European Control Conference (ECC'01), Porto, Portugal, 2001.

S. Kanev and M. Verhaegen,
An Approach to the Isolation of Sensor and Actuator Faults Based on Subspace Identification,
ESA Workshop on "On-Board Autonomy", Noordwijk, The Netherlands, 2001.

S. Kanev and M. Verhaegen,
Reconfigurable Robust Fault-Tolerant Control and State Estimation,
Proceedings of the 15th Triennial World Congress of IFAC (b'02), Barcelona, Spain, 2002.

S. Mesic, V. Verdult, M. Verhaegen and S. Kanev,
Estimation and Robustness Analysis of Actuator Faults Based on Kalman Filtering,
Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003), Washington, D.C., USA, 2003.

V. Verdult, S. Kanev, J. Breeman and M. Verhaegen,
Estimating Multiple Sensor and Actuator Scaling Faults Using Subspace Identification,
Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003), Washington, D.C., USA, 2003.

2

Probabilistic Approach to Passive State-Feedback FTC

In the introductory Chapter 1 there were two main classes of FTC approaches that were discussed, namely passive and active methods. Passive approaches are off-line methods to FTC that are based on robust controller design algorithms, i.e. a controller needs to be designed that is insensitive to some preselected class of anticipated system faults, viewed as uncertainties. Such passive FTC methods are suitable in the time interval between the detection of a fault and its diagnosis, or in cases when no FDD scheme is present. After the fault has been diagnosed, controller reconfiguration can take place to further improve the performance of the faulty closed-loop system.

In this chapter a new probabilistic approach is proposed that is applicable to any robust controller/filter design problem that is representable as an LMI problem. Given an initial ellipsoid that contains the solution set, the approach proposed here iteratively generates a sequence of ellipsoids with decreasing volumes, all containing the solution set. A method for finding an initial ellipsoid is also given. The proposed approach is illustrated on a real-life diesel actuator benchmark model with real parametric uncertainty, for which a $\mathcal{H}_2$ robust state-feedback controller is designed.

2.1 Introduction

Recently, a new approach for probabilistic design of LQ regulators was proposed in the literature (Polyak and Tempo 2001), to which we will refer to as the *Subgradient Iteration Algorithm* (SIA), which was later on extended to deal with general robust LMIs (Calafiore and Polyak 2001). The main advantage of this approach over the existing deterministic approaches to robust controller design is that it can handle very general uncertainty structures, where the uncertainty can enter the system in any, possibly non-linear, fashion. In addition to that, this approach does not need to solve simultaneously a number of LMIs, whose dimension grows exponentially with the number of uncertain parameters, but rather solves one LMI at each iteration. This turns out to be a very powerful feature when one observes that even for ten real uncertain parameters most of the existing LMI solvers will be unable to handle the resulting number of LMIs. For an overview of the literature on probabilistic design the reader is referred to (Calafiore and Polyak 2001; Polyak and Tempo 2001; Stengel and Ray 1991; Tempo and Dabbene 2001; Ugrinovskii 2001; Vidyasagar 1998; Öhrn et al. 1995; Chen and Zhou 1998; Fujisaki et al. 2001), and the references therein.

While enjoying these nice properties, the major drawback of the SIA is that the radius of a ball contained in the solution set (the set of all feasible solutions to the problem) is required to be known *a-priori*. This radius is used at each iteration of the SIA to compute the size of the step which will be made in the direction of the anti-gradient of a suitably defined convex function. It will be shown later in this chapter that not knowing such a radius r can result in the SIA failing to find a feasible solution. Knowing r, on the other hand, guarantees that the algorithm will terminate in a feasible solution in a finite number of iterations with probability one, provided that the solution set has a non-empty interior (Polyak and Tempo 2001; Calafiore and Polyak 2001). The purpose of this chapter is to develop a new probabilistic approach that no longer necessitates the knowledge of r, while keeping the above-mentioned advantages and the convergence property of SIA.

To circumvent the lack of knowledge of r, it is proposed in (Calafiore and Polyak 2001; Kushner and Yin 1997) that one can substitute this number with a sequence $\{\epsilon_s\}$ such that $\epsilon_s > 0$, $\epsilon \to 0$ and $\sum_{s=0}^{\infty} \epsilon_s = \infty$. While this indeed releases the assumption that the radius r is known, it increases the number of iterations necessary to arrive at a feasible solution. In addition to that the choice of an appropriate sequence $\{\epsilon_s\}$ remains an open question.

An interesting result concerning the algorithm in (Calafiore and Polyak 2001) appeared recently in (Oishi and Kimura 2001), where it is shown that the expected time to achieve a solution is infinite. In (Oishi and Kimura 2001) the authors also propose a slight modification of the approach from (Calafiore and Polyak 2001) that results in an algorithm with finite expected achievement time. Yet, this modified algorithm suffers from the “curse of dimensionality”, i.e. the expected achievement time grows (faster than) exponentially with the number of uncertain parameters.

The approach proposed in this chapter is based on the *Ellipsoid Algorithm* (EA). The algorithm can be used for finding exact or approximate solutions to

LMI optimization problems, like those arising from many (robust) controller and filter design problems. The uncertainty Δ is assumed to be bounded in the structured uncertainty set $\mathbf{\Delta}$, and to be coupled with a probability density function $f_{\mathbf{\Delta}}(\Delta)$. It is further assumed that it is possible to generate samples of Δ according to $f_{\mathbf{\Delta}}(\Delta)$. The interested reader is referred to (Calafiore et al. 2000) for more details on the available algorithms for uncertainty generation. Then, similarly to the SIA, at each iteration of the EA two steps are performed. In the first step a random uncertainty sample $\Delta^{(i)} \in \mathbf{\Delta}$ is generated according to the given probability density function $f_{\mathbf{\Delta}}(\Delta)$. With this generated uncertainty a suitably defined convex function is parametrized so that at the second step of the algorithm an ellipsoid is computed, in which the solution set is guaranteed to lie. The EA thus produces a sequence of ellipsoids with decreasing volumes, all containing the solution set. Using some existing facts, and provided that the solution set has a non-empty interior, it will be established that this algorithm converges to a feasible solution in a finite number of iterations with probability one. To initialize the algorithm, a method is presented for obtaining an initial ellipsoid that contains the solution set. It is also shown that even if the solution set has a zero volume, the EA converges to the solution set when the iteration number tends to infinity, a property not possessed by the SIA.

The remaining part of the chapter is organized as follows. In the next Section the problem is formulated, and the SIA is summarized. In Section 2.3 the EA is developed and its convergence is established. In Section 2.4 a possible method for finding an initial ellipsoid containing the solution set is presented. The complete EA method is illustrated in Section 2.6 on the design of a robust $\mathcal{H}_2$ state-feedback controller for a real-life diesel actuator benchmark model, taken from (Blanke et al. 1995). Finally, Section 2.7 concludes the chapter.

2.2 Preliminaries

2.2.1 Notation and Problem Formulation

The notation used in the chapter is as follows. I_n denotes the identity matrix of dimension $n \times n$, $I_{n\times m}$ is a matrix of dimension $n \times m$ with ones on its main diagonal. The dimensions will often be omitted in cases where they can be implied from the context. For two matrices A and B of appropriate dimension, $\langle A, B\rangle \doteq \mathbf{trace}(A^T B)$. $\|.\|_F$ denotes the Frobenius norm, defined for an $n \times m$ matrix A with elements a_{ij} as

$$\|A\|_F^2 \doteq \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}^2.$$

The Frobenius norm has the following useful properties, needed in the sequel,

$$\|A\|_F^2 = \langle A, A\rangle = \sum_{i=1}^{\min\{n,m\}} \sigma_i^2(A) = \sum_{i=1}^{n} \lambda_i(A^T A), \tag{2.1}$$

where $\sigma_i(A)$ are the singular values of the matrix A and $\lambda_i(A^TA)$ are the eigenvalues of the matrix (A^TA). In addition to that, for any two matrices A and B of equal dimensions it holds that

$$\|A+B\|_F^2 = \|A\|_F^2 + 2\langle A, B\rangle + \|B\|_F^2. \tag{2.2}$$

$A > 0$ ($A \geq 0$) means that A is positive definite (positive semi-definite). We also introduce the notation $\|x\|_Q^2 \doteq x^TQx$ for $x \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n\times n}$ with $Q \geq 0$, which should not be mistaken with the standard notation for the vector p-norm ($\|x\|_p$). In LMIs, the symbols $\bullet$ will be used to indicate entries readily implied from symmetry. A vector of dimension n with all elements equal to zero will be denoted as $\mathbf{0}_n$. Futher, the volume of a closed set A is denoted as $\mathbf{vol}(A)$.

Let $\mathcal{C}_n^+$ denote the cone of symmetric non-negative definite n-by-n matrices, i.e.

$$\mathcal{C}_n^+ \doteq \{A \in \mathbb{R}^{n\times n} :\ A = A^T,\ A \geq 0\}.$$

For a symmetric matrix A we define the projection onto $\mathcal{C}_n^+$ as follows

$$\mathbf{\Pi}^+A \doteq \arg\min_{X\in\mathcal{C}_n^+} \|A-X\|_F. \tag{2.3}$$

Similarly, denoting

$$\mathcal{C}_n^- \doteq \{A \in \mathbb{R}^{n\times n} :\ A = A^T,\ A \leq 0\},$$

then the projection onto the cone of symmetric negative-definite matrices is defined as

$$\mathbf{\Pi}^-A \doteq \arg\min_{X\in\mathcal{C}_n^-} \|A-X\|_F. \tag{2.4}$$

Note that these two projections are uniquely defined. They have the following properties Calafiore and Polyak (2001).

Lemma 2.1 (Properties of the projection) *For a symmetric matrix A, the following properties hold*

(P1) $\mathbf{\Pi}^+A + \mathbf{\Pi}^-A = A$.

(P2) $\langle \mathbf{\Pi}^+A, \mathbf{\Pi}^-A\rangle = 0$.

(P3) *Let* $A = U\Lambda U^T$*, where* U *is an orthogonal matrix containing the eigenvectors of* A*, and* Λ *is a diagonal matrix with the eigenvalues* $\lambda_i,\ i = 1,\ldots,n,$ *of* A *appearing on its diagonal. Then*

$$\mathbf{\Pi}^+A = U\mathit{diag}\{\lambda_1^+,\ldots,\lambda_n^+\}U^T,$$

with $\lambda_i^+ \doteq \max(0,\lambda_i)$, $i = 1,\ldots,n$. *Equivalently,*

$$\mathbf{\Pi}^-A = U\mathit{diag}\{\lambda_1^-,\ldots,\lambda_n^-\}U^T,$$

with $\lambda_i^- \doteq \min(0,\lambda_i)$, $i = 1,\ldots,n$.

(P4) $\mathbf{\Pi}^+A$ *and* $\mathbf{\Pi}^-A$ *are continuous in* A.

In this chapter we consider the following uncertain transfer function

$$M_\Delta(\sigma) : \begin{bmatrix} u \\ \xi \end{bmatrix} \mapsto \begin{bmatrix} z \\ y \end{bmatrix},$$

defined as

$$M_\Delta(\sigma) = \begin{bmatrix} C_z^\Delta \\ C_y^\Delta \end{bmatrix} (\sigma I_n + A^\Delta)^{-1} \begin{bmatrix} B_u^\Delta & B_\xi^\Delta \end{bmatrix} + \begin{bmatrix} D_{zu}^\Delta & D_{z\xi}^\Delta \\ D_{yu}^\Delta & D_{y\xi}^\Delta \end{bmatrix}. \tag{2.5}$$

where $A^\Delta \in \mathbb{R}^{n\times n}$, $B_u^\Delta \in \mathbb{R}^{n\times m}$, $B_\xi^\Delta \in \mathbb{R}^{n\times n_\xi}$, $C_z^\Delta \in \mathbb{R}^{n_z\times n}$, $C_y^\Delta \in \mathbb{R}^{p\times n}$, $D_{zu}^\Delta \in \mathbb{R}^{n_z\times m}$, $D_{z\xi}^\Delta \in \mathbb{R}^{n_z\times n_\xi}$, $D_{yu}^\Delta \in \mathbb{R}^{p\times m}$, $D_{y\xi}^\Delta \in \mathbb{R}^{p\times n_\xi}$, $u \in \mathbb{R}^m$ is the control action, $y \in \mathbb{R}^p$ is the measured output, $z \in \mathbb{R}^{n_z}$ is the controlled output of the system, and $\xi \in \mathbb{R}^{n_\xi}$ is the disturbance to the system, and where the symbol σ represents the s-operator (i.e. the time-derivative operator) for continuous-time systems, and the z-operator (i.e. the shift operator) for discrete-time systems. The uncertainty Δ is assumed to be such that it

1. belongs to the uncertainty set $\boldsymbol{\Delta}$, and
2. is coupled with some probability density function $f_{\boldsymbol{\Delta}}(\Delta)$ inside the uncertainty set $\boldsymbol{\Delta}$.

There are further no restrictions on $\boldsymbol{\Delta}$ besides that the elements of the state-space matrices of the system should not become unbounded, i.e. it should hold that

$$\left\| \begin{bmatrix} A^\Delta & B_\xi^\Delta & B_u^\Delta \\ C_z^\Delta & D_{z\xi}^\Delta & D_{zu}^\Delta \\ C_y^\Delta & D_{y\xi}^\Delta & D_{yu}^\Delta \end{bmatrix} \right\|_F < \infty,\ \forall \Delta \in \boldsymbol{\Delta}. \tag{2.6}$$

Remark 2.1 *Whenever the uncertainty is fully deterministic or no a-priori information is available about its statistical properties, uniform distribution could be selected, i.e.*

$$f_{\boldsymbol{\Delta}}(\Delta) = \frac{1}{\mathbf{vol}(\boldsymbol{\Delta})},\ \forall \Delta \in \boldsymbol{\Delta}.$$

The following mild assumptions need to be imposed.

Assumption 2.1 *It is assumed that random samples of Δ can be generated inside $\boldsymbol{\Delta}$ with the specified probability distribution $f_{\boldsymbol{\Delta}}(\Delta)$.*

For certain probability density functions there exist algorithms in the literature for generation of random samples of Δ. For instance, in Calafiore et al. (1999) the authors consider the problem of generations of (real and complex) vectors samples uniformly in the ball $\mathcal{B}(r) = \{x : \|x\|_p \leq r\}$. This is consequently extended for the matrix case, but only the 1-norm and the ∞-norm are considered. The important case of matrix 2-norm is considered later on in Calafiore et al. (2000). The reader is referred to Calafiore et al. (2000, 1999) for more details on the available algorithms for uncertainty generation.

In Chapter 1 the optimization problem (1.19) on page 26 was defined for achieving passive FTC. In this optimization problem a cost function $J(\cdot)$ needs to be optimized for the worst-case model uncertainty δ and the worst case fault f. Both f and δ are considered in this chapter as one uncertainty

$$\Delta = \begin{bmatrix} \delta \\ f \end{bmatrix},$$

so that problem (1.19) becomes equivalent to the following *optimization problem*

$$(\mathcal{P}_{\mathcal{O}}) : \ K^* = \arg \min_{K} \max_{\Delta \in \boldsymbol{\Delta}} J(G_{\Delta}(\sigma), K). \tag{2.7}$$

Furthermore, the problem $(\mathcal{P}_{\mathcal{O}})$ is equivalent to the minimization of a scalar $\gamma > 0$ over γ and K subject to the constraint

$$(\mathcal{P}_{\mathcal{F}}) : \ \max_{\Delta \in \boldsymbol{\Delta}} J(G_{\Delta}(\sigma), K) \leq \gamma. \tag{2.8}$$

For a fixed γ, the problem $(\mathcal{P}_{\mathcal{F}})$ defined in equation (2.8) is called a *feasibility problem.* When there is a method that can solve the feasibility problem $(\mathcal{P}_{\mathcal{F}})$, then solving the original optimization problem $(\mathcal{P}_{\mathcal{O}})$ only requires a bisection algorithm on γ where at each iteration $(\mathcal{P}_{\mathcal{F}})$ is solved for fixed γ. For that reason we consider now the feasibility problem $(\mathcal{P}_{\mathcal{F}})$ only.

The feasibility problem (2.8) is a problem of robust controller design. Many controller and filter design problems are known to be representable in terms of LMIs (Boyd et al. 1994) in the form

Control Problem: Find a feasible solution to the LMI $U_{\gamma}(x, \Delta) \leq 0, \ x \in \mathcal{X} \subseteq \mathbb{R}^N$, for all $\Delta \in \boldsymbol{\Delta}$, (2.9)

where $U_{\gamma}(x, \Delta) = U_{\gamma}^T(x, \Delta) \in \mathbb{R}^{q \times q}$ is affine in the vector of variables x, and where the set $\mathcal{X}$ is assumed to be convex. The controller is then parametrized by any solution x^* to (2.9).

Remark 2.2 *Note that the vector x in this chapter represents the unknown variables in the control problem given in (2.9). It should not be mistaken with the notation for the state vector used in other chapters of this thesis.*

It should be noted here that when dealing with uncertain systems in the general *output-feedback* case the feasibility problem $(\mathcal{P}_{\mathcal{F}})$ cannot be represented as a robust LMI problem, but as a BMI problem[1]. Such BMI problems are non-convex, NP hard problems that cannot be treated by the approaches discussed in this chapter. BMI problems are discussed in the next chapter. In contrast to the output-feedback, in the *state-feedback* case most design problems (including LQR, $\mathcal{H}_2$, $\mathcal{H}_\infty$, pole-placement, etc.) can be written in the form (2.9).

To motivate the probabilistic framework used in this chapter we note that the deterministic methods to robust LMI problems of the form (2.9) usually assume that the set $\{U_{\gamma}(\bar{x}, \Delta) : \ \Delta \in \boldsymbol{\Delta}\}$ is a convex polytope for any fixed $\bar{x} \in \mathcal{X}$. This

---

[1] When no uncertainty is present in the model description many output-feedback problems can also be equivalently transformed to LMIs.

makes it possible to represent the infinite set of matrix inequalities $U_\gamma(x,\Delta) \le 0$ by a finite system of LMIs $U_\gamma^{(i)}(x) \le 0$, $i = 1, 2, \ldots, K$, defined on the vertexes of the polytope, i.e.

$$U_\gamma(x,\Delta) \le 0 \iff \begin{cases} U_\gamma^{(1)}(x) \le 0 \\ U_\gamma^{(2)}(x) \le 0 \\ \vdots \\ U_\gamma^{(K)}(x) \le 0 \end{cases}$$

There are very efficient and fast LMI solvers nowadays for solving such systems of LMIs (Gahinet et al. 1995).

Whenever the set $\{U_\gamma(\bar{x},\Delta) : \Delta \in \mathbf{\Delta}\}$ is not a convex polytope, however, this approach can only be applied after accepting a certain amount of conservatism by over-bounding the set by a convex polytope. To avoid such conservatism the robust LMI problem is addressed here in a probabilistic framework.

The set of all feasible solutions to the control problem is called the *solution set,* and is denoted as

$$\mathcal{S}_\gamma \doteq \{x \in \mathcal{X} : U_\gamma(x,\Delta) \le 0, \ \forall \Delta \in \mathbf{\Delta}\}. \tag{2.10}$$

The goal is the development of an iterative algorithm capable of finding a solution to the control problem defined (2.9). To this end the following *cost function* is defined

$$v_\gamma(x,\Delta) \doteq \|\mathbf{\Pi}^+[U_\gamma(x,\Delta)]\|_F^2 \ge 0, \tag{2.11}$$

which is non-negative for any $x \in \mathcal{X}$ and $\Delta \in \mathbf{\Delta}$. The usefulness of the so-defined function $v_\gamma(x,\Delta)$ stems from the following fact.

Lemma 2.2 *For a given pair* $(\bar{x},\bar{\Delta}) \in \mathcal{X} \times \mathbf{\Delta}$ *it holds that* $U_\gamma(\bar{x},\bar{\Delta}) \in \mathcal{C}_q^-$ *if and only if* $v_\gamma(\bar{x},\bar{\Delta}) = 0$.

Proof:
Using the third property in Lemma 2.1 on page 36 we note that $U_\gamma(\bar{x},\bar{\Delta}) \in \mathcal{C}_q^-$ holds if and only if

$$\mathbf{\Pi}^-[U_\gamma(\bar{x},\bar{\Delta})] = U_\gamma(\bar{x},\bar{\Delta})$$

Making use of the first property in Lemma 2.1 we then observe that

$$\mathbf{\Pi}^+[U_\gamma(\bar{x},\bar{\Delta})] = 0,$$

or equivalently, that $v_\gamma(\bar{x},\bar{\Delta}) = 0$. □

Using the result from Lemma 2.2 it follows that

$$\{x \in \mathcal{X} : v_\gamma(x,\Delta) = 0, \ \forall \Delta \in \mathbf{\Delta}\} \equiv \mathcal{S}_\gamma$$

holds. In other words $v_\gamma(x,\Delta) = 0$ for all $\Delta \in \mathbf{\Delta}$ if and only if $x \in \mathcal{S}_\gamma$.

In this way the initial problem is reformulated to the following optimization problem

$$x^* = \arg\min_{x\in\mathcal{X}} \sup_{\Delta\in\mathbf{\Delta}} v_\gamma(x,\Delta). \tag{2.12}$$

In the algorithms presented in this chapter the gradient of the function $v_\gamma(\cdot,\cdot)$ will be needed. In order to derive an analytic expression for it, we first note that since $U_\gamma(x,\Delta)$ is affine in x it can be written in the form

$$U_\gamma(x,\Delta) = U_{\gamma,0}(\Delta) + \sum_{i=1}^{N} U_{\gamma,i}(\Delta)x_i,$$

where x_i is the i-th element of the vector x, and where

$$\begin{aligned}
&U_{\gamma,0}(\Delta) = U_\gamma(\mathbf{0}_N,\Delta),\\
&U_{\gamma,i}(\Delta) = U_\gamma(e_i,\Delta) - U_{\gamma,0}(\Delta),\ i = 1,2,\ldots,N,\\
&e_i^T = \left[\begin{array}{ccc} \mathbf{0}_{i-1}^T, & 1, & \mathbf{0}_{N-i}^T \end{array}\right]
\end{aligned} \tag{2.13}$$

Then the following result holds.

Lemma 2.3 *The function $v_\gamma(x,\Delta)$, defined in equation (2.11), is convex and differentiable in x and its gradient is given by*

$$\nabla v_\gamma(x,\Delta) = 2\left[\begin{array}{c} \textbf{trace}\left(U_{\gamma,1}(\Delta)\mathbf{\Pi}^+[U_\gamma(x,\Delta)]\right)\\ \vdots \\ \textbf{trace}\left(U_{\gamma,N}(\Delta)\mathbf{\Pi}^+[U_\gamma(x,\Delta)]\right)\end{array}\right] \tag{2.14}$$

Proof: By using the properties of the projection in Lemma 2.1 we observe that for some symmetric matrices R and ΔR it can be written that

$$\begin{aligned}
\|\mathbf{\Pi}^+[R+\Delta R]\|_F^2 &\overset{\text{(P1)}}{=} \|R+\Delta R-\mathbf{\Pi}^-[R+\Delta R]\|_F^2\\
&\overset{\text{(P1)}}{=} \|\mathbf{\Pi}^+R+\mathbf{\Pi}^-R+\Delta R-\mathbf{\Pi}^-[R+\Delta R]\|_F^2\\
&\overset{(2.2)}{=} \|\mathbf{\Pi}^+R\|_F^2 + 2\left\langle\mathbf{\Pi}^+R,\Delta R\right\rangle + 2\underbrace{\left\langle\mathbf{\Pi}^+R,\mathbf{\Pi}^-R\right\rangle}_{=0}\\
&\quad +\|\mathbf{\Pi}^-R+\Delta R-\mathbf{\Pi}^-[R+\Delta R]\|_F^2\\
&\quad +2\underbrace{\left\langle\mathbf{\Pi}^+R,-\mathbf{\Pi}^-[R+\Delta R]\right\rangle}_{\geq 0}\\
&\overset{\text{(P2),(P3)}}{\geq} \|\mathbf{\Pi}^+R\|_F^2 + \left\langle 2\mathbf{\Pi}^+R,\Delta R\right\rangle
\end{aligned}$$

In addition to that, noting that from (2.4) on page 36 it follows that

$$\|A-\mathbf{\Pi}^+A\|_F^2 = \min_{X\in\mathcal{C}_n^-}\|A-X\|_F^2, \tag{2.15}$$

we can write that

$$\begin{aligned}
\|\mathbf{\Pi}^+[R+\Delta R]\|_F^2 &\overset{\text{(P1)}}{=} \|R+\Delta R-\mathbf{\Pi}^-[R+\Delta R]\|_F^2\\
&\overset{(2.15)}{=} \min_{S\in\mathcal{C}_n^-}\|R+\Delta R-S\|_F^2\\
&\leq \|R+\Delta R-\mathbf{\Pi}^-R\|_F^2 \overset{\text{(P1)}}{=} \|\mathbf{\Pi}^+R+\Delta R\|_F^2\\
&\overset{(2.2)}{=} \|\mathbf{\Pi}^+R\|_F^2 + \left\langle 2\mathbf{\Pi}^+R,\Delta R\right\rangle + \|\Delta R\|_F^2.
\end{aligned}$$

It thus follows that

$$\|\mathbf{\Pi}^{+}[R+\Delta R]\|_F^2 = \|\mathbf{\Pi}^{+}R\|_F^2 + \langle 2\mathbf{\Pi}^{+}R, \Delta R\rangle + O(\|\Delta R\|_F^2).$$

Now, substitute $R = U_\gamma(x,\Delta)$ and $\Delta R = \sum_{i=1}^N U_{\gamma,i}(\Delta)\Delta x_i$ to obtain

$$v_\gamma(x+\Delta x,\Delta) \geq v_\gamma(x,\Delta) + \sum_{i=1}^N \langle 2\mathbf{\Pi}^{+}[U_\gamma(x,\Delta)]U_{\gamma,i}(\Delta), \Delta x_i\rangle \tag{2.16}$$

$$\begin{aligned} & v_\gamma(x+\Delta x,\Delta) \\ & = v_\gamma(x,\Delta) + \sum_{i=1}^N \langle 2\mathbf{\Pi}^{+}[U_\gamma(x,\Delta)], U_{\gamma,i}(\Delta)\rangle\,\Delta x_i + O(\|\Delta x\|_2^2), \end{aligned} \tag{2.17}$$

The convexity follows from inequality (2.16), while the differentiability – from equation (2.17). The gradient of $v_\gamma(x,\Delta)$ is then given by (2.14). □

Now that the gradient of the function $v_\gamma(x,\Delta)$ is derived analytically we are ready to proceed to the probabilistic approaches to controller design.

2.2.2 The Subgradient Iteration Algorithm

For finding a feasible solution to the optimization problem (2.12), an algorithm was proposed in Calafiore and Polyak (2001). It originated in Polyak and Tempo (2001), where it was developed specifically for the design of a state-feedback LQ regulator. We will refer to this algorithm as the *Subgradient Iteration Algorithm* due to the fact that it is based on subgradient iterations.

Define the operator $\mathbf{\Pi}_{\mathcal{X}} : \mathbb{R}^N \mapsto \mathcal{X}$ as follows

$$\mathbf{\Pi}_{\mathcal{X}} x \doteq \arg\min_{y\in\mathcal{X}} \|x-y\|_2.$$

Further, the following assumption is imposed for the SIA.

Assumption 2.2 (Strong Feasibility Condition) *A scalar $r > 0$ is known for which there exists $x^* \in \mathcal{X}$ such that*

$$\mathcal{B}(x) \doteq \{x \in \mathcal{X} : \|x - x^*\| \leq r\} \subseteq \mathcal{S}_\gamma.$$

Assumption 2.2 implies that the solution set $\mathcal{S}_\gamma$ has a non-empty interior, and that a radius r of a ball contained in $\mathcal{S}_\gamma$ is known. This is often is a rather restrictive assumption due to the fact that usually no *a-priori* information about the solution set $\mathcal{S}_\gamma$ is available. This assumption will be released in the next section where the newly proposed algorithm is presented.

The SIA is then summarized in Algorithm 2.1 (see Polyak and Tempo (2001); Calafiore and Polyak (2001) for more details). As an initial condition $x^{(0)}$ to the algorithm can be selected any element of the set $\mathcal{X}$. As a stopping criterion one may, for instance, select the condition that for a given number of iterations L (usually $L \gg 1$) the step-size $\mu_{i-k} = 0$ (or equivalently $v_\gamma(x^{(i-k)}, \Delta^{(i-k)}) = 0$) for $k = 0, 1, \ldots, L$. A "weaker" stopping condition could be that the vector $x^{(i)}$

Algorithm 2.1 (Subgradient Iteration Algorithm)

```
Initialization: $i = 0$, $x^{(0)}$, $P_0 = P_0^T > 0$, $\varepsilon > 0$ small, $0 < \eta < 2$, integer $L > 0$.
Step 1. Set $i \leftarrow i + 1$.
Step 2. Generate a random sample $\Delta^{(i)}$ with probability distribution $f_\Delta$.
Step 3. If $v_\gamma(x^{(i)}, \Delta^{(i)}) \neq 0$ then take
            $x^{(i+1)} = \mathbf{\Pi}_{\mathcal{X}}[x^{(i)} - \mu_i \nabla v_\gamma(x^{(i)}, \Delta^{(i)})].$   (2.20)
        with
            $\mu_i = \eta \frac{v_\gamma(x^{(i)}, \Delta^{(i)}) + r\|\nabla v_\gamma(x^{(i)}, \Delta^{(i)})\|_2}{\|\nabla v_\gamma(x^{(i)}, \Delta^{(i)})\|_2^2}$   (2.21)
        else take $x^{(i+1)} = x^{(i)}$.
Step 4. If $\{v_\gamma(x^{(i+j-L)}, \Delta^{(i+j-L)}) = 0$ for $j = 0, 1, \ldots, L\}$ then Stop
        else Goto Step 1.
```

did not change "significantly" in the last L iterations. Once the algorithm has terminated, a Monte-Carlo simulation could be performed to estimate the empirical probability of robust feasibility (Calafiore and Polyak 2001). Whenever the obtained probability is unsatisfactory, the number L can be increased and the algorithm can be continued until a better solution (achieving higher empirical probability of robust feasibility) is found.

For proving the convergence of the algorithm, the following technical assumption needs to be additionally imposed.

Assumption 2.3 *For any* $x^{(i)} \notin \mathcal{S}_\gamma$ *there is a non-zero probability to generate a sample* $\Delta^{(i)}$ *for which* $v_\gamma(x^{(i)}, \Delta^{(i)}) > 0$, *i.e.*

$$\mathbf{Prob}(v_\gamma(x^{(i)}, \Delta^{(i)}) > 0) > 0.$$

This assumption is not restrictive and needs to hold also for the algorithm, proposed in the next section. Note that a sufficient for the assumption to hold is that the density function f_Δ is nonzero everywhere. The assumption is needed to make sure that the algorithm will not terminate at an infeasible point $x^{(i)} \notin \mathcal{S}_\gamma$ at which there is a zero probability for a *correction step* to be executed. By correction step it is meant an iteration (2.20) with $x^{(i+1)} \neq x^{(i)}$.

It is shown in Calafiore and Polyak (2001) that for any initial condition $x^0 \in \mathcal{X}$, the SIA finds a feasible solution with probability one in a finite number of iterations, provided that Assumptions 2.2 and 2.3 hold. It is also shown that the number

$$I_{SIA} = \|x^{(0)} - x^*\|^2/(r^2\eta(2-\eta)) \tag{2.22}$$

provides an upper bound on the maximum number of correction steps that will be executed before a feasible solution is reached. However, the relation 2.22 cannot be directly used to compute the bound I_{SIA} since x^* is unknown.

Although there are a lot of applications for which the subgradient algorithm performs well, in general it possesses the weakness that Assumption 2.2 is too restrictive, i.e. the number r is not known. As it is demonstrated below, if it is selected not small enough, so that the condition in Assumption 2.2 does not hold, then Algorithm SIA results in an oscillatory sequence $\{x^{(i)}\}_{i=1,2,\dots}$ that actually *diverges* from the solution set. On the other hand, if r is selected too small to make sure that Assumption 2.2 is satisfied, then the convergence rate of the algorithm can drastically slow down since the maximum number of correction steps is reversely proportional to r^2. To experimentally illustrate this discussion we consider an example. Before we proceed with this example, however, we define the level set $LS_\gamma(c, \Delta^*)$ for the function $v_\gamma(x, \Delta)$ for some given $\Delta^* \in \boldsymbol{\Delta}$ and a given scalar $c > 0$ as follows

$$LS_\gamma(c, \Delta^*) \doteq \{x \in \mathcal{X} : \ v_\gamma(x, \Delta^*) \le c\}. \tag{2.23}$$

Example 2.1 *Consider the discrete-time system*

$$\mathcal{M} : \ x_{k+1} = x_k + u_k, \tag{2.24}$$

and the following standard LQ cost function

$$J_{LQR} = \sum_{i=1}^{\infty} \|x_{k+i}\|_Q^2 + \|u_{k+i}\|_R^2,$$

for some $Q, R > 0$. *It is shown in Kothare et al. (1996) that the control action*

$$u_k = Fx_k = YX^{-1}x_k$$

achieves an upper bound of $x_k^T X^{-1} x_k$ *on the cost function if and only if* $X = X^T > 0$ *and* Y *are such that*

$$\begin{bmatrix} X & (AX+BY)^T & XQ^{1/2} & Y^TR^{1/2} \\ \star & X & 0 & 0 \\ \star & \star & I & 0 \\ \star & \star & \star & I \end{bmatrix} \ge 0. \tag{2.25}$$

By (randomly) selecting $Q = 1$, $R = 10$, $r = 1$, $\eta = 1$, $X_0 = 0.1545$, $Y_0 = -1.7073$, *the subgradient iteration algorithm does not converge to the solution set, but rather begins to oscillate, as it can be seen from Figure 2.1. The feasibility set is represented by the innermost contour in Figure 2.1 (left). The contours in Figure 2.1 represent different level sets. The reason for these oscillations is that there exists no ball of radius* $r = 1$ *inside the solution set, as required by 2.2. Clearly,*

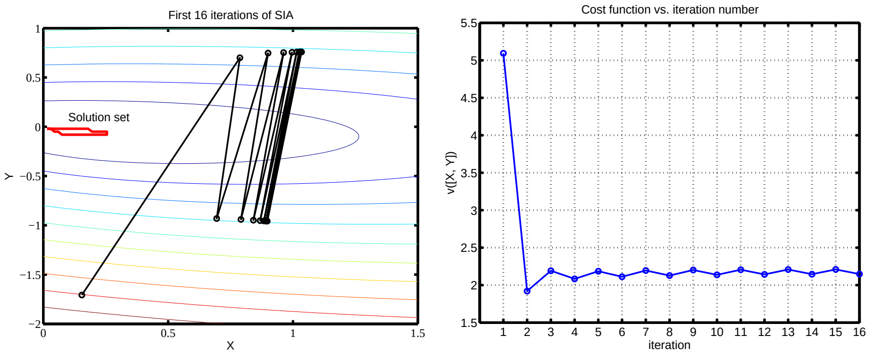


Figure 2.1: Performance of the Subgradient Iteration Algorithm for system $\mathcal{M}$: (left) level curves of $v_\gamma([X\ Y]^T)$ together with a plot of the sequence $\{[X^{(i)},\ Y^{(i)}]^T\}_{i=1}^{16}$, (right) plot of $v_\gamma([X^{(i)},\ Y^{(i)}]^T)$ versus the iteration number i.

for this trivial example one can obtain convergence by simply reducing r a bit (for instance, taking $r = 0.5$ results in convergence to a solution in six iterations), but in general for larger systems of LMIs simple trial-and-error method with different values of the radius r may not be the best option.

As proposed in Calafiore and Polyak (2001); Kushner and Yin (1997), one way to circumvent the lack of knowledge of r is to substitute it with a sequence $\{\epsilon_s\}$ such that $\epsilon_s > 0$, $\epsilon \to 0$ and $\sum_{s=0}^{\infty} \epsilon_s = \infty$. This releases the assumption that the radius r is known while at the same time retaining the property of guaranteed convergence in a finite number of iterations with probability one. However, this approach increases the number of iterations necessary to arrive at a feasible solution. In addition to that the choice of an appropriate sequence $\{\epsilon_s\}$ remains an open question.

The approach that we propose in this chapter is based on the *Ellipsoid Algorithm* (EA). The starting point in EA is the computation of an initial ellipsoid that contains the solution set $\mathcal{S}_\gamma$. Then, similarly to the SIA method, at each iteration of the EA two steps are performed. In the first step a random uncertainty sample $\Delta^{(i)} \in \mathbf{\Delta}$ is generated according to the given probability density function $f_{\mathbf{\Delta}}(\Delta)$. With this generated uncertainty the convex function $U_\gamma(x, \Delta^{(i)})$ is parametrized and used at the second step of the algorithm where an ellipsoid is computed, in which the solution set is guaranteed to lie. In this way the EA produces a sequence of ellipsoids with decreasing volumes, all containing the solution set. Using some existing facts, and provided that the solution set has a non-empty interior, it will be established that this algorithm converges to a feasible solution in a finite number of iterations with probability one. To initialize the algorithm, a method is presented for obtaining an initial ellipsoid that contains the solution set. It is also shown that even if the solution set has a zero volume, the EA converges to the solution set when the iteration number tends to infinity, which is a property not possessed by the SIA.

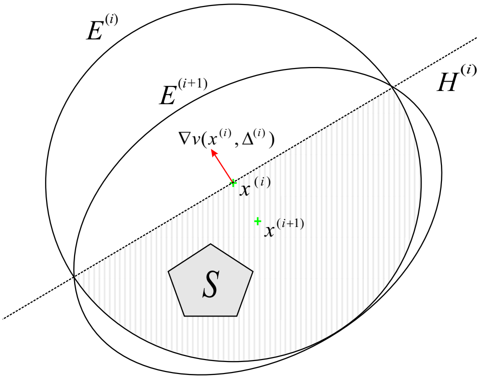


Figure 2.2: One iteration of the ellipsoid method in the two-dimensional case.

2.3 The Ellipsoid Algorithm: Feasibility

The algorithm presented below releases the restrictive Assumption 2.2, and retains only Assumption 2.3. Convergence in a finite number of iterations with probability one is also guaranteed.

Assume that an initial ellipsoid $E^{(0)}$, that contains the solution set $\mathcal{S}_\gamma$, is given

$$E^{(0)} = \{x \in \mathcal{X} :\ (x - x^{(0)})^T P_0^{-1}(x - x^{(0)}) \leq 1\} \supseteq \mathcal{S}_\gamma \tag{2.26}$$

described by its center $x^{(0)} \in \mathcal{X}$ and the matrix $P_0 \in \mathcal{C}_N^+$ related to its shape and orientation. We further assume that the dimension N of the vector of unknowns is is larger than one[2]. The problem of finding such an initial ellipsoid will be discussed in the next section. Define

$$H^{(0)} \doteq \{x \in \mathcal{X} :\ \nabla^T v_\gamma(x^{(0)}, \Delta)(x - x^{(0)}) \leq 0\}.$$

Due to the convexity of the function $v_\gamma(x, \Delta)$ we know that $H^{(0)}$ also contains the solution set $\mathcal{S}_\gamma$, and therefore $\mathcal{S}_\gamma \subseteq H^{(0)} \cap E^{(0)}$. We can then construct a new ellipsoid, $E^{(1)}$, as the *minimum volume* ellipsoid such that $E^{(1)} \supseteq H^{(0)} \cap E^{(0)} \supseteq \mathcal{S}_\gamma$, and such that the volume of $E^{(1)}$ is less than the volume of $E^{(0)}$. This, repeated iteratively, represents the main idea behind the Ellipsoid Algorithm (Boyd et al. 1994; Grötschel et al. 1988).

Suppose that after iteration i we have $x^{(i)} \in \mathcal{X}$ and $P_i = P_i^T > 0$ such that

$$E^{(i)} = \{x \in \mathcal{X} :\ (x - x^{(i)})^T P_i^{-1}(x - x^{(i)}) \leq 1\} \supseteq \mathcal{S}_\gamma.$$

The Ellipsoid algorithm, visualized in the two-dimensional case in Figure 2.2, is then summarized in Algorithm 2.2.

The algorithm terminates when the value of the function $v_\gamma(x^{(\cdot)}, \Delta^{(\cdot)})$ remains equal to zero for L successieve iterations or when the volume of the ellipsoid (which is proportional to $\det(P)^{1/2}$) becomes smaller than a pre-defined

---

[2] With $N = 1$ the algorithm simplifies to a bisection algorithm.

Algorithm 2.2 (The Ellipsoid Algorithm for $(\mathcal{P}_{\mathcal{F}})$)

Initialization: $i = 0$, $x^{(0)}$, $P_0 = P_0^T > 0$, $\varepsilon > 0$ small, integer $L > 0$.

Step 1. Set $i \leftarrow i + 1$.

Step 2. Generate a random sample $\Delta^{(i)}$ with probability distribution f_Δ.

Step 3. If $v_\gamma(x^{(i)}, \Delta^{(i)}) \neq 0$ then take

$$\begin{aligned} x^{(i+1)} &= x^{(i)} - \frac{1}{N+1} \frac{P_i \nabla v_\gamma(x^{(i)}, \Delta^{(i)})}{\sqrt{\nabla^T v_\gamma(x^{(i)}, \Delta^{(i)}) P_i \nabla v_\gamma(x^{(i)}, \Delta^{(i)})}} \\ P_{i+1} &= \frac{N^2}{N^2 - 1} \left(P_i - \frac{2}{N+1} \frac{P_i \nabla v_\gamma(x^{(i)}, \Delta^{(i)}) \nabla^T v_\gamma(x^{(i)}, \Delta^{(i)}) P_i^T}{\nabla^T v_\gamma(x^{(i)}, \Delta^{(i)}) P_i \nabla v_\gamma(x^{(i)}, \Delta^{(i)})} \right) \end{aligned}$$

else take $x^{(i+1)} = x^{(i)}$, $P_{i+1} = P_i$.

Step 4. Form the ellipsoid

$$E^{(i+1)} = \{x : \ (x - x^{(i+1)})^T P_{i+1}^{-1} (x - x^{(i+1)}) \leq 1\} \supseteq \mathcal{S}_\gamma.$$

Step 5. If $\left(\sqrt{\det(P)} < \varepsilon\right)$ or $\big(v_\gamma(x^{(i+j-L)}, \Delta^{(i+j-L)}) = 0$ for $j = 0, 1, \ldots, L\big)$ then **Stop** else **Goto Step 1.**

small positive number ε. In the latter case no feasible solution is found (for instance due to the fact that the solution set has an empty interior, i.e. **vol**$(\mathcal{S}_\gamma) = 0$). In such case γ has to be increased in the feasibility problem (2.8) on page 38 and Algorithm 2.2 has to be started again until a feasible solution is found. Note that if the feasibility problem (2.8) is feasible for some γ^*, then it is also feasible for any $\gamma > \gamma^*$. It should also be noted that, due to the probabilistic nature of the algorithm, the fact that the algorithm terminates due to the cost function being equal to zero for a finite number L of successive iterations does not necessarily imply that a feasible solution is found. In practice, however, choosing L sufficiently large ensures the feasibility of the solution.

The convergence of the approach is established immediately, provided that Assumption 2.3 holds, which implies that for any $x^{(i)} \notin \mathcal{S}_\gamma$ there exists a non-zero probability for the execution of a correction step (i.e. there is a non-zero probability for generation of $\Delta^{(i)} \in \mathbf{\Delta}$ such that $v_\gamma(x^{(i)}, \Delta^{(i)}) > 0$).

Lemma 2.4 (Convergence of Algorithm EA) *Consider Algorithm 2.2 without the stopping condition in Step 5 (or with $\varepsilon = 0$ and $L \to \infty$), and suppose that Assumption 2.3 holds. Suppose also that*

(i) **vol**$(\mathcal{S}_\gamma) > 0$. *Then a feasible solution will be found in a finite number of itera-*

tions with probability one.

(ii) **vol**$(\mathcal{S}_\gamma) = 0$. *Then*

$$\lim_{i\to\infty} x^{(i)} = x^* \in \mathcal{S}_\gamma$$

with probability one.

Proof: Suppose that at the i-th iteration of Algorithm EA $k(i)$ correction steps have been performed. Algorithm EA generates ellipsoids with geometrically decreasing volumes so that for the i-th iteration we can write (Boyd et al. 1994)

$$\textbf{vol}(E^{(i)}) \le e^{-\frac{k(i)}{2N}} \textbf{vol}(E^{(0)}),$$

Due to Assumption 2.3, for any $x^{(i)} \notin \mathcal{S}_\gamma$ there exists a non-zero probability for the execution of a correction step. Therefore, at any infeasible point $x^{k(i)}$ the algorithm will execute a correction step after a finite number of iterations with probability one. This implies that

$$\lim_{i\to\infty} \textbf{vol}(E^{(i)}) = 0. \tag{2.27}$$

(i) If we then suppose that the solution set $\mathcal{S}_\gamma$ has a non-empty interior, i.e. **vol**$(\mathcal{S}) > 0$, then from equation (2.27) and due to the fact that $E^{(i)} \supseteq \mathcal{S}_\gamma$ for all $i = 0, 1, \dots$, it follows that in a finite number of iterations with probability one the algorithm will terminate at a feasible solution.
(ii) If we now suppose that **vol**$(\mathcal{S}) = 0$, then due to the convexity of the function, and due to equation (2.27), the algorithm will converge to a point in $\mathcal{S}_\gamma$ with probability one. □

The result in Lemma 2.4 outlines the advantages of Algorithm EA over the previously proposed Algorithm SIA. While in the case **vol**$(\mathcal{S}_\gamma) > 0$ Algorithm EA preserves the property of guaranteed convergence with probability one in a finite number of iterations, it offers the advantages over Algorithm SIA that

- no a-priori knowledge about a number $r > 0$ satisfying the condition in Assumption 2.2 is necessary (we will discuss how to find an initial ellipsoid in the next Section), and
- it converges (although at infinity) even in the case that the set $\mathcal{S}_\gamma$ has an empty interior.

Remark 2.3 *It needs to be noted, however, the Lemma 2.4 considers Algorithm EA with* $L \to \infty$*, which in practice is never the case. For finite* L *the solution found by the algorithm can only be guaranteed to be* ϵ*-suboptimal with some probability. To be more specific, let some scalars* $\epsilon \in (0,1)$ *and* $\delta \in (0,1)$ *be given, and let* x^* *be the output of Algorithm EA for* $\varepsilon = 0$ *and* $L \ge \ln\frac{1}{\delta}/\ln\frac{1}{1-\epsilon}$*. Then (Dabbene 1999; Fujisaki and Kozawa 2003)*

$$\mathit{Prob}\{\mathit{Prob}\{v_\gamma(x^*, \Delta) > 0\} \le \epsilon\} \ge 1 - \delta.$$

Therefore, if we want with high confidence (e.g. $\delta = 0.01$*) that the probability that* x^* *is an optimal solution is very high (*$1-\epsilon = 0.999$*) then we need to select* L *larger than* 4603*. In practice, however, a much smaller value for* L *suffices.*

Finally, similarly to the bound I_{SIA} on the maximum number of correction steps for the Subgradient Iteration Algorithm (see equation (2.22) on page 43), we can derive such an upper bound for the proposed Ellipsoid method.

Lemma 2.5 *Consider Algorithm EA, and suppose that Assumption 2.3 holds. Suppose further that the solution set has a non-empty interior, i.e.* $\mathbf{vol}(\mathcal{S}_\gamma) > 0$. *Then the number*

$$I_{EA} = 2N \left\lceil \ln \frac{\mathbf{vol}(E^{(0)})}{\mathbf{vol}(\mathcal{S}_\gamma)} \right\rceil \tag{2.28}$$

is an upper bound on the maximum number of correction steps that can be performed starting from any ellipsoid $E^{(0)} \supseteq \mathcal{S}_\gamma$, *where* $\lceil a \rceil$, $a \in \mathbb{R}$, *denotes the minimum integer number larger than or equal to* a.

Proof: It is shown in Boyd et al. (1994) that for the $k(i)$-th correction step one can write

$$\mathbf{vol}(E^{(k(i))}) \leq e^{-\frac{k(i)}{2N}} \mathbf{vol}(E^{(0)}).$$

Since the volume of the consecutive ellipsoids tends to zero, and since $\mathbf{vol}(\mathcal{S}_\gamma) > 0$, there exists an correction step number I_{EA} such that

$$e^{-\frac{k(i)}{2N}} \mathbf{vol}(E^{(0)}) \leq \mathbf{vol}(\mathcal{S}_\gamma), \text{ for } \{\forall i : \ k(i) \geq I_{EA}\}.$$

Therefore, we could obtain the number I_{EA} from the following relation

$$\frac{\mathbf{vol}(\mathcal{S}_\gamma)}{\mathbf{vol}(E^{(0)})} \geq e^{-\frac{k(i)}{2N}} \quad \Longleftarrow \quad \{\forall i : \ k(i) \geq I_{EA}\}.$$

Now, by taking the natural logarithm on both sides one obtains

$$\ln \frac{\mathbf{vol}(\mathcal{S}_\gamma)}{\mathbf{vol}(E^{(0)})} \geq -\frac{k(i)}{2N} \quad \Longleftarrow \quad \{\forall i : \ k(i) \geq I_{EA}\}$$

or

$$k(i) \geq 2N \ln \frac{\mathbf{vol}(E^{(0)})}{\mathbf{vol}(\mathcal{S}_\gamma)} \quad \Longleftarrow \quad \{\forall i : \ k(i) \geq I_{EA}\}$$

Therefore, equation (2.28) is proven. □

We would like to point out that usually $I_{EA} \ll I_{SIA}$. This is demonstrated in the following example.

Example 2.2 (Comparison between the bounds I_{EA} and I_{SIA}) *Suppose that the dimension of our vector of unknowns* x *is 10 (i.e.* $N = 10$*), and that the solution set is a ball of radius* 1.1 *and center* $x^* \in \mathbb{R}^{10}$

$$\mathcal{S}_\gamma = \{x \in \mathbb{R}^{10} : \ \|x - x^*\| \leq 1.1\}.$$

To make a fair comparison between the SIA and the newly proposed EA we proceed as follows: we assume that the initial condition $x^{(0)}$ *for SIA is at a distance* $d > 1.1$ *from the center of* $\mathcal{S}_\gamma$, *i.e.* $\|x^{(0)} - x^*\| = d$, *and that the initial ellipsoid for EA is a ball of radius* d. *Since for SIA the number* r *in Assumption 2.2 should be*

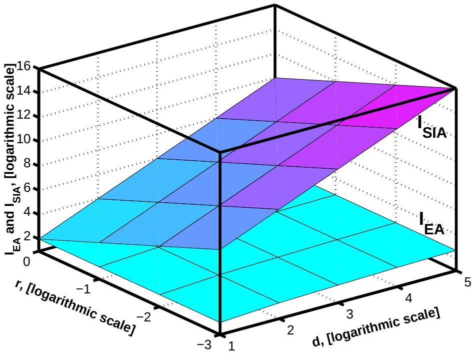


Figure 2.3: Comparison between the upper bounds I_{EA} and I_{SIA} for the algorithms SIA and EA.

known, we will make several experiments with $r = \{0.001, 0.01, 0.1, 1\}$*. For these values of* r*, and for* $d = \{10, 10^2, 10^3, 10^4, 10^5\}$ *the two upper bounds* I_{EA} *and* I_{SIA} *on the maximum numbers of possible correction steps for the two algorithms were computed. Figure 2.3 represents the results (note that all the three axes are in logarithmic scale). Clearly,* $I_{EA} \ll I_{SIA}$*. It should be pointed out that even if one selects the initial ellipsoid for the EA to be a ball of radius* $10d$*, or even* $100d$*, one still gets* $I_{EA} \ll I_{SIA}$*.*

Example 2.3 *Let us consider again Example 2.1 on page 43. Suppose that we select the initial ellipsoid (2.26) on page 45 for the EA as follows*

$$E^{(0)} = \left\{ x \in \mathcal{X} : \left(x - \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \right)^T \begin{bmatrix} 4 & 0 \\ 0 & 4.25 \end{bmatrix}^{-1} \left(x - \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \right) \leq 1 \right\}.$$

Then the EA terminates in 16 iterations at a feasible solution. Figure 2.4 visualizes the convergence process by depicting four ellipsoids: the initial one, and the ellipsoids obtained at iterations 7, 13, and 16. The figure also shows the performance of the SIA (when executed on this example with constant parameter $r = 1$*) for comparison.*

It should be pointed out here that the initial ellipsoid used in this example has been chosen so that it "embraces" the trajectory made by the SIA algorithm. If the method for finding the initial ellipsoid from the next section was used instead then the EA would terminate in one iteration, i.e. the center of the initial ellipsoid lies in the feasibility set.

In the next Section we present a method to obtain an initial ellipsoid.

2.4 Finding an Initial Ellipsoid $E^{(0)}$

In this section we consider the problem of finding initial ellipsoid that contains the solution set $\mathcal{S}_\gamma$, that is needed to initialize Algorithm EA. The approach that

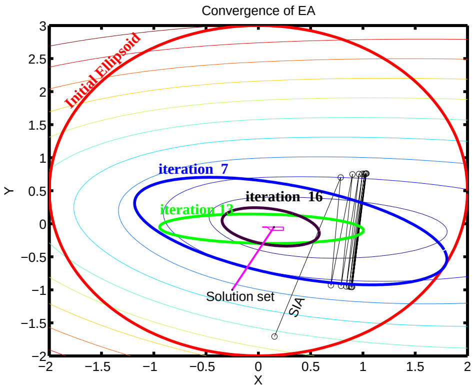


Figure 2.4: Performance of the EA on the problem in Example 2.4.

will first be presented in the next subsection is applicable to general LMI problems of the form (2.9) on page 38. Afterwards we will concentrate on a more specific problem, namely the problem of constrained robust linear least squares (LLS). This problem is a special case of (2.9) on page 38 and is also in the basis of the well-known Model Predictive Control (MPC) strategy, discussed later on in Chapter 5 of this thesis. The reason for considering this problem separately is that due to its structure the initial ellipsoid can be formed in an easier and more natural way.

2.4.1 Procedure for General LMI Problems

Before the method for obtaining an initial ellipsoid is presented, some additional notation must be introduced. In addition to the solution set $\mathcal{S}_\gamma$ and the level sets $LS_\gamma(c, \Delta)$, we now define the *local solution sets* for any fixed $\Delta_i \in \mathbf{\Delta}$ as the level set at zero

$$S^0(\Delta_i) \doteq LS_\gamma(0, \Delta_i). \tag{2.29}$$

Therefore, any $x^* \in \mathcal{S}_\gamma$ is such that $x^* \in S^0(\Delta)$ for all $\Delta \in \mathbf{\Delta}$. Also the solution set $\mathcal{S}_\gamma$ is the intersection of all local solution sets

$$\mathcal{S}_\gamma = \bigcap_{\Delta_i \in \mathbf{\Delta}} S^0(\Delta_i).$$

Note also, that for any $c \geq 0$ it holds that $LS_\gamma(c, \Delta) \supseteq S^0(\Delta) \supseteq \mathcal{S}_\gamma$. Figure 2.5 provides a two-dimensional visualization. Due to the convexity of the functions $v_\gamma(x, \Delta_i)$ (consult Lemma 2.3 on page 40), the solution set is clearly convex.

The following additional assumption needs to be imposed.

Assumption 2.4 *It is assumed that the level set* $\mathcal{X} \bigcap LS_\gamma(0, \Delta)$ *is a bounded set for all* $\Delta \in \mathbf{\Delta}$.

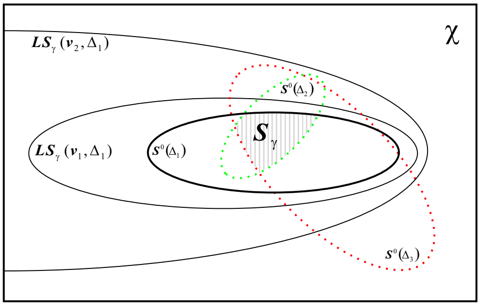


Figure 2.5: Level sets $LS_\gamma(v_i, \Delta_i)$, local solution sets $S^0(\Delta_i) = LS_\gamma(0, \Delta_i)$, and the (global) solution set $\mathcal{S}_\gamma$.

Assumption 2.4 can be ensured by means of selecting the set $\mathcal{X}$ to be bounded as would, for instance, be the case when one selects $\mathcal{X}$ to be a bounded box (see Figure 2.5). From practical point of view this assumption is not very restrictive since the box can be selected "large enough" to encompass (at least a part of) the solution set. Notice also, that from numerical point of view the introduction of such hard constraints on the entries of the vector x is not unreasonable since, as discussed in the beginning of this chapter, the optimal solution is often needed to parametrize a controller or an observer, and as a result very large entries in x may lead to numerical problems. It should also be pointed out that such an assumption is not imposed in Algorithm SIA; in fact, as shown by Liberzon and Tempo (2003), cases in which the solution set is not bounded are even favorable for SIA. For instance, considering the problem of finding a Lyapunov matrix P for a stable linear system $\dot{x} = Ax$ by means of solving the inequality $PA + A^T P < 0$ makes it clear, that if P^* is a solution then αP^* is also a solution for any $\alpha >$, so that for this problem Assumption 2.2 is satisfied for any $r > 0$.

Let us now again concentrate on the problem of finding the initial ellipsoid containing the solution set $\mathcal{S}_\gamma$ under Assumption 2.4. For this purpose we will make use of the fact that $\mathcal{S}_\gamma$ is contained in any local solution set S_Δ, and therefore in any level set $LS_\gamma(c, \Delta)$ for any $c > 0$ and $\Delta \in \boldsymbol{\Delta}$. It is, therefore, contained in $LS_\gamma(0, \Delta^{(0)})$, for some (possibly randomly generated) $\Delta^{(0)} \in \boldsymbol{\Delta}$, i.e. $\mathcal{S}_\gamma \subseteq LS_\gamma(0, \Delta^{(0)})$. The idea is then to find an ellipsoid that contains the level set $LS_\gamma(0, \Delta^{(0)})$. To this end we will first bound the set $LS_\gamma(0, \Delta^{(0)})$ with a rectangular parallelepiped, and then we build an ellipsoid around it as shown in Figure 2.6, which we will use as an initial ellipsoid to start Algorithm EA. In order to find a bounding rectangular parallelepiped, we need to find solutions to the following constrained optimization problems

$$\bar{x}_i = \max_{x \in \mathcal{X}} x_i, \text{ subject to } x \in LS_\gamma(0, \Delta^{(0)}),\ i = 1, 2, \ldots, N,$$
$$\underline{x}_i = \min_{x \in \mathcal{X}} x_i, \text{ subject to } x \in LS_\gamma(0, \Delta^{(0)}),\ i = 1, 2, \ldots, N,$$

These can be rewritten as LMI problems by noting that

$$\{x \in LS_\gamma(0, \Delta^{(0)})\} \equiv \{x \in \mathcal{X} :\ v_\gamma(x, \Delta^{(0)}) = 0\} \equiv \{x \in \mathcal{X} :\ U_\gamma(x, \Delta^{(0)}) \leq 0\}.$$

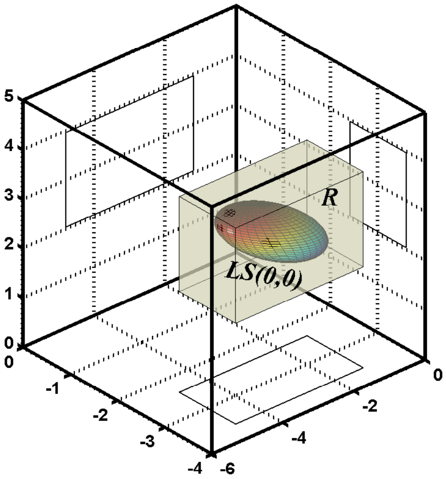


Figure 2.6: The initial ellipsoid is computed by first bounding the level set $LS_\gamma(0,0)$ with a box, and then obtaining an ellipsoid that embraces it (not drawn on the figure).

Note that under Assumption 2.4 it holds that $-\infty < \underline{x}_i \leq \bar{x}_i < \infty$. Hence, the box

$$R = \{x : \underline{x} \leq x \leq \bar{x}\} \supseteq LS_\gamma(0,0) \supseteq \mathcal{S}_\gamma. \tag{2.30}$$

$\bar{x} = [\bar{x}_1, \ldots, \bar{x}_N]^T$ and $\underline{x} = [\underline{x}_1, \ldots, \underline{x}_N]^T$, contains the solution set. Then the following result holds.

Lemma 2.6 *The ellipsoid* $E_P^{(0)} = \{x : (x - x^{(0)})^T P_0^{-1}(x - x^{(0)})\}$ *with*

$$x^{(0)} = \frac{1}{2}(\overline{x} + \underline{x}), \;\; P_0 = \frac{\dim x}{4}\left[\mathbf{diag}(\overline{x} - \underline{x})\right]^2 \tag{2.31}$$

contains the solution set $\mathcal{S}$*, where* $\underline{x}$ *and* $\overline{x}$ *are defined as the vertexes of the box* R *in equation (2.30).*

Proof: It can easily be verified that the ellipsoid

$$E_{in} \doteq \{x : (x - x^{(0)})^T Z^{-1}(x - x^{(0)}) \leq 1\}$$

with $x^{(0)} = \frac{1}{2}(\overline{x} + \underline{x})$ and $Z = \left[\frac{1}{2}\mathbf{diag}(\overline{x} - \underline{x})\right]^2$ is inside R and its axes are perpendicular to the faces of R. This ellipsoid can be equivalently represented as

$$E_{in} = \{x : \|Z^{-1/2}x - Z^{-1/2}x^{(0)}\|_2^2 \leq 1\}.$$

Stretching the ellipsoid E_{in} by $\alpha > 1$ results in

$$E_{out} = \{x : \alpha^{-1}\|Z^{-1/2}x - Z^{-1/2}x^{(0)}\|_2^2 \leq 1\},$$

Algorithm 2.3 (Initial Ellipsoid Computation)

INITIALIZATION: SELECT ANY $\Delta^{(0)} \in \boldsymbol{\Delta}$.

Step 1. FIND SOLUTIONS TO THE LMI PROBLEMS

$$\begin{aligned}\bar{x}_i &= \max_{x\in\mathcal{X}} x_i, \ \text{SUBJECT TO } U_\gamma(x, \Delta^{(0)}) \leq 0, \ i = 1, 2, \ldots, N, \\ \underline{x}_i &= \min_{x\in\mathcal{X}} x_i, \ \text{SUBJECT TO } U_\gamma(x, \Delta^{(0)}) \leq 0, \ i = 1, 2, \ldots, N.\end{aligned}$$

Step 2. TAKE $\bar{x} = [\bar{x}_1, \ldots, \bar{x}_N]^T$ AND $\underline{x} = [\underline{x}_1, \ldots, \underline{x}_N]^T$.

Step 3. TAKE $E^{(0)}$ WITH $x^{(0)}$ AND P_0 DEFINED IN (2.31) AS INITIAL ELLIPSOID.

which we want to make such that it contains the vertex points $\overline{x}$ and $\underline{x}$ of the box R. Therefore we select α such that the vertex points of the box $R = \{x : \ \underline{x} \leq x \leq \overline{x}\}$ lie on the surface of E_{out}, i.e.

$$\begin{aligned}\alpha &= \|Z^{-1/2}\overline{x} - Z^{-1/2}x^{(0)}\|_2^2 \\ &= \|Z^{-1/2}\tfrac{1}{2}(\overline{x} - \underline{x})\|_2^2 \\ &= \left\| [\mathbf{diag}(\overline{x} - \underline{x})]^{-1} (\overline{x} - \underline{x}) \right\|_2^2 \\ &= \dim \boldsymbol{d}.\end{aligned}$$

Therefore the ellipsoid

$$E_{out} = \{x : \ (x - x^{(0)})^T (\alpha Z)^{-1} (x - x^{(0)}) \leq 1\},$$

embraces the box R and the initial ellipsoid (2.26) parameterized by (2.31) contains the box R, that on its turn contains the solution set $\mathcal{S}$. $\square$

Algorithm 2.3 summarizes the procedure for initial ellipsoid computation.

The initial ellipsoid computation procedure is next illustrated with a simple example.

Example 2.4 (Initial Ellipsoid Computation) *To illustrate the algorithm for initial ellipsoid computation, proposed in the previous Section, we consider the following system*

$$\begin{aligned}\dot{x}(t) &= -x(t) + u(t) + \xi(t) \\ z(t) &= x(t)\end{aligned}$$

for which a constant state-feedback controller has to be designed such that the squared $\mathcal{H}_\infty$-norm of the resulting closed-loop system is less than $\gamma = 10^{-5}$. Using the results in (Boyd et al. 1994), this would be the case if there exist $Q \in \mathbb{R}$, $R \in \mathbb{R}$,

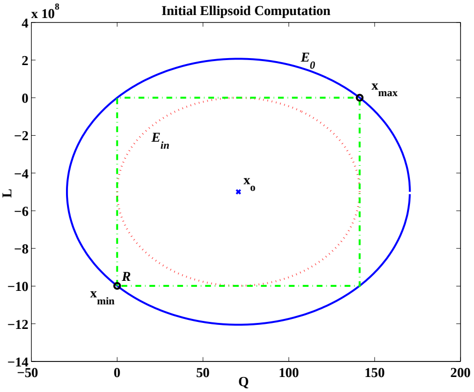


Figure 2.7: Illustration of the algorithm for initial ellipsoid computation.

and $L \in \mathbb{R}$ *such that*

$$\begin{bmatrix} Q & 0 & 0 & 0 \\ \star & 2Q - L - L^T & 1 & Q \\ \star & \star & 1 & 0 \\ \star & \star & \star & \gamma \end{bmatrix} > 0$$

Figure 2.7 visualizes the initial ellipsoid that was generated by Algorithm 2.3 on page 53.

2.4.2 The Constrained Robust Least-Squares Problem

The linear least squares problem arises in a wide variety of engineering application, ranging from data fitting to controller and filter design. It is in the basis of the well-known Model Predictive Control (MPC) strategy, an industrially very relevant control technique due to its ability to handle constraints on the inputs and outputs of the controlled system.

In this subsection the following *robust constrained linear least-squares problem* is considered: Find $x \in \mathbb{R}^N$ that achieves

$$\left\{ \begin{array}{l} \textbf{Optimization problem } (\mathcal{P})\textbf{:} \\ \text{Find } x \in \mathbb{R}^N \text{ that achieves} \\ \\ \gamma_{opt} = \min\limits_{x} \max\limits_{\Delta \in \boldsymbol{\Delta}} \|b(\Delta) - A(\Delta)x\|_2^2, \text{ subject to} \\ F(x, \Delta) \doteq F_0(\Delta) + \sum\limits_{i=1}^{N} F_i(\Delta)x_i \geq 0, \ \forall \Delta \in \boldsymbol{\Delta} \end{array} \right. \tag{2.32}$$

where $x = [x_1, \dots, x_N]^T$ denotes the vector of unknowns, $b(\Delta) \in \mathbb{R}^p$ and $A(\Delta) \in \mathbb{R}^{p \times N}$ are known functions of the uncertainty $\Delta \in \boldsymbol{\Delta}$. Similarly to (2.8) on page 38 we first concentrate on the feasibility problem, that now has the form

$$\left\{\begin{array}{l}\textbf{Feasibility problem } (\mathcal{FP})\textbf{:}\\ \text{Given } \gamma > 0, \text{ find } x \in \mathbb{R}^N \text{ that achieves}\\ \\ \max\limits_{\Delta \in \boldsymbol{\Delta}} \|b(\Delta) - A(\Delta)x\|_2^2 \leq \gamma,\\ F(x,\Delta) \geq 0, \ \forall \Delta \in \boldsymbol{\Delta}.\end{array}\right.$$

Clearly,

$$(\mathcal{P}) \Longleftrightarrow \min \gamma \text{ s.t. } (\mathcal{FP})\text{-feasible}.$$

We proceed by recasting the feasibility problem $(\mathcal{FP})$ to an LMI feasibility problem. To this end, note that the first inequality in $(\mathcal{FP})$ is equivalent to

$$(b(\Delta) - A(\Delta)x)^T I^{-1} (b(\Delta) - A(\Delta)x) \leq \gamma,$$

so that by using the Schur complement it becomes an LMI

$$G_\gamma(x,\Delta) \doteq \begin{bmatrix} I & b(\Delta) - A(\Delta)x \\ \star & \gamma \end{bmatrix} \geq 0,$$

where $\star$ denotes entries in LMIs that follow by symmetry. As a result, the feasibility problem $(\mathcal{FP})$ becomes

$$(\mathcal{FP}) \Longleftrightarrow \begin{bmatrix} G_\gamma(x,\Delta) & \\ & F(x,\Delta) \end{bmatrix} > 0, \ \forall \Delta \in \boldsymbol{\Delta}.$$

For solving this (robust) LMI problem with the probabilistic approach, we begin by defining the following function

$$w_\gamma(x,\Delta) \doteq \|\boldsymbol{\Pi}^-[G_\gamma(x,\Delta)]\|_F^2 + \|\boldsymbol{\Pi}^-[F(x,\Delta)]\|_F^2. \tag{2.33}$$

Note that $w_\gamma(x,\Delta)$ has the same form as $v_\gamma(x,\Delta)$ defined in equation (2.11) on page 39 besides that now the projection $\boldsymbol{\Pi}^-$ is used instead of $\boldsymbol{\Pi}^+$. Similarly, for this function the following result holds.

Lemma 2.7 *The function $w_\gamma(x,\Delta)$ is convex and differentiable, and its gradient is given by*

$$\begin{aligned}\nabla w_\gamma(x,\Delta) &= \left(\begin{bmatrix} \mathbf{0}_N & -4 \end{bmatrix} \boldsymbol{\Pi}^-[G_\gamma(x,\Delta)] \begin{bmatrix} A(\Delta) \\ 0 \end{bmatrix}\right)^T + \\ & \quad 2\begin{bmatrix} \mathbf{trace}(F_1(\Delta)\boldsymbol{\Pi}^-[F(x,\Delta)]) \\ \vdots \\ \mathbf{trace}(F_N(\Delta)\boldsymbol{\Pi}^-[F(x,\Delta)]) \end{bmatrix}.\end{aligned} \tag{2.34}$$

Proof: Since both $G_\gamma(x,\Delta)$ and $F(x,\Delta)$ are affine in x, then following the same reasoning as in the proof of Lemma 2.3 on page 40 it can be shown that the function $w_\gamma(x,\Delta)$ is also convex and differentiable, and that

$$\begin{aligned}&\nabla w_\gamma(x,\Delta)\\ &= \underbrace{2\begin{bmatrix} \mathbf{trace}(G_{\gamma,1}(\Delta)\boldsymbol{\Pi}^-[G_\gamma(x,\Delta)]) \\ \vdots \\ \mathbf{trace}(G_{\gamma,N}(\Delta)\boldsymbol{\Pi}^-[G_\gamma(x,\Delta)]) \end{bmatrix}}_{\nabla\|\boldsymbol{\Pi}^-[G_\gamma(x,\Delta)]\|_F^2} + \underbrace{2\begin{bmatrix} \mathbf{trace}(F_1(\Delta)\boldsymbol{\Pi}^-[F(x,\Delta)]) \\ \vdots \\ \mathbf{trace}(F_N(\Delta)\boldsymbol{\Pi}^-[F(x,\Delta)]) \end{bmatrix}}_{\nabla\|\boldsymbol{\Pi}^-[F(x,\Delta)]\|_F^2}\end{aligned}$$

where $G_{\gamma,i}(\Delta) = G_\gamma(e_i,\Delta) - G_\gamma(0,\Delta)$ with $e_i \in \mathbb{R}^N$ defined in equation 2.13 on page 40. Making use of the special structure of the matrix $G_\gamma(x,\Delta)$, and letting $A_i(\Delta) = A(\Delta)e_i$ denote the i-th column of the matrix $A(\Delta)$, we can then write that

$$\begin{aligned}
&\mathbf{trace}(G_{\gamma,i}(\Delta)\mathbf{\Pi}^-[G_\gamma(x,\Delta)]) \\
&= -\mathbf{trace}\left(\mathbf{\Pi}^-[G_\gamma(x,\Delta)]\begin{bmatrix} 0 & A_i(\Delta) \\ A_i^T(\Delta) & 0 \end{bmatrix}\right) \\
&= -2\mathbf{trace}\left(\begin{bmatrix} \mathbf{0}_N^T, & 1 \end{bmatrix}\mathbf{\Pi}^-[G_\gamma(x,\Delta)]\begin{bmatrix} A_i(\Delta) \\ 0 \end{bmatrix}\right)
\end{aligned}$$

from where it follows that

$$\nabla\|\mathbf{\Pi}^-[G_\gamma(x,\Delta)]\|_F^2 = \left(\begin{bmatrix} \mathbf{0}_N & -4 \end{bmatrix}\mathbf{\Pi}^-[G_\gamma(x,\Delta)]\begin{bmatrix} A(\Delta) \\ 0 \end{bmatrix}\right)^T,$$

which completes the proof. □

An initial ellipsoid can be found by making use the fact that any ellipsoid that contains the set

$$\left\{x: \max_{\Delta\in\mathbf{\Delta}} \|b(\Delta) - A(\Delta)x\|_2^2 \le \gamma\right\} \tag{2.35}$$

also contains the solution set

$$\mathcal{S}_\gamma = \left\{x: \max_{\Delta\in\mathbf{\Delta}} \|b(\Delta) - A(\Delta)x\|_2^2 \le \gamma,\ F(x,\Delta) \ge 0,\ \forall\Delta\in\mathbf{\Delta}\right\}.$$

On the other hand we note that for any $\hat{\Delta} \in \mathbf{\Delta}$ the set

$$\mathcal{J}(\hat{\Delta}) \doteq \left\{x: \|b(\hat{\Delta}) - A(\hat{\Delta})x\|_2^2 \le \gamma\right\} \tag{2.36}$$

contains the set defined in equation (2.35). Therefore, it will suffice to find an initial ellipsoid such that

$$E^{(0)} \supseteq \mathcal{J}(\hat{\Delta})$$

for some $\hat{\Delta} \in \mathbf{\Delta}$ in order to be sure that $E^{(0)}$ will also contain $\mathcal{S}_\gamma$. Usual choice for $\hat{\Delta}$ is $\hat{\Delta} = 0$ (provided that $0 \in \mathbf{\Delta}$, of course), but in practice any other (i.e. randomly generated) element $\hat{\Delta}$ from the set $\mathbf{\Delta}$ can be used.

For simplicity of notation we also define the ellipsoid

$$\mathcal{E}(\bar{x},\bar{P}) \doteq \{x: (x-\bar{x})^T\bar{P}^{-1}(x-\bar{x}) \le 1\},$$

with center $\bar{x}$ and with the matrix $\bar{P} = \bar{P}^T > 0$ defining its shape and orientation. The following cases, related to the rank and dimension of the matrix A can be differentiated:

Case 1. $p = N$ **and** $A(\hat{\Delta})$ **is invertible.**

In this case

$$\mathcal{J}(\hat{\Delta}) = \left\{x: \left(x - A^{-1}(\hat{\Delta})b(\hat{\Delta})\right)^T \frac{A^T(\hat{\Delta})A(\hat{\Delta})}{\gamma}\left(x - A^{-1}(\hat{\Delta})b(\hat{\Delta})\right) \le 1\right\}$$

so that $E^{(0)} = \mathcal{E}\left(A^{-1}(\hat{\Delta})b(\hat{\Delta}), \frac{A^T(\hat{\Delta})A(\hat{\Delta})}{\gamma}\right)$.

Case 2. $p > N$ **and** $A(\hat{\Delta})$ **is left-invertible.**

We can thus factorize $A(\hat{\Delta})$ (e.g. by using the singular value decomposition) as

$$A(\hat{\Delta}) = U_\gamma \begin{bmatrix} A_1(\hat{\Delta}) \\ 0 \end{bmatrix},$$

where U_γ is a unitary matrix and $A_1(\hat{\Delta})$ is a square non-singular matrix. Denoting

$$\begin{bmatrix} b_1(\hat{\Delta}) \\ b_2(\hat{\Delta}) \end{bmatrix} = U_\gamma^T b(\hat{\Delta}),$$

we can then write

$$\begin{aligned} \|(b(\hat{\Delta}) - A(\hat{\Delta})x)\|_2^2 &= \left\| U_\gamma \begin{bmatrix} b_1(\hat{\Delta}) - A_1(\hat{\Delta})x \\ b_2(\hat{\Delta}) \end{bmatrix} \right\|_2^2 \\ &= \|b_1(\hat{\Delta}) - A_1(\hat{\Delta})x\|_2^2 + \|b_2(\hat{\Delta})\|_2^2 \leq \gamma. \end{aligned}$$

Therefore, we take

$$\begin{aligned} \mathcal{J}(\hat{\Delta}) &= \left\{ x : \|b(\hat{\Delta}) - A(\hat{\Delta})x\|_2^2 \leq \gamma \right\} \\ &= \left\{ x : \|b_1(\hat{\Delta}) - A_1(\hat{\Delta})x\|_2^2 \leq \gamma - \|b_2(\hat{\Delta})\|_2^2 \right\} \\ &= \left\{ x : \left(x - A_1^{-1}(\hat{\Delta})b_1(\hat{\Delta}) \right)^T \frac{A_1^T(\hat{\Delta})A_1(\hat{\Delta})}{\gamma - \|b_2(\hat{\Delta})\|_2^2} \left(x - A_1^{-1}(\hat{\Delta})b_1(\hat{\Delta}) \right) \leq 1 \right\}, \end{aligned}$$

so that $E^{(0)} = \mathcal{E}\left(A_1^{-1}(\hat{\Delta})b_1(\hat{\Delta}), \frac{A_1^T(\hat{\Delta})A_1(\hat{\Delta})}{\gamma - \|b_2(\hat{\Delta})\|_2^2} \right)$.

Case 3. $A(\hat{\Delta})$ **is not full column rank.**

In this case we cannot obtain an analytic expression for the initial ellipsoid, which could be computed by directly solving all N optimization problems in Algorithm 2.3 on page 53. However, the computational burden can be reduced here by solving $N - K$ instead of N optimization problems in Algorithm 2.3 on page 53, where K is the rank of the matrix $A(\hat{\Delta})$. To this end we proceed as follows. First, use the singular value decomposition to find a unitary matrix V such that

$$A(\hat{\Delta})V = \begin{bmatrix} \bar{A}, & 0 \end{bmatrix},$$

where $\bar{A} \in \mathbb{R}^{p \times K}$ is full column rank matrix (with $K < N$). Define

$$\bar{x} = V^T x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix},$$

with $x^{(1)} \in \mathbb{R}^K$ and $x^{(2)} \in \mathbb{R}^{N-K}$. Then

$$\begin{aligned} \mathcal{J}(\hat{\Delta}) &= \left\{ x : \|b(\hat{\Delta}) - A(\hat{\Delta})x\|_2^2 \leq \gamma \right\} \\ &= \left\{ V \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} : \|b(\hat{\Delta}) - \hat{A}x^{(1)}\|_2^2 \leq \gamma \right\} \end{aligned}$$

Then we use either Case 1 or Case 2 (depending on whether $\bar{A}$ is square or not) to form an ellipsoid $E_1 = \mathcal{E}\left(\bar{x}_1, \bar{P}_1\right)$ based on $\bar{A}$. In this way we have defined an ellipsoid in which $x^{(1)}$ should lie. The goal is to find another ellipsoid $E_2 = \mathcal{E}\left(\bar{x}_2, \bar{P}_2\right)$ for $x^{(2)}$ and to subsequently merge these two ellipsoids.

The second ellipsoid, $E_2 = \mathcal{E}\left(\bar{x}_2, \bar{P}_2\right)$, can be found using Algorithm 2.3 on page 53 of the previous subsection under assumption 2.4.

Given the two ellipsoids $E_1 = \mathcal{E}\left(\bar{x}_1, \bar{P}_1\right)$ and $E_2 = \mathcal{E}\left(\bar{x}_2, \bar{P}_2\right)$ we can merge them into one by observing that for all

$$\left[\begin{array}{c} x^{(1)} \\ x^{(2)} \end{array}\right] \in \mathcal{E}\left(\left[\begin{array}{c} \bar{x}_1 \\ \bar{x}_2 \end{array}\right], \left[\begin{array}{cc} 2\bar{P}_1 & \\ & 2\bar{P}_2 \end{array}\right]\right)$$

it holds that

$$\begin{aligned} x^{(1)} &\in \mathcal{E}\left(\bar{x}_1, \bar{P}_1\right), \\ x^{(2)} &\in \mathcal{E}\left(\bar{x}_2, \bar{P}_2\right). \end{aligned}$$

By going back to the original variables, the initial ellipsoid for this Case 3 is taken as

$$E^{(0)} = \mathcal{E}\left(V \left[\begin{array}{c} \bar{x}_1 \\ \bar{x}_2 \end{array}\right], V \left[\begin{array}{cc} 2\bar{P}_1 & \\ & 2\bar{P}_2 \end{array}\right] V^T\right).$$

In this way the initial ellipsoid for the feasibility problem corresponding to the special case of constrained robust least squares can be computed in order to be subsequently used in the probabilistic Algorithm EA.

2.5 The Ellipsoid Algorithm: Optimization

In Section 2.3 we focused our attention of the feasibility problem for a fixed value of γ in (2.8) on page 38, and briefly discussed that once it has been solved a bisection algorithm on γ can be used to solve the initial optimization problem (2.7) on page 38. This is now summarized in Algorithm 2.4.

The algorithm begins by checking whether a feasible solution to (2.9) for $\gamma = 1$ can be found by means of Algorithm EA. If not, γ is increased ten times to $\gamma = 10$ and Algorithm EA is run again. In this way Algorithm 2.4 iterates between **Step 1** and **Step 7** until a feasible solution for some γ is found. After that Algorithm 2.4 begins to iterate between **Step 3** and **Step 8**, so that at each cycle either γ_{UB} or γ_{LB} is set equal to the current γ, depending on whether this γ is feasible or not. In this way $[\gamma_{LB},\ \gamma_{UB}]$ is a constantly decreasing interval inside which the optimal γ lies. The algorithm is terminated once the length of this interval becomes smaller than the selected tolerance.

It should be born in mind that the smaller the selected tolerance Tol in Algorithm 2.4 the larger the value of the parameter L needs to be selected. If fact, one could derive a lower bound for L so that the obtained solution is an optimal solution with a given probability for some desired confidence (see Remark 2.3). Again, such bounds are usually conservative and a much smaller number L suffices in practice.

Algorithm 2.4 (Ellipsoid Algorithm for ($\mathcal{P}_\mathcal{O}$))

```
Initialization: real numbers $Tol > 0$ and $\gamma_{max} > 0$ (sufficiently
    large). Set $\gamma_1 = 1$, $\gamma_{LB} = 0$, $\gamma_{UB} \leftarrow \infty$, and $k = 1$.
Step 1. Find initial ellipsoid $\mathcal{E}_k^{(0)}(x_k^{(0)}, P_k^{(0)})$ for the (2.9) with $\gamma =$
    $\gamma_k$ using the methods of Section 2.4.
Step 2. Set $\mathcal{E}_{opt}(x_{opt}, P_{opt}) = \mathcal{E}_k^{(0)}(x_k^{(0)}, P_k^{(0)})$.
Step 3. Run Algorithm 2.2 on the (2.9) with $\gamma = \gamma_k$ and with initial ellipsoid $\mathcal{E}_{opt}(x_{opt}, P_{opt})$.
Step 4. Denote $\mathcal{E}_k^*(x_k^*, P_k^*)$ as the ellipsoid at the final iteration of
    Algorithm 2.2.
Step 5. If $(x_k^* \notin \mathcal{S}_{\gamma_k})$ then $(\gamma_{LB} = \gamma)$
    else $(\gamma_{UB} = \gamma$ and $\mathcal{E}_{opt}(x_{opt}, P_{opt}) = \mathcal{E}_k^*(x_k^*, P_k^*))$.
Step 6. Set $k \leftarrow k + 1$.
Step 7. If $(\gamma_{UB} = \infty)$ then $(\gamma_k = 10\gamma_{k-1}$ and goto Step 1.)
    else (if $\gamma_{LB} = 0$ then $\gamma_k = 0.1\gamma_{k-1}$ else $\gamma_k = \frac{\gamma_{LB}+\gamma_{UB}}{2}$)
Step 8. If $\frac{(\gamma_{UB}-\gamma_{LB})}{\gamma_{UB}} > Tol$ and $\gamma_{LB} < \gamma_{max}$ goto Step 3.
Step 9. Exit the algorithm with $\gamma_{opt} = \gamma_{UB}$ achieved by $x_{opt}$.
```

2.6 Experimental part

Next, we present an example illustrating the probabilistic approach developed in this chapter used to design a robust $\mathcal{H}_2$ state-feedback controller for a model, representing a real-life diesel actuator benchmark system, taken from (Blanke et al. 1995). The model represents the behavior of a brushless DC motor, which is the actuator part of a real-life speed governor for large diesel engines. A block-schematic representation of the system is given on Figure 2.8.

A linear, continuous-time model of the system can be written in state-space form as

$$\begin{array}{rcl} \dot{x}(t) & = & A(\delta)x(t) + B_u(\delta)\alpha u(t) + B_\xi(\delta)\xi(t) \\ z(t) & = & C_z x(t) \end{array} \tag{2.37}$$

with

$$A(\delta) = \begin{bmatrix} 0 & -\frac{K_v}{T_v} & 0 \\ \frac{\delta_1\delta_4}{\delta_2} & -\frac{\delta_3+K_v\delta_1\delta_4}{\delta_2} & 0 \\ 0 & \frac{1}{N} & 0 \end{bmatrix}, \ B_u(\delta) = \begin{bmatrix} \frac{K_v}{T_v} \\ \frac{K_v\delta_1\delta_4}{\delta_2} \\ 0 \end{bmatrix}, \ B_\xi(\delta) = \begin{bmatrix} 0 \\ \frac{1}{N\delta_2} \\ 0 \end{bmatrix},$$

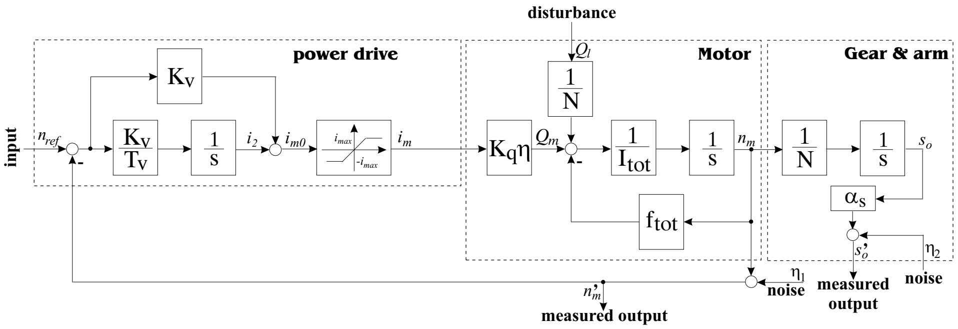


Figure 2.8: Block scheme of the Diesel Engine Actuator.

Par	Nom. value	Unit	Physical meaning
f_{tot}	19.7×10^{-3}	$Nm/rad/s$	Total friction
I_{tot}	2.53×10^{-3}	$kg.m^2$	Total inertia
K_q	0.54	Nm/A	Torque constant of servo motor
K_v	0.9	$A/rad/s$	Gain of the speed controller
N	89	-	Gear ratio
α_s	0.987	-	Measurement scaling factor
η	0.85	–	Gear efficiency
T_v	8.8×10^{-3}	s	Integral time of the speed controller
α	1	–	multiplicative actuator fault

Table 2.1: Nominal values of the parameters in the state-space model of the diesel engine actuator benchmark example.

where $x^T = [i_2,\ n_m,\ s_o]$ is the state vector, $u(t) = n_{ref}$ is the vector of control actions applied to the system, $y^T = [n'_m, s'_o]$ is the vector of measured outputs, $z = i_2$ is the controlled output, and $\xi = Q_l$ is a disturbance signal. The nominal values of the parameters, as well as their physical meaning, are given in Table 2.1. The variables in the state-space model (2.37) with their ranges and physical meaning are summarized in Table 2.2. The vector $\delta = [\eta,\ I_{tot},\ f_{tot},\ K_q]^T$ represents the uncertain parameters in the system. Multiplicative uncertainty representation is used so that

$$\delta = (I + \Delta)\delta^{nom}, \tag{2.38}$$

where

$$\delta^{nom} \doteq [0,775,\ 2,53.10^{-3},\ 3,45.10^{-2},\ 0.54]^T, \tag{2.39}$$

and where

$$\Delta \in \mathbf{\Delta} \doteq \{\mathbf{diag}(p_1, p_2, p_3, p_4) :\ |p_i| \leq \bar{p}_i\} \tag{2.40}$$

The scalar $0 \leq \alpha \leq 1$ in (2.37) is used to represent partial and total multiplicative actuator faults in the system. α and the uncertainty set $\mathbf{\Delta}$ in which Δ lies are

Variable	Range	Unit	Physical meaning
i_m	$\lvert i_m \rvert \leq 30$	A	Motor current
n_m	$\lvert n_m \rvert \leq 314$	rad/s	shaft speed of servo motor
n_{ref}	$\lvert n_{ref} \rvert \leq 314$	rad/s	shaft speed reference
Q_l	$\lvert Q_l \rvert \leq 6$	Nm	load torque
Q_m	$\lvert Q_m \rvert \leq 16$	Nm	torque from servo motor
s_o	$\lvert s_o \rvert \leq 0,4$	rad	shaft angular position

Table 2.2: The variables in the state-space model of the diesel engine actuator benchmark example.

defined below for the two simulations performed. The purpose of the first simulation is to make a simple comparison with the existing SIA method. The second simulation illustrates the design of a passive FTC controller.

The goal in both examples is design a state-feedback controller that guarantees robust stability of the closed loop system and ensures that with minimal energy of the input to the motor ($i_m(t)$) for impulse disturbance (load) $\xi(t)$. To this end we need to design an $\mathcal{H}_2$ robust state-feedback controller for the uncertain system (2.37). As shown in Boyd et al. (1994), if the matrices $Q = Q^T > 0$, $R = R^T$, and L are such that for all possible values of the parameters δ the following system of LMIs is feasible

$$\begin{aligned} \text{trace}(R) &< \gamma \\ \begin{bmatrix} R & C_z Q \\ \star & Q \end{bmatrix} &> 0 \\ \begin{bmatrix} -A(\delta)Q - QA(\delta)^T - B_u(\delta)L - L^T B_u(\delta)^T & B_\xi(\delta) \\ \star & I \end{bmatrix} &> 0 \end{aligned} \tag{2.41}$$

then the state-feedback control law $u(t) = Fx(t)$ with $F = LQ^{-1}$ results in a closed-loop system $T_{cl}(s,\delta) = C_z(sI - A(\delta) - B(\delta)Fx(t))^{-1}B_\xi(\delta)$ with $\mathcal{H}_2$-norm $\|T_{cl}(s,\delta)\|_2^2 \leq \gamma$ for all $\delta \in \mathbf{\Delta}$.

2.6.1 Comparison with SIA

The goal of this comparison is to show that the newly proposed EA might be a good alternative to the existing SIA for some applications. We here select the size of the uncertainty set as originally proposed in Blanke et al. (1995), see Table 2.3.

Parameter:	**Value:**
$(\bar{p}_1, \bar{p}_2, \bar{p}_3, \bar{p}_4)$	$(\frac{3}{31}, 0.15, \frac{5}{7}, 0.05)$
α	1

Table 2.3: Model parameters used for the comparison example in Section 2.6.1.

In this example the feasibility problem is considered of finding state-feedback gain matrix F that achieves an upper bound of $\gamma = 1$ on the performance index. Application of the proposed approach, summarized in Algorithms 2.4 and 2.2, resulted in the state-feedback gain matrix

$$F = \left[\begin{array}{ccc} -318.4153 & -44.6374 & -1.3393 \end{array} \right].$$

This solution was found by the EA method in about 100 iterations. This controller was computed in MatLab® in a few seconds on a 400MHz computer. For comparison, the SIA was terminated after 500 iterations having found no feasible solution (it was run for $r = 1$, $r = 0.1$, and $r = 0.01$).

2.6.2 Passive FTC Design

In this section the uncertainty set is further increased and, in addition, partial actuator faults represented by α in (2.37) are included in order to make a more challenging example, that can later on in Chapter 4 be compared to an active method for FTC. To this end the model parameters in this example are selected as shown in Table 2.4. We note that in this example some or all of the uncertain parameters can also be viewed as possible faults; they are treated by this passive approach in the same way as uncertainties.

Par.	**Value**
$(\bar{p}_1, \bar{p}_2, \bar{p}_3, \bar{p}_4)$	$(\frac{13}{31}, \frac{1}{2}, \frac{5}{7}, \frac{1}{2})$
α	$[0.5,\ 1]$

Table 2.4: Model parameters used for the passive FTC design in Section 2.6.2.

In this example the optimization problem is considered of minimizing γ subject to the system of LMIs (2.41). Running Algorithm 2.4 on the considered system resulted in the following optimal state-feedback gain

$$F = \left[\begin{array}{ccc} -13.6338 & -14.3643 & -3.8083 \times 10^{-7} \end{array} \right]$$

that achieves $\gamma_{opt} = 0.83125$. The parameters used for Algorithms 2.4 and 2.2 are $Tol = 0.1$, $\gamma_{max} = 100$, $\varepsilon = 0$, and $L = 40$. The required precision was achieved in 6 iterations of Algorithm 2.4 (each consisting of multiple sub-iterations in Algorithm 2.2, called at Step 3 of Algorithm 2.4). A summary of the convergence process is given in Table 2.5, where for each iteration the values for γ_k and its upper (γ_{UB}) and lower (γ_{LB}) bounds are provided, the volume of the final ellipsoid $\mathcal{E}_k^*$, and the status (feasibility or infeasibility) of the computed solution.

2.7 Conclusions

In this chapter a new approach was proposed to the probabilistic design of robust controllers (state estimators), based on the Ellipsoid Algorithm. It features

iter.	γ_k	γ_{LB}	γ_{UB}	**vol**$(\mathcal{E}_k^*)$	**status**
1	**1**	0	2	9.384×10^4	feas
2	0.1	0	**1**	1.528×10^1	infeas
3	0.55	0.1	**1**	4.891×10^3	infeas
4	0.775	0.55	**1**	1.795×10^4	infeas
5	**0.8875**	0.775	1	8.487×10^4	feas
6	**0.83125**	0.775	0.8875	8.487×10^4	feas

Table 2.5: Summary of the iterations performed by Algorithm 2.4. The optimal feasible value for γ at each iteration is written in boldface.

a number of advantages over the probabilistic Subgradient Iteration Algorithm, recently proposed in (Polyak and Tempo 2001; Calafiore and Polyak 2001). Although the latter possessed a number of useful properties, namely guaranteed convergence in a finite number of iterations with probability one, applicability to general uncertainty structures and to large numbers of uncertain parameters, it has the strong disadvantage that the radius of a non-empty ball contained in the solution set must be known. This drawback is removed in the EA approach proposed in this chapter, while still retaining the advantages of the SIA method. Similarly to the SIA method, at each iteration of the EA two steps are performed. In the first step a random uncertainty sample $\Delta^{(i)} \in \mathbf{\Delta}$ is generated according to the given probability density function $f_{\mathbf{\Delta}}(\Delta)$. With this generated uncertainty a suitably defined convex function is parametrized so that at the second step of the algorithm an ellipsoid is computed, in which the solution set is guaranteed to lie. As a result, the EA algorithm produces a sequence of ellipsoids with decreasing volumes, all containing the solution set. An efficient method for obtaining an initial ellipsoid is also proposed in the chapter. The approach is illustrated by means of a case study with a real-life diesel actuator benchmark model with four real uncertain parameters, for which an $\mathcal{H}_2$ robust state-feedback controller was designed.

3

BMI Approach to Passive Robust Output-Feedback FTC

As discussed in Chapter 1, the FTC methods are divided into passive and active ones. In Chapter 2 a probabilistic approach was presented to passive FTC where the starting point was a robust LMI. It was discussed that the problem of robust state-feedback controller design is representable in terms of such robust LMIs. The robust output-feedback controller design problem, on the other hand, is a nonconvex problem that cannot be addressed by the methods of Chapter 2. For most standard design objectives, including $\mathcal{H}_2$ and $\mathcal{H}_\infty$-norm minimization, this problem is representable as bilinear matrix inequality (BMI) optimization problem. Being able to solve such BMI problems is therefore important for passive output-feedback FTC design.

The contribution of this chapter is twofold. First, a new approach is proposed to the design of *locally optimal* robust output-feedback controllers. Starting from any initial feasible controller it performs local optimization over a suitably defined non-convex function. The approach features the properties of guaranteed convergence to a local optimum as well as applicability to a very wide range of problems, namely such representable as BMI problems. The second contribution in the chapter is the development of a fast procedure for computing an initial feasible controller. The design objectives considered are $\mathcal{H}_2$, $\mathcal{H}_\infty$, and pole-placement constraints. This procedure consists of two steps: first an optimal robust state-feedback gain F is designed, which is consequently kept fixed at the second step where the remaining controller matrices are designed. The complete output-feedback controller design approach is demonstrated on a model of one joint of a real-life space robotic manipulator.

3.1 Introduction

In the last decade much research was focused on the development of new approaches to controller design (Boyd et al. 1994; Scherer et al. 1997; Gahinet 1996; Gahinet et al. 1995; Palhares et al. 1996; Oliveira et al. 1999b; Kothare et al. 1996), state estimation (Geromel 1999; Geromel et al. 2000; Geromel and Oliveira 2001; Cuzzola and Ferrante 2001; Palhares et al. 1999), and system performance analysis (Oliveira et al. 1999a; Palhares et al. 1997; Zhou et al. 1995) on the basis of LMIs due to the recent development of computationally fast and numerically reliable algorithms for solving convex optimization problems subject to LMI constraints. In the cases when no uncertainty is considered in the model description, numerous LMI-based approaches exist that address the problems of state-feedback (Oliveira et al. 1999b; Palhares et al. 1996; Peres and Palhares 1995) and dynamic output-feedback controller (Apkarian and Gahinet 1995; Gahinet 1996; Geromel et al. 1999; Oliveira et al. 1999b) design for different design objectives. In these approaches, in general, the controller state-space matrices are parametrized by a set of matrices representing a feasible solution to a system of LMIs that describes the control objective, plus (often) the state-space matrices of the controlled system. For an overview of the LMI methods for analysis and design of control systems the reader is referred to Boyd et al. (1994); Scherer et al. (1997) and the references therein.

Whenever the controller parametrization is not explicitly dependent on the state-space matrices of the controlled system, generalization to polytopic uncertainties is trivial. Such cases include the LMI-based state-feedback controller design approaches to $\mathcal{H}_2$-control (Palhares et al. 1996; Kothare et al. 1996), $\mathcal{H}_\infty$-control (Palhares et al. 1996; Peres and Palhares 1995; Zhou et al. 1995), pole-placement in LMI regions (Chilali et al. 1999; Scherer et al. 1997), etc. These, however, require that the system state is measurable, thus imposing a severe restriction on the class of systems to which they are applicable.

Similar extension of most of the output-feedback controller design methods to the *structured uncertainty* case is, unfortunately, not that simple due to the fact that the controller parametrization explicitly depends on the state-space matrices of the system, which are unknown (Apkarian and Gahinet 1995; Gahinet 1996; Masubuchi et al. 1998; Scherer et al. 1997). Clearly, whenever the uncertainty is *unstructured* (e.g. high-frequency unmodelled dynamics), it can be recast into the general linear fractional transformation (LFT) representation and using the small gain theorem the design objective can be translated into controller design in the absence of uncertainty (Zhou and Doyle 1998). Application of this approach to systems with structured uncertainty, i.e. disregarding the structure of the uncertainty, often turns out to be excessively conservative. To overcome this conservatism μ-synthesis was developed (Zhou and Doyle 1998; Balas et al. 1998), which consists of an iterative procedure (known as $D - K$ iteration) where at each iteration two convex optimizations are executed - one in which the controller K is kept fixed, and one in which a certain diagonal scaling matrix D is kept fixed. This procedure, however, is not guaranteed to converge to a local optimum because optimality in two fixed directions does not imply optimality in all possible directions, and it may therefore lead to conservative results

(VanAntwerp et al. 1997).

Recently, some attempts have been made towards the development of LMI-based approaches to output-feedback controller design for systems with structured uncertainties in the context of robust quadratic stability with disturbance attenuation (Kose and Jabbari 1999b), linear parameter-varying (LPV) systems (Kose and Jabbari 1999a), positive real synthesis (Mahmoud and Xie 2000), and $\mathcal{H}_\infty$ control (Xie et al. 1992). In Kose and Jabbari (1999b) the authors develop a two-step procedure for the design of output-feedback controllers for continuous-time systems and provide conditions under which the two stages of the design can be solved sequentially. These conditions, however, restrict the class of systems that can be dealt with by the proposed approach to minimum-phase, left-invertible systems. The same idea has been used in Kose and Jabbari (1999a), but extended to deal with LPV systems in which only some of the parameters are measured and the others are treated as uncertainty. In Mahmoud and Xie (2000) the output-feedback design of positive real systems is investigated by expressing the uncertainty in an LFT form and recasting the problem to a simplified, but still non-linear, problem independent of the uncertainties. A possible way, based on eigenvalue assignment, to solve the non-linear optimization problem is proposed that determines the output-feedback controller. This approach is applicable to square systems only. In the case when the uncertainty consists of one full uncertainty block it was shown in Xie et al. (1992) how the problem can be transformed into a standard $\mathcal{H}_\infty$ problem along a line search for a single scalar. However, as argued in Kose and Jabbari (1999b), this approach may turn out to be too conservative in cases when the uncertainty contains repeated real scalars.

It is well-known that most of the output-feedback controller design problems are representable in terms of bilinear (or rather bi-affine) matrix inequalities (VanAntwerp and Braatz 2000), which however are in general NP-hard (Toker and Özbay 1995). This means that any algorithm which is guaranteed to find the global optimum cannot be expected to have a polynomial time complexity.

The method proposed in this chapter belongs to the class of approaches that directly aim at solving the BMI optimization problem at hand. There exist different approaches to the solution of this problem, which can be classified into global (Beran et al. 1997; Fukuda and Kojima 2001; Goh et al. 1994; Tuan and Apkarian 2000; Tuan et al. 2000a,b; VanAntwerp et al. 1997; Yamada and Hara 1998; Yamada et al. 2001) and local (Ibaraki and Tomizuka 2001; Iwasaki 1999; Iwasaki and Rotea 1997; Hassibi et al. 1999; Grigoradis and Skelton 1996). Most of the global algorithms to the BMI problem are variations of the Branch and Bound Algorithm (Tuan and Apkarian 2000; Goh et al. 1994; Fukuda and Kojima 2001; VanAntwerp et al. 1997; Beran et al. 1997). Although the major focus of all global search algorithms is the computational complexity, none of them is polynomial-time due to the NP-hardness of the problem. As a result, these approaches can currently be applied only to problems of modest size (VanAntwerp et al. 1997) with no more than just a few "complicating variables"[1] (Tuan and Apkarian 2000). Thus, the global algorithms are not practical to output-feedback

---

[1] Generally speaking, this is the minimal number of variables in the BMI problem that, if kept fixed, results in an LMI problem.

controller design problems for polytopic systems, where even small problems can result in lots of such complicating variables (for instance, in the case study presented in Section 3.6 there are 40 complicating variables).

Most of the existing local approaches, on the other hand, are computationally faster but, depending on the initial condition, may not converge to the global optimum. The simplest local approach makes use of the fact that by fixing some of the variables, the BMI problem becomes convex in the remaining variables, and vice versa, and it iterates between them (Iwasaki 1999). This is also the idea behind the well-known $D-K$ iteration for μ-synthesis (Doyle 1983). In some papers (Iwasaki 1999; Iwasaki and Rotea 1997; Iwasaki and Skelton 1995) the search is performed in other, more suitably defined search directions. Nevertheless, these type of algorithms, called coordinate descent methods in Iwasaki (1999), alternating SDP method in Fukuda and Kojima (2001), and the dual iteration in Iwasaki (1999), are not guaranteed to converge to a local solution (Goh et al. 1994; Fukuda and Kojima 2001; Yamada and Hara 1998).

Recently, interior point methods have also been developed for nonconvex semidefinite programming (SDP) problems (Leibfritz and Mostafa 2002; Hol et al. 2003; Forsgren 2000). The interior point approach tries to find an approximate solution to the nonconvex SDP problem by rewriting it as logarithmic barrier function optimization problem. The approach then finds approximate solutions to a sequence of barrier problems and in this way produces an approximate solution to the original nonconvex SDP problem. In Leibfritz and Mostafa (2002) a trust region method is proposed for the design of optimal static output-feedback gains. This is a nonconvex BMI problem (Leibfritz 2001).

Another local approach is the so-called path-following method (Hassibi et al. 1999), which is based on linearization. The idea is that under the assumption of small search steps the BMI problem can be approximated as an LMI problem by making use of the first-order perturbation approximation (Hassibi et al. 1999). In practise this approach can be used for problems where the required closed-loop performance is not drastically better than the open-loop system performance, to solve the actuator/sensor placement problem, as well as the controller topology design problem (Hassibi et al. 1999). Similar is the continuation algorithm proposed in Collins et al. (1999) that basically consists in iterating between two LMI problems each obtained by linearization using first order perturbation approximations. Yet another local approach is the rank-minimization method (Ibaraki and Tomizuka 2001). Although convergence is established for a suitably modified problem, there are no guarantees that the solution to this modified problem will be feasible for the original BMI problem. The XY-centering algorithm, proposed in Iwasaki and Skelton (1995) is also an alternative local approach, which focusses on a subclass of BMI problems in which the non-convexity can be expressed in the form $X=Y^{-1}$, and is thus applicable to a restricted class of controller design problems. Finally, the method of centers (Goh et al. 1994) has guaranteed local convergence provided that a feasible initial condition is given. It is, however, the computationally most involving approach, and it is also known that it can experience numerical problems during some iterations (Fukuda and Kojima 2001).

Similarly to the method of centers, the approach in this chapter performs local optimization over a suitably defined non-convex function at each iteration. It enjoys the property of guaranteed convergence to a local optimum, while at the same time is computationally faster and numerically more reliable than the method of centers. In addition to that, a two-step procedure is proposed for the design of an initially feasible controller. At the first step an optimal robust mixed $\mathcal{H}_2/\mathcal{H}_\infty$/pole-placement state-feedback gain F is designed. This gain F is consequently kept fixed during the design of the remaining state-space matrices of the dynamic output-feedback controller. Although the first step is convex, the second one remains non-convex. However, by constraining a Lyapunov function for the closed-loop system to have a block-diagonal structure, this second step is easily transformed into an LMI optimization problem.

The chapter is organized as follows. In Section 3.2 the notation is defined and the problem is formulated. The proposed algorithm for locally optimal controller design is next presented in Section 3.3. For the purposes of its initialization, a computational scheme is proposed to find an initial feasible controller in Section 3.4. Here a mixed $\mathcal{H}_2/\mathcal{H}_\infty$/pole-placement criterion is considered. A summary of the complete algorithm is given in Section 3.5. In Section 3.6 the design approach is tested on a case study with a diesel actuator benchmark model and, in addition, a comparison is made between several existing methods for local BMI optimization. Finally, Section 3.7 concludes the chapter.

3.2 Preliminaries and Problem Formulation

3.2.1 Notation

The symbol $\star$ in LMIs will denote entries that follow from symmetry. In addition to that the notation $Sym(A) = A + A^*$ will also be used. Boldface capital letters denote variable matrices appearing in matrix inequalities, and boldface small letters – vector variables. The convex hull of a set of matrices $\mathcal{S} = \{M_1, \dots, M_N\}$ is denoted as $\mathbf{co}\{\mathcal{S}\}$, and is defined as the intersection of all convex sets containing all elements of $\mathcal{S}$. Also used is the notation $\langle A, B\rangle = \mathbf{trace}(A^TB)$ for any matrices A and B of appropriate dimensions, and $\|A\|_F$ denotes the Frobenius norm of A. $\mathcal{L}_2$ is the space of square integrable signals. The notation $\|x\|_2$ is used for the vector 2-norm (i.e. $(x^Tx)^{1/2}$) as well as for the signal 2-norm (i.e. $(\int_0^\infty x(t)^Tx(t)dt)^{1/2}$ for continuous-time signal x, and $(\sum_0^\infty x(k)^Tx(k))^{1/2}$). The set of eigenvalues of a matrix A will be denoted as $\lambda(A)$, while for a complex number $z \in \mathbb{C}$, the complex conjugate is denoted as $\bar{z}$. The direct sum of matrices A_i, $i = 1, 2, \dots, n$ will be denoted as

$$\bigoplus_{i=1}^{n} A_i = A_1 \oplus \cdots \oplus A_n \triangleq \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_n \end{bmatrix}.$$

Also, v_i will denote the i-th element of the vector v.

The projection onto the cone of symmetric positive-definite matrices is defined as

$$\mathbf{\Pi}^+[A] \doteq \arg\min_{\boldsymbol{S}\geq 0} \|A - \boldsymbol{S}\|_F. \tag{3.1}$$

Similarly, the projection onto the cone of symmetric negative-definite matrices is defined as

$$\mathbf{\Pi}^{-}[A] \doteq \arg\min_{\boldsymbol{S}\leq 0} \|A - \boldsymbol{S}\|_F. \tag{3.2}$$

These projections have some useful properties summarized in Chapter 2, Lemma 2.1 on page 36.

Finally, for two matrices $A = [a_{ij}] \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, the Kronecker product of A and B is defined as

$$A \otimes B \doteq \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{mp\times nq}.$$

In the remaining part of this section we summarize some existing results for system analysis and controller synthesis which lie at the basis of the developments in the next Section.

3.2.2 Output-Feedback Passive FTC

In the introductory Chapter 1 the problem of passive FTC was defined in (1.19) on page 26 as the problem of designing a controller that achieves robust closed-loop stability and performance for certain faults f and model uncertainties δ. As in Chapter 2, we represent here f and δ as one uncertainty

$$\Delta = \begin{bmatrix} \delta \\ f \end{bmatrix},$$

so that problem (1.19) becomes equivalent to the following worst-case minimization problem

$$(\mathcal{P}_{\mathcal{O}}): \ K^* = \arg\min_{K} \max_{\Delta\in\boldsymbol{\Delta}} J(G_\Delta(\sigma), K). \tag{3.3}$$

In Chapter 2 this problem was considered in the state-feedback case for which it can be rewritten in the form of a robust LMI. As argued in the introduction above, such LMI representation is, unfortunately, not possible in the output-feedback case. The later is considered in this chapter in a deterministic setting (as opposed by the probabilistic setting from Chapter 2).

To be more specific, this chapter focuses on the problem (3.3) in the case when the controller $K = K(\sigma)$ is a dynamic system with the same order as that of the plant (i.e. *full order* controller). Its input is the measured output of the controlled system. The plant $G_\Delta(\sigma)$ may either be discrete-time or continuous-time system. Furthermore, *polytopic uncertainty* representation is assumed; in other words, if $(A^\Delta, B^\Delta, C^\Delta, D^\Delta)$ is the state-space representation of $G_\Delta(\sigma)$, then it is assumed that N ($N < \infty$) vertex systems (A_i, B_i, C_i, D_i) are given such that

$$\underbrace{\left\{\begin{bmatrix} A^\Delta & B^\Delta \\ C^\Delta & D^\Delta \end{bmatrix} : \Delta \in \boldsymbol{\Delta}\right\}}_{\mathcal{M}} \subseteq \underbrace{\mathbf{co}\left\{\begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} : i = 1, 2, \dots, N.\right\}}_{\mathbb{S}_{\mathbf{co}}}$$

Remark 3.1 *We note here that assuming the uncertainty as polytopic might lead to the introduction of conservatism when the polytopic set* $\mathbb{S}_{\mathbf{co}}$ *is not exactly equal to the real uncertainty set* $\mathcal{M}$*, i.e. in cases when* $\mathbb{S}_{\mathbf{co}}/\mathcal{M} \neq \emptyset$*.*

Finally, the performance index $J(\cdot, \cdot)$ in (3.3) considered in this chapter represents a multiobjective $\mathcal{H}_2$/$\mathcal{H}_\infty$/pole-placement design problem. Before formulating this worst-case optimization problem mathematically we proceed in the next section by presenting some existing results for robust performance analysis of uncertain systems that will be useful later on in the chapter.

3.2.3 $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Norm Computation for Uncertain Systems

Consider the uncertain state-space model

$$\mathcal{S}_a(\sigma, \Delta) : \left\{ \begin{array}{rcl} \sigma x & = & A^\Delta x + B^\Delta \xi \\ z & = & C^\Delta x + D^\Delta \xi \end{array} \right. \tag{3.4}$$

where $x(t) \in \mathbb{R}^n$ is the system state, $z(t) \in \mathbb{R}^{n_z}$ is the controlled output of the system, and $\xi(t) \in \mathbb{R}^{n_\xi}$ is the disturbance to the system, and where the symbol σ represents the s-operator (*i.e.* the time-derivative operator) for continuous-time systems, and the z-operator (i.e. the shift operator) for discrete-time systems. Define the matrix

$$M_{an}^\Delta \doteq \left[\begin{array}{cc} A^\Delta & B^\Delta \\ C^\Delta & D^\Delta \end{array} \right] \tag{3.5}$$

where the subscript “an” denotes that it will be used for the purposes of analysis only. Later on, a similar matrix for the synthesis problem will be defined. The matrices $(A^\Delta, B^\Delta, C^\Delta, D^\Delta)$ in (3.4) are assumed unknown, not measurable, and possibly time-varying, but are known to lie in a given convex set $\mathcal{M}_{an}$, defined as

$$\mathcal{M}_{an} \doteq \mathbf{co} \left\{ \left[\begin{array}{cc} A_1 & B_1 \\ C_1 & D_1 \end{array} \right], \ldots, \left[\begin{array}{cc} A_N & B_N \\ C_N & D_N \end{array} \right] \right\}. \tag{3.6}$$

The following Lemma, which can be found in e.g. (Chilali et al. 1999), can be used to check whether the eigenvalues of a matrix are all located inside an LMI region.

Lemma 3.1 *Let* $\mathcal{A}$ *be a real matrix, and define the LMI region*

$$\mathcal{D} \doteq \{z \in \mathbb{C} : \ L_{\mathcal{D}} + Sym(zM_{\mathcal{D}}) < 0\}, \tag{3.7}$$

for some given real matrices $L_{\mathcal{D}} = L_{\mathcal{D}}^T$ *and* $M_{\mathcal{D}}$*. Then* $\lambda(\mathcal{A}) \subset \mathcal{D}$ *if and only if there exists a matrix* $\boldsymbol{P} = \boldsymbol{P}^T > 0$ *such that*

$$L_{\mathcal{D}} \otimes \boldsymbol{P} + Sym(M_{\mathcal{D}} \otimes (\boldsymbol{P}\mathcal{A})) < 0. \tag{3.8}$$

The class of LMI regions, defined in Equation (3.8), is fairly general – it can represent convex regions that are symmetric with respect to the real axis (Gahinet et al. 1995).

In order to be able to deal with linear time-varying systems (LTV), like those that are subject to faults, we need to extend the notions of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms

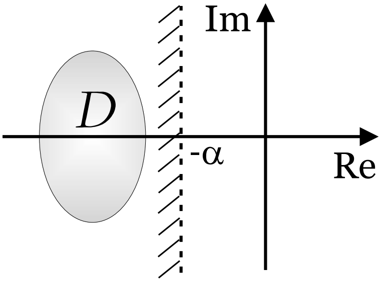


Figure 3.1: $\mathcal{D}$-stability region.

that are usually defined for linear time-invariant (LTI) systems. In addition to that, since the notion of pole is also not defined for LTV systems we will make use of the so-called $\mathcal{D}$-stability.

Consider the LTV system (3.4) described by the operator $\mathcal{S}_a(\sigma, \Delta)$ that maps the system input ξ to the system output z. The $\mathcal{H}_\infty$-norm of $\mathcal{S}_a(\sigma, \Delta)$ is then defined as the $\mathcal{L}_2$-induced gain (Apkarian and Gahinet 1995; Scherer 1996)

$$\|\mathcal{S}_a(\sigma, \Delta)\|_\infty \doteq \sup_{\xi \in \mathcal{L}_2} \frac{\|\mathcal{S}_a(\sigma, \Delta)\xi\|_2}{\|\xi\|_2}.$$

For LTI systems and in the absence of the uncertainty the $\mathcal{L}_2$-induced gain coincides with the standard $\mathcal{H}_\infty$-norm.

The $\mathcal{H}_2$-norm can also be extended to LTV systems in the spirit of Peters and Stoorvogel (1994); Scherer (1996). This can be done by using the stochastic interpretation of the $\mathcal{H}_2$-norm. With the input signal ξ being a white Gaussian noise the $\mathcal{H}_2$-norm of the operator $\mathcal{S}_a(\sigma, \Delta)$ is defined as

$$\|\mathcal{S}_a(\sigma, \Delta)\|_2^2 \doteq \sup \mathcal{E} \left\langle \mathcal{S}_a(\sigma, \Delta)\xi, \mathcal{S}_a(\sigma, \Delta)\xi \right\rangle,$$

The so defined $\mathcal{H}_2$-norm represents the maximum output variance when the input is a white Gaussian noise, and is thus a generalization of the genuine $\mathcal{H}_2$-norms of LTI systems (Scherer 1996).

For LTV systems the notion of pole (or pole-placement) is undefined. For that reason we make use of the notion of $\mathcal{D}$-stability of a time-varying matrix $A(\Delta)$ which is equivalent to the requirement that at each time instant the eigenvalues of the matrix are located in the LMI region $\mathcal{D}$. For continuous-time systems, in particular, if the region $\mathcal{D}$ is contained in the half-plane $\{z \in \mathbb{C} : \ 2\alpha + z + \bar{z} < 0\}$ (see Figure 3.1) for some positive scalar α, then exponential decay with decay rate α of the transients is guaranteed for all possible trajectories $\Delta(t)$ (Chilali et al. 1999), i.e. $\exists M > 0$ such that

$$\|e^{A(\Delta(t))t}\| \leq Me^{-\alpha t}.$$

In (Scherer et al. 1997; Masubuchi et al. 1998) LMI conditions are provided for the evaluation of the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norm of the system (3.4) in the case when there is no uncertainty present in the system, i.e. for the case when the matrix M_{an}^{Δ} in (3.5) is exactly known and time-invariant. The following two results are immediate generalizations to the case when M_{an}^{Δ} is only known to lie in a certain convex set $\mathcal{M}_{an}$, and as such will be left without proof. We note that these results are also applicable to LTV systems in view of the extensions of the norm definitions above (Apkarian and Gahinet 1995; Apkarian and Adams 1998; Scherer et al. 1997; Scherer 1996; Chilali et al. 1999). Define

$$\begin{aligned} \mathcal{L}(C^{\Delta}, \boldsymbol{W}, \boldsymbol{P}, \gamma) &= (\gamma - \mathbf{trace}(\boldsymbol{W})) \oplus \begin{bmatrix} \boldsymbol{W} & C^{\Delta} \\ \star & \boldsymbol{P} \end{bmatrix}, \\ \mathcal{M}_{CT}(A^{\Delta}, B^{\Delta}, \boldsymbol{P}) &= \begin{bmatrix} -Sym(\boldsymbol{P}A^{\Delta}) & \boldsymbol{P}B^{\Delta} \\ \star & I \end{bmatrix}, \\ \mathcal{M}_{DT}(A^{\Delta}, B^{\Delta}, \boldsymbol{P}) &= \begin{bmatrix} \boldsymbol{P} & \boldsymbol{P}A^{\Delta} & \boldsymbol{P}B^{\Delta} \\ \star & \boldsymbol{P} & 0 \\ \star & \star & I \end{bmatrix}. \end{aligned} \tag{3.9}$$

Lemma 3.2 ($\mathcal{H}_2$ norm) *Consider the system (3.4) with $D^{\Delta} = 0$. Then*

$$\sup_{M_{an}^{\Delta} \in \mathcal{M}_{an}} \|\mathcal{S}_a(\sigma, \Delta)\|_2^2 < \gamma$$

if there exist matrices $\boldsymbol{P} = \boldsymbol{P}^T$ and $\boldsymbol{W} = \boldsymbol{W}^T$ such that for all $M_{an}^{\Delta} \in \mathcal{M}_{an}$

$$\begin{aligned} &\mathcal{L}(C^{\Delta}, \boldsymbol{W}, \boldsymbol{P}, \gamma) \oplus \mathcal{M}_{CT}(A^{\Delta}, B^{\Delta}, \boldsymbol{P}) > 0, \ \textit{(continuous case)}, \\ &\mathcal{L}(C^{\Delta}, \boldsymbol{W}, \boldsymbol{P}, \gamma) \oplus \mathcal{M}_{DT}(A^{\Delta}, B^{\Delta}, \boldsymbol{P}) > 0, \ \textit{(discrete case)}. \end{aligned} \tag{3.10}$$

Lemma 3.3 ($\mathcal{H}_\infty$ norm) *Consider the system (3.4). Then*

$$\sup_{M_{an}^{\Delta} \in \mathcal{M}_{an}} \|\mathcal{S}_a(\sigma, \Delta)\|_\infty^2 < \gamma$$

if there exists a matrix $\boldsymbol{P} = \boldsymbol{P}^T$ such that for all $M_{an}^{\Delta} \in \mathcal{M}_{an}$

$$\begin{aligned} &\boldsymbol{P} \oplus \begin{bmatrix} \mathcal{M}_{CT}(A^{\Delta}, B^{\Delta}, \boldsymbol{P}) & \begin{bmatrix} C^{\Delta}, & D^{\Delta} \end{bmatrix}^T \\ \star & \gamma I \end{bmatrix} > 0, \ \textit{(cont. case)}, \\ &\begin{bmatrix} \mathcal{M}_{DT}(A^{\Delta}, B^{\Delta}, \boldsymbol{P}) & \begin{bmatrix} 0, & C^{\Delta}, & D^{\Delta} \end{bmatrix}^T \\ \star & \gamma I \end{bmatrix} > 0, \ \textit{(discrete case)}. \end{aligned} \tag{3.11}$$

The infinite number of LMIs in Lemmas 3.2 and 3.3 over all possible elements of the set $\mathcal{M}_{an}$ can be substituted by a finite number of LMIs by using the fact that the set $\mathcal{M}_{an}$ is convex. This can be achieved by substituting the matrices (A_i, B_i, C_i, D_i) from $(A^{\Delta}, B^{\Delta}, C^{\Delta}, D^{\Delta})$ in the LMIs (3.10) and (3.11), and then searching for a feasible solution for all $i = 1, \ldots, N$.

Remark 3.2 *Note that due to the fact that Lemmas 3.2 and 3.3 provide only sufficient conditions. The reason for that is that the same Lyapunov matrix $\boldsymbol{P}$ is used for all values of the uncertainties. This constraint on the Lyapunov matrix leads to the introduction of conservatism that in some applications might be too high.*

3.2.4 Problem Formulation

We next focus our attention to the synthesis problem. To this end, consider the following uncertain system

$$\mathcal{S}_s(\sigma,\Delta):\left\{\begin{array}{rcl}\sigma x & = & A^{\Delta}x+B^{\Delta}_{\xi}\xi+B^{\Delta}_{u}u\\ z & = & C^{\Delta}_{z}x+D^{\Delta}_{z\xi}\xi+D^{\Delta}_{zu}u\\ y & = & C^{\Delta}_{y}x+D^{\Delta}_{y\xi}\xi+D^{\Delta}_{yu}u\end{array}\right. \tag{3.12}$$

where the signals x, ξ, and z have the same meaning and the same dimensions as in (3.4), and where $u \in \mathbb{R}^m$ is the control action, and $y \in \mathbb{R}^p$ is the measured output.

Similarly as in Section 3.2.3, we define the matrix

$$M^{\Delta}_{syn} \doteq \begin{bmatrix} A^{\Delta} & B^{\Delta}_{\xi} & B^{\Delta}_{u}\\ C^{\Delta}_{z} & D^{\Delta}_{z\xi} & D^{\Delta}_{zu}\\ C^{\Delta}_{y} & D^{\Delta}_{y\xi} & D^{\Delta}_{yu}\end{bmatrix} \tag{3.13}$$

where the subscript "syn" denotes that it will now be used for the purposes of synthesis. We also define the convex set

$$\mathcal{M}_{syn} \doteq \textbf{co}\left\{\begin{bmatrix} A_i & B_{\xi,i} & B_{u,i}\\ C_{z,i} & D_{z\xi,i} & D_{zu,i}\\ C_{y,i} & D_{y\xi,i} & D_{yu,i}\end{bmatrix} : i=1,2,\ldots,N.\right\}. \tag{3.14}$$

Interconnected to system (3.12) is the following full-order dynamic output-feedback controller

$$\mathcal{C}_{\sigma}:\left\{\begin{array}{rcl}\sigma x^c & = & A_c x^c + B_c y\\ u & = & F x^c\end{array}\right. \tag{3.15}$$

with $x^c \in \mathbb{R}^n$ its state. This yields the closed-loop system

$$\mathcal{S}_{cl}(\sigma,\Delta):\left\{\begin{array}{rcl}\sigma x_{cl} & = & A^{\Delta}_{cl}x_{cl}+B^{\Delta}_{cl}\xi\\ z & = & C^{\Delta}_{cl}x_{cl}+D^{\Delta}_{cl}\xi\end{array}\right. \tag{3.16}$$

where it is denoted $x_{cl}^T = [x^T,\ (x^c)^T]$, and

$$\left[\begin{array}{c|c} A^{\Delta}_{cl} & B^{\Delta}_{cl}\\ \hline C^{\Delta}_{cl} & D^{\Delta}_{cl}\end{array}\right] \doteq \left[\begin{array}{cc|c} A^{\Delta} & B^{\Delta}_{u}F & B^{\Delta}_{\xi}\\ B_c C^{\Delta}_{y} & A_c + B_c D^{\Delta}_{yu}F & B_c D^{\Delta}_{y\xi}\\ \hline C^{\Delta}_{z} & D^{\Delta}_{zu}F & D^{\Delta}_{z\xi}\end{array}\right]. \tag{3.17}$$

This chapter addresses the following problem.

Multiobjective Design: Consider the system (3.12). Given positive scalars α_2 and α_∞ and a convex set $\mathcal{M}_{syn}$, defined in Equation (3.14), find constant matrices A_c, B_c, and F, parametrizing the controller (3.15), that solve the following constrained optimization problem

$$\begin{array}{ll} & \min\limits_{\gamma_2,\gamma_\infty,A_c,B_c,F} \alpha_2\gamma_2+\alpha_\infty\gamma_\infty\\ \text{subject to:} & \\ \mathcal{H}_2\text{ objective:} & \sup\limits_{M^{\Delta}_{syn}\in\mathcal{M}_{syn}} \|L_2(\mathcal{S}_{cl}(\sigma,\Delta)-D^{\Delta}_{cl})R_2\|_2^2 < \gamma_2,\\ \mathcal{H}_\infty\text{ objective:} & \sup\limits_{M^{\Delta}_{syn}\in\mathcal{M}_{syn}} \|L_\infty\mathcal{S}_{cl}(\sigma,\Delta)R_\infty\|_\infty^2 < \gamma_\infty,\\ \text{Pole-placement:} & \lambda(A^{\Delta}_{cl})\in\mathcal{D},\ \forall M^{\Delta}_{syn}\in\mathcal{M}_{syn}.\end{array} \tag{3.18}$$

where the operator $\mathcal{S}_{cl}(\sigma, \Delta)$ is defined in (3.16), and where the matrices L_2, R_2, L_∞, and R_∞, are used to select the desired input-output channels that need to satisfy the required constraint in (3.18).

As discussed in the introduction, this problem is not convex and is NP-hard. In the next section we will present a new algorithm which can be used for finding a locally optimal solution to the problem defined in (3.18). As most local approaches, this approach requires an initially feasible solution from which the local optimization is initiated. For the purposes of its initialization, a computationally fast approach based on LMIs for finding an initially feasible controller is later on proposed in Section 3.4. A summary of the complete algorithm is given in Section 3.5.

3.3 Locally Optimal Robust Controller Design

It is well-known that for systems with polytopic uncertainty the output-feedback controller design problem can be written as BMIs in the general form (3.20) (VanAntwerp and Braatz 2000). In this section a method for solving BMI problems is proposed. To this end, define the following N biaffine functions

$$\mathcal{BMI}^{(k)}(\boldsymbol{x}, \boldsymbol{y}) \doteq F_{00}^{(k)} + \sum_{i=1}^{N_1} F_{i0}^{(k)} \boldsymbol{x}_i + \sum_{j=1}^{N_2} F_{0j}^{(k)} \boldsymbol{y}_j + \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} F_{ij}^{(k)} \boldsymbol{x}_i \boldsymbol{y}_j, \qquad (3.19)$$

where $F_{ij}^{(k)} = (F_{ij}^{(k)})^T$, $i = 0, 1, \ldots, N_1$, $j = 0, 1, \ldots, N_2$, $k = 1, \ldots, M$ are given symmetric matrices. In this chapter we consider the following BMI optimization problem

$$(\mathcal{P}) : \left\{ \begin{array}{ll} & \min \gamma, \ \text{ over } \boldsymbol{x}, \ \boldsymbol{y}, \text{ and } \gamma \\ & \\ \text{subject to} & \mathcal{BMI}^{(k)}(\boldsymbol{x}, \boldsymbol{y}) \leq 0, \ \ k = 1, 2, \ldots, M, \\ & \langle c, \boldsymbol{x} \rangle + \langle d, \boldsymbol{y} \rangle \leq \gamma, \\ & \underline{x} \leq \boldsymbol{x} \leq \overline{x}, \\ & \underline{y} \leq \boldsymbol{y} \leq \overline{y} \end{array} \right. \qquad (3.20)$$

where $\underline{x}, \overline{x} \in \mathbb{R}^{N_1}$ and $\underline{y}, \overline{y} \in \mathbb{R}^{N_2}$ are given vectors with finite elements. This problem is known to be NP-hard (Toker and Özbay 1995). The bounds on the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ in (3.20) are included here for technical reasons that will become clear shortly. The problem of selecting these bounds in practise is not critical – taking the upper bounds large enough (e.g. 10^{10}), and the lower bounds small enough is often sufficient. Notice that in this way one could also ensure, for implementation reasons, that the resulting controller does not have excessively large entries in its state-space matrices.

It should also be pointed out that the BMI problem defined in (3.20) actually addresses a wider class of problems than those represented by (3.18), e.g. the design of reduced order output-feedback control (Safonov et al. 1994). However, the focus of the chapter is restricted to (3.18) since the initial controller design method, discussed later on in Section 3.4, is developed only for the case of full order output-feedback control problems.

Let us, for now, consider the feasibility problem for a fixed γ. Denote

$$\begin{aligned}&\mathcal{BMI}^{(M+1)}(\boldsymbol{x},\boldsymbol{y}) \doteq \langle c,\boldsymbol{x}\rangle + \langle d,\boldsymbol{y}\rangle - \gamma,\\ &\mathcal{BMI}^{(M+2)}(\boldsymbol{x},\boldsymbol{y}) \doteq \boldsymbol{x} - \overline{x},\ \mathcal{BMI}^{(M+3)}(\boldsymbol{x},\boldsymbol{y}) \doteq \underline{x} - \boldsymbol{x},\\ &\mathcal{BMI}^{(M+4)}(\boldsymbol{x},\boldsymbol{y}) \doteq \boldsymbol{y} - \overline{y},\ \mathcal{BMI}^{(M+5)}(\boldsymbol{x},\boldsymbol{y}) \doteq \underline{y} - \boldsymbol{y},\end{aligned}$$

and let $N = M + 5$. The feasibility problem is then defined as

$$(\mathcal{FP}) : \begin{cases} \text{Find } (\boldsymbol{x},\boldsymbol{y}) \\ \\ \text{such that } \bigoplus_{k=1}^{N} \mathcal{BMI}^{(k)}(\boldsymbol{x},\boldsymbol{y}) \leq 0. \end{cases} \tag{3.21}$$

Define the following cost function

$$v_\gamma(\boldsymbol{x},\boldsymbol{y}) \doteq \left\| \boldsymbol{\Pi}^+ \left[\bigoplus_{k=1}^{N} \mathcal{BMI}^{(k)}(\boldsymbol{x},\boldsymbol{y}) \right] \right\|_F^2 \geq 0. \tag{3.22}$$

From the definition of the projection $\boldsymbol{\Pi}^+[\cdot]$, and from the properties of the Frobenius norm we can write

$$v_\gamma(\boldsymbol{x},\boldsymbol{y}) = \sum_{k=1}^{N} \left\| \boldsymbol{\Pi}^+ \left[\mathcal{BMI}^{(k)}(\boldsymbol{x},\boldsymbol{y}) \right] \right\|_F^2 \doteq \sum_{k=1}^{N} v_\gamma^{(k)}(\boldsymbol{x},\boldsymbol{y}). \tag{3.23}$$

It is therefore clear that

$$(\mathcal{FP}) \text{ is feasible } \Leftrightarrow\ 0 = \min_{\boldsymbol{x},\boldsymbol{y}} v_\gamma(\boldsymbol{x},\boldsymbol{y}).$$

In this way we have rewritten the BMI feasibility problem $(\mathcal{FP})$ as an optimization problem, where the goal is now to find a local minimum of v_γ. However, the function $v_\gamma(\boldsymbol{x},\boldsymbol{y})$ is not convex. Even worse, it may have multiple local minima. Now, if $(\boldsymbol{x}_{opt},\boldsymbol{y}_{opt})$ is a local minimum for v_γ and is such that $v_\gamma(\boldsymbol{x}_{opt},\boldsymbol{y}_{opt}) = 0$, then $(\boldsymbol{x}_{opt},\boldsymbol{y}_{opt})$ is also a feasible solution to $(\mathcal{FP})$. However, if $(\boldsymbol{x}_{opt},\boldsymbol{y}_{opt})$ is such that $v_\gamma(\boldsymbol{x}_{opt},\boldsymbol{y}_{opt}) > 0$, then we cannot say anything about the feasibility of $(\mathcal{FP})$. The idea is then to start from a feasible solution for a given γ, and then apply the method of bisection over γ to achieve a local minimum with a desired precision, at each iteration searching for a feasible solution to $(\mathcal{FP})$. A more extensive description of this bisection algorithm is provided in Section 3.5.

Let us now concentrate on the problem of finding a local solution to

$$\min_{\boldsymbol{x},\boldsymbol{y}} v_\gamma(\boldsymbol{x},\boldsymbol{y}). \tag{3.24}$$

The goal is to develop an approach that has a guaranteed convergence to a local optimum of $v_\gamma(\boldsymbol{x},\boldsymbol{y})$. To this end, we first note that the function $v_\gamma(\boldsymbol{x},\boldsymbol{y})$ is differentiable, and we derive an expression for its gradient.

Theorem 3.1 *With continuously differentiable* $G : \mathbb{R}^{N_v} \mapsto \mathbb{R}^{q\times q}$, $G = G^T$, *and* $f : \mathbb{R}^{q\times q} \mapsto \mathbb{R}$ *defined as* $f(M) = \|\mathbf{\Pi}^{+}[M]\|_F^2$, *the function*

$$(f \circ G)(\boldsymbol{v}) \doteq \|\mathbf{\Pi}^{+}[G(\boldsymbol{v})]\|_F^2,$$

is differentiable, and its gradient

$$\nabla(f\circ G)(\boldsymbol{v}) \doteq \left[\frac{\partial}{\partial \boldsymbol{v}_1}, \frac{\partial}{\partial \boldsymbol{v}_2}, \dots \frac{\partial}{\partial \boldsymbol{v}_{N_v}}\right]^T (f\circ G)(\boldsymbol{v}),$$

is given by

$$\frac{\partial}{\partial \boldsymbol{v}_i}(f\circ G)(\boldsymbol{v}) = 2\left\langle \mathbf{\Pi}^{+}[G(\boldsymbol{v})], \frac{\partial}{\partial \boldsymbol{v}_i}G(\boldsymbol{v})\right\rangle. \tag{3.25}$$

Proof: Using the properties of the projection $\mathbf{\Pi}^{+}[\cdot]$ (see Lemma 2.1 on page 36) we infer for any symmetric matrices G and ΔG, that

$$\begin{aligned} f\circ(G+\Delta G) &= \|\mathbf{\Pi}^{+}[G+\Delta G]\|_F^2 \\ &= \|G+\Delta G - \mathbf{\Pi}^{-}[G+\Delta G]\|_F^2 = \min_{\boldsymbol{S}\le 0}\|G+\Delta G-\boldsymbol{S}\|_F^2 \\ &\le \|G+\Delta G-\mathbf{\Pi}^{-}[G]\|_F^2 = \|\mathbf{\Pi}^{+}[G]+\Delta G\|_F^2 \\ &= \|\mathbf{\Pi}^{+}[G]\|_F^2 + 2\langle \mathbf{\Pi}^{+}[G], \Delta G\rangle + \|\Delta G\|_F^2. \end{aligned}$$

On the other hand,

$$\begin{aligned} & f\circ(G+\Delta G) = \|\mathbf{\Pi}^{+}[G+\Delta G]\|_F^2 = \|G+\Delta G-\mathbf{\Pi}^{-}[G+\Delta G]\|_F^2 \\ & = \|\mathbf{\Pi}^{+}[G]+\mathbf{\Pi}^{-}[G]+\Delta G-\mathbf{\Pi}^{-}[G+\Delta G]\|_F^2 \\ & \ge \|\mathbf{\Pi}^{+}[G]\|_F^2 + 2\langle\mathbf{\Pi}^{+}[G],\Delta G\rangle + 2\langle\mathbf{\Pi}^{+}[G],\mathbf{\Pi}^{-}[G]\rangle + 2\langle\mathbf{\Pi}^{+}[G], -\mathbf{\Pi}^{-}[G+\Delta G]\rangle \\ & \ge \|\mathbf{\Pi}^{+}[G]\|_F^2 + 2\langle\mathbf{\Pi}^{+}[G],\Delta G\rangle. \end{aligned}$$

Thus we have $f\circ(G+\Delta G) = f\circ G + 2\langle\mathbf{\Pi}^{+}[G],\Delta G\rangle + o(\|\Delta G\|_F)$ for any symmetric ΔG.

Now, take $\Delta G(\boldsymbol{v}) \doteq G(\boldsymbol{v}+\Delta\boldsymbol{v}) - G(\boldsymbol{v})$. Since $G(\boldsymbol{v})$ is continuously differentiable it follows that

$$G(\boldsymbol{v}+\Delta\boldsymbol{v}) = G(\boldsymbol{v}) + \sum_{i=1}^{N_v}\left(\frac{\partial}{\partial\boldsymbol{v}_i}G(\boldsymbol{v})\right)\Delta\boldsymbol{v}_i + o(\|\Delta\boldsymbol{v}\|_2).$$

Therefore

$$(f\circ G)(\boldsymbol{v}+\Delta\boldsymbol{v}) = (f\circ G)(\boldsymbol{v}) + 2\sum_{i=1}^{N_v}\left(\left\langle\mathbf{\Pi}^{+}[G(\boldsymbol{v})], \frac{\partial}{\partial\boldsymbol{v}_i}G(\boldsymbol{v})\right\rangle\Delta\boldsymbol{v}_{\boldsymbol{i}}\right) + o(\|\Delta\boldsymbol{v}\|_2).$$

Hence $(f\circ G)$ is differentiable and its partial derivatives are given by the expressions (3.25). □

The partial derivatives of our original function $v_\gamma(\boldsymbol{x},\boldsymbol{y})$ can then be directly derived using the result of Theorem 3.1:

$$\frac{\partial}{\partial\boldsymbol{x}_i}v_\gamma(\boldsymbol{x},\boldsymbol{y}) = 2\sum_{k=1}^{N}\left\langle\mathbf{\Pi}^{+}\left[\mathcal{BMI}^{(k)}(\boldsymbol{x},\boldsymbol{y})\right], F_{i0}^{(k)} + \sum_{j=1}^{N_2}F_{ij}^{(k)}\boldsymbol{y}_j\right\rangle \tag{3.26}$$

$$\frac{\partial}{\partial\boldsymbol{y}_j}v_\gamma(\boldsymbol{x},\boldsymbol{y}) = 2\sum_{k=1}^{N}\left\langle\mathbf{\Pi}^{+}\left[\mathcal{BMI}^{(k)}(\boldsymbol{x},\boldsymbol{y})\right], F_{0j}^{(k)} + \sum_{i=1}^{N_1}F_{ij}^{(k)}\boldsymbol{x}_i\right\rangle \tag{3.27}$$

Note that these partial derivatives are continuous functions (see Lemma 2.1), so that $v_\gamma \in C^1$. Note also, that a lower bound on the cost function in (3.20) can always be obtained by solving the so-called relaxed LMI optimization problem (Tuan and Apkarian 2000)

$$\begin{cases} \gamma_{LB} = \min\limits_{\boldsymbol{x},\boldsymbol{y}} \langle c, \boldsymbol{x} \rangle + \langle d, \boldsymbol{y} \rangle, \\ \\ \text{subject to: } \boldsymbol{x} \in [\underline{x}, \overline{x}], \boldsymbol{y} \in [\underline{y}, \overline{y}], \boldsymbol{w}_{ij} \in [\underline{w}_{ij}, \overline{w}_{ij}] \\ F_{00}^{(k)} + \sum\limits_{i=1}^{N_1} F_{i0}^{(k)} \boldsymbol{x}_i + \sum\limits_{j=1}^{N_2} F_{0j}^{(k)} \boldsymbol{y}_j + \sum\limits_{i=1}^{N_1} \sum\limits_{j=1}^{N_2} F_{ij}^{(k)} \boldsymbol{w}_{ij} \leq 0, \\ \text{for } k = 1, 2, \ldots, M, \end{cases} \tag{3.28}$$

where

$$\begin{aligned} \underline{w}_{ij} &= \min\{\underline{x}_i \underline{y}_j, \underline{x}_i \overline{y}_j, \overline{x}_i \underline{y}_j, \overline{x}_i \overline{y}_j\}, \\ \overline{w}_{ij} &= \max\{\underline{x}_i \underline{y}_j, \underline{x}_i \overline{y}_j, \overline{x}_i \underline{y}_j, \overline{x}_i \overline{y}_j\}. \end{aligned}$$

If this problem is not feasible, then the original BMI problem is also not feasible.

Now that it was shown that the function v_γ is C^1 and an expression for its gradient has been derived, a quasi-Newton type optimization algorithm, adopted from (Li and Fukushima 2001), can be used for finding a local minimum of $v_\gamma \in C^1$. It is summarized in Algorithm 3.1 on page 79.

As a stopping condition usually $\|g^{(k)}\| \leq \epsilon$ is used for some sufficiently small scalar ϵ.

The convergence of this is established in (Li and Fukushima 2001) under the assumption that,

(a) the level set $\Omega = \{\boldsymbol{x}, \boldsymbol{y} : \ v_\gamma(\boldsymbol{x}, \boldsymbol{y}) \leq v_\gamma(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}))\}$ is bounded,

(b) $v_\gamma(\boldsymbol{x}, \boldsymbol{y})$ is continuously differentiable on Ω, and

(c) there exists a constant $L > 0$ such that the global Lipschitz condition holds:

$$\|g(\boldsymbol{x}, \boldsymbol{y}) - g(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})\|_2 \leq L \left\| \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} - \begin{bmatrix} \bar{\boldsymbol{x}} \\ \bar{\boldsymbol{y}} \end{bmatrix} \right\|_2, \ \forall (\boldsymbol{x}, \boldsymbol{y}), (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \in \Omega.$$

For the problem considered in this section the level set Ω is compact (see equation (3.20)), so that condition (a) holds. Condition (b) was shown in Theorem 3.1. Condition (c) follows by observing that the projection $\boldsymbol{\Pi}^+[\cdot]$ is Lipschitz, and hence, since $\mathcal{BMI}^{(k)}(\boldsymbol{x}, \boldsymbol{y})$ is smooth, the functions in (3.26) and (3.27) satisfy a local Lipschitz condition. The compactness of the set Ω then implies the desired global Lipschitz condition.

Note that the optimization problem discussed above applies to a more general class of problems with smooth nonlinear matrix inequality (NMI) constraints. However, finding an initially feasible solution to start the local optimization is a rather difficult problem, for which reason NMI problems fall outside the scope of this chapter. It also needs to be noted here that any algorithm with guaranteed convergence to a local minimum could be used instead of the one presented in Algorithm 3.1.

In the next section we focus on the problem of finding an initial feasible solution to the BMI optimization problem.

Algorithm 3.1 (Cautious BFGS method (Li and Fukushima 2001))

Initialization: $(x^{(0)}, y^{(0)})$, symmetric positive definite matrix $B^{(0)} \in \mathbb{R}^{(N_1+N_2)\times(N_1+N_2)}$, constants $0 < \sigma_1, \rho < 1$, $\alpha > 0$, and $\varepsilon > 0$. Set $k = 0$, denote the gradient of $v_\gamma(\boldsymbol{x}, \boldsymbol{y})$ evaluated at $(\boldsymbol{x}^{(\boldsymbol{k})}, \boldsymbol{y}^{(\boldsymbol{k})})$ as

$$\begin{aligned} g^{(k)} &= g(\boldsymbol{x}^{(\boldsymbol{k})}, \boldsymbol{y}^{(\boldsymbol{k})}) \\ &= \begin{bmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_{N_1}} & \frac{\partial}{\partial y_1} & \cdots & \frac{\partial}{\partial y_{N_2}} \end{bmatrix}^T v_\gamma(\boldsymbol{x}^{(\boldsymbol{k})}, \boldsymbol{y}^{(\boldsymbol{k})}), \end{aligned}$$

with the partial derivatives given by the expressions (3.26) and (3.27). Perform the steps

Step 1. Solve the equation $B^{(k)}p^{(k)} + g^{(k)} = 0$ to get $p^{(k)} \in \mathbb{R}^{(N_1+N_2)}$. Partition $p^{(k)} = \begin{bmatrix} p^x \\ p^y \end{bmatrix}$ with $p^x \in \mathbb{R}^{N_1}$ and $p^y \in \mathbb{R}^{N_2}$.

Step 2. Determine a step-size $\lambda^{(k)} > 0$ by using the Armijo-type line search, i.e. take $\lambda^{(k)}$ as the largest value in the set $\{\rho^i : i = 0, 1, \dots\}$ such that the following inequality holds:

$$v_\gamma(\boldsymbol{x}^{(\boldsymbol{k})} + \lambda^{(k)}p^x, \boldsymbol{y}^{(\boldsymbol{k})} + \lambda^{(k)}p^y) \le v_\gamma(\boldsymbol{x}^{(\boldsymbol{k})}, \boldsymbol{y}^{(\boldsymbol{k})}) + \sigma_1\lambda^{(k)}(g^{(k)})^T p^{(k)}.$$

Step 3. Take

$$\begin{aligned} \boldsymbol{x}^{(\boldsymbol{k}+\boldsymbol{1})} &= \boldsymbol{x}^{(\boldsymbol{k})} + \lambda^{(k)}p^x, \\ \boldsymbol{y}^{(\boldsymbol{k}+\boldsymbol{1})} &= \boldsymbol{y}^{(\boldsymbol{k})} + \lambda^{(k)}p^y. \end{aligned}$$

Step 4. If $\left(\frac{(t^{(k)})^T s^{(k)}}{\|s^{(k)}\|_2^2} \ge \varepsilon \|g^{(k)}\|_2^\alpha\right)$ then compute

$$\begin{aligned} s^{(k)} &= [(\lambda^{(k)}p^x)^T, (\lambda^{(k)}p^y)^T]^T, \\ t^{(k)} &= g^{(k+1)} - g^{(k)}, \\ B^{(k+1)} &= B^{(k)} - \frac{B^{(k)}s^{(k)}(s^{(k)})^T B^{(k)}}{(s^{(k)})^T B^{(k)} s^{(k)}} + \frac{t^{(k)}(t^{(k)})^T}{(t^{(k)})^T s^{(k)}} \end{aligned}$$

else take $B^{(k+1)} = B^{(k)}$.

Step 4. Set $k \leftarrow k + 1$ and go to Step 1.

3.4 Initial Robust Multiobjective Controller Design

In this Section, a two-step procedure is presented for the design of an initial feasible robust output-feedback controller. It can be summarized as follows:

Step 1: Design a robust state-feedback gain matrix $\boldsymbol{F}$ such that the multiobjective criterion of the form (3.18) is satisfied for the closed-loop system with state-feedback control $u = Fx$. This problem is convex and is considered in Subsection Section 3.4.1.

Step 2: Plug the state-feedback gain matrix F, computed at Step 1, into the original closed-loop system (3.16), and search for a solution to the multiobjective control problem, defined in Equation (3.18), in terms of the remaining unknown controller matrices $\boldsymbol{A_c}$ and $\boldsymbol{B_c}$. This problem, in contrast to the one in Step 1 above, remains non-convex. It is discussed in Section 3.4.2.

In the remaining part of this Section we proceed with proposing a solution to the problems in the two steps above.

3.4.1 Step 1: Robust Multiobjective State-Feedback Design

The state-feedback case for the system (3.12) is equivalent to taking $C_y^\Delta = I_n$, $D_{y\xi}^\Delta = 0_{n\times n_\xi}$, $D_{yu}^\Delta = 0_{n\times m}$, so that $y \equiv x$. Furthermore, we consider the constant state-feedback controller $u = Fx$, which results in the closed-loop system

$$T_{sf}(\sigma,\Delta) : \left\{ \begin{array}{rcl} \sigma x_{sf} & = & (A^\Delta + B_u^\Delta F)x_{sf} + B_\xi^\Delta \xi, \\ z & = & (C_z^\Delta + D_{zu}^\Delta F)x_{sf} + D_{z\xi}^\Delta \xi. \end{array} \right. \tag{3.29}$$

The following Theorem can be used for robust multiobjective state-feedback design for discrete-time and continuous-time systems. The proof follows after rewriting Lemmas 3.2 and 3.3 for the closed-loop system (3.29) as LMIs in

$$\boldsymbol{Q} = \boldsymbol{P}^{-1},$$

with subsequent change of variables. It will be omitted here (Scherer et al. 1997; Oliveira et al. 1999b).

Theorem 3.2 (Robust Multiobjective State-Feedback Control) *Consider the system (3.12), and assume that $C_y^\Delta = I_n$, $D_{y\xi}^\Delta = 0_{n\times n_\xi}$, $D_{yu}^\Delta = 0_{n\times m}$. Consider the controller $u = Fx$ resulting in the closed-loop system $T_{sf}(\sigma,\Delta)$ in equation (3.29). Given matrices L_2, R_2, L_∞, and R_∞, the conditions*

$$\begin{array}{l} \sup\limits_{M_{syn}^\Delta \in \mathcal{M}_{syn}} \|L_2(T_{sf}(\sigma,\Delta) - D_{z\xi}^\Delta)R_2\|_2^2 < \gamma_2, \\ \sup\limits_{M_{syn}^\Delta \in \mathcal{M}_{syn}} \|L_\infty T_{sf}(\sigma,\Delta)R_\infty\|_\infty^2 < \gamma_\infty, \\ \lambda(A^\Delta + B_u^\Delta F) \in \mathcal{D},\ \forall M_{syn}^\Delta \in \mathcal{M}_{syn}. \end{array} \tag{3.30}$$

hold if there exist matrices $\boldsymbol{Q} = \boldsymbol{Q}^T$, $\boldsymbol{W} = \boldsymbol{W}^T$, $\boldsymbol{R} = \boldsymbol{R}^T$, and $\boldsymbol{L}$ such that for all

$i = 1, \ldots, N$ *the following LMIs hold*

$$\text{PP:} \quad (-\boldsymbol{Q}) \oplus (L_{\mathcal{D}} \otimes \boldsymbol{Q} + Sym(M_{\mathcal{D}} \otimes (A_i\boldsymbol{Q} + B_{u,i}\boldsymbol{L}))) < 0, \tag{3.31}$$

$$\mathcal{H}_2\text{:} \quad (\gamma_2 - \mathbf{trace}(\boldsymbol{R})) \oplus \begin{bmatrix} \boldsymbol{R} & L_2(C_{z,i}\boldsymbol{Q} + D_{zu,i}\boldsymbol{L}) \\ \star & \boldsymbol{Q} \end{bmatrix} \oplus$$
$$\oplus \begin{cases} \begin{bmatrix} -Sym(A_i\boldsymbol{Q} + B_{u,i}\boldsymbol{L}) & B_{\xi,i}R_2 \\ \star & I \end{bmatrix} > 0 & \text{(cont. case)} \\[2ex] \begin{bmatrix} \boldsymbol{Q} & A_i\boldsymbol{Q} + B_{u,i}\boldsymbol{L} & B_{\xi,i}R_2 \\ \star & \boldsymbol{Q} & 0 \\ \star & \star & I \end{bmatrix} > 0 & \text{(discr. case).} \end{cases} \tag{3.32}$$

$$\mathcal{H}_\infty\text{:} \quad \begin{cases} \boldsymbol{Q} \oplus \\ \begin{bmatrix} -Sym(A_i\boldsymbol{Q} + B_{u,i}\boldsymbol{L}) & B_{\xi,i}R_\infty & (\boldsymbol{Q}C_{z,i}^T + \boldsymbol{L}^T D_{zu,i}^T)L_\infty^T \\ \star & I & R_\infty^T D_{z\xi,i}^T L_\infty^T \\ \star & \star & \gamma_\infty I \end{bmatrix} > 0 \\ \text{(continuous case)} \\[2ex] \begin{bmatrix} \boldsymbol{Q} & A_i\boldsymbol{Q} + B_{u,i}\boldsymbol{L} & B_{\xi,i}R_\infty & 0 \\ \star & \boldsymbol{Q} & 0 & (\boldsymbol{Q}C_{z,i}^T + \boldsymbol{L}^T D_{zu,i}^T)L_\infty^T \\ \star & \star & I & R_\infty^T D_{z\xi,i}^T L_\infty^T \\ \star & \star & \star & \gamma_\infty I \end{bmatrix} > 0. \\ \text{(discrete case)} \end{cases} \tag{3.33}$$

The state-feedback gain matrix F is then given by $F = \boldsymbol{L}\boldsymbol{Q}^{-1}$.

3.4.2 Step 2: Robust Multiobjective Output-Feedback Design

In what follows we assume that the optimal state-feedback gain F has already been computed at Step 1. In contrast to Step 1, the problem defined in Step 2 of the algorithm at the beginning of Section 3.4 is certainly non-convex in the variables $\boldsymbol{P}$, $\boldsymbol{W}$, $\boldsymbol{A_c}$, and $\boldsymbol{B_c}$ since application of Lemmas 3.2 and 3.3 to the closed-loop system in Equation (3.17) leads to non-linear matrix inequalities due to the fact that the variables $\boldsymbol{A_c}$ and $\boldsymbol{B_c}$ appear in the closed-loop system matrices $\boldsymbol{A}_{cl}^{\Delta}$ and $\boldsymbol{B}_{cl}^{\Delta}$ (for which reason the last two are typed in boldface).

Note that the function $V = x_{cl}^T \boldsymbol{P} x_{cl}$ acts as a Lyapunov function for the closed-loop system. This can easily be seen by observing that the matrix inequalities in Lemmas 3.2 and 3.3, when applied to the closed-loop system (3.16) imply $(\boldsymbol{A}_{cl}^{\Delta})^T \boldsymbol{P} \boldsymbol{A}_{cl}^{\Delta} - \boldsymbol{P} < 0$ for the discrete-time case, and $\boldsymbol{P}\boldsymbol{A}_{cl}^{\Delta} + (\boldsymbol{A}_{cl}^{\Delta})^T \boldsymbol{P} < 0$ for the continuous-time case.

The purpose of this section is to show how by introducing some conservatism by means of constraining the Lyapunov matrix $\boldsymbol{P}$ to have block-diagonal structure

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{X} & \\ & \boldsymbol{Y} \end{bmatrix}, \tag{3.34}$$

the nonlinear matrix inequalities in question can be written as LMIs. However, it can easily be seen that a necessary condition for the existence of a structured

Lyapunov matrix of the form (3.34) for $\boldsymbol{A}_{cl}^{\Delta}$ defined in (3.17) is that the matrix A^{Δ} is stable for all $M_{syn}^{\Delta} \in \mathcal{M}_{syn}$. However, this restriction can be removed by introducing a change of basis of the state vector of the closed-loop system

$$\bar{x}_{cl} = T x_{cl} = \begin{bmatrix} x \\ x - x^c \end{bmatrix}, \tag{3.35}$$

represented by the similarity transformation matrix

$$T = \begin{bmatrix} I_n & 0 \\ I_n & -I_n \end{bmatrix} = T^{-1}.$$

This changes the state-space matrices of the closed-loop system to

$$(\bar{\boldsymbol{A}}_{cl}^{\Delta}, \bar{\boldsymbol{B}}_{cl}^{\Delta}, \bar{C}_{cl}^{\Delta}, \bar{D}_{cl}^{\Delta}) = (T\boldsymbol{A}_{cl}^{\Delta}T, T\boldsymbol{B}_{cl}^{\Delta}, C_{cl}^{\Delta}T, D_{cl}^{\Delta}),$$

with

$$\begin{aligned}
\bar{\boldsymbol{A}}_{cl}^{\Delta} &= \begin{bmatrix} A^{\Delta} + B_u^{\Delta}F & -B_u^{\Delta}F \\ A^{\Delta} + B_u^{\Delta}F - \boldsymbol{B_c}C_y^{\Delta} - \boldsymbol{A_c} - \boldsymbol{B_c}D_{yu}^{\Delta}F & \boldsymbol{A_c} + \boldsymbol{B_c}D_{yu}^{\Delta}F - B_u^{\Delta}F \end{bmatrix} \\
\bar{\boldsymbol{B}}_{cl}^{\Delta} &= \begin{bmatrix} B_{\xi}^{\Delta} \\ B_{\xi}^{\Delta} - \boldsymbol{B_c}D_{y\xi}^{\Delta} \end{bmatrix} \\
\bar{C}_{cl}^{\Delta} &= \begin{bmatrix} C_z^{\Delta} + D_{zu}^{\Delta}F & -D_{zu}^{\Delta}F \end{bmatrix} \\
\bar{D}_{cl}^{\Delta} &= D_{z\xi}^{\Delta}
\end{aligned} \tag{3.36}$$

Now, searching for a structured Lyapunov matrix for this (equivalent) closed-loop system only necessitates the stability of the matrix $(A^{\Delta} + B_u^{\Delta}F)$ for all $M_{syn}^{\Delta} \in \mathcal{M}_{syn}$, which is guaranteed to hold by the design of the state-feedback gain F.

Remark 3.3 *An interesting interpretation of the transformation (3.35) and the structural constraint on $\boldsymbol{P}$ can be given as follows. If we consider the closed-loop system in the new state basis (3.35), and we restrict the Lyapunov matrix to have a block-diagonal structure as in (3.34) then the quadratic Lyapunov function takes the form*

$$V = \bar{x}_{cl}^T \boldsymbol{P} \bar{x}_{cl} = x^T \boldsymbol{X} x + (x - x^c)^T \boldsymbol{Y} (x - x^c).$$

Therefore, quadratic stability would imply that the controller state x^c converges to the system state x, so that the transformation (3.35) with the structural constraint (3.34) could be viewed as imposing an "observer structure" in the controller. For instance, for LTI systems with no uncertainty the well-known LQG controller has such an observer structure since it is based on Kalman filter and a state-feedback gain matrix. It is well known that due to the separation principle these two components of the LQG controller can be designed independently from each other. Since the state-feedback gain stabilizes the system it means that there exists $\boldsymbol{X} > 0$ such that $x^T \boldsymbol{X} x > 0$ is a Lyapunov function for the system. On the other hand, the Kalman filter also guarantees stability of the estimation error model, so that there *exists* $\boldsymbol{Y} > 0$ *such that* $(x - x^c)^T \boldsymbol{Y}(x - x^c) > 0$ *is a Lyapunov function for the estimation error model. Hence the LQG controller also results in such a diagonally-structured Lyapunov matrix as in equation (3.34) for the closed-loop system state formed by augmenting the system state with the estimation error (3.35). In the uncertainty case, considered here, the observer and the state-feedback controller are coupled (i.e. they will not be designed independent of each other), but still imposing such an "observer structure" in the controller could be motivated from the uncertainty-free case.*

We are now ready to present the following result.

Theorem 3.3 (Robust Multiobjective Output-Feedback Control) *Consider the closed-loop system* $\mathcal{S}_{cl}(\sigma, \Delta)$ *(3.16), formed by interconnecting the plant (3.12) with the dynamic output-feedback controller (3.15), in which the state-feedback gain matrix* F *is given. Then given matrices* L_2*,* R_2*,* L_∞*, and* R_∞ *of appropriate dimensions, the conditions*

$$\begin{aligned} &\sup_{M_{an}^{\Delta} \in \mathcal{M}_{an}} \|L_2(\mathcal{S}_{cl}(\sigma, \Delta) - D_{cl}^{\Delta})R_2\|_2^2 < \gamma_2, \\ &\sup_{M_{an}^{\Delta} \in \mathcal{M}_{an}} \|L_\infty \mathcal{S}_{cl}(\sigma, \Delta) R_\infty\|_\infty^2 < \gamma_\infty, \\ &\lambda(A_{cl}^{\Delta}) \in \mathcal{D},\ \forall M_{an}^{\Delta} \in \mathcal{M}_{an}. \end{aligned} \tag{3.37}$$

hold if there exist matrices $\boldsymbol{W} = \boldsymbol{W}^T$*,* $\boldsymbol{X} = \boldsymbol{X}^T$*,* $\boldsymbol{Y} = \boldsymbol{Y}^T$*,* $\boldsymbol{Z}$ *and* $\boldsymbol{G}$ *such that the following system of LMIs has a feasible solution for all* $i = 1, \ldots, N$

$$\textit{PP:} \qquad (-\boldsymbol{P}) \oplus (L_{\mathcal{D}} \otimes \boldsymbol{P} + Sym(M_{\mathcal{D}} \otimes \boldsymbol{M}_i)) < 0, \tag{3.38}$$

$$\begin{aligned} \mathcal{H}_2\text{:} \qquad & (\gamma_2 - \mathbf{trace}(\boldsymbol{W})) \oplus \begin{bmatrix} \boldsymbol{W} & L_2 \bar{C}_{cl,i} \\ \star & \boldsymbol{P} \end{bmatrix} \oplus \\ & \oplus \begin{cases} \begin{bmatrix} -Sym(\boldsymbol{M}_i) & \boldsymbol{N}_i R_2 \\ \star & I \end{bmatrix} > 0 & \textit{(continuous case)} \\ \\ \begin{bmatrix} \boldsymbol{P} & \boldsymbol{M}_i & \boldsymbol{N}_i R_2 \\ \star & \boldsymbol{P} & 0 \\ \star & \star & I \end{bmatrix} > 0 & \textit{(discrete case).} \end{cases} \end{aligned} \tag{3.39}$$

$$\mathcal{H}_\infty\text{:} \qquad \begin{cases} \boldsymbol{P} \oplus \begin{bmatrix} -Sym(\boldsymbol{M}_i) & \boldsymbol{N}_i R_\infty & \bar{C}_{cl,i}^T L_\infty^T \\ \star & I & R_\infty^T D_{z\xi,i}^T L_\infty^T \\ \star & \star & \gamma_\infty I \end{bmatrix} > 0 \\ \textit{(continuous case)} \\ \\ \begin{bmatrix} \boldsymbol{P} & \boldsymbol{M}_i & \boldsymbol{N}_i R_\infty & 0 \\ \star & \boldsymbol{P} & 0 & \bar{C}_{cl,i}^T L_\infty^T \\ \star & \star & I & R_\infty^T D_{z\xi,i}^T L_\infty^T \\ \star & \star & \star & \gamma_\infty I \end{bmatrix} > 0 \\ \textit{(discrete case)} \end{cases} \tag{3.40}$$

where the matrices $\boldsymbol{M}_i$, $\boldsymbol{N}_i$, $\boldsymbol{P}$, *and* $\bar{C}_{cl,i}$ *are defined as*

$$
\begin{gathered}
\boldsymbol{M}_i = \\
\begin{bmatrix} \boldsymbol{X}(A_i + B_{u,i}F) & -\boldsymbol{X}B_{u,i}F \\ \boldsymbol{Y}(A_i + B_{u,i}F) - \boldsymbol{Z} - \boldsymbol{G}(C_{y,i} + D_{yu,i}F) & \boldsymbol{Z} + \boldsymbol{G}D_{yu,i}F - \boldsymbol{Y}B_{u,i}F \end{bmatrix} \\
\boldsymbol{N}_i = \begin{bmatrix} \boldsymbol{X}B_{\xi,i} \\ \boldsymbol{Y}B_{\xi,i} - \boldsymbol{G}D_{y\xi,i} \end{bmatrix}, \ \boldsymbol{P} = \begin{bmatrix} \boldsymbol{X} & \\ & \boldsymbol{Y} \end{bmatrix}, \\
\bar{C}_{cl,i} = \begin{bmatrix} C_{z,i} + D_{zu,i}F & -D_{zu,i}F \end{bmatrix}.
\end{gathered}
\tag{3.41}
$$

Furthermore, the unknown matrices $\boldsymbol{A_c}$ *and* $\boldsymbol{B_c}$ *of the controller (3.15) are given by*

$$
\begin{aligned}
\boldsymbol{A_c} &= \boldsymbol{Y}^{-1}\boldsymbol{Z} \\
\boldsymbol{B_c} &= \boldsymbol{Y}^{-1}\boldsymbol{G}
\end{aligned}
\tag{3.42}
$$

Proof: From Lemmas 3.1, 3.2 and 3.3 it follows that a sufficient condition for (3.37) is that the following matrix inequalities are feasible for all values of the uncertainty

$$
\text{PP:} \qquad (-\boldsymbol{P}) \oplus (L_{\mathcal{D}} \otimes \boldsymbol{P} + Sym(M_{\mathcal{D}} \otimes (\boldsymbol{P}\bar{A}^{\Delta}_{cl}))) < 0, \tag{3.43}
$$

$$
\begin{aligned}
\mathcal{H}_2\text{:} \qquad & \mathcal{L}(L_2\bar{C}^{\Delta}_{cl}, \boldsymbol{W}, \boldsymbol{P}, \gamma) \oplus \\
& \begin{cases} \mathcal{M}_{CT}(\bar{A}^{\Delta}_{cl}, \bar{B}^{\Delta}_{cl}R_2, \boldsymbol{P}) > 0 & \text{(continuous case)} \\ \mathcal{M}_{DT}(\bar{A}^{\Delta}_{cl}, \bar{B}^{\Delta}_{cl}R_2, \boldsymbol{P}) > 0 & \text{(discrete case)} \end{cases}
\end{aligned}
\tag{3.44}
$$

$$
\mathcal{H}_\infty\text{:} \qquad
\begin{cases}
\boldsymbol{P} \oplus \begin{bmatrix} \mathcal{M}_{CT}(\bar{A}^{\Delta}_{cl}, \bar{B}^{\Delta}_{cl}R_\infty, \boldsymbol{P}) & \begin{bmatrix} (\bar{C}^{\Delta}_{cl})^T L_\infty^T \\ R_\infty^T (\bar{D}^{\Delta}_{cl})^T L_\infty^T \end{bmatrix} \\ \star & \gamma I \end{bmatrix} > 0 \\
\text{(continuous case)} \\
\\
\begin{bmatrix} \mathcal{M}_{DT}(\bar{A}^{\Delta}_{cl}, \bar{B}^{\Delta}_{cl}R_\infty, \boldsymbol{P}) & \begin{bmatrix} 0 \\ (\bar{C}^{\Delta}_{cl})^T L_\infty^T \\ R_\infty^T (\bar{D}^{\Delta}_{cl})^T L_\infty^T \end{bmatrix} \\ \star & \gamma I \end{bmatrix} > 0 \\
\text{(discrete case)}
\end{cases}
\tag{3.45}
$$

Next, let the matrices $(\bar{\boldsymbol{A}}_{cl,i}, \bar{\boldsymbol{B}}_{cl,i}, \bar{C}_{cl,i}, \bar{D}_{cl,i})$ denote the closed-loop system (3.36) that correspond to the i-th vertex of the convex polytope (3.14). Then, with $\boldsymbol{P}$ defined as in (3.41) we can write

$$
\begin{gathered}
\boldsymbol{P}\bar{A}^{\Delta}_{cl} = \\
\begin{bmatrix} \boldsymbol{X}(A^{\Delta} + B^{\Delta}_{u}F) & -\boldsymbol{X}B^{\Delta}_{u}F \\ \boldsymbol{Y}(A^{\Delta} + B^{\Delta}_{u}F) - \boldsymbol{Y}\boldsymbol{B_c}(C^{\Delta}_{y} + D^{\Delta}_{yu}F) - \boldsymbol{Y}\boldsymbol{A_c} & \boldsymbol{Y}\boldsymbol{A_c} + \boldsymbol{Y}\boldsymbol{B_c}D^{\Delta}_{yu}F - \boldsymbol{Y}B^{\Delta}_{u}F \end{bmatrix},
\end{gathered}
$$

$$
\boldsymbol{P}\bar{B}^{\Delta}_{cl} = \begin{bmatrix} \boldsymbol{X}B^{\Delta}_{\xi} \\ \boldsymbol{Y}B^{\Delta}_{\xi} - \boldsymbol{Y}\boldsymbol{B_c}D^{\Delta}_{y\xi} \end{bmatrix}.
$$

Making the one-to-one change of variables

$$
\boldsymbol{Y}\begin{bmatrix} \boldsymbol{A_c} & \boldsymbol{B_c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Z} & \boldsymbol{G} \end{bmatrix}
$$

Algorithm 3.2 (Robust Output-Feedback Controller Design) Use the results in Theorems 3.2 and 3.3 to find an initially feasible controller, represented by the variables $(\boldsymbol{x}_0, \boldsymbol{y}_0, \gamma_0)$ related to the corresponding BMI problem (3.20). Set $(\boldsymbol{x}^*, \boldsymbol{y}^*, \gamma_{UB}^{(0)}) = (\boldsymbol{x}_0, \boldsymbol{y}_0, \gamma_0)$. Solve the relaxed LMI problem (3.28) to obtain $\gamma_{LB}^{(0)}$. Select the desired precision (relative tolerance) Tol and the maximum number of iterations allowed k_{max}. Set $k = 1$.

Step 1. Take $\gamma_k = \frac{\gamma_{UB}^{(k-1)} + \gamma_{LB}^{(k-1)}}{2}$, and solve the problem $(\boldsymbol{x}_k, \boldsymbol{y}_k) = \arg\min v_{\gamma_k}(\boldsymbol{x}, \boldsymbol{y})$ starting with initial condition $(\boldsymbol{x}^*, \boldsymbol{y}^*)$.

Step 2. If $v_{\gamma_k}(\boldsymbol{x}_k, \boldsymbol{y}_k) = 0$ Then set $(\boldsymbol{x}^*, \boldsymbol{y}^*, \gamma_{UB}^{(k)}) = (\boldsymbol{x}_k, \boldsymbol{y}_k, \gamma_k)$ Else set $\gamma_{LB}^{(k)} = \gamma_k$.

Step 3. If $|\gamma_{UB}^{(k)} - \gamma_{LB}^{(k)}| < \text{Tol}|\gamma_{UB}^{(k)}|$ Or $k \geq k_{max}$ Then **Stop** ($(\boldsymbol{x}^*, \boldsymbol{y}^*, \gamma_{UB}^{(k)})$ is the best (locally) feasible solution with the desired tolerance) Else Set $k \leftarrow k + 1$ and go to **Step 1.**

results in $\boldsymbol{P}\bar{\boldsymbol{A}}_{cl,i} = \boldsymbol{M}_i$, and $\boldsymbol{P}\bar{\boldsymbol{B}}_{cl,i} = \boldsymbol{N}_i$, where with the matrices $\boldsymbol{M}_i$ and $\boldsymbol{N}_i$ defined as in (3.41), being linear in the new variables. Therefore the feasibility of (3.43) is equivalent to feasibility of (3.38) for all $i = 1, 2, \ldots, N$.

Further, let R be either R_2 (in the $\mathcal{H}_2$ case) or R_∞ (in the $\mathcal{H}_\infty$ case), and consider the matrices $\mathcal{L}(\cdot)$, $\mathcal{M}_{CT}(\cdot)$, and $\mathcal{M}_{DT}(\cdot)$, as defined in (3.9). With the notation introduced above we can then write that

$$\begin{aligned}
\mathcal{L}(L_2\bar{C}_{cl,i}, \boldsymbol{W}, \boldsymbol{P}, \gamma) &= (\gamma - \mathbf{trace}(\boldsymbol{W})) \oplus \begin{bmatrix} \boldsymbol{W} & L_2\bar{C}_{cl,i} \\ \star & \boldsymbol{P} \end{bmatrix}, \\
\mathcal{M}_{CT}(\bar{A}_{cl,i}, \bar{B}_{cl,i}R, \boldsymbol{P}) &= \begin{bmatrix} -Sym(\boldsymbol{M}_i) & \boldsymbol{N}_i R \\ \star & I \end{bmatrix}, \\
\mathcal{M}_{DT}(\bar{A}_{cl,i}, \bar{B}_{cl,i}R, \boldsymbol{P}) &= \begin{bmatrix} \boldsymbol{P} & \boldsymbol{M}_i & \boldsymbol{N}_i R \\ \star & \boldsymbol{P} & 0 \\ \star & \star & I \end{bmatrix}.
\end{aligned}$$

With this it follows that equations (3.44)-(3.45) are equivalent to (3.39)-(3.40). □

3.5 Summary of the Approach

We next summarize the proposed approach to robust dynamic output-feedback controller design.

Note, that γ_{LB} at each iteration represents an infeasible value for γ, while γ_{UB} is a feasible one. At each iteration of the algorithm the distance between these

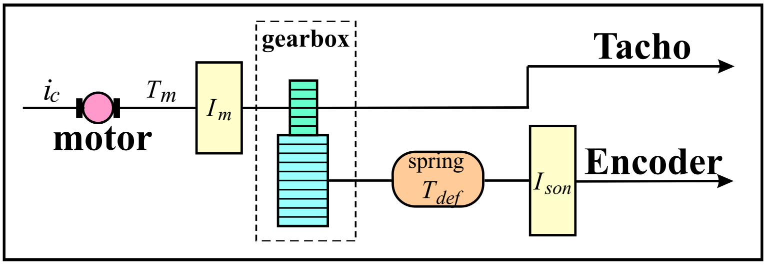


Figure 3.2: Schematic representation of one joint of a space robotic manipulator.

two bounds is reduced in two. It should again be noted that if for a given γ_k the optimal value for the cost function $v_{\gamma_k}(\boldsymbol{x}_k, \boldsymbol{y}_k)$ is nonzero then the algorithm assumes γ_k as infeasible. Since the algorithm converges to a local minimum it may happen that the original BMI problem is actually feasible for this γ_k (e.g. corresponding to the global optimum) but the local optimization is unable to confirm feasibility – an effect that cannot be circumvented.

3.6 Illustrative Examples

3.6.1 Locally Optimal Robust Multiobjective Controller Design

The example considered consists of a linear model of one joint of a real-life space robot manipulator (SRM) system, taken from Kanev and Verhaegen (2000b). A schematic representation of the system is given in Figure 3.2. The equations of motion of the SRM are as follows:

$$\begin{gathered} N^2 I_m \ddot{\Omega} + I_{son}(\ddot{\Omega} + \ddot{\epsilon}) + \beta(\dot{\Omega} + \dot{\epsilon}) = T_j^{eff}, \\ I_{son}(\ddot{\Omega} + \ddot{\epsilon}) + \beta(\dot{\Omega} + \dot{\epsilon}) = T_{def}. \end{gathered}$$

The actuator model of the motor plus the gearbox is:

$$T_j^{eff} = N T_m, \qquad T_m = K_t i_c,$$

and the deformation torque T_{def} is described as

$$T_{def} = c\epsilon$$

Denote $x = [\Omega, \dot{\Omega}, \epsilon, \dot{\epsilon}]^T$ as the state, $y = \begin{bmatrix} \Omega + \epsilon, & N\dot{\Omega} \end{bmatrix}^T$, as the measured output, and $u = i_c$ as the input, then the state-space model of the system is given by

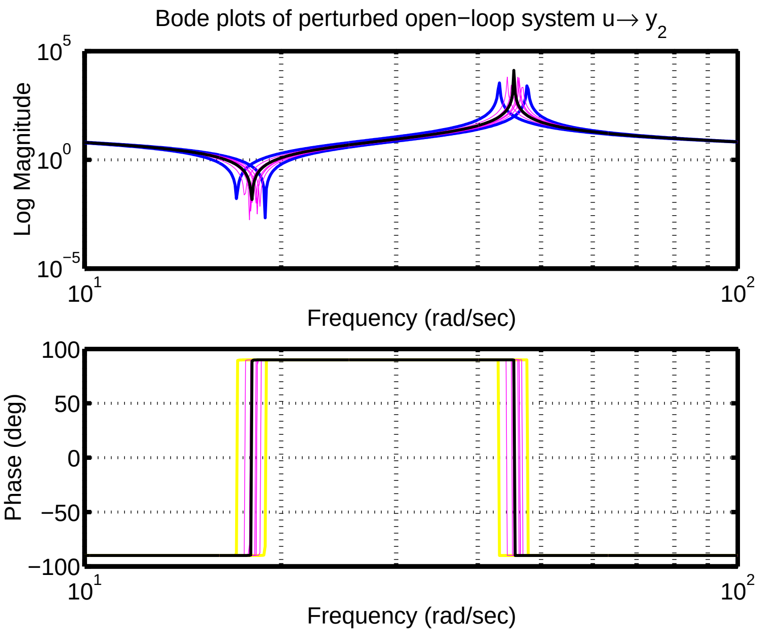


Figure 3.3: Bode plot of the perturbed open-loop transfer from u to y_2.

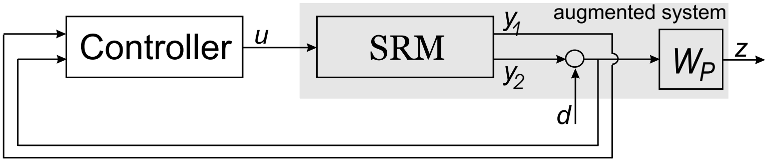


Figure 3.4: Closed-loop system with the selected weighting function $W_p(s)$.

$$\begin{aligned}
\dot{x}(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{c}{N^2 I_m} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{\beta}{I_{son}} & -\frac{c}{N^2 I_m} - \frac{c}{I_{son}} & -\frac{\beta}{I_{son}} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{K_t}{N I_m} \\ 0 \\ -\frac{K_t}{N I_m} \end{bmatrix} u(t) \\
y(t) &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & N & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \xi(t) \\
z(t) &= \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} x(t) + \xi(t)
\end{aligned} \tag{3.46}$$

The system parameters are given in Table 3.1.

The damping coefficient β and the spring constant c are considered as component (parameter) faults in this example. A Bode plot of the open-loop system for different values of β and c is given in Figure 3.3. The objective (see Figure

Parameter:	**Symbol:**	**Value:**
gearbox ratio	N	-260.6
joint angle of inertial axis	Ω	variable
effective joint input torque	T_j^{eff}	variable
motor torque constant	K_t	0.6
the damping coefficient	β	$[0.36,\ 0.44]$
deformation torque of the gearbox	T_{def}	variable
inertia of the input axis	I_m	0.0011
inertia of the output system	I_{son}	400
joint angle of the output axis	ϵ	variable
motor current	i_c	variable
spring constant	c	$[1.17 \times 10^5,\ 1.43 \times 10^5]$

Table 3.1: The values of the parameters in the linear model of one joint of the SRM.

3.4) is to find a controller that achieves for all considered component faults a disturbance rejection of at least 1:100 for constant disturbances on the shaft angular position ($\Omega + \epsilon$) of the motor (such as, e.g., load), and a bandwidth of at least 1 [rad/sec]. This can be achieved by selecting the following performance weighting function (see the upper curve on Figure 3.5)

$$W_p(s) = \frac{1}{s + 0.01},$$

and then requiring that

$$\|W_p(s)S(s)\|_\infty < 1$$

holds for all considered component faults, where $S(s)$ is the transfer function from the disturbance d to the angular velocity $y_2 = N\dot{\Omega}$. In other words, the design specifications would be achieved with a given controller $K(s)$ if the closed-loop transfer function from the disturbance d to the controlled output y_2 lies below the Bode magnitude plot of $W_p^{-1}(s)$.

It should be noted here that this problem is of a rather large scale: the BMI optimization problem (3.20) consists of 4 bilinear matrix inequalities, each of dimension 12×12, and each a function of 95 variables (40 for the controller parameters, and 55 for the closed-loop Lyapunov matrix). Also note, that the number of complicating variables, defined in Tuan and Apkarian (2000) as the number $\min\{\dim(\boldsymbol{x}), \dim(\boldsymbol{y})\}$, in this example equals 40. This makes it clear that the problem is far beyond the capabilities of the global approaches to solving the underlying BMI problem, which can at present deal with no more than just a few complicating variables.

First, using the result in Theorem 3.3 an initial controller was found achieving an upper bound of $\gamma_{\infty,init} = 1.0866$, which was subsequently used to initialize the newly proposed BMI optimization (see Algorithm 3.2). The tolerance of $\text{TOL} = 10^{-3}$ was selected. The new algorithm converged in 10 iterations to

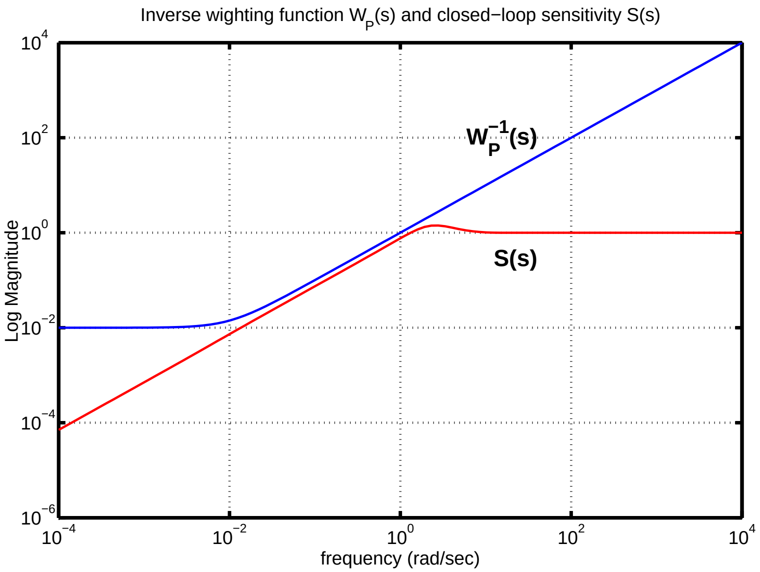


Figure 3.5: Sensitivity function of the closed-loop system for the nominal values of the parameters and the inverse of the weighting function W_p.

$\gamma_{\infty,NEW} = 0.6356$. The computation took about 100 minutes on a PC with Intel(R) Pentium IV CPU 1500 MHz and 1 Gb RAM. Next, four other algorithms were tested on this example with the same initial controller, the same tolerance and the same stopping conditions. These algorithms were Rank Minimization Approach (RMA) (Ibaraki and Tomizuka 2001), the Method of Centers (MC) (Goh et al. 1994), the Path-Following Method (PATH) (Hassibi et al. 1999), and the Alternating coordinate method (DK) (Iwasaki 1999). The results are summarized in Table 3.2. From among these four approaches only two were able to improve the initial controller, namely the MC which achieved $\gamma_{\infty,MC} = 0.8114$ in about 610 minutes, and the DK iteration that terminated in about 20 minutes with $\gamma_{\infty,DK} = 0.8296$. The MC method was unable to improve the performance further due to numerical problems. Similar problems were reported in (Fukuda and

method	achieved γ_{opt}
NEW	0.6356
RMA	-
MC	0.8114
PATH	infeas.
DK	0.8296

Table 3.2: Performance achieved by the five local BMI approaches applied to the model of SRM (3.46).

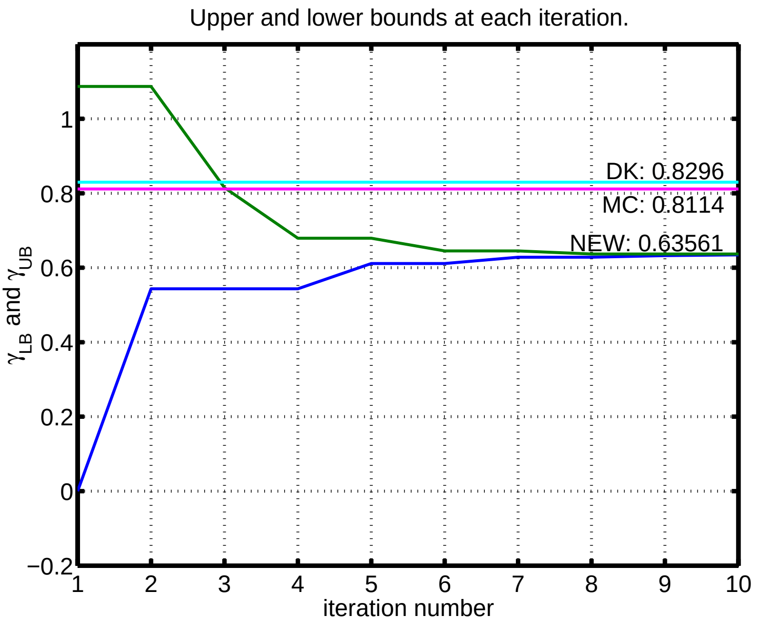


Figure 3.6: Upper and lower bounds on γ during the BMI optimization.

Kojima 2001). The PATH converged to an infeasible solution due to the fact that the initial condition is not "close enough" to the optimal one, so that the first order approximation that is made at each iteration is not accurate. Finally, the RMA method was also unable to find a feasible solution.

This experiment shows that after initializing all BMI approaches with the same controller, the newly proposed method outperforms the other compared methods by achieving the lowest value for the cost function. On the other hand, the initial controller itself also achieves a value for the cost function that is rather close to the optimal cost obtained by the DK and the MC methods, i.e. these methods were not able to significantly improve the initial solution. This implies that the initial controller design method could provide good initial point for starting a local optimization. For the newly proposed method, the upper and the lower bounds on γ at each iteration are plotted in Figure 3.6. Note that at each iteration the upper bound represents a feasible value for γ, and the lower bound- an infeasible one. Also plotted on the same figure are the values achieved by the DK iteration and the MC methods.

The optimal controller obtained after the execution of the newly proposed method has the form (3.15). With this optimal controller, the closed-loop sensitivity function is depicted in Figure 3.5, together with the inverse of the selected performance weighting function $W_p^{-1}(s)$. It can be seen from the figure that the sensitivity function remains below $W_p^{-1}(s)$, implying that the desired robust performance has been achieved.

The difficulties that some of the other local approaches experienced is mainly due to the large scale of the problem that causes numerical difficulties and very slow convergence. In order to further analyze the compared approaches we present a simpler example in the next subsection. This allows us to perform a series of experiments and to compare both the convergence properties as well as the computational speed of the methods.

3.6.2 A Comparison Between Some Local BMI Approaches

In this subsection a comparison is made between the five above-mentioned local approaches to BMI optimization. These approaches will now be tested on the following simple example, taken from Goh et al. (1994): $\min_{x,y} \Lambda(x,y)$ with $\Lambda(x,y) = \max\{y - 2x, x - 2y, xy - 6\}$. The global minimum is -2 achieved at $(x,y) = (2,2)$. This problem can be equivalently rewritten in the form (3.20) as follows

$$\min_{x,y,\gamma} \gamma, \text{ subject to } \mathbf{diag}\{y - 2x - \gamma, x - 2y - \gamma, xy - 6 - \gamma\} \leq 0. \tag{3.47}$$

Only the RMA method does not require an initial condition. All of the four other methods require a feasible initial condition. To make the comparison as fair as possible it is performed in the following way.

- 100 experiments were made. At each experiment a random pair (x,y) was generated, with x and y in the interval $[-3,3]$, and the four algorithms (without RMA) were initialized with the same initial condition.
- In order to guarantee that the initial condition is feasible, the parameter γ was selected in each experiment as $\gamma = 1.1\Lambda(x,y)$.
- The same stopping condition was used for all methods. The stopping condition was selected as in Step 3 of Algorithm 3.2 with $\text{TOL} = 10^{-3}$ and $k_{max} = 20$.
- The time needed for convergence is also computed for each experiment. The algorithms were programmed in Matlab and executed on a computer with Pentium II processor running at 450 MHz.

The results from the comparison are given in Table 3.3. It becomes clear from the Table 3.3 that the newly proposed method is the only one with 100 % convergence to the global optimum. Its best competitor is the MC with 86 % global convergence, which is however much slower. It should be pointed that the performances of these local approaches can vary quite a lot from one application to the other. Thus the results presented in Table 3.3 should not be misinterpreted, but should be considered as representative for the example considered in this subsection. The newly proposed approach is then to be viewed as an alternative to the existing methods that might be useful for some applications. Figure 3.7 visualizes the performance of the five compared approaches starting from initial condition $(x^{(0)}, y^{(0)}) = (-0.5384, 2.3619)$. It can be seen how the DK iteration method converges to a point that is optimal in the directions of x and y, but is still not a local optimum (and is actually pretty far away from that). The RMA

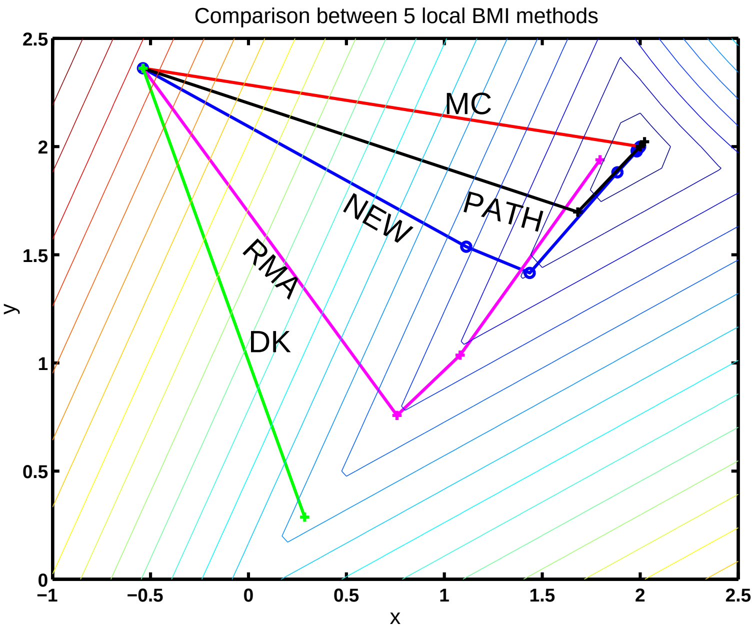


Figure 3.7: Performance of the five compared local BMI approaches.

method	convergence to the global optimum	aver. comp. time [sec]
NEW	100 %	2.542
RMA*	0 %	32.630
MC	86 %	19.656
PATH	69 %	3.304
DK	16 %	0.612

Table 3.3: Comparison between five local approaches to BMI optimization. *The RMA method does not require the initial condition, and as such was executed only one time.

approach also did not manage to converge to the optimum, but still performed better than the DK iteration. The other three approaches, namely NEW, PATH, and MC, all converged to the optimum. However, the newly proposed approach was the fastest one.

3.7 Conclusions

The passive FTC system design is an approach where a controller needs to be designed that makes the closed-loop system insensitive to certain class of faults.

This can be achieved by viewing the faults as uncertainty in the system and then designing a robust controller that can guarantee some satisfactory performance in the worst-case uncertainty (fault scenario). In this way sensor, actuator and component faults are represented as parametric *(structured)* uncertainty, and one single controller is used for all fault scenarios. This can be viewed as a trade-off between performance and increased robustness to faults, which can be desirable when initially no information about the fault is available. The FDD part could in this way gain time to do a more accurate fault diagnosis, after which the actual reconfiguration can take place.

To this end, a new approach was presented in this chapter to the design of locally optimal robust dynamic output-feedback controllers for systems with structured uncertainties was presented. The uncertainty is allowed to have a very general structure and is only assumed to be such that the state-space matrices of the system belong to a certain convex set. The approach is based on BMI optimization that is guaranteed to converge to a locally optimal solution provided that an initially feasible controller is given. This algorithm enjoys the useful properties of computational efficiency and guaranteed convergence to a local optimum. An algorithm for fast computation of an initially feasible controller is also provided and is based on a two-step procedure, where at each step an LMI optimization problem is solved – one to find the optimal state-feedback gain and one to find the remaining state-space matrices of the output-feedback controller. The design objectives considered are $\mathcal{H}_2$, $\mathcal{H}_\infty$, and pole-placement in LMI regions. The approach was tested on a model of one joint of a real-life space robotic manipulator, for which a robust $\mathcal{H}_\infty$ controller was designed. In addition, the proposed approach was compared to several existing approaches on a simpler BMI optimization and it became clear that it can act as a good alternative for some applications.

4

LPV Approach to Robust Active FTC

In the passive approaches to FTC, considered in Chapters 2 and 3, the goal was to design one robust controller that achieves satisfactory performance for a specific class of possible faults. These passive FTC methods do not require fault diagnosis; they trade performance for increased robustness with respect to faults. In the next chapters we focus on active FTC methods which can improve the performance that can be achieved by the passive FTC methods by means of using estimates of the faults, provided by some FDD scheme. The main focus is on developing *robust* methods to active FTC that deal with both *model uncertainty* as well as uncertainty in the fault estimates (also called *FDD uncertainty*). Furthermore, the size of the FDD uncertainty is allowed to vary with time, making it possible to consider more uncertainty immediately after the occurrence of a fault due to the initial lack of enough measurement data from the faulty system. To this end it is assumed that an FDD scheme is present that provides the FTC scheme with both fault estimates as well as the uncertainty intervals of these estimates, as illustrated on Figure 4.1 on page 96.

The active FTC methods, considered in this chapter, are based on parameter-varying controller design. Two methods are presented. First, in Section 4.2 we propose a deterministic approach to active FTC design that can deal with multiplicative sensor and actuator faults. This method designs off-line a bank of LPV controllers for specific fault scenarios. Then, based on the fault estimates, the controller that achieves the best performance is switched on. This LPV controller is subsequently scheduled by the size of the uncertainty in the fault estimate.

The second method, developed in Section 4.3, is based on the probabilistic framework of Chapter 2. This probabilistic design method makes it possible to consider, in addition to sensor and actuator faults, also component faults, as well as to schedule the LPV by both the fault estimates and their uncertainty sizes. In this way the bank of controllers from the deterministic method is replaced by only one LPV controller. This second approach also considers (structured) model uncertainty in addition to the FDD uncertainty. Both approaches can be used for state-feedback as well as output-feedback design.

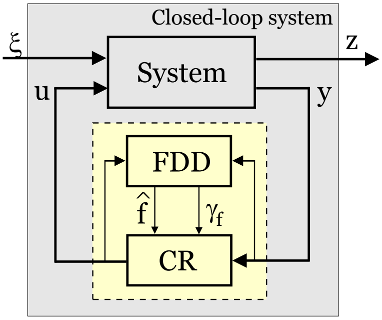


Figure 4.1: The FDD scheme provides to the controller reconfiguration scheme not only the estimates of the faults $\hat{f}$, but also the sizes γ_f of the uncertainties in these estimates.

4.1 Introduction

As opposed to the passive FTC methods, presented in Chapters 2 and 3 of this thesis, the active methods to FTC usually require the presence of an FDD scheme that provides estimates of the faults. These fault estimates can then be used in an active FTC approach in order to improve the performance that is achievable by the passive methods. Additionally, active methods can deal with a wider class of system faults. This chapter proposes two methods to active FTC based on parameter-dependent controller design with the clear focus on dealing with imprecise (uncertain) fault estimates provided by the FDD scheme. To this end, it is assumed that on-line estimates of the faults are available, but that the real values of the faults lie within some given intervals around their estimates. The length of these intervals is considered time-varying. This makes it possible to model more accurately a real-life FDD scheme, where after the occurrence of an abrupt fault it can first only provide rough fault estimates with big uncertainty, that are later on fine-tuned as more measurements become available from the system.

The first approach to FTC, proposed in Section 4.2 of this chapter, can deal with multiplicative sensor and actuator faults. The approach consists of the off-line design of a set of suitably selected parameter-dependent controllers, in which the scheduling parameters are the sizes of the uncertainty intervals. After a fault has been diagnosed, the controller that achieves the best performance for the current *total* fault estimates is switched on. This controller is then scaled to accommodate the *partial faults* that are currently in effect. The resulting LPV controller is subsequently scheduled by the sizes of the uncertainties in the fault estimates. Although a finite set of controllers are initially designed, the reconfiguration scheme is not restricted to a finite set of anticipated faults, but deals with an arbitrary combination of multiplicative sensor and actuator faults.

The second approach, proposed in Section 4.3, is developed in the probabilistic framework from Chapter 2. This makes it possible to consider, in addition to sensor and actuator faults, also component faults, as well as to schedule the LPV by both the fault estimates and their uncertainty sizes. In this way the bank of controllers from the deterministic method of Section 4.2 is replaced by only one LPV controller. This second approach also considers (structured) model uncertainty in addition to the FDD uncertainty.

Some previous work on the use of linear parameter-varying control methods for active FTC design are Bennani et al. (1999); Ganguli et al. (2002); Shin et al. (2002). The main contribution of the methods presented in this chapter is that they consider time-varying FDD uncertainty. Additionally, the probabilistic method is applicable to a much wider class of faults as the fault estimate signal is allowed to enter the state-space matrices of the system in any way as long as the matrices remain bounded.

This chapter is organized as follows. The next section begins with the description of the deterministic approach to LPV-based robust active FTC for sensor and actuator faults. The second, probabilistic design approach for component faults is subsequently presented in Section 4.3. In Section 4.4 some examples are provided to illustrate the developed methods. Finally, Section 4.5 concludes the chapter.

4.2 Deterministic Method for Multiplicative Sensor and Actuator Faults

This section considers the problem of controller reconfiguration (CR) in cases of multiplicative sensor and actuator faults. It is assumed that on-line estimates of the faults are provided by some fault detection and diagnosis scheme as shown in Figure 4.1. In order to model uncertainty in the FDD process, the true faults are further assumed to lie inside given uncertainty intervals around the estimates. Additionally, the lengths of these intervals are allowed to be time-varying and are also assumed provided by the FDD scheme. The approach is demonstrated on the diesel engine actuator benchmark model of Section 2.6.

4.2.1 Problem Formulation

Consider the following discrete-time linear system

$$\mathcal{S}_{nom}: \left\{ \begin{array}{rcl} x_{k+1} & = & Ax_k + B_\xi \xi_k + B_u u_k \\ z_k & = & C_z x_k + D_{z\xi}\xi_k + D_{zu} u_k \\ y_k & = & C_y x_k + D_{y\xi}\xi_k, \end{array} \right. \tag{4.1}$$

where $x_k \in \mathbb{R}^n$ is the state of the system, $u \in \mathbb{R}^m$ is the control action, $y \in \mathbb{R}^p$ is the measured output, $z \in \mathbb{R}^{n_z}$ represents the controlled output of the system, and $\xi \in \mathbb{R}^\xi$ is the disturbance to the system.

In this section we consider multiplicative sensor and actuator faults, as modelled in (1.6) on page 7. The offsets $\bar{u}$ and $\bar{y}$ in (1.6) will be considered equal to zero in the sequel. When nonzero, their effect on the controlled output z_k can be minimized by including them in the disturbance signal ξ_k. Replacing u_k and y_k in (4.1) with the faulty signals u_k^f in (1.2) and y_k in (1.4) on page 7, and subsequent substitution $\bar{u} = 0$ and $\bar{y} = 0$, results in the following model describing multiplicative simultaneous sensor and actuator faults

$$\mathcal{S}_F : \left\{ \begin{array}{rcl} x_{k+1} & = & Ax_k + B_\xi \xi_k + B_u \Sigma_A u_k \\ z_k & = & C_z x_k + D_{z\xi} \xi_k + D_{zu} \Sigma_A u_k \\ y_k & = & \Sigma_S C_y x_k + \Sigma_S D_{y\xi} \xi_k, \end{array} \right. \tag{4.2}$$

As discussed in the introductory chapter 1, it is assumed that $\Sigma_A \in \boldsymbol{\Sigma_A}$ and $\Sigma_S \in \boldsymbol{\Sigma_S}$, where

$$\begin{array}{l} \boldsymbol{\Sigma_A} \doteq \left\{ \ \Sigma_A = \text{diag}(\sigma_a^1, \ldots, \sigma_a^m) : \ (A, B_u \Sigma_A) \text{ is stabilizable } \ \right\} \\ \boldsymbol{\Sigma_S} \doteq \left\{ \ \Sigma_S = \text{diag}(\sigma_s^1, \ldots, \sigma_s^p) : \ (A, \Sigma_S C_y) \text{ is detectable } \ \right\}. \end{array} \tag{4.3}$$

In other words, only faults that do not affect the stabilizability and the detectability of the system are considered.

Note that the quantities Σ_A and Σ_S are allowed to be time varying, and that we require that the conditions $\Sigma_A \in \boldsymbol{\Sigma_A}$ and $\Sigma_S \in \boldsymbol{\Sigma_S}$ hold at each time instant k. For simplicity of the notations, however, we will not explicitly write the time dependence in Σ_A and Σ_S.

As already discussed, the focus of this section is the development of a controller reconfiguration technique applicable to multiplicative sensor and actuator faults. The detection and isolation of these faults is not the purpose of this chapter, and it is assumed that a fault detection and isolation (FDD) scheme is available and produces online both estimates of the faults $\hat{\Sigma}_A$, $\hat{\Sigma}_S$, as well as of the uncertainty intervals around them Γ_A and Γ_A so that

$$\begin{array}{l} \Sigma_A \in \hat{\Sigma}^A (I + \Gamma_A \Delta_A), \ 0 \leq \Gamma_A^L \leq \Gamma_A \leq \Gamma_A^U \\ \Sigma_S \in \hat{\Sigma}^S (I + \Gamma_S \Delta_S), \ 0 \leq \Gamma_S^L \leq \Gamma_S \leq \Gamma_S^U, \end{array} \tag{4.4}$$

where Δ_A and Δ_S are two real diagonal matrices with $\|\Delta_A\|_2 \leq 1$ and $\|\Delta_S\|_2 \leq 1$, representing the uncertainty. The real diagonal matrices Γ_A and Γ_S, on the other hand, are used to represent the size of the uncertainty intervals around the fault estimates since it can be written that

$$\begin{array}{l} \hat{\Sigma}_A (I - \Gamma_A) \leq \Sigma_A \leq \hat{\Sigma}_A (I + \Gamma_A), \\ \hat{\Sigma}_S (I - \Gamma_S) \leq \Sigma_S \leq \hat{\Sigma}_S (I + \Gamma_S). \end{array} \tag{4.5}$$

We denote the matrices Γ_A and Γ_S and their bounds as

$$\begin{array}{l} \Gamma_A \doteq \textbf{diag}\left(\gamma_{a,1}, \ldots, \gamma_{a,m}\right), \\ \Gamma_S \doteq \textbf{diag}\left(\gamma_{s,1}, \ldots, \gamma_{s,p}\right), \\ \Gamma_A^L \doteq \textbf{diag}\left(\gamma_{a,1}^l, \ldots, \gamma_{a,m}^l\right), \\ \Gamma_A^U \doteq \textbf{diag}\left(\gamma_{a,1}^u, \ldots, \gamma_{a,m}^u\right), \\ \Gamma_S^L \doteq \textbf{diag}\left(\gamma_{s,1}^l, \ldots, \gamma_{s,p}^l\right), \\ \Gamma_S^U \doteq \textbf{diag}\left(\gamma_{s,1}^u, \ldots, \gamma_{s,p}^u\right), \end{array} \tag{4.6}$$

so that $\gamma_{a,i} \in [\gamma_{a,i}^l, \ \gamma_{a,i}^u]$, and $\gamma_{s,j} \in [\gamma_{s,j}^l, \ \gamma_{s,j}^u]$, as implied from (4.4). In practice it could be expected that Γ_A and Γ_S would be large immediately after the occurrence of a fault (i.e. they equal their upper bounds), and then gradually become smaller as more data becomes available from the system.

Interconnected to the fault-free system $\mathcal{S}_{nom}$ is the full order fault-free controller

$$\mathcal{K}_{nom}: \left\{ \begin{array}{rcl} x^c_{k+1} & = & A^{nom}_c x^c_k + B^{nom}_c y_k \\ u_k & = & C^{nom}_c x^c_k + D^{nom}_c y_k, \end{array} \right. \tag{4.7}$$

resulting in the fault-free closed-loop system $T_{cl} = \mathcal{F}_L(\mathcal{S}_{nom}, \mathcal{K}_{nom})$, where $\mathcal{F}_L(\cdot,\cdot)$ is used to denote the lower linear fractional transformation (Zhou and Doyle 1998). The control objective considered is the minimization of the $\mathcal{H}_\infty$-norm of the closed-loop system T_{cl}, i.e.

$$\min_{\mathcal{K}_{nom}} \|\mathcal{F}_L(\mathcal{S}_{nom}, \mathcal{K}_{nom})\|_\infty. \tag{4.8}$$

When a given combination of sensor and actuator faults occurs in the system, the model of the faulty system $\mathcal{S}_F$ becomes uncertain, as is clear after combining equations (4.2) and (4.4). If the uncertainty intervals were constant in time (or, equivalently, if in (4.4) we take $\Gamma^L_A = \Gamma^U_A$ and $\Gamma^L_S = \Gamma^U_S$), then for any fixed matrices $\hat{\Sigma}_A \in \mathbf{\Sigma_A}$ and $\hat{\Sigma}_S \in \mathbf{\Sigma_S}$ the controller reconfiguration problem could be defined as

$$\min_{\mathcal{K}} \sup_{\Delta_A, \Delta_S} \|\mathcal{F}_L(\mathcal{S}_F, \mathcal{K})\|_\infty.$$

However, in the more general case when $\Gamma^L_A < \Gamma^U_A$ and $\Gamma^L_S < \Gamma^U_S$, the controller could be made dependent on Γ_A and Γ_S. To this end, the problem is formulated as follows: given any fixed $\hat{\Sigma}_A \in \mathbf{\Sigma_A}$ and $\hat{\Sigma}_S \in \mathbf{\Sigma_S}$, design a parameter-dependent controller $\mathcal{K}(\Gamma_A, \Gamma_S)$, that achieves

$$\min_{\mathcal{K}(\Gamma_A,\Gamma_S)} \sup_{\substack{\Delta_A,\Delta_S \\ \Gamma_A,\Gamma_S}} \|\mathcal{F}_L(\mathcal{S}_F(\Gamma_A,\Gamma_S), \mathcal{K}(\Gamma_A,\Gamma_S))\|_\infty. \tag{4.9}$$

Note that in the deterministic approach presented in section 4.2, as opposed to the probabilistic approach in section 4.3, the controller is not directly scheduled by the the fault estimates $\hat{\Sigma}_A$ and $\hat{\Sigma}_S$. The reason for that is that if we make the controller dependent on $\hat{\Sigma}_A$ and $\hat{\Sigma}_S$ this results in an infinite number of LMIs that need to be solved to compute the controller. This is due to the fact that the LMIs that we will derive in the sequel are affine in the fault estimates $\hat{\Sigma}_A$ and $\hat{\Sigma}_S$ (for fixed Γ_A and Γ_S) and in the uncertainty intervals Γ_A and Γ_S (for fixed $\hat{\Sigma}_A$ and $\hat{\Sigma}_S$), but are not affine in all $\hat{\Sigma}_A$, $\hat{\Sigma}_S$, Γ_A and Γ_S at the same time. Therefore, the well-known vertex LMI property cannot be applied here to equivalently represent the infinite number of LMIs by a finite number of vertex LMIs.

To circumvent this problem we impose the restriction that the controller is scheduled only by the uncertainty intervals Γ_A and Γ_S. In order to be able to deal with the fault estimates $\hat{\Sigma}_A$ and $\hat{\Sigma}_S$ in this Section 4.2 we proceed as follows (see Figure 4.2). First, a well-chosen set of parameter dependent local controllers is designed off-line by solving the problem (4.9) for some given pairs $\hat{\Sigma}_A \in \mathbf{\Sigma_A}$ and $\hat{\Sigma}_S \in \mathbf{\Sigma_S}$. Then, after each occurrence of a combination of sensor and actuator faults the controller is reconfigured by means of input/output scaling of one of the pre-designed controllers. This reconfigured controller is subsequently scheduled by current values for Γ_A and Γ_S. The question of how to select the $\hat{\Sigma}_A$'s and $\hat{\Sigma}_S$'s for which to design the set of local LPV controllers will be considered in the next subsection.

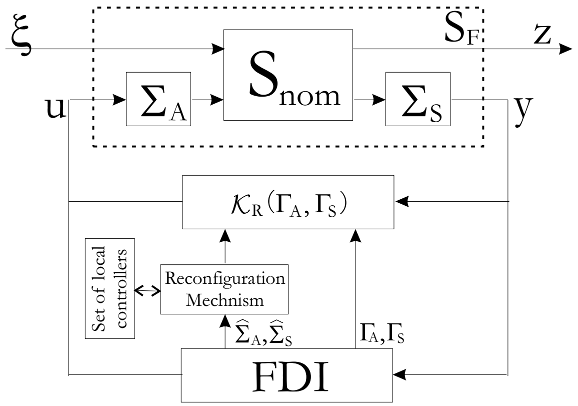


Figure 4.2: Block-schematic representation of the reconfiguration of the overall fault-tolerant system.

4.2.2 LPV Controllers Design

In this subsection we concentrate on the design of a LPV controller for certain fixed and given $\hat{\Sigma}_A \in \boldsymbol{\Sigma_A}$ and $\hat{\Sigma}_S \in \boldsymbol{\Sigma_S}$ by solving the problem (4.9). The resulting LPV controller will be scheduled by the uncertainty sizes Γ_A and Γ_S. In the next subsection we will discuss on how to select the $\hat{\Sigma}_A$'s and $\hat{\Sigma}_S$'s for which we need to design the set of local LPV controllers.

Consider the faulty system (4.2) interconnected with the parameter dependent controller

$$\mathcal{K}(\Gamma_A, \Gamma_S) : \left\{ \begin{array}{rcl} x_{k+1}^c & = & A_c(\Gamma_A, \Gamma_S)x_k^c + B_c(\Gamma_A, \Gamma_S)y_k \\ u_k & = & C_c(\Gamma_A, \Gamma_S)x_k^c + D_c(\Gamma_A, \Gamma_S)y_k, \end{array} \right. \tag{4.10}$$

where $x_k^c \in \mathbb{R}^n$. Further, observing the equivalence between the two block-schemes on Figure 4.3, it becomes clear that the faulty system $\mathcal{S}_F$ can be written in the form

$$\mathcal{M} : \left\{ \begin{array}{l} \left[\begin{array}{c} x_{k+1} \\ \hline z_{a,k} \\ z_{s,k} \\ z_k \\ \hline y_k \end{array} \right] = \left[\begin{array}{ccc} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{array} \right] \left[\begin{array}{c} x_k \\ \hline w_{a,k} \\ w_{s,k} \\ \xi_k \\ \hline u_k \end{array} \right] \\ \\ w_{a,k} = \Delta_A z_{a,k}, \\ w_{s,k} = \Delta_S z_{s,k} \end{array} \right. \tag{4.11}$$

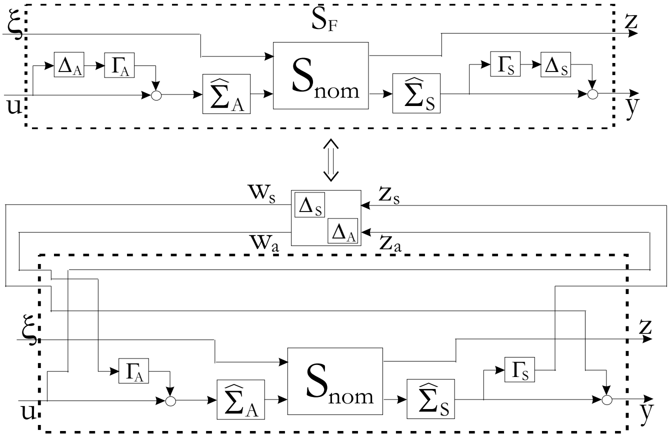


Figure 4.3: Pulling out the uncertainties.

where it is denoted

$$\begin{aligned}
B_1(\Gamma_A) &\doteq \left[\begin{array}{ccc} B_u\hat{\Sigma}_A\Gamma_A & 0 & B_\xi \end{array}\right], \; B_2 \doteq B_u\hat{\Sigma}_A, \\
C_1(\Gamma_S) &\doteq \left[\begin{array}{c} 0 \\ \Gamma_S\hat{\Sigma}_S C_y \\ C_z \end{array}\right], \; C_2 \doteq \hat{\Sigma}_S C_y, \\
D_{11}(\Gamma_A,\Gamma_S) &\doteq \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & \Gamma_S\hat{\Sigma}_S D_{y\xi} \\ D_{zu}\hat{\Sigma}_A\Gamma_A & 0 & D_{z\xi} \end{array}\right], \\
D_{12} &\doteq \left[\begin{array}{c} I \\ 0 \\ D_{zu}\hat{\Sigma}_A \end{array}\right], \; D_{21} \doteq \left[\begin{array}{ccc} 0 & I & \hat{\Sigma}_S D_{y\xi} \end{array}\right].
\end{aligned}$$

Note, that the matrices B_2, C_2, D_{12} and D_{21} are independent on the variables Γ_A and Γ_S, an important fact that would be exploited in what will follow. Note also, that the matrices B_1, C_1, and D_{11} are *affine* in Γ_A and Γ_S.

In this way, the uncertainty in the system was "pulled out", that resulted in an augmented model $\mathcal{M}$ that depends on the known variables Γ_A and Γ_S, so that

$$\mathcal{S}_F(\Delta_A,\Delta_S) = \mathcal{F}_U\left(\mathcal{M}, \left[\begin{array}{cc} \Delta_A & \\ & \Delta_S \end{array}\right]\right)$$

Here $\mathcal{F}_U(\cdot,\cdot)$ denotes the upper linear fractional transformation. Therefore, the faulty closed-loop system can be rewritten as

$$\begin{aligned}
&\mathcal{F}_L(\mathcal{S}_F(\Gamma_A,\Gamma_S),\mathcal{K}(\Gamma_A,\Gamma_S)) = \\
&\quad \mathcal{F}_L\left(\mathcal{F}_U\left(\mathcal{M}, \left[\begin{array}{cc} \Delta_A & \\ & \Delta_S \end{array}\right]\right), \mathcal{K}(\Gamma_A,\Gamma_S)\right).
\end{aligned}$$

Using the small gain theorem it then follows that for any given $\gamma > 0$

$$\sup_{\Delta_A,\Delta_S} \|\mathcal{F}_L(\mathcal{S}_F(\Gamma_A,\Gamma_S),\mathcal{K}(\Gamma_A,\Gamma_S))\|_\infty \le \gamma^{-1}$$

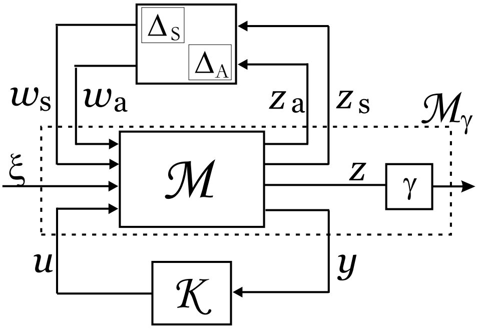


Figure 4.4: The system $\mathcal{M}_\gamma$.

if

$$\|\mathcal{F}_L\left(\mathcal{M}_\gamma, \mathcal{K}(\Gamma_A, \Gamma_S)\right)\|_\infty \leq 1, \tag{4.12}$$

where $\mathcal{M}_\gamma$ is obtained by multiplying the output z_k of the system $\mathcal{M}$ by γ, as demonstrated on Figure 4.4. We note here that by making use of the small gain theorem we "destroy" the block-diagonal structure of the uncertainty. What we gain is a convex LMI problem that can be solved very efficiently. The prise that we have to pay for this convexity is conservatism in the resulting controller.

Thus, the problem defined in (4.9) is reduced to maximization of γ under the constraint (4.12). In this way it is enough to consider the case of $\gamma = 1$, i.e. $\mathcal{M}_\gamma = \mathcal{M}$ in (4.12), and then a simple bisection type of algorithm can be used to solve the problem of minimizing γ^{-1} subject to the constraint (4.12). Before we proceed with a result that can be used to find a parameter dependent controller $\mathcal{K}(\Gamma_A, \Gamma_S)$ such that (4.12) holds for $\gamma = 1$ we define the matrices B_1^i, C_1^j, and D_{11}^l, $i = 0, 1, \ldots, (p+m)$, such that

$$\begin{aligned}
B_1(\Gamma_A) &= B_1^0 + \sum_{i=1}^{m} B_1^i \gamma_{a,i}, \\
C_1(\Gamma_S) &= C_1^0 + \sum_{j=1}^{p} C_1^j \gamma_{s,j} \\
D_{11}(\Gamma_A, \Gamma_S) &= D_{11}^0 + \sum_{i=1}^{m} D_{11}^i \gamma_{a,i} + \sum_{j=1}^{p} D_{11}^{m+j} \gamma_{s,j}
\end{aligned} \tag{4.13}$$

The LPV controller design is next considered for the state-feedback and output-feedback cases.

State-feedback case

In the state-feedback case the state x_k is assumed given and the measurement y_k in (4.1) is no longer considered. Sensor faults, which are faults in the measurements y_k, have therefore also no effect on the optimal state-feedback design. In this case the controller takes the form

$$\mathcal{K}_{SF}(\Gamma_A):\ u_k = F(\Gamma_A)x_k,$$

and the faulty system $\mathcal{S}_F$ can be written in the form

$$\mathcal{M}_{sf}:\ \begin{cases} \left[\begin{array}{c} x_{k+1} \\ \hline z_{a,k} \\ z_k \end{array}\right] = \begin{bmatrix} A & \bar{B}_1 & B_2 \\ \bar{C}_1 & \bar{D}_{11} & \bar{D}_{12} \end{bmatrix} \left[\begin{array}{c} x_k \\ \hline w_{a,k} \\ \xi_k \\ \hline u_k \end{array}\right] \\ \\ w_{a,k} = \Delta_A z_{a,k}, \end{cases} \tag{4.14}$$

where

$$\begin{aligned} &\bar{B}_1(\Gamma_A) \doteq \begin{bmatrix} B_u\hat{\Sigma}_A\Gamma_A & B_\xi \end{bmatrix},\ \bar{B}_2 \doteq B_u\hat{\Sigma}_A, \\ &\bar{C}_1 \doteq \begin{bmatrix} 0 \\ C_z \end{bmatrix},\ \bar{D}_{11}(\Gamma_A) \doteq \begin{bmatrix} 0 & 0 \\ D_{zu}\hat{\Sigma}_A\Gamma_A & D_{z\xi} \end{bmatrix}, \\ &\bar{D}_{12} \doteq \begin{bmatrix} I \\ D_{zu}\hat{\Sigma}_A \end{bmatrix}. \end{aligned}$$

Then, similarly to (4.13), we define

$$\begin{aligned} \bar{B}_1(\Gamma_A) &= \bar{B}_1^0 + \sum_{i=1}^{m} \bar{B}_1^i \gamma_{a,i}, \\ \bar{D}_{11}(\Gamma_A) &= \bar{D}_{11}^0 + \sum_{i=1}^{m} \bar{D}_{11}^i \gamma_{a,i} \end{aligned} \tag{4.15}$$

The following result can then be used for the design of the state-feedback gain $F(\Gamma_A)$.

Lemma 4.1 *Consider the system $\mathcal{M}$ in Equation (4.14) with the matrix Γ_A being bounded as in (4.4). Let the matrices $\mathbf{X} = \mathbf{X}^T \in \mathbb{R}^{n\times n}$ and $\mathbf{L}_i \in \mathbb{R}^{m\times n}$, $i = 0, 1, \ldots, m$ be such that the for all $\gamma_{a,i} \in \{\gamma_{a,i}^l; \gamma_{a,i}^u\}$ the following linear matrix inequalities are feasible*

$$\begin{bmatrix} \mathbf{X} & A + B_2\mathbf{L} & \bar{B}_1(\Gamma_A) & 0 \\ \star & \mathbf{X} & 0 & \mathbf{L}^T D_{12}^T + \mathbf{X}C_1 \\ \star & \star & I & D_{11}(\Gamma_A)^T \\ \star & \star & \star & I \end{bmatrix} > 0, \tag{4.16}$$

where

$$\mathbf{L} \doteq \mathbf{L}_0 + \sum_{i=1}^{m} \mathbf{L}_i \gamma_{a,i}. \tag{4.17}$$

Then the parameter-dependent state-feedback matrix

$$F(\Gamma_A) = F_0 + \sum_{i=1}^{m} F_i \gamma_{a,i} \tag{4.18}$$

with $F_i = \mathbf{L}_i\mathbf{X}^{-1}$*,* $i = 0, 1, \ldots, m$*, results in the closed-loop system*

$$T_{cl} : \begin{cases} x_{cl,k+1} &= \bar{A}_{cl}x_{cl,k} + \bar{B}_{cl}\begin{bmatrix}\omega_{a,k}\\ \xi_k\end{bmatrix} \\ \begin{bmatrix}z_{a,k}\\ z_k\end{bmatrix} &= \bar{C}_{cl}x_{cl,k} + \bar{D}_{cl}\begin{bmatrix}\omega_{a,k}\\ \xi_k\end{bmatrix} \end{cases} \tag{4.19}$$

with

$$\begin{array}{ll} \bar{A}_{cl} = A + B_2F(\Gamma_A), & \bar{B}_{cl} = \bar{B}_1(\Gamma_A), \\ \bar{C}_{cl} = \bar{C}_1 + \bar{D}_{12}F(\Gamma_A), & \bar{D}_{cl} = \bar{D}_{11}(\Gamma_A), \end{array} \tag{4.20}$$

for which $\|T_{cl}\|_\infty \leq 1$.

Proof: Consider the closed-loop system (4.20). Then $\|T_{cl}\|_\infty \leq 1$ will hold for all $\gamma_{a,i} \in \{\gamma^l_{a,i}; \gamma^u_{a,i}\}$ if the following matrix inequalities holds (consult Lemma 3.3 on page 73)

$$\begin{bmatrix} \mathbf{Y} & \mathbf{Y}\bar{A}_{cl} & \mathbf{X}\bar{B}_{cl} & 0 \\ \star & \mathbf{Y} & 0 & \bar{C}^T_{cl} \\ \star & \star & I & \bar{D}^T_{cl} \\ \star & \star & \star & I \end{bmatrix} > 0, \tag{4.21}$$

is feasible for some matrix $\mathbf{Y} = \mathbf{Y}^T$ or, equivalently, is there exists a symmetric matrix $\mathbf{X}$ such that

$$\begin{bmatrix} \mathbf{X} & \bar{A}_{cl}\mathbf{X} & \mathbb{X}\bar{B}_{cl} & 0 \\ \star & \mathbf{X} & 0 & \mathbf{X}\bar{C}^T_{cl} \\ \star & \star & I & \bar{D}^T_{cl} \\ \star & \star & \star & I \end{bmatrix} > 0, \tag{4.22}$$

holds for all $\gamma_{a,i} \in \{\gamma^l_{a,i}; \gamma^u_{a,i}\}$. After substitution of the closed-loop system matrices in (4.20), and subsequently performing the one-to-one change of variables $\mathbf{L} = F\mathbf{X}$ results in the system of matrix inequalities (4.16). Therefore, taking $\mathbf{L}$ as in equation (4.17) results in state-feedback gain $F = \mathbf{L}\mathbf{X}^{-1}$, defined in (4.18), so that for the closed-loop system it will hold that $\|T_{cl}\|_\infty \leq 1$ for all $\gamma_{a,i} \in \{\gamma^l_{a,i}; \gamma^u_{a,i}\}$. □

Output-feedback case

In the output-feedback case the following result can be used for the LPV controller design.

Theorem 4.1 *Consider the system* $\mathcal{M}$ *in Equation (4.11) with the matrices* Γ_A *and* Γ_S *being bounded as in (4.4). Let the matrices* $\mathbf{X} = \mathbf{X}^T \in \mathbb{R}^{n\times n}$*,* $\mathbf{Y} = \mathbf{Y}^T \in \mathbb{R}^{n\times n}$*,* $\mathbf{K}_i \in \mathbb{R}^{n\times n}$*,* $\mathbf{L}_i \in \mathbb{R}^{n\times p}$*,* $\mathbf{M}_i \in \mathbb{R}^{m\times n}$*,* $\mathbf{N}_i \in \mathbb{R}^{m\times p}$*,* $i = 0, 1, \ldots, (m+p)$ *be such that the for all* $\gamma_{a,i} \in \{\gamma^l_{a,i}; \gamma^u_{a,i}\}$*, and* $\gamma_{s,j} \in \{\gamma^l_{s,j}; \gamma^u_{s,j}\}$ *the linear matrix inequalities*

$$\begin{bmatrix} \boldsymbol{\mathcal{Y}} & \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}} & 0 \\ \star & \boldsymbol{\mathcal{Y}} & 0 & \boldsymbol{\mathcal{C}} \\ \star & \star & I & \boldsymbol{\mathcal{D}} \\ \star & \star & \star & I \end{bmatrix} > 0 \tag{4.23}$$

hold, where it is denoted

$$\boldsymbol{\mathcal{Y}} = \begin{bmatrix} \mathbf{Y} & I \\ \star & \mathbf{X} \end{bmatrix}, \quad \boldsymbol{\mathcal{A}} = \begin{bmatrix} A\mathbf{Y} + B_2\mathbf{M} & A + B_2\mathbf{N}C_2 \\ \mathbf{K} & \mathbf{X}A + \mathbf{L}C_2 \end{bmatrix},$$

$$\boldsymbol{\mathcal{B}} = \begin{bmatrix} B_1(\Gamma_A) + B_2\mathbf{N}D_{21} \\ \mathbf{X}B_1(\Gamma_A) + \mathbf{L}D_{12} \end{bmatrix}, \quad \boldsymbol{\mathcal{C}} = \begin{bmatrix} (C_1(\Gamma_S)\mathbf{Y} + D_{12}\mathbf{M})^T \\ (C_1(\Gamma_S) + D_{12}\mathbf{N}C_2)^T \end{bmatrix},$$

$$\boldsymbol{\mathcal{D}} = (D_{11}(\Gamma_A, \Gamma_S) + D_{12}\mathbf{N}D_{21})^T, \tag{4.24}$$

$$\begin{bmatrix} \mathbf{K} & \mathbf{L} \\ \mathbf{M} & \mathbf{N} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_0 & \mathbf{L}_0 \\ \mathbf{M}_0 & \mathbf{N}_0 \end{bmatrix} + \sum_{i=1}^{m} \gamma_{a,i} \begin{bmatrix} \mathbf{K}_i & \mathbf{L}_i \\ \mathbf{M}_i & \mathbf{N}_i \end{bmatrix} + \sum_{j=1}^{p} \gamma_{s,m+j} \begin{bmatrix} \mathbf{K}_{m+j} & \mathbf{L}_{m+j} \\ \mathbf{M}_{m+j} & \mathbf{N}_{m+j} \end{bmatrix}$$

Then for any nonsingular matrices $\mathbf{U}$ *and* $\mathbf{V}$*, such that* $\mathbf{U}\mathbf{V}^T = I - \mathbf{X}\mathbf{Y}$*, a controller* $\mathcal{K}(\Gamma_A, \Gamma_S)$ *that achieves* $\|\mathcal{F}_L(\mathcal{M}, \mathcal{K}(\Gamma_A, \Gamma_S))\|_\infty \leq 1$ *is parametrized by*

$$\begin{bmatrix} A_c & B_c \\ C_c & D_c \end{bmatrix} = \begin{bmatrix} \mathbf{U} & \mathbf{X}B_2 \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{K} - \mathbf{X}A\mathbf{Y} & \mathbf{L} \\ \mathbf{M} & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{V}^T & 0 \\ C_2\mathbf{Y} & I \end{bmatrix}^{-1} \tag{4.25}$$

Proof: Consider the interconnection of the system $\mathcal{M}$ (4.11) with the controller (4.10) that results in closed-loop system $\mathbb{T}_{cl} = \mathcal{F}_L(\mathcal{M}, \mathcal{K}(\Gamma_A, \Gamma_S))$ with state-space matrices

$$\mathbb{A}_{cl} = \begin{bmatrix} A + B_2D_cC_2 & B_2C_c \\ B_cC_2 & A_c \end{bmatrix}, \quad \mathbb{B}_{cl} = \begin{bmatrix} B_1(\Gamma_A) + B_2D_cD_{21} \\ B_cD_{21} \end{bmatrix}$$

$$\mathbb{C}_{cl} = \begin{bmatrix} C_1(\Gamma_S) + D_{12}D_cC_2 & D_{12}C_c \end{bmatrix}$$

$$\mathbb{D}_{cl} = D_{11}(\Gamma_A, \Gamma_S) + D_{12}D_cD_{21}$$

Then it is a fact (see Lemma 3.3 on page 73) that $\|\mathbb{T}_{cl}\|_\infty \leq 1$ will hold for all Γ_A and Γ_S if there exists a matrix $\mathbb{X} = \mathbb{X}^T$ and controller matrices (A_c, B_c, C_c, D_c) such that

$$\begin{bmatrix} \mathbb{X} & \mathbb{X}\mathbb{A}_{cl} & \mathbb{X}\mathbb{B}_{cl} & 0 \\ \star & \mathbb{X} & 0 & \mathbb{C}_{cl}^T \\ \star & \star & I & \mathbb{D}_{cl}^T \\ \star & \star & \star & I \end{bmatrix} > 0. \tag{4.26}$$

Note that the above inequality implies $\mathbb{X} > 0$, and that it is nonlinear in the unknowns. In order to linearize it we will perform a certain one-to-one change of variables. To this end denote

$$\mathbb{X} = \begin{bmatrix} \mathbf{X} & \mathbf{U} \\ \mathbf{U}^T & \bar{X} \end{bmatrix}, \quad \mathbb{X}^{-1} = \begin{bmatrix} \mathbf{Y} & \mathbf{V} \\ \mathbf{V}^T & \bullet \end{bmatrix}, \quad \mathbb{Y} = \begin{bmatrix} \mathbf{Y} & I \\ \mathbf{V}^T & 0 \end{bmatrix},$$

so that from $\mathbb{X}\mathbb{X}^{-1} = I$ it follows that

$$\begin{aligned} \mathbf{X}\mathbf{Y} + \mathbf{U}\mathbf{V}^T &= I \\ \mathbf{Y}\mathbf{U} + \mathbf{V}\bar{X} &= 0. \end{aligned}$$

Above, $\bullet$ denote entries that will be of no importance in the sequel. Next, note that

$$\mathbb{Y}^T\mathbb{X} = \begin{bmatrix} \mathbf{YX}+\mathbf{VU}^T & \mathbf{YU}+\mathbf{V}^T\bar{X} \\ \mathbf{X} & \mathbf{U} \end{bmatrix} = \begin{bmatrix} I & 0 \\ \mathbf{X} & \mathbf{U} \end{bmatrix},$$

$$\mathbb{Y}^T\mathbb{X}\mathbb{Y} = \begin{bmatrix} \mathbf{Y} & I \\ \mathbf{XY}+\mathbf{UV}^T & \mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{Y} & I \\ I & \mathbf{X} \end{bmatrix}.$$

are affine in $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{U}$. Multiplication of the first two (block) rows in (4.26) by $\mathbb{Y}^T$, and the first two columns by $\mathbb{Y}$ results in the equivalent inequality

$$\begin{bmatrix} \mathbb{Y}^T\mathbb{X}\mathbb{Y} & \mathbb{Y}^T\mathbb{X}\mathbb{A}_{cl}\mathbb{Y} & \mathbb{Y}^T\mathbb{X}\mathbb{B}_{cl} & 0 \\ \star & \mathbb{Y}^T\mathbb{X}\mathbb{Y} & 0 & \mathbb{Y}^T\mathbb{C}_{cl} \\ \star & \star & I & \mathbb{D}_{cl} \\ \star & \star & \star & I \end{bmatrix} > 0,$$

in which the matrices of interest can be written as

$$\left[\begin{array}{c|c} \mathbb{Y}^T\mathbb{X}\mathbb{A}_{cl}\mathbb{Y} & \mathbb{Y}^T\mathbb{X}\mathbb{B}_{cl} \\ \hline \mathbb{Y}^T\mathbb{C}_{cl} & \mathbb{D}_{cl} \end{array}\right] = \left[\begin{array}{cc|c} A\mathbf{Y} & A & B_1(\Gamma_A) \\ 0 & \mathbf{X}A & \mathbf{X}B_1(\Gamma_S) \\ \hline C_1(\Gamma_S)\mathbf{Y} & C_1(\Gamma_S) & D_{11}(\Gamma_A,\Gamma_S) \end{array}\right] +$$

$$\left[\begin{array}{cc} 0 & B_2 \\ I & 0 \\ \hline 0 & D_{12} \end{array}\right] \begin{bmatrix} \mathbf{K} & \mathbf{L} \\ \mathbf{M} & \mathbf{N} \end{bmatrix} \left[\begin{array}{cc|c} I & 0 & 0 \\ 0 & C_2 & D_{12} \end{array}\right]$$

where the linearizing change of variables was introduced

$$\begin{bmatrix} \mathbf{K} & \mathbf{L} \\ \mathbf{M} & \mathbf{N} \end{bmatrix} = \begin{bmatrix} \mathbf{U} & \mathbf{X}B_2 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_c & B_c \\ C_c & D_c \end{bmatrix} \begin{bmatrix} \mathbf{V}^T & 0 \\ C_2\mathbf{Y} & I \end{bmatrix} + \begin{bmatrix} \mathbf{X}A\mathbf{Y} & 0 \\ 0 & 0 \end{bmatrix}$$

This one-to-one reparametrization is used to convexify the problem, i.e. it allows us to solve a convex LMI optimization problem in the new variables $\mathbf{K}, \mathbf{L}, \mathbf{M}, \mathbf{N}$, and subsequently compute the controller matrices uniquely using the inverse transformation.

Note that equation (4.25) follows from here. Due to the fact that the closed-loop matrices are affine in the elements of Γ_A and Γ_S, the matrix inequality will hold for all $\gamma_{a,i} \in [\gamma^l_{a,i},\ \gamma^u_{a,i}]$, and $\gamma_{s,j} \in [\gamma^l_{s,j},\ \gamma^u_{s,j}]$, if it holds at the vertexes of the intervals $\gamma_{a,i} \in \{\gamma^l_{a,i}; \gamma^u_{a,i}\}$, and $\gamma_{s,j} \in \{\gamma^l_{s,j},\ \gamma^u_{s,j}\}$. By observing that the matrices B_2, C_2, and D_{12} are independent on Γ_A and Γ_S it becomes clear, that taking the matrices $\mathbf{K}$, $\mathbf{L}$, $\mathbf{M}$, and $\mathbf{N}$ as in (4.24) results in (4.23). □

Dealing with parametric uncertainty

Parametric uncertainty in the system matrices can also be considered and the same results, derived above for the state-feedback and output-feedback cases, can be used for LPV controller design. However, it should be pointed out that the structure of the parametric uncertainty is lost when one applies the results above due to the fact that they both make use of the small gain theorem.

To show how parametric uncertainty can be dealt with, consider the following fault-free uncertain system

$$\mathcal{S}(p):\left\{\begin{array}{rcl} x_{k+1} &=& \mathcal{A}(p)x_k+\mathcal{B}_\xi(p)\xi_k+\mathcal{B}_u(p)u_k \\ z_k &=& \mathcal{C}_z(p)x_k+\mathcal{D}_{z\xi}(p)\xi_k+\mathcal{D}_{zu}(p)u_k \\ y_k &=& \mathcal{C}_y(p)x_k+\mathcal{D}_{y\xi}(p)\xi_k, \end{array}\right. \tag{4.27}$$

where the parameter vector $p=[p_1,p_2,\ldots,p_N]^T$, with $p_i\in[-1,1]$ and

$$\left[\begin{array}{ccc} \mathcal{A}(p) & \mathcal{B}_\xi(p) & \mathcal{B}_u(p) \\ \mathcal{C}_z(p) & \mathcal{D}_{z\xi}(p) & \mathcal{D}_{zu}(p) \\ \mathcal{C}_y(p) & \mathcal{D}_{y\xi}(p) & 0 \end{array}\right] \doteq \left[\begin{array}{ccc} \mathcal{A}^0 & \mathcal{B}_\xi^0 & \mathcal{B}_u^0 \\ \mathcal{C}_z^0 & \mathcal{D}_{z\xi}^0 & \mathcal{D}_{zu}^0 \\ \mathcal{C}_y^0 & \mathcal{D}_{y\xi}^0 & 0 \end{array}\right] + \sum_{i=1}^{N} p_i \left[\begin{array}{ccc} \mathcal{A}^i & \mathcal{B}_\xi^i & \mathcal{B}_u^i \\ \mathcal{C}_z^i & \mathcal{D}_{z\xi}^i & \mathcal{D}_{zu}^i \\ \mathcal{C}_y^i & \mathcal{D}_{y\xi}^i & 0 \end{array}\right].$$

The idea is to pull the uncertain parameters p_i out of the system like we did above with the FDD uncertainties Γ_A and Γ_S. To this end let

$$r_i=\mathbf{rank}\left[\begin{array}{ccc} \mathcal{A}^i & \mathcal{B}_\xi^i & \mathcal{B}_u^i \\ \mathcal{C}_z^i & \mathcal{D}_{z\xi}^i & \mathcal{D}_{zu}^i \\ \mathcal{C}_y^i & \mathcal{D}_{y\xi}^i & 0 \end{array}\right],$$

and define (for instance, by using the singular value decomposition) for $i=1,2,\ldots,N$ the matrices

$$\left[\begin{array}{c} E_i \\ F_i^z \\ F_i^y \end{array}\right]\left[\begin{array}{ccc} G_i & H_i^\xi & F_i^u \end{array}\right] \doteq \left[\begin{array}{ccc} \mathcal{A}^i & \mathcal{B}_\xi^i & \mathcal{B}_u^i \\ \mathcal{C}_z^i & \mathcal{D}_{z\xi}^i & \mathcal{D}_{zu}^i \\ \mathcal{C}_y^i & \mathcal{D}_{y\xi}^i & 0 \end{array}\right],$$

where

$$\begin{array}{ll} E_i\in\mathbb{R}^{n\times r_i}, & G_i\in\mathbb{R}^{r_i\times n}, \\ F_i^z\in\mathbb{R}^{n_z\times r_i}, & H_i^\xi\in\mathbb{R}^{r_i\times n_\xi}, \\ F_i^y\in\mathbb{R}^{p\times r_i}, & H_i^u\in\mathbb{R}^{r_i\times m}. \end{array}$$

It can then easily be verified that the system

$$\mathcal{S}_{unc}:\left\{\begin{array}{rcl} x_{k+1} &=& Ax_k+B_\xi\bar{\xi}_k+B_u u_k \\ \bar{z}_k &=& C_z x_k+D_{z\xi}\bar{\xi}_k+D_{zu}u_k \\ y_k &=& C_y x_k+D_{y\xi}\bar{\xi}_k \\ && \\ \bar{\xi}_k &=& \mathbf{diag}\{p_1 I_{r_1},\ \ldots,\ p_N I_{r_N},\ I_{n_z}\}\bar{z}_k \end{array}\right. \tag{4.28}$$

with matrices

$$A\doteq\ \mathcal{A}^0 \qquad B_\xi\doteq\left[\begin{array}{cccc} E_1 & \ldots & E_N & \mathcal{B}_\xi^0 \end{array}\right] \qquad B_u\doteq\ \mathcal{B}_u^0$$

$$C_z\doteq\left[\begin{array}{c} G_1 \\ \vdots \\ G_N \\ \mathcal{C}_z^0 \end{array}\right] \qquad D_{z\xi}\doteq\left[\begin{array}{cccc} 0 & \ldots & 0 & H_1^\xi \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & H_N^\xi \\ F_1^z & \ldots & F_N^z & \mathcal{D}_{z\xi}^0 \end{array}\right] \qquad D_{zu}\doteq\left[\begin{array}{c} H_1^u \\ \vdots \\ H_N^u \\ \mathcal{D}_{zu}^0 \end{array}\right]$$

$$C_y\doteq\ \mathcal{C}_y^0 \qquad D_{y\xi}\doteq\left[\begin{array}{cccc} F_1^y & \ldots & F_N^y & \mathcal{D}_{y\xi}^0 \end{array}\right]$$

is equivalent to the system (4.27). Therefore, by substituting the system (4.1) with (4.28) we can make use of the same approach to the design of the state-feedback and the output-feedback LPV controllers, presented above.

4.2.3 Controller Reconfiguration Strategy

Next, we consider the problem of controller reconfiguration. As already discussed, the approach is based on a predesigned set $\mathcal{S}_{\mathcal{K}}$ of parameter dependent controllers. Each controller is designed for a given faulty model by making use of the results in the previous section. The reconfigured controller is then taken as a scaled version of an appropriately selected parameter dependent controller from the set $\mathcal{S}_{\mathcal{K}}$.

Define the following two sets of total faults

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{A}}^{\boldsymbol{t}} &\doteq \{\Sigma_A \in \boldsymbol{\Sigma}_{\boldsymbol{A}} : \Sigma_A = \Sigma_A^{\dagger}\} \subset \boldsymbol{\Sigma}_{\boldsymbol{A}} \\ \boldsymbol{\Sigma}_{\boldsymbol{S}}^{\boldsymbol{t}} &\doteq \{\Sigma_S \in \boldsymbol{\Sigma}_{\boldsymbol{S}} : \Sigma_S = \Sigma_S^{\dagger}\} \subset \boldsymbol{\Sigma}_{\boldsymbol{S}}, \end{aligned} \tag{4.29}$$

where the notation $A^{\dagger}$ denotes the pseudo-inverse of A, i.e. if the singular value decomposition of A is $U\mathbf{diag}\,(\lambda_1, \ldots, \lambda_r, 0, \ldots, 0)\, V^T$, then

$$A^{\dagger} = U\mathbf{diag}\left(\lambda_1^{-1}, \ldots, \lambda_r^{-1}, 0, \ldots, 0\right) V^T.$$

The set $\boldsymbol{\Sigma}_{\boldsymbol{A}}^{\boldsymbol{t}}$ ($\boldsymbol{\Sigma}_{\boldsymbol{S}}^{\boldsymbol{t}}$) represents all possible combinations of *total* actuator (sensor) faults that do not affect the stabilizability (detectability) of the system.

Let us suppose that a set of N controllers, each dependent on the time-varying parameters Γ_A and Γ_S, has been designed

$$\mathcal{S}_{\mathcal{K}} = \{\mathcal{K}_1(\Gamma_A, \Gamma_S), \mathcal{K}_2(\Gamma_A, \Gamma_S), \ldots, \mathcal{K}_N(\Gamma_A, \Gamma_S)\}, \tag{4.30}$$

where the i-th controller is designed for the faulty system (4.2)-(4.4) for some given $\hat{\Sigma}_A^i \in \boldsymbol{\Sigma}_{\boldsymbol{A}}^{\boldsymbol{t}}$ and $\hat{\Sigma}_S^i \in \boldsymbol{\Sigma}_{\boldsymbol{S}}^{\boldsymbol{t}}$, by making use of the results in Section 4.2.2. In order to be able to deal with any possible combination of sensor and actuator faults, the matrices $(\hat{\Sigma}_A^i, \hat{\Sigma}_S^i)$, need to be selected such that

1. $\hat{\Sigma}_A^i \neq 0$, $\hat{\Sigma}_S^i \neq 0$, for all $i = 1, 2, \ldots, N$, and
2. for any $\tilde{\Sigma}_A \in \boldsymbol{\Sigma}_{\boldsymbol{A}}^{\boldsymbol{t}}$ and $\tilde{\Sigma}_S \in \boldsymbol{\Sigma}_{\boldsymbol{S}}^{\boldsymbol{t}}$ there exists an index i for which

$$\begin{aligned} \tilde{\Sigma}_A \hat{\Sigma}_A^i &= \hat{\Sigma}_A^i, \text{ and} \\ \tilde{\Sigma}_S \hat{\Sigma}_S^i &= \hat{\Sigma}_S^i. \end{aligned} \tag{4.31}$$

In other words, for any pair of total faults $(\tilde{\Sigma}_A, \tilde{\Sigma}_S) \in \boldsymbol{\Sigma}_{\boldsymbol{A}}^{\boldsymbol{t}} \times \boldsymbol{\Sigma}_{\boldsymbol{S}}^{\boldsymbol{t}}$ there should be at least one controller $\mathcal{K}_i(\cdot, \cdot)$ that does not use the totally failed sensors and actuators described by the diagonal elements of the matrices $(\tilde{\Sigma}_A, \tilde{\Sigma}_S)$. This is illustrated in the following example.

Example 4.1 *Consider only actuator faults, and let the system have three actuators. Suppose also that the system is controllable by each individual actuator. Then the set of admissible actuator faults is given by (see equation (4.3) on page 98)*

$$\boldsymbol{\Sigma}_{\boldsymbol{A}} = \left\{ \begin{bmatrix} \alpha_1 & & \\ & \alpha_2 & \\ & & \alpha_3 \end{bmatrix} : \sum_{i=1}^{3} |\alpha_i| \neq 0 \right\}.$$

In other words only the case when all three actuators are totally failed does not belong to $\mathbf{\Sigma_A}$ *since this is the only case when the system is not stabilizable. Therefore, the set of total actuator faults* $\mathbf{\Sigma}^t_{\mathbf{A}}$*, defined in (4.29), in this example takes the form*

$$\mathbf{\Sigma}^t_{\mathbf{A}} = \left\{ \begin{bmatrix} \alpha_1 & & \\ & \alpha_2 & \\ & & \alpha_3 \end{bmatrix} : \alpha_i \in \{0; 1\}, \sum_{i=1}^{3} \alpha_i \neq 0 \right\}.$$

In this case the minimum number of controllers in the bank $\mathcal{S}_{\mathcal{K}}$ *is three and corresponds to the following actuator fault patterns*

$$\hat{\Sigma}^1_A = \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix}, \ \hat{\Sigma}^2_A = \begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix}, \ \hat{\Sigma}^3_A = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix}.$$

Indeed, now for any total actuator fault $\tilde{\Sigma}_A$ *from the set* $\mathbf{\Sigma}^t_{\mathbf{A}}$*, there exists at least one* $\hat{\Sigma}^i_A$ *for which (4.31) holds. For instance, if* $\tilde{\Sigma}_A = \mathbf{diag}\{[1,\ 0,\ 1]\}$*, then both* $\hat{\Sigma}^1_A$ *and* $\hat{\Sigma}^3_A$ *are such that* $\tilde{\Sigma}_A \hat{\Sigma}^1_A = \hat{\Sigma}^1_A$ *and* $\tilde{\Sigma}_A \hat{\Sigma}^3_A = \hat{\Sigma}^3_A$*, so that either controller* $\mathcal{K}_1$ *or* $\mathcal{K}_3$ *can be used (none of them uses the totally failed second actuator). The idea exploited below is then to select the controller that can achieve a better performance for the closed-loop system.*

We note here that in addition to the three $\hat{\Sigma}^i_A$*'s, defined above, one may also include four additional: three corresponding to the three individual total actuator faults, and one for the fault-free case. This would further improve the achievable performance for a particular combination of total faults.*

Further than that there is no restriction on how to select the controllers. It should be noted, that if the control system $\mathcal{S}$ is stabilizable by each single input, and is detectable from each single output, then the minimal number of controllers that will be needed is (mp), i.e. there should be one controller for each input-output channel.

Thus, the i-th controller $\mathcal{K}_i(\Gamma_A, \Gamma_S)$ is such that

$$\sup_{\substack{\Delta_A, \Delta_S \\ \Gamma_A, \Gamma_S}} \|\mathcal{F}_L(\mathcal{S}_F(\hat{\Sigma}^i_A, \hat{\Sigma}^i_S, \Gamma_A, \Gamma_S), \mathcal{K}_i(\Gamma_A, \Gamma_S))\|_\infty \leq \frac{1}{\gamma_i}$$

for some $\gamma_i > 0$.

Now that the set of controllers has been defined, we are ready to present the reconfiguration scheme. Consider the system (4.2), and suppose that a combination of sensor and actuator faults has occurred in the system, represented by the diagonal matrices Σ_A and Σ_S as in (4.4). Consider also the set of local controllers (4.30). Define the matrices

$$\begin{aligned} \hat{\Sigma}^t_A &= \hat{\Sigma}_A \hat{\Sigma}^\dagger_A, \text{ and} \\ \hat{\Sigma}^t_S &= \hat{\Sigma}_S \hat{\Sigma}^\dagger_S, \end{aligned}$$

which carry information about the *total* faults only. Let $\hat{\Sigma}^p_A$ and $\hat{\Sigma}^p_S$ are any nonsingular matrices of appropriate dimensions such that

$$\begin{aligned} \hat{\Sigma}_A &= \hat{\Sigma}^p_A \hat{\Sigma}^t_A, \text{ and} \\ \hat{\Sigma}_S &= \hat{\Sigma}^p_S \hat{\Sigma}^t_S. \end{aligned} \tag{4.32}$$

In this way we have actually split the faults into total and partial.

Let

$$i_{opt} \in \left\{ \arg\max_{i} \{\gamma_i : \ \hat{\Sigma}_A^t \hat{\Sigma}_A^i = \hat{\Sigma}_A^i, \ \hat{\Sigma}_S^t \hat{\Sigma}_S^i = \hat{\Sigma}_S^i, \} \right\}.$$

Then the controller

$$\mathcal{K}_R(\Gamma_A, \Gamma_S) = \left(\hat{\Sigma}_S^p\right)^{-1} \mathcal{K}_{i_{opt}}(\Gamma_A, \Gamma_S) \left(\hat{\Sigma}_A^p\right)^{-1} \tag{4.33}$$

achieves

$$\sup_{\substack{\Delta_A, \Delta_S \\ \Gamma_A, \Gamma_S}} \left\| \mathcal{F}_L \left(\mathcal{S}_F(\hat{\Sigma}_A, \hat{\Sigma}_S, \Gamma_A, \Gamma_S), \mathcal{K}_R(\Gamma_A, \Gamma_S) \right) \right\|_\infty \leq \frac{1}{\gamma_{i_{opt}}}$$

This becomes clear by observing that for any $i = 1, 2, \ldots, N$

$$\begin{aligned} &\mathcal{F}_L \left(\mathcal{S}_F(\hat{\Sigma}_A, \hat{\Sigma}_A, \Gamma_A, \Gamma_S), (\hat{\Sigma}_S^p)^{-1} \mathcal{K}_i(\Gamma_A, \Gamma_S)(\hat{\Sigma}_A^p)^{-1} \right) \\ &= \mathcal{F}_L \left(\mathcal{S}_F(\hat{\Sigma}_A^t, \hat{\Sigma}_S^t, \Gamma_A, \Gamma_S), \mathcal{K}_i(\Gamma_A, \Gamma_S) \right) \end{aligned}$$

Since the i-th controller is designed for total faults represented by the matrices $(\hat{\Sigma}_A^i, \hat{\Sigma}_S^i)$ we can write that

$$\mathcal{K}_i(\Gamma_A, \Gamma_S) = \hat{\Sigma}_A^i \mathcal{K}_i(\Gamma_A, \Gamma_S) \hat{\Sigma}_S^i,$$

and thus

$$\begin{aligned} &\mathcal{F}_L \left(\mathcal{S}_F(\hat{\Sigma}_A^t, \hat{\Sigma}_S^t, \Gamma_A, \Gamma_S), \mathcal{K}_i(\Gamma_A, \Gamma_S) \right) \\ &= \mathcal{F}_L \left(\mathcal{S}_F(\hat{\Sigma}_A^t \hat{\Sigma}_A^i, \hat{\Sigma}_S^i \hat{\Sigma}_S^t, \Gamma_A, \Gamma_S), \mathcal{K}_i(\Gamma_A, \Gamma_S) \right) \end{aligned}$$

so that for any i, for which $\hat{\Sigma}_A^t \hat{\Sigma}_A^i = \hat{\Sigma}_A^i$ and $\hat{\Sigma}_S^t \hat{\Sigma}_S^i = \hat{\Sigma}_S^i$ hold, the controller $\mathcal{K}_i(\Gamma_A, \Gamma_S)$ is such that

$$\sup_{\substack{\Delta_A, \Delta_S \\ \Gamma_A, \Gamma_S}} \left\| \mathcal{F}_L \left(\mathcal{S}_F(\hat{\Sigma}_A^i, \hat{\Sigma}_S^i, \Gamma_A, \Gamma_S), \mathcal{K}_R(\Gamma_A, \Gamma_S) \right) \right\|_\infty \leq \frac{1}{\gamma_i}.$$

is satisfied.

Of course, after the controller $\mathcal{K}_R(\Gamma_A, \Gamma_S)$ has been switched on, if the performance, measured by the number $1/\gamma_{i_{opt}}$ is not satisfactory, then one may wish to solve the optimization problem in Section 4.2.2 on-line. While this may be time consuming, the computational time is no longer of critical importance.

This deterministic method will be illustrated on a case study in Section 4.4. In the next section we present the probabilistic design approach to LPV-based robust active FTC, that can deal with component faults in addition to sensor and actuator faults.

4.3 Probabilistic Method for Component Faults

In this second section of this chapter we develop a scheme that implements a robust active FTC design applicable to systems with *sensor, actuator and component faults* in the presence of model uncertainty. The estimates of the faults in the system, obtained from the FDD scheme, are considered imprecise (uncertain). These estimates are here again assumed to lie inside some uncertainty intervals, the sizes of which are assumed given. The parameter-varying robust active FTC that is derived, however, is scheduled not only by the sizes of the uncertainty intervals γ_f (see Figure 4.1) as in the previous section, but also by the fault estimates $\hat{f}$. this is made possible by making use of the probabilistic framework from Chapter 2. It is this probabilistic design method that also allows us here to consider a much wider class of faults as well as structured model uncertainties.

4.3.1 Problem Formulation

Below we consider the nominal system $\mathcal{S}_{nom}$ defined in (4.1) on page 97, in which sensor, actuator and/or component faults can occur. The faulty system description is assumed in the form

$$\mathcal{S}_F : \left\{ \begin{array}{rcl} x_{k+1} & = & A^{\Delta}(f)x_k + B^{\Delta}_{\xi}(f)\xi_k + B^{\Delta}_u(f)u_k \\ z_k & = & C^{\Delta}_z(f)x_k + D^{\Delta}_{z\xi}(f)\xi_k + D^{\Delta}_{zu}(f)u_k \\ y_k & = & C^{\Delta}_y(f)x_k + D^{\Delta}_{y\xi}(f)\xi_k + D^{\Delta}_{yu}(f)u_k, \end{array} \right. \tag{4.34}$$

where $x_k \in \mathbb{R}^n$ is the state of the system, $u \in \mathbb{R}^m$ is the control action, $y \in \mathbb{R}^p$ is the measured output, $z \in \mathbb{R}^{n_z}$ represents the controlled output of the system, and $\xi \in \mathbb{R}^{\xi}$ is the disturbance to the system. The vector $f(k) \in \mathcal{F} \subset \mathbf{R}^{n_f}$ represents faults in the system. Note that this general representation could be used to model a wide class of sensor, actuator and component faults. Δ represents the uncertainty in the system, which is assumed as in Chapter 2 to belong to some bounded set $\mathbf{\Delta}$ with a given probability density function $f_{\mathbf{\Delta}}(\Delta)$ inside $\mathbf{\Delta}$. In addition to that, since the probabilistic framework is used in this section, it is assumed that random uncertainty samples can be generated with the specified probability density function $f_{\mathbf{\Delta}}(\Delta)$. Furthermore, Δ is assumed to be such that

$$\left\| \begin{bmatrix} A^{\Delta}(f) & B^{\Delta}_{\xi}(f) & B^{\Delta}_u(f) \\ C^{\Delta}_z(f) & D^{\Delta}_{z\xi}(f) & D^{\Delta}_{zu}(f) \\ C^{\Delta}_y(f) & D^{\Delta}_{y\xi}(f) & D^{\Delta}_{yu}(f) \end{bmatrix} \right\|_2 < \infty, \ \forall \Delta \in \mathbf{\Delta}, \ f \in \mathcal{F}.$$

As discussed above, the estimates $\hat{f}(k)$ are assumed to be imprecise, so that the i-th entry of f is represented as

$$f_i(k) = (1 + \gamma_{f,i}(k)\hat{\Delta}_i)\hat{f}_i(k), \ i = 1, 2, \dots, n_f, \tag{4.35}$$

where $|\hat{\Delta}_i| \leq 1$ represents the FDD uncertainty, and where $\gamma_{f,i}(k)$ defines the size of the uncertainty in the sense that (4.35) is equivalent to $f_i(k) = (1+\bar{\Delta}_i(k))\hat{f}_i(k)$ with $|\bar{\Delta}_i(k)| \leq \gamma_{f,i}(k)$. We will however make use of the FDD uncertainty representation in equation (4.35) where the uncertainty $\hat{\Delta}_i$ is normalized since we will later on design a controller scheduled by both the fault estimates $\hat{f}(k)$ and the uncertainty sizes $\gamma_{f,i}(k)$. The uncertainty sizes $\gamma_{f,i}$ are allowed to be time varying, they are provided by the FDD scheme together with the fault estimates (see

Figure 4.1 on page 96). In addition, we will denote the set in which the matrix $\hat{\Delta}$, representing the FDD uncertainty, can take values as

$$\hat{\mathbf{\Delta}} \doteq \left\{ \hat{\Delta} = \text{diag}(\hat{\Delta}_1, \ldots, \hat{\Delta}_{n_f}) : \, |\hat{\Delta}_i| \leq 1, \, i = 1, 2, \ldots, n_f \right\}.$$

Both the fault estimates $\hat{f}(k)$ and the uncertainty sizes $\gamma_f(k)$ are assumed to belong to some known interval sets,

$$\begin{array}{l} \gamma_f(k) \in \Omega_\gamma = \{ w \in \mathbb{R}^{n_f} \, : \, \gamma_{f,min} \leq w \leq \gamma_{f,max} \} \\ \hat{f}(k) \in \Omega_f = \{ w \in \mathbb{R}^{n_f} \, : \, f_{min} \leq w \leq f_{max} \}. \end{array}$$

In this subsection the control objective is specified as an LPV design problem, where the goal is to design a controller that can be scheduled by the fault estimates $\hat{f}_i$ and the FDD uncertainty sizes $\gamma_{f,i}$, i.e. an LPV controller of the form $\mathcal{K} = \mathcal{K}(\hat{f}, \gamma_f)$.

For a reason that will become clear shortly we split the controlled output vector z into two vectors

$$z(k) = \left[\begin{array}{c} z_1(k) \\ z_2(k) \end{array} \right] \begin{array}{l} \} \, n_{z1} \\ \} \, n_{z2} \end{array}, \quad \text{with } n_z = n_{z1} + n_{z2}. \tag{4.36}$$

For some given bounded functions $g_i(\hat{f}, \gamma_f) \, : \, \Omega_f \times \Omega_\gamma \mapsto \mathbb{R}$ we consider the following parametrization of the LPV controller:

$$\mathcal{K}(\hat{f}, \gamma_f) = \mathcal{K}_0 + \sum_{i=1}^{n_f} g_i(\hat{f}, \gamma_f) \mathcal{K}_i, \tag{4.37}$$

where

$$\mathcal{K}(\hat{f}, \gamma_f) = \left[\begin{array}{cc} A^c(\hat{f}, \gamma_f) & B^c(\hat{f}, \gamma_f) \\ F(\hat{f}, \gamma_f) & 0 \end{array} \right], \, \mathcal{K}_i = \left[\begin{array}{cc} A_i^c & B_i^c \\ F_i & 0 \end{array} \right], \, i = 0, 1, \ldots, n_f, \tag{4.38}$$

with $A^c \in \mathbb{R}^{n \times n}$, $B^c \in \mathbb{R}^{n \times p}$, and $F \in \mathbb{R}^{m \times n}$. Thus, only strictly proper full-order controllers are considered.

Let $T^{\Delta}_{\xi \mapsto z_1}(f)$ and $T^{\Delta}_{\xi \mapsto z_2}(f)$ denote the closed-loop system from the disturbance signal $\xi(k)$ to $z_1(k)$ and $z_2(k)$, respectively. Note that, in view of (4.35), these operators depend on $\hat{f}$ and γ_f. The problem considered in this paper is formulated as follows (see Figure 4.5): find an LPV controller (4.37) by solving the problem

$$\begin{array}{lcl} \min\limits_{F(\hat{f}, \gamma_f)} & \sup\limits_{\substack{\Delta \in \mathbf{\Delta}, \, \hat{\Delta} \in \hat{\mathbf{\Delta}} \\ \hat{f} \in \Omega_f, \, \gamma_f \in \Omega_\gamma}} & \|T^{\Delta}_{\xi \mapsto z_1}(\hat{f}, \gamma_f)\|_\infty \\ \text{subject to} & \sup\limits_{\substack{\Delta \in \mathbf{\Delta}, \, \hat{\Delta} \in \hat{\mathbf{\Delta}} \\ \hat{f} \in \Omega_f, \, \gamma_f \in \Omega_\gamma}} & \|T^{\Delta}_{\xi \mapsto z_2}(\hat{f}, \gamma_f)\|_\infty \leq 1 \end{array} \tag{4.39}$$

We note here that in the optimization (4.39) instead of the $\mathcal{H}_\infty$-norm one may prefer to use the $\mathcal{H}_2$-norm. One would then need to follow the same lines of

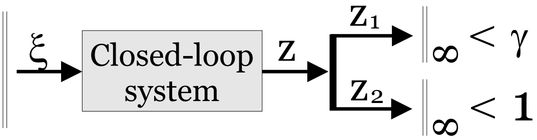


Figure 4.5: Design objective specification.

reasoning as below to derive the corresponding $\mathcal{H}_2$ controller. In this Section 4.3 we focus on the $\mathcal{H}_\infty$-norm.

The term that is optimized in (4.39) defines the performance index, i.e. the cost function, while the second term can be used to handle input constraints. If this is not desired and all controlled outputs $z(k)$ need to be included into the cost function, one should simply choose $n_{z2} = 0$, so that $z_1 \equiv z$.

For convenience we write the optimization problem (4.39) in the following form

$$\begin{array}{lll} \min\limits_{F(\hat{f},\gamma_f)} & \gamma & \\ \text{subject to} & \sup\limits_{\substack{\Delta \in \mathbf{\Delta},\ \hat{\Delta} \in \hat{\mathbf{\Delta}} \\ \hat{f} \in \Omega_f,\ \gamma_f \in \Omega_\gamma}} & \left\| \begin{bmatrix} \gamma^{-1} T^{\Delta}_{\xi \mapsto z_1}(\hat{f},\gamma_f) \\ T^{\Delta}_{\xi \mapsto z_2}(\hat{f},\gamma_f) \end{bmatrix} \right\|_\infty \leq 1 \end{array} \tag{4.40}$$

Two main difficulties arise when one tries to solve problem (4.40). The first difficulty in solving the problem (4.40) lies in the fact that the function is not convex in the parameters $\hat{f}$ and γ_f since both the system and the controller depend on them. For this reason the standard LMI techniques cannot be used to solve the problem. For that reason the approach for probabilistic design of Chapter 2 will be used here.

The second difficulty appears when *output-feedback* LPV controller needs to be designed. In the output-feedback case the problem (4.40) is not convex in the design variables and it is known that there exist no algorithms that can find the global optimum in polynomial-time, i.e. the problem is NP-hard (see Chapter 3). In this section we will address this problem by making use of the method for initial robust controller computation derived in Section 3.4 of Chapter 3. In this method the output-feedback case is solved in two steps by designing a state-feedback gain in the first step that is then used at the second step where the remaining controller matrices are computed. We remind here that since in the second step the problem remains nonconvex, a constraint on the structure of the Lyapunov matrix is introduced to convexify it, which however increases the conservatism of the method. Although this method converges, there are no guarantees that it will converge to the global or even to a local optimum of the cost function (see Chapter 3 for more details on this method). Still, numerous experiments performed with this method so far yield very good results in almost all cases.

4.3.2 State-Feedback Case

We will begin by considering the state-feedback case, assuming that the state is directly measured. The controller is in this case taken of the form (4.37) with

$$\mathcal{K}(\hat{f},\gamma_f) = F(\hat{f},\gamma_f),\ \mathcal{K}_i = F_i,\ i = 0, 2, \ldots, n_f.$$

Let us also, in accordance with the optimization problem defined in (4.40), define the matrices

$$\begin{bmatrix} \bar{C}_z^{\Delta} & \bar{D}_{z\xi}^{\Delta} & \bar{D}_{zu}^{\Delta} \end{bmatrix}(f) \doteq \begin{bmatrix} \gamma^{-1} I_{n_{z1}} & \\ & I_{n_{z2}} \end{bmatrix} \begin{bmatrix} C_z^{\Delta} & D_{z\xi}^{\Delta} & D_{zu}^{\Delta} \end{bmatrix}(f). \tag{4.41}$$

Applying the state-feedback control law

$$u(k) = F(\hat{f},\gamma_f)x(k) = \left(F_0 + \sum_{i=1}^{n_f} g_i(\hat{f},\gamma_f) F_i \right) x(k) \tag{4.42}$$

results in the following closed-loop system

$$T_{st.fb.}^{\Delta}(\hat{f},\gamma_f) : \left\{ \begin{array}{r} x_{k+1} = (A^{\Delta}(f) + B_u^{\Delta}(f)F(\hat{f},\gamma_f))x_k + B_{\xi}^{\Delta}(f)\xi_k \\ \bar{z}_k = (\bar{C}_z^{\Delta}(f) + \bar{D}_{zu}^{\Delta}(f)F(\hat{f},\gamma_f))x_k + \bar{D}_{z\xi}^{\Delta}(f)\xi_k \end{array} \right. \tag{4.43}$$

The following result then holds (consult, for instance, Lemma 3.3 on page 73).

Lemma 4.2 *Suppose that the matrices* $\boldsymbol{P} = \boldsymbol{P}^T$ *and* $\mathbf{L_i}$, $i = 0, 1, \ldots, n_f$, *be such that for all* $\Delta \in \boldsymbol{\Delta}$, $\hat{\Delta} \in \hat{\boldsymbol{\Delta}}$, $\hat{f} \in \Omega_f$, $\gamma_f \in \Omega_\gamma$ *it holds that*

$$\begin{bmatrix} \mathbf{P} & A^{\Delta}(f)\mathbf{P} + B_u^{\Delta}(f)\mathbf{L}(\hat{f},\gamma_f) & B_{\xi}^{\Delta}(f) & 0 \\ \bullet & \mathbf{P} & 0 & (\bar{C}_z^{\Delta}(f)\mathbf{P} + \bar{D}_{zu}^{\Delta}(f)\mathbf{L}(\hat{f},\gamma_f))^T \\ \bullet & \bullet & I & (\bar{D}_{z\xi}^{\Delta}(f))^T \\ \bullet & \bullet & \bullet & I \end{bmatrix} > 0, \tag{4.44}$$

where it is denoted $\mathbf{L}(\hat{f},\gamma_f) = \mathbf{L_0} + \sum_{i=1}^{N_g} g_i(\hat{f},\gamma_f)\mathbf{L_i}$, *and where the elements of the vector* f *are defined in (4.35). Then the state-feedback gain (4.42) with* $F_i = \mathbf{L_i}\mathbf{P}^{-1}$, $i = 0, 1, \ldots, n_f$, *achieves*

$$\sup_{\substack{\Delta \in \boldsymbol{\Delta},\ \hat{\Delta} \in \hat{\boldsymbol{\Delta}} \\ \hat{f} \in \Omega_f,\ \gamma_f \in \Omega_\gamma}} \|T_{st.fb.}^{\Delta}(\hat{f},\gamma_f)\|_{\infty} \leq 1. \tag{4.45}$$

Proof: The proof follows by applying the discrete-time version of Lemma 3.3 on page 73 on the closed-loop system (4.43) and then performing the convexifying change of variables $\mathbf{L_i} = F_i\mathbf{P}^{-1}$. □

This result can be used to compute a feasible solution (if such exists) to the problem (4.40) in the state-feedback case with fixed γ (note that γ enters the matrix inequality (4.44) via the matrices $\bar{C}_z^{\Delta}(f)$ $\bar{D}_{zu}^{\Delta}(f)$, and $\bar{D}_{z\xi}^{\Delta}(f)$. A bisection algorithm on γ can then be used to approach the solution to the original optimization problem (4.40).

Note that the matrix inequality (4.44) is linear in the unknowns, but not necessarily convex in $\hat{f}$, Δ, $\hat{\Delta}$, and γ_f. For that reason the standard LMI tools cannot be used. Instead, problem (4.44) fits very nicely in the probabilistic framework developed in Chapter 2 of this thesis, which will therefore be used to solve it. Before that, however, we first pay attention to the output-feedback case.

4.3.3 Output-Feedback Case

We next consider the output-feedback case and we will be searching for controller of the form (4.37)-(4.38) needs to be designed. This controller results in the following closed-loop system

$$T^{\Delta}_{out.fb.}(f) : \left\{ \begin{array}{rcl} x_{k+1} & = & \mathbb{A}_{cl} x_k + \mathbb{B}_{cl} \xi_k \\ \bar{z}_k & = & \mathbb{C}_{cl} x_k + \mathbb{D}_{cl} \xi_k, \end{array} \right. \tag{4.46}$$

with

$$\mathbb{A}_{cl} = \left[\begin{array}{cc} A^{\Delta}(f) & B^{\Delta}_{u}(f) F(\hat{f}, \gamma_f) \\ B_c(\hat{f}, \gamma_f) C^{\Delta}_{y}(y) & A_c(\hat{f}, \gamma_f) + B_c(\hat{f}, \gamma_f) D^{\Delta}_{yu}(f) F(\hat{f}, \gamma_f) \end{array} \right],$$

$$\mathbb{B}_{cl} = \left[\begin{array}{c} B^{\Delta}_{\xi}(f) \\ B_c(\hat{f}, \gamma_f) D^{\Delta}_{y\xi}(f) \end{array} \right],$$

$$\mathbb{C}_{cl} = \left[\begin{array}{cc} \bar{C}^{\Delta}_{y}(f) & \bar{D}^{\Delta}_{yu}(f) F(\hat{f}, \gamma_f) \end{array} \right], \mathbb{D}_{cl} = \bar{D}^{\Delta}_{z\xi}(f) \xi_k,$$

where the matrices $\bar{C}^{\Delta}_{y}(f)$, $\bar{D}^{\Delta}_{yu}(f)$, and $\bar{D}^{\Delta}_{z\xi}(f)$ are defined in (4.41).

Suppose that a state-feedback controller has been found in the form (4.42) by using the result in Lemma 4.2. The remaining matrices $A_c(\hat{f}, \gamma_f)$ and $B_c(\hat{f}, \gamma_f)$ of the LPV controller (4.38) can be found using the following result

Lemma 4.3 *Suppose that the matrices* $\mathbf{X} = \mathbf{X}^T$, $\mathbf{Y} = \mathbf{Y}^T$, $\mathbf{Z}_\mathbf{i}$ *and* $\mathbf{G}_\mathbf{i}$, $i = 0, 1, \ldots, n_f$, *are such that for all* $\Delta \in \boldsymbol{\Delta}$, $\hat{\Delta} \in \hat{\boldsymbol{\Delta}}$, $\hat{f} \in \Omega_f$, $\gamma_f \in \Omega_\gamma$ *it holds that*

$$\left[\begin{array}{cccc} \mathbf{P} & \mathbf{M}(\hat{f}, \gamma_f) & \mathbf{N}(\hat{f}, \gamma_f) & 0 \\ \bullet & \mathbf{P} & 0 & R(\hat{f}, \gamma_f)^T \\ \bullet & \bullet & I & D^T_{z\xi} \\ \bullet & \bullet & \bullet & I \end{array} \right] > 0 \tag{4.47}$$

where the elements of the vector f *are defined in (4.35), and where the matrices*

$\mathbf{M}(\hat{f},\gamma_f)$, $\mathbf{N}(\hat{f},\gamma_f)$, $R(\hat{f},\gamma_f)$, *and* $\mathbf{P}$ *are defined as*

$$\begin{array}{rcl}
\mathbf{M}(\hat{f},\gamma_f) & = & \left[\begin{array}{cc} \mathbf{M}_{11}(\hat{f},\gamma_f) & \mathbf{M}_{12}(\hat{f},\gamma_f) \\ \mathbf{M}_{21}(\hat{f},\gamma_f) & \mathbf{M}_{22}(\hat{f},\gamma_f) \end{array}\right], \\
\mathbf{M}_{11}(\hat{f},\gamma_f) & = & \mathbf{X}(A^{\Delta}(f) + B_u^{\Delta}(f)F(\hat{f},\gamma_f)) \\
\mathbf{M}_{21}(\hat{f},\gamma_f) & = & \mathbf{Y}(A^{\Delta}(f) + B_u^{\Delta}(f)F(\hat{f},\gamma_f)) - \mathbf{Z}(\hat{f},\gamma_f) - \\
 & & \mathbf{G}(\hat{f},\gamma_f)(\bar{C}_y^{\Delta}(f) + \bar{D}_{yu}^{\Delta}(f)F(\hat{f},\gamma_f)) \\
\mathbf{M}_{12}(\hat{f},\gamma_f) & = & -\mathbf{X}B_u^{\Delta}(f)F(\hat{f},\gamma_f) \\
\mathbf{M}_{22}(\hat{f},\gamma_f) & = & \mathbf{Z}(\hat{f},\gamma_f) + \mathbf{G}(\hat{f},\gamma_f)\bar{D}_{yu}^{\Delta}(f)F(\hat{f},\gamma_f) - \mathbf{Y}B_u^{\Delta}(f)F(\hat{f},\gamma_f) \\
\mathbf{N}(\hat{f},\gamma_f) & = & \left[\begin{array}{c} \mathbf{X}B_{\xi}^{\Delta}(f) \\ \mathbf{Y}B_{\xi}^{\Delta}(f) - \mathbf{G}(\hat{f},\gamma_f)D_{y\xi,i}^{\Delta}(f) \end{array}\right], \ \mathbf{P} = \left[\begin{array}{cc} \mathbf{X} & \\ & \mathbf{Y} \end{array}\right], \\
R(\hat{f},\gamma_f) & = & \left[\begin{array}{cc} \bar{C}_z^{\Delta}(f) + \bar{D}_{zu}^{\Delta}(f)F(\hat{f},\gamma_f) & -\bar{D}_{zu}^{\Delta}(f)F(\hat{f},\gamma_f) \end{array}\right], \\
\mathbf{G}(\hat{f},\gamma_f) & = & \mathbf{G_0} + \sum_{i=1}^{n_f} g_i(\hat{f},\gamma_f)\mathbf{G_i}, \ \ \mathbf{Z}(\hat{f},\gamma_f) = \mathbf{Z_0} + \sum_{i=1}^{n_f} g_i(\hat{f},\gamma_f)\mathbf{Z_i}.
\end{array} \tag{4.48}$$

Then the LPV controller (4.37)-(4.38) with

$$\begin{array}{l}
A_c(\hat{f},\gamma_f) = \mathbf{Y}^{-1}\mathbf{Z}(\hat{f},\gamma_f), \\
B_c(\hat{f},\gamma_f) = \mathbf{Y}^{-1}\mathbf{G}(\hat{f},\gamma_f),
\end{array}$$

is such that

$$\sup_{\substack{\Delta \in \mathbf{\Delta},\ \hat{\Delta} \in \hat{\mathbf{\Delta}} \\ \hat{f} \in \Omega_f,\ \gamma_f \in \Omega_\gamma}} \|T_{out.fb.}^{\Delta}(f)\|_\infty \le 1. \tag{4.49}$$

Proof: The proof follows the same lines as the proof of Theorem 3.3 on page 83 for the discrete-time $\mathcal{H}_\infty$ case. □

Similarly to the state-feedback case, the standard LMI solvers are not directly applicable to this problem due to the nonlinear dependence of (4.47) on the parameters $\hat{f}$ and γ_f. For that reason both (4.44) and (4.47) will be solved by using the probabilistic method of Chapter 2. This is explained in the next subsection.

4.3.4 The Probabilistic Approach to the LPV Design

Next we propose an approach to the solution of the feasibility problem (4.44) based on the probabilistic framework developed in Chapter 2. The algorithm is iterative, and at each iteration two basic steps are performed. In the first step of the i-th iteration the proposed probabilistic approach generates randomly

$$\{\Delta^{(i)}, \hat{\Delta}^{(i)}, \hat{f}^{(i)}, \gamma_f^{(i)}\} \in \mathbf{\Delta} \times \hat{\mathbf{\Delta}} \times \Omega_f \times \Omega_\gamma,$$

where $\Delta^{(i)}$ is generated according to the specified probability probability distribution $f_{\mathbf{\Delta}}(\Delta)$, while $\hat{\Delta}^{(i)}$, $\hat{f}^{(i)}$, and $\gamma_f^{(i)}$ – according to a uniform distribution. This generated parameters are used at the second step of the algorithm where an ellipsoid is computed that (a) has a smaller volume than the ellipsoid from the previous iteration, and (b) contains the solution set (i.e. the set of all feasible points to the feasibility problem). In this way a sequence of ellipsoids with

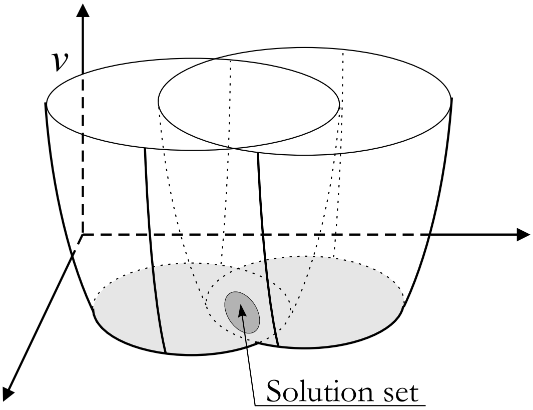


Figure 4.6: The function $v(\cdot)$ is convex in the variables $\mathbf{d}$. In the simple visualization here the function is plotted for two different values of the parameters, say $\{\Delta^{(i)}, \hat{\Delta}^{(i)}, \hat{f}^{(i)}, \gamma_f^{(i)}\}$ and $\{\Delta^{(i+1)}, \hat{\Delta}^{(i)}, \hat{f}^{(i+1)}, \gamma_f^{(i+1)}\}$. The solution set is exactly the intersection of all sub-optimal sets for $v(.)$ computed for all possible values of the parameters.

decreasing volumes is iteratively generated all containing the solution set. The center of the ellipsoid then represents the current update on the variables in the problem. The convergence of this algorithm in a finite number of iterations with probability one is established under the assumptions that the solution set has a non-empty interior and that the algorithm is initialized with an initial ellipsoid that contains the solution set. A method for finding an initial ellipsoid will also be proposed.

Before one can apply the algorithm from Chapter 2 one needs to transform the variables (e.g. all free entries in the matrices $(\mathbf{P}, \mathbf{L_i})$ in (4.44), or $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z_i}, \boldsymbol{G_i})$ in (4.47)) into one vector of unknowns $\mathbf{d} \in \mathbf{R}^{N_d}$. The considered feasibility problem can then be rewritten in the form

$$U(\mathbf{d}, \Delta, \hat{\Delta}, \hat{f}, \gamma_f) = U_0(\Delta, \hat{\Delta}, \hat{f}, \gamma_f) + \sum_{i=1}^{N_d} U_i(\Delta, \hat{f}, \gamma_f)\mathbf{d_i} > 0. \tag{4.50}$$

We proceed by defining the following function

$$v(\mathbf{d}, \Delta, \hat{\Delta}, \hat{f}, \gamma_f) \doteq ||\mathbf{\Pi}^-[U(\mathbf{d}, \Delta, \hat{\Delta}, \hat{f}, \gamma_f)]||_F^2,$$

where $\mathbf{\Pi}^-[\cdot]$ denotes the projection onto the cone of symmetric negative-definite matrices, defined in (2.4) on page 36. Therefore the so-defined function $v(\cdot)$ is such that

$$\left\{\mathbf{d} : \ v(\mathbf{d}, \Delta, \hat{\Delta}, \hat{f}, \gamma_f) = 0, \ \forall\{\Delta, \hat{\Delta}, \hat{f}, \gamma_f\} \in \mathbf{\Delta} \times \hat{\mathbf{\Delta}} \times \Omega_f \times \Omega_\gamma\right\}$$
$$\Updownarrow$$
$$\mathcal{S} = \left\{\mathbf{d} : \ U(\mathbf{d}, \Delta, \hat{\Delta}, \hat{f}, \gamma_f) > 0, \ \forall\{\Delta, \hat{\Delta}, \hat{f}, \gamma_f\} \in \mathbf{\Delta} \times \hat{\mathbf{\Delta}} \times \Omega_f \times \Omega_\gamma\right\}$$

	state-feedback	**output-feedback**
sensor and actuator faults	P-LPV (less conservative)	D-LPV (P-LPV is too slow)
sensor, actuator and component faults	P-LPV (D-LPV is n.a.)	P-LPV (D-LPV is n.a.)

Table 4.1: Comparison between the deterministic LPV (D-LPV) method and the probabilistic LPV (P-LPV) method. The preferred methods for each of the four different classes of problems are given.

where $\mathcal{S}$ is the *solution set*. Thus, similarly to what was done in Chapter 2, we have transformed the original feasibility problem (4.50), or equivalently (4.44), into an optimization problem (see Figure 4.6), so that the ellipsoid algorithm as presented in Algorithm 2.4 on page 59 can be used for solving it. The algorithm for initial ellipsoid computation of Chapter 2, developed in Section 2.4 on page 49 is also directly applicable to the problem in this section. These algorithms will be omitted here to avoid unnecessary repetition.

4.4 Illustrative Examples

This section presents some examples that illustrate the capabilities of the two methods presented in this chapter, i.e. the deterministic method from Section 4.2 and the probabilistic one from Section 4.3. Before proceeding with the examples, however, we want to point out again that the deterministic method deals only with sensor and actuator faults, while the probabilistic method can deal with component faults. Moreover, the deterministic method usually results in very conservative controller due to the fact that the diagonal structure of the FDD uncertainty is neglected by considering unstructured uncertainty in Lemma 4.1 and Theorem 4.1. For that reason the probabilistic method performs usually better even though the deterministic method is based on a set of LPV controllers designed for specific fault scenarios. A strong disadvantage of the probabilistic method is that the computational speed strongly increases with the number of variables in the optimization as a result of the significant increase of the volume of the initial ellipsoid (see equation (2.28) on page 48). For that reason, the deterministic method would be the preferred method only in the output-feedback case for systems with sensor and actuator fault (see Table 4.1). In the state-feedback case the number of optimization variables is significantly smaller so that the computational burden is less an issue; the less conservative probabilistic method is, hence, a better candidate for such problems.

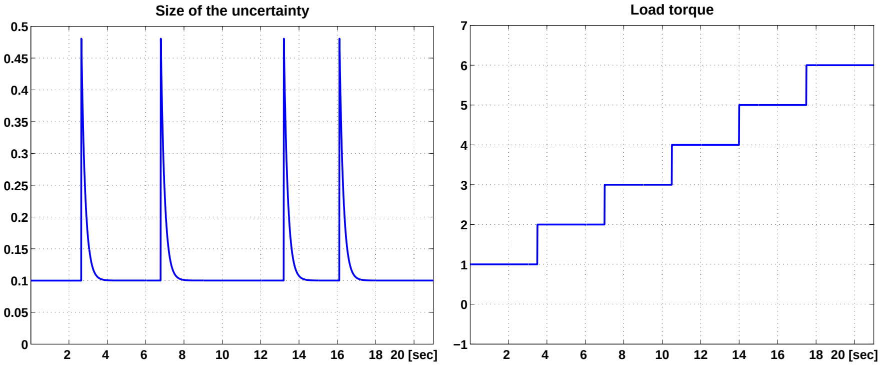


Figure 4.7: Uncertainty size and load torque used in the simulation.

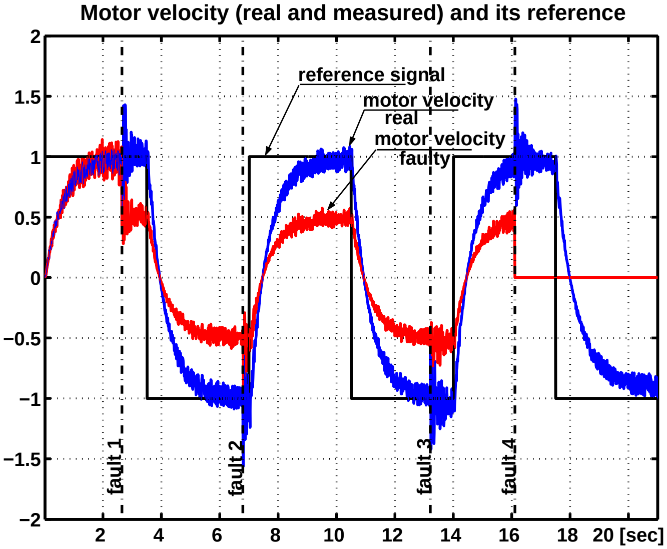


Figure 4.8: The controlled output (real and measured) and the reference signal.

4.4.1 Deterministic Approach to LPV-based FTC

In this section the proposed approach is demonstrated on the diesel engine actuator benchmark model from Chapter 2. The model represents the behavior of a brushless DC motor, which is the actuator part of a real-life speed governor for large diesel engines. Inputs are the motor velocity reference n_{ref} and the load torque (disturbance) Q_l, while the measured outputs are the motor velocity n_m and the gear output position s_o. As controlled signal considered here is the motor velocity.

Note that the approach from this section is not capable of dealing with model uncertainties, and therefore a comparison with the results from Chapter 2 is not possible. The approach presented in the next section, however, can deal with both FDD and model uncertainties and will therefore be compared to the passive method of Chapter 2.

In the simulation made here no model uncertainty is considered in the model (2.37) on page 59, i.e. $\Delta = 0$ so that $\delta = \delta^{nom}$.

The simulated fault scenario is as follows:

- at $t = 2.65$ a partial sensor fault $\sigma_s^1 = 0.5$ in n_m occurs,
- at $t = 6.79$ a partial sensor fault $\sigma_s^2 = 0.7$ in s_o occurs,
- at $t = 13.21$ a partial actuator fault $\sigma_a^1 = 0.4$ in n_{ref} occurs,
- at $t = 16.11$ a total fault $\sigma_s^1 = 0$ in n_m occurs.

All faults remain active until the end of the simulation, so that after the occurrence of the fourth fault at time $t = 16.11$ all sensors and actuators are faulty.

The goal is that the motor velocity follows a reference signal. Note that all sensors and actuators are faulty after $t = 16.11$. The uncertainty in the fault estimates (see Figure 4.7 (left)) increases abruptly to its maximum value of 0.5 after each fault occurrence, and then gradually reduces to its minimum of 0.1. In addition, a step-wise load torque (see Figure 4.7 (right)) is introduced throughout the simulation. A set of controllers has been designed: a nominal parameter dependent controller for the fault-free system, and a controller for the system with a total fault in the measurement of the motor velocity. A total fault in the other measurement (i.e. in s_o) is not considered as it results in an undetectable system. Figure 4.8 depicts the measured value of the controlled output n_m, its real value, and the reference trajectory. It can be observed how after each fault the performance degrades, and then gradually improves as the uncertainties in the fault estimates become smaller. Note also that the true velocity follows the reference trajectory even after the complete loss of its measurement.

4.4.2 Probabilistic Approach to LPV-based FTC

Example 1: Nominal LPV Control

The purpose of this first example is to illustrate the design of an LPV controller of the form (4.42) on page 114, where the functions $g_i(\cdot,\cdot)$ may have a very general, nonlinear structure. We note here, that in the standard methods to LPV controller design there are usually restriction on the structure (e.g. affine in the scheduling parameters) of the LPV controller. No such restrictions are imposed here.

Consider the problem of Section 2.6.2 on page 62, where a robust passive FTC was designed for the diesel benchmark model (2.37) on page 59. We consider the same example here with the parameters of the model defined in Tables 2.1 and 2.4.

We assume in this example that estimate of the vector $\delta = [\eta,\ I_{tot},\ f_{tot},\ K_q]^T$ will be available on-line on the basis of which the LPV controller can be scheduled. The following structure (in terms of the functions $g_i(\cdot,\cdot)$ in (4.42) on page 114) for the LPV controller was selected

$$F(\delta) = F_0 + \frac{\delta_1\delta_4}{\delta_2}F_1 + \frac{1}{\delta_2}F_2 + \frac{\delta_3}{\delta_2}F_3. \tag{4.51}$$

Note that in this example:

- The fault vector $\hat{f}$ is represented by the parameter vector δ in the example of Section 2.6.2;
- There is no FDD uncertainty, i.e. the size γ_f of the FDD uncertainty is equal to zero. Later on, in Chapter 6 the same method will be applied to a real-life brushless DC motor experimental setup and the LPV controller where also FDD uncertainty with time-varying size will be considered;
- The parameter α, representing multiplicative actuator faults in the benchmark model (2.37), is here taken as uncertainty, i.e. it is assumed that the FDD scheme does not provide on-line estimates of α. For that reason the LPV controller needs to be robust with respect to variations in α in the interval specified in Table 2.4 on page 62. In the next example the actuator fault α will be the scheduling parameter, and δ will be the model uncertainty. Of course, one may wish to take both α and δ as scheduling parameters that would improve the attainable performance even more. This would, however, introduce more decision variables into the optimization problem that would result in increased computational burden.

The probabilistic Algorithm 2.4 on page 59 is executed on this problem starting with an initial upper bound γ_{UB} equal to the optimal value for γ that was achieved by the robust controller in from Chapter 2. This performance is here also achievable since by taking $F_1 = F_2 = F_3 = 0$ in (4.51) one transforms the LPV design problem to the robust controller design problem considered in Chapter 2. Algorithm 2.4 was initiated with the following parameters: $Tol = 0.1$, $\varepsilon = 0$, $L = 100$, $\gamma_{max} = 100$, and $\gamma_{UB} = 0.83125$ (instead of infinity as suggested in the initialization step of Algorithm 2.4). Table 4.2 summarizes the intermediate results of the 10 performed iterations before convergence. The table provides at each iteration the tested value for γ as well as its lower and upper bounds, γ_{LB} and γ_{UB}, respectively. In addition to that the volume of the optimal ellipsoid at each iteration is reported.

The optimal state-feedback LPV controller (4.51) computed by the algorithm has matrices

$$\begin{aligned}
F_0 &= \begin{bmatrix} -5.311172 \times 10^{2} & -7.29898 \times 10^{1} & -2.0506 \times 10^{-5} \end{bmatrix} \\
F_1 &= \begin{bmatrix} 0.31225 & 4.4286 \times 10^{-2} & 3.0043 \times 10^{-7} \end{bmatrix} \\
F_2 &= \begin{bmatrix} -0.2124 & -3.0124 \times 10^{-2} & -2.0436 \times 10^{-7} \end{bmatrix} \\
F_3 &= \begin{bmatrix} -2.0867 \times 10^{-4} & -2.9595 \times 10^{-5} & -2.0077 \times 10^{-10} \end{bmatrix}
\end{aligned}$$

The optimal value for γ achieved by this LPV controller is $\gamma_{opt} = 0.03198$. In comparison, the robust controller designed in Section 2.6.2 of Chapter 2 achieved $\gamma = 0.83125$. The performance achieved by the LPV controller is, therefore, improved by a factor of more than 20.

iter.	γ_k	γ_{LB}	γ_{UB}	**vol**$(\mathcal{E}_k^*)$	**status**
1.	$\mathbf{0.08312}$	0	0.8312	5.520×10^5	feas
2.	0.008312	0	$\mathbf{0.08312}$	2.629×10^1	infeas
3.	$\mathbf{0.04572}$	0.00831	0.08312	1.621×10^4	feas
4.	0.02702	0.00831	$\mathbf{0.04572}$	3.123×10^2	infeas
5.	$\mathbf{0.03637}$	0.02702	0.04572	4.461×10^3	feas
6.	0.03169	0.02702	$\mathbf{0.03637}$	2.705×10^3	infeas
7.	$\mathbf{0.03403}$	0.03169	0.03637	1.228×10^3	feas
8.	$\mathbf{0.03286}$	0.03169	0.03403	7.253×10^2	feas
9.	$\mathbf{0.03228}$	0.03169	0.03286	6.436×10^1	feas
10.	$\mathbf{0.03198}$	0.03169	0.03228	4.336×10^1	feas

Table 4.2: Summary of the iterations performed by Algorithm 2.4 for the design of a nominal LPV controller. The optimal feasible value for γ at each iteration is written in boldface.

Example 2: Robust Active LPV-based FTC

In this second example we consider again the same diesel benchmark example of Section 2.6.2 of Chapter 2. The parameters of the model are again defined in Tables 2.1 and 2.4.

Opposite to the example of the previous subsection, we assume here that the vector $\delta = [\eta,\ I_{tot},\ f_{tot},\ K_q]^T$ represents the model uncertainty. The FDD scheme will provide no estimates of δ, but of the multiplicative actuator fault represented by the scalar α in the benchmark model. The LPV controller will, therefore, have to be scheduled by the the actuator fault signal α and needs at the same time to be robust with respect to parametric uncertainties, as represented by equations (2.38), (2.39) and (2.40) on page 60.

There is only one function $g_1(\cdot,\cdot)$ (see (4.42) on page 114) that defines the structure of the LPV controller in this example which is selected to be equal to α, so that

$$F(\alpha) = F_0 + \alpha F_1. \tag{4.52}$$

Again, the probabilistic Algorithm 2.4 on page 59 is executed on this problem beginning with $\gamma_{UB} = 0.83125$, i.e. an upper bound equal to the optimal value for γ that was achieved by the robust controller from Chapter 2. Algorithm 2.4 on page 59 was initiated with the same parameters as in the previous example: $Tol = 0.1$, $\varepsilon = 0$, $L = 100$, $\gamma_{max} = 100$, and $\gamma_{UB} = 0.83125$. Table 4.3 summarizes the results of the optimization. In this example the optimal state-feedback gain is $F(\alpha) = F_0 + \alpha F_1$ with

$$\begin{aligned} F_0 &= \left[\begin{array}{ccc} -2.0692783 \times 10^3 & -1.693148 \times 10^2 & -7.4361 \times 10^{-5} \end{array}\right], \\ F_1 &= \left[\begin{array}{ccc} 7.261707 \times 10^2 & 6.02549 \times 10^1 & 1.6434 \times 10^{-4} \end{array}\right]. \end{aligned}$$

The optimal achieved value for the cost function γ_{opt} is 0.01138. As expected, there is again improvement in the achievable performance by the LPV controller as compared to the passive robust FTC designed in 2.6.2 on page 62. The achieved

iter.	γ_k	γ_{LB}	γ_{UB}	**vol**$(\mathcal{E}_k^*)$	**status**
1.	**0.08312**	0	0.8312	1.493×10^5	feas
2.	0.00831	0	**0.08312**	8.789×10^0	infeas
3.	**0.04572**	0.008312	0.08312	9.052×10^4	feas
4.	**0.02702**	0.008312	0.04572	4.670×10^3	feas
5.	**0.01766**	0.008312	0.02702	3.407×10^2	feas
6.	**0.01299**	0.008312	0.01766	6.459×10^0	feas
7.	0.01065	0.008312	**0.01299**	1.484×10^{-1}	infeas
8.	**0.01182**	0.01065	0.01299	8.018×10^{-2}	feas
9.	0.01123	0.01065	**0.01182**	2.005×10^{-2}	infeas
10.	**0.01153**	0.01123	0.01182	3.062×10^{-2}	feas
11.	**0.01138**	0.01123	0.01153	2.728×10^{-2}	feas
12.	0.01131	0.01123	**0.01138**	5.849×10^{-3}	infeas

Table 4.3: Summary of the iterations performed by Algorithm 2.4 for the design of robust active FTC. The optimal feasible value for γ at each iteration is written in boldface.

performance by the LPV controller from this example is about a factor of 3 better than the optimal one from the previous example, where the model parameters δ (instead of the actuator fault signal α) were used as scheduling parameters, and about a factor of 80 better than the performance achieved by the passive FTC from Chapter 2.

4.5 Conclusion

This chapter presented a deterministic and a probabilistic approach to robust active FTC design.

The deterministic approach to robust active FTC, presented in Section 4.2, is applicable to systems in which multiplicative sensor and actuator faults can occur, i.e. faults represented by scalings on the inputs and the outputs of the system. The approach assumes that the real faults lie within given time-varying intervals around their estimates. It consists of the off-line design of a set of parameter-dependent controllers, where the sizes of the uncertainty intervals are the scheduling parameters. The reconfigured controller is taken as a scaled version of one of the predesigned controllers. The approach was demonstrated in a case study with the diesel engine actuator benchmark model.

In Section 4.3 the idea exploited in Section 4.2 is further improved by making use of the probabilistic framework of Chapter 2 that allows us to design the LPV controller to depend not only on the sizes of the FDD uncertainties, but also directly on the fault estimates. Furthermore, the probabilistic setting made it possible to consider a much wider class of faults, namely faults that change the state-space matrices of the system in a very general way. In addition to that structured parametric uncertainty is considered by this method. The method was illustrated by two examples, were a comparison was made with the results from Chapter 2 where a passive FTC was designed. The method from this chapter clearly outperformed the passive method. This method will later on be tested on a real-life experimental setup in Chapter 6, where an LPV controller will be designed to be scheduled by not only the fault estimates (as in the examples above), but also by the sizes of the FDD uncertainties.

5

Robust Output-Feedback MPC Method for Active FTC

Model predictive control is a very suitable control methodology for active FTC due to its self-reconfigurability property. In this chapter we propose a new technique to robust active FTC based on robust output-feedback MPC for systems with parametric uncertainty. The main advantage of this method is that one no longer needs to solve NP-hard optimization problems in the output-feedback case like those considered in Chapter 3. Instead, the method of this chapter derives the control action at each time instant by solving a robust linear least squares problem using the probabilistic method of Chapter 2. The approach consists in a combination of a Kalman filter and a finite-horizon MPC into one min-max (worst-case) optimization problem. This problem is *affine* in the variables, being the system state and the control action in a future interval of time. The solution to this optimization problem achieves both state estimation and reference trajectory tracking. Given the state covariance matrix, the optimization problem is solved in the probabilistic framework of Chapter 2. Additionally, two methods for finding a covariance matrix that is compatible with all values of the uncertainty were presented. The first one aims at minimization of the trace of the covariance matrix and is computationally more involving, while the second method is more conservative but much faster. The complete MPC approach is tested on a case study with a model of one joint of a real-life space robotic manipulator.

5.1 Introduction

Model predictive control (MPC) is an industrially relevant control strategy that has been receiving a lot of attention lately. Due to the underlying optimization that needs to be executed at each time instant, it is an attractive method mainly for slower processes such as those encountered in the chemical industry (Kothare et al. 1996). This optimization is based on matching (in the vector 2-norm sense) a prediction of the system output to some desired reference trajectory. The latter is assumed to be known in advance. In addition, MPC features the property that it can handle constraints on the inputs, outputs and states of the system in an explicit way by incorporating them into the optimization problem (see e.g. (Cuzzola and Ferrante 2001; Bemporad et al. 2002; Bemporad and Garulli 2000)). For an overview of the approaches based on MPC the reader is referred to (Clarke and Mohtadi 1989; Bemporad and Morari 1999; Bemporad et al. 2002; Scokaert and Mayne 1998; Nikolaou 2001; Bemporad and Garulli 2000) and the references therein.

Most robust state-space approaches to MPC are derived under the assumption, which is rather restrictive in practice, that the system state is exactly known (measured) (Badgwell 1997; Bemporad et al. 2002; Kothare et al. 1996; Cuzzola and Ferrante 2001; Casavola et al. 2000). Under this assumption robust MPC techniques based on LMIs have also been proposed in the literature for systems with polytopic uncertainty (Kothare et al. 1996; Casavola et al. 2000; Cuzzola and Ferrante 2001). In contrast to the uncertainty-free case a state observer cannot be used to supply the missing state information in the MPC when model uncertainty is considered. This problem is not trivial due to the fact that the separation principle is no longer valid. It is a well-known fact in the field of robust control theory that the problem of designing an output-feedback controller for systems with structured uncertainty is an NP-hard Bilinear Matrix Inequality (BMI) optimization problem (Toker and Özbay 1995). On the other hand, using an unstructured uncertainty representation usually results in a convex problem, but this is often achieved only at the expense of unnecessary conservatism, that for some applications might be overly disadvantageous.

In this chapter we present an approach to the design of finite-horizon robust output-feedback MPC. The approach integrates the Kalman filter recursions in the finite-horizon MPC optimization. Including uncertainty results in a min-max (worst-case) optimization problem. This problem is *affine* in the variables, being the current system state and the control action in a future interval of time. Solving this optimization problem achieves both state estimation and reference trajectory tracking. The main advantage of this problem formulation is that one does no longer have to solve difficult BMI optimization problems to find a controller. Another advantage is that constraints in the control action can easily be incorporated. The drawback is that the robust stability of the resulting closed-loop system cannot be theoretically guaranteed.

In order to solve the underlying min-max optimization problem the probabilistic framework developed in Chapter 2 is used. The advantages of using this probabilistic approach can be outlined as follows. On the first place, at each iteration the problem remains of a reasonably small size since only one parametrization of the uncertain system is used (and not a large amount of vertex systems as is often the case with LMI approaches applied to systems with polytopic uncertainty). Secondly, a rather general class of uncertainties can be considered – the only assumption that is imposed is that the system state-space matrices remain bounded over all possible values of the uncertainties. And third, as will become clear later, the approach allows us to significantly reduce the number of variables in the underlying optimization problem. Similarly to the other approaches to MPC, a drawback is the computational complexity.

In the Kalman filter recursion at each iteration the covariance matrix of the state vector is required. Whenever the covariance matrix is given, the resulting problem is a robust LMI problem that may depend on the uncertainty in a very general, nonlinear way. This problem can also be solved by making use of the ellipsoid algorithm for probabilistic design from Chapter 2. However, if the covariance matrix of the state is unknown, it needs to be computed at each time instant on-line so that it is compatible with all possible values of the uncertainty. It is shown in the chapter that the minimum-trace covariance matrix can be found by again using the probabilistic ellipsoid algorithm. For this purpose a separate optimization for the covariance matrix needs to be performed on-line. This optimization, however, additionally increases the computational burden that might sometimes be undesirable. Whenever it is not required that the covariance matrix at each iteration has the minimal trace, a second (faster) algorithm is also proposed for the computation of a covariance matrix so that it is compatible with all values of the uncertainty.

The chapter is organized as follows. In Section 5.2 the notation is explained and the problem is formulated. In Section 5.3 it is rewritten as a least squares optimization problem and it is discussed how it can be solved by making use of the probabilistic framework given the covariance matrix of the state vector. The computation of the covariance matrix is next considered in Section 5.4. The method is tested in a case study with a model of one joint of a real-life space robotic manipulator in Section 5.5. Finally, Section 5.6 concludes the chapter.

5.2 Notation and Problem Formulation

The notation used in this chapter is as follows. For a vector x in an n-th dimensional Euclinean space, $x \in \mathbb{R}^n$, and an n-by-n symmetric matrix $W \in \mathbb{R}^{n\times n}$, the weighted norm is denoted as $\|x\|_W = x^T W x$. The symbol $\otimes$ denotes the Kronecker product, and symbol $\star$ will denote entries in LMIs that follow from symmetry. Further, for matrices A and B of appropriate dimension, the inner product is defined as $\langle A, B\rangle = \mathbf{trace}(AB)$. The notation $x \sim \mathcal{N}(\bar{x}, S)$ will be used to make clear that x is a random Gaussian process with mean $\bar{x}$ and covariance S.

For a symmetric matrix A, the projection onto the cone of symmetric positive-definite matrices is defined as

$$\mathbf{\Pi}^{+}[A] \doteq \arg\min_{\boldsymbol{S}\geq 0} \|A - \boldsymbol{S}\|_F. \tag{5.1}$$

Similarly, the projection onto the cone of symmetric negative-definite matrices is defined as

$$\mathbf{\Pi}^{-}[A] \doteq \arg\min_{\boldsymbol{S}\leq 0} \|A - \boldsymbol{S}\|_F. \tag{5.2}$$

These projections have some useful properties summarized in Chapter 2, Lemma 2.1 on page 36.

Finally, for a random variable x_k, $\hat{x}_{k+i|k}$ will denote the prediction of x_{k+i} made at time instant k (i.e. by using the input-output measurements up to time instant k).

In this chapter we consider the following discrete-time linear system

$$\mathcal{S}: \left\{ \begin{array}{rcl} x_{k+1} & = & A^{\Delta}x_k + B_u^{\Delta}u_k + Q^{\Delta}\xi_k^x \\ z_k & = & C_z^{\Delta}x_k + D_{zu}^{\Delta}u_k + R_z^{\Delta}\xi_k^z \\ y_k & = & C_y^{\Delta}x_k + D_{yu}^{\Delta}u_k + R_y^{\Delta}\xi_k^y, \end{array} \right. \tag{5.3}$$

where $x \in \mathbb{R}^n$ is the state of the system which is here not assumed to be measured, $u \in \mathbb{R}^m$ is the control action, $y \in \mathbb{R}^p$ is the measured output, $z \in \mathbb{R}^{n_z}$ represents the controlled output of the system, and ξ^x, ξ^z and ξ^y are white Gaussian noises (i.e. random zero-mean processes with covariance matrices equal to the identity matrix). The matrices Q^{Δ}, R_z^{Δ} and R_y^{Δ} are symmetric positive-definite matrices. The model uncertainty Δ is assumed to belong to some (possibly structured) bounded uncertainty set $\boldsymbol{\Delta}$ and to have a known probability density function $f_{\boldsymbol{\Delta}}$ inside this uncertainty set. There are no further restrictions on the way in which the uncertainty Δ enters the state-space matrices of the system, as long as these remain bounded over all $\Delta \in \boldsymbol{\Delta}$ (see (2.6) on page 37). It is further assumed that samples of Δ can be generated with the selected probability density function $f_{\boldsymbol{\Delta}}$. The reader is referred to (Calafiore et al. 2000, 1999) for more details on the available algorithms for generation of random uncertainty samples.

This chapter is concerned with the problem of finding a control action u_k that makes the controlled output z_k of the uncertain system (5.3) track a reference trajectory signal r_k, which is assumed to be known in advance.

Remark 5.1 *We note here that the algorithm in this chapter can be used for both passive FTC (by means of representing the possible faults as uncertainties as in Chapters 2 and 3) or as active FTC (by changing the internal model (5.3) after the occurrence of a fault in the system). Due to that fact that the predictive controller at each time instant depends directly on the state-space matrices of the system we will not explicitly write the dependence on the fault signal f. Instead, model (5.3) is assumed in the sequel to represent either the nominal or some faulty mode of operation of the system. The model is allowed to change during the operation of the system.*

Remark 5.2 *We also note that FDD uncertainty can also be treated by this method. To this end one could follow the same idea that was exploited in Chapter 4, namely to consider the uncertain faulty system (4.34) on page 111 with some uncertainty representation of the fault signal f as in equation (4.35). This would result in an uncertain system where the uncertainty is a combination of the original model uncertainty and the FDD uncertainty represented by equation (4.35).*

5.3 Integrated State Prediction and Trajectory Tracking Control

In this section a robust output-feedback MPC control algorithm will be proposed by first writing both Kalman filtering problem and the finite-horizon MPC as least squares problems, and then combining them into one min-max optimization problem. The probabilistic approach from Chapter 2 will subsequently be used for solving it.

5.3.1 The Kalman filter over a finite-time horizon

Suppose, to begin with, that no uncertainty is present in the system that the input u_k to the system (5.3) is given, and consider the state-estimation problem from available input-output measurements. In (Verhaegen and Verdult 2003), the Kalman filtering problem over a finite time horizon is formulated as a least-squares problem. To summarize this, let the system state x_k be first written in the general covariance representation as $x_k \sim \mathcal{N}(\hat{x}_{k|k-1}, S_{k|k-1})$, i.e. a random gaussian process with mean $\hat{x}_{k|k-1}$ and covariance $P_{k|k-1} = S_{k|k-1}S_{k|k-1}^T > 0$. It is assumed that $\hat{x}_{0|-1}$ and $P_{0|-1}$ are given. Combining this representation of the state x_k with the dynamic equations of the system (5.3) results in

$$\begin{aligned} \hat{x}_{k|k-1} &= x_k + S_{k|k-1}n_k \\ y_k &= C_y x_k + D_{yu}u_k + R_y \xi_k^y \\ -B_u u_k &= A x_k - x_{k+1} + Q\xi_k^x \end{aligned}$$

where n_k is a zero-mean stochastic variable with covariance matrix equal to the identity matrix. Writing these equations over a finite-time window $[k-N,\ k]$ leads to the set of equations

$$\tilde{Y} = \tilde{F}\tilde{\boldsymbol{X}} + \tilde{L}\tilde{\Xi}, \tag{5.4}$$

where

$$\tilde{Y} = \begin{bmatrix} \hat{x}_{k-N|k-N-1} \\ y_{k-N} - D_{yu}u_{k-N} \\ -B_u u_{k-N} \\ y_{k-N+1} - D_{yu}u_{k-N+1} \\ -B_u u_{k-N+1} \\ \vdots \\ y_k - D_{yu}u_k \\ -B_u u_k \end{bmatrix}, \ \tilde{F} = \begin{bmatrix} I_n & 0 & 0 & \dots & 0 & 0 \\ C_y & 0 & 0 & \dots & 0 & 0 \\ A & -I & 0 & \dots & 0 & 0 \\ 0 & C_y & 0 & \dots & 0 & 0 \\ 0 & A & -I & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & C_y & 0 \\ 0 & 0 & 0 & \dots & A & -I \end{bmatrix}$$

$$\tilde{\boldsymbol{X}} = \begin{bmatrix} x_{k-N} \\ x_{k-N+1} \\ x_{k-N+2} \\ \vdots \\ x_{k+1} \end{bmatrix}, \ \tilde{\Xi} = \begin{bmatrix} n_{k-N} \\ \xi_{k-N}^y \\ \xi_{k-N}^x \\ \vdots \\ \xi_k^y \\ \xi_k^x \end{bmatrix}, \ \tilde{L} = \begin{bmatrix} S_{k-N|k-N-1} & \\ & I_{N+1} \otimes \begin{bmatrix} R_y & \\ & Q \end{bmatrix} \end{bmatrix}.$$

Above, the vector $\tilde{Y}$ contains only signals available at time instant k. The Kalman filter over a finite time horizon is then formulated as the following least-squares problem (Verhaegen and Verdult 2003)

$$\begin{bmatrix} \hat{x}_{k-N|k} \\ \vdots \\ \hat{x}_{k+1|k} \end{bmatrix} = \arg\min_{\tilde{\boldsymbol{X}}} \|\tilde{L}^{-1}\tilde{Y} - \tilde{L}^{-1}\tilde{F}\tilde{\boldsymbol{X}}\|_2^2. \tag{5.5}$$

For more details the interested reader is referred to (Verhaegen and Verdult 2003), where the authors propose a recursive-time solution to this problem by exploiting its structure. This approach will not be pursued here since it requires the exact knowledge of the system matrices. Instead, it will be combined with MPC into one robust least-squares optimization problem, which will subsequently be addressed in the probabilistic framework.

5.3.2 Combination of the Kalman filter and MPC

Let us now go back to the original problem of trajectory tracking for the uncertain system (5.3). In this section we combine the Kalman filter with a finite-horizon MPC. For our purposes it will prove to be sufficient to consider a filtering interval consisting of only one time instant k, i.e. $N = 0$ in (5.4). This is done here for the sake of brevity.

Provided that the reference trajectory signal r_k is given (at least N_2 samples ahead in the future), in the conventional MPC the control action u_k is computed as the solution to the following optimization problem

$$\min_{u_{k+i},\, i=1,\ldots,N_u} J_k = \sum_{i=N_1}^{N_2} \|r_{k+i} - \hat{z}_{k+i|k}\|_W^2 + \sum_{i=1}^{N_u} \|u_{k+i}\|_{W_U}^2,$$

where N_1, and N_2 define the prediction horizon, and $N_u \leq N_2$ - the control horizon, and where $W = W^T > 0$ and $W_U = W_U^T \geq 0$ are weighting matrices. The standard assumption is imposed that the control action remains constant after the control horizon (Clarke and Mohtadi 1989).

Assumption 5.1 $u_{k+i} = u_{k+N_u}$ *for* $i \geq N_u$.

Remark 5.3 *It needs to be pointed out here that if the system matrices depend on some fault signal f as in (4.34) on page 111 then the additional assumption will have to be imposed either that the fault signal f remains constant during the prediction horizon*[1] *or that it varies but its variations are known in the interval* $[k,\ k+k_2]$. *The reason for that assumption is that the MPC is based on a prediction of the output signal in the future. Below we assume that f remains constant in the future as this significantly simplifies the expressions that will follow.*

---

[1] A similar assumption to that will be imposed later on in Chapter 7 where the mode probabilities of the Interacting Multiple-Model estimator are assumed constant over the control horizon (see Assumption 7.2 on page 180).

In the sequel we will suppose, without any loss of generality, that the weighting matrix $W_U = 0$. This is done only to keep the notations as simple as possible; deriving the results for nonzero W_U is straightforward.

Given $\hat{x}_{k+1|k}$ and $\{u_{k+1}, \ldots, u_{k+i}\}$, we can represent the prediction of the controlled output z_{k+i} $(i \geq 2)$ as follows

$$\hat{z}_{k+i} = C_z^{\Delta}(A^{\Delta})^{i-1}\hat{x}_{k+1|k} + C_z^{\Delta}\sum_{j=1}^{i-1}(A^{\Delta})^{i-j-1}B_u^{\Delta}u_{k+j} + D_{zu}^{\Delta}u_{k+i},$$

Since we want the controlled output to track the reference trajectory signal r_k then, under Assumption 5.1, we write

$$\left[\begin{array}{c} \hat{x}_{k|k-1} \\ y_k - D_{yu}^{\Delta}u_k \\ -B_u^{\Delta}u_k \\ \hline r_{k+N_1} \\ r_{k+N_1+1} \\ \vdots \\ r_{k+N_2} \end{array}\right] = \left[\begin{array}{c|c} M_{11}^{\Delta} & 0 \\ \hline M_{21}^{\Delta} & M_{22}^{\Delta} \end{array}\right] \left[\begin{array}{c} x_k \\ x_{k+1} \\ \hline u_{k+1} \\ \vdots \\ u_{k+N_1-1} \\ \vdots \\ u_{k+N_u} \end{array}\right] + L^{\Delta} \left[\begin{array}{c} n_k \\ \xi_k^y \\ \xi_k^x \\ \hline \tilde{\xi}_{k+N_1} \\ \vdots \\ \tilde{\xi}_{k+N_2} \end{array}\right], \tag{5.6}$$

where

$$M_{11}^{\Delta} = \begin{bmatrix} I & 0 \\ C_y^{\Delta} & 0 \\ A^{\Delta} & -I \end{bmatrix}, \; M_{21}^{\Delta} = \begin{bmatrix} 0 & C_z^{\Delta}(A^{\Delta})^{N_1-1} \\ 0 & C_z^{\Delta}(A^{\Delta})^{N_1} \\ \vdots & \vdots \\ 0 & C_z^{\Delta}(A^{\Delta})^{N_2-1} \end{bmatrix},$$

$$L^{\Delta} = \left[\begin{array}{ccc|c} S_{k|k-1} & & & \\ & R_y^{\Delta} & & 0 \\ & & Q^{\Delta} & \\ \hline & 0 & & \tilde{I}_{N_2-N_1+1} \otimes W^{-1} \end{array}\right] \tag{5.7}$$

$$M_{22}^{\Delta} = \begin{bmatrix} \mathbb{C}_{N_1-2}^{\Delta} & \cdots & D_{zu}^{\Delta} & \cdots & 0 \\ \mathbb{C}_{N_1-1}^{\Delta} & \cdots & \mathbb{C}_0^{\Delta} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbb{C}_{N_2-2}^{\Delta} & \cdots & \mathbb{C}_{N_2-N_1}^{\Delta} & \cdots & \sum_{i=N_u+1}^{N_2-1} \mathbb{C}_{N_2-i-1}^{\Delta} + D_{zu}^{\Delta} \end{bmatrix},$$

$$\mathbb{C}_k^{\Delta} = C_z^{\Delta}(A^{\Delta})^k B_u^{\Delta}.$$

The goal is to solve (5.6) in a least-squares sense in order to estimate the state and at the same time to push the predicted output as close as possible to the reference signal r_{k+i}. Note that the state x_{k+1} affects the predicted output $\hat{z}_{k+i}$ during the prediction horizon. The idea is that when the predicted output $\hat{z}_{k+i}$ approximately equals the reference trajectory r_{k+i} over the prediction horizon

$[k+N_1,\ k+N_2]$ we could update the state x_{k+1} not only by looking at the past data (the third row in equation (5.6)) but also by looking in the future, i.e. by accounting for the output that we expect in the future. Note that if one only considers the past data one would then want to take $x_{k+1} = A^{\Delta}x_k + B_u^{\Delta}u_k$ which due to the presence of uncertainty defines here a set for x_{k+1}, which one could try to reduce by looking at the expected output in the future. However, it still needs to be investigated under what conditions (e.g. on the reference signal, problem parameters, etc.) this approach would provide beter results that when x_{k+1} is computed by only accounting for the past data. For instance, if due to the initial state the predicted output is initially very different from the reference signal the algorithm might experience difficulties. Note also, that the weighting matrix W could be used to put more weighting on the control problem and less on the estimation, and vise versa.

The over-determined system of equations (5.6) can be written more compactly as

$$a^{\Delta} = F^{\Delta}\boldsymbol{b} + L^{\Delta}\Xi.$$

We denote the dimensions of the vectors a^{Δ} and $\boldsymbol{b}$ as

$$\begin{aligned} N_a &= \dim a^{\Delta} = 2n + p + (N_2 - N_1 - 1)n_z, \\ N_b &= \dim \boldsymbol{b} = 2n + N_u m, \end{aligned}$$

and we assume that the matrix F^{Δ} is full column rank for every $\Delta \in \boldsymbol{\Delta}$.

Assumption 5.2 *rank* $F^{\Delta} = N_b,\ \forall\Delta \in \boldsymbol{\Delta}$.

When a specific application is considered this can be ensured by appropriate selection of the parameters N_1, N_2, and N_u (provided that the system is left-invertible). Note also that the matrix L^{Δ} is invertible as it contains positive definite matrices on its diagonal.

The problem of combined robust state estimation and MPC is then defined as the following min-max optimization problem

$$\hat{b} = \arg\min_{\boldsymbol{b}} \max_{\Delta\in\boldsymbol{\Delta}} \|(L^{\Delta})^{-1}(a^{\Delta} - F^{\Delta}\boldsymbol{b})\|_2^2. \tag{5.8}$$

Once this optimization problem is solved we take

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{x}_{k+1|k} \\ u_{k+1} \end{bmatrix} = [I_{2n+m}\ 0]\hat{b}.$$

Note that the matrix $S_{k+1|k}$ also needs to be computed in order to be used at the next time instant. Its computation is addressed in Section 5.4.

Remark 5.4 *It should be pointed out that whenever the system matrices are* affinely *dependent on a number of uncertain parameters then this results in a polytopic system. Using the state equation in (5.3), the set of equations (5.6) can be extended* *to*

$$
\left[\begin{array}{c}
\hat{x}_{k|k-1} \\
y_k - D^{\Delta}_{yu} u_k \\
-B^{\Delta}_u u_k \\
\hline
r_{k+1} \\
0 \\
\vdots \\
r_{k+N_2} \\
0
\end{array}\right]
=
\left[\begin{array}{cccccc|ccc}
I & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\
C^{\Delta}_y & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\
A^{\Delta} & -I & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\
\hline
0 & C^{\Delta}_z & 0 & \dots & 0 & 0 & D^{\Delta}_{zu} & \dots & 0 \\
0 & A^{\Delta} & -I & \dots & 0 & 0 & B^{\Delta}_u & \dots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \dots & C^{\Delta}_z & 0 & 0 & \dots & D^{\Delta}_{zu} \\
0 & 0 & 0 & \dots & A^{\Delta} & -I & 0 & \dots & B^{\Delta}_u
\end{array}\right]
\left[\begin{array}{c}
x_k \\
x_{k+1} \\
x_{k+2} \\
\vdots \\
x_{k+N_2} \\
x_{k+N_2+1} \\
\hline
u_{k+1} \\
\vdots \\
u_{k+N_u}
\end{array}\right]
$$

$$
+
\left[\begin{array}{c|c}
\begin{matrix} S_{k|k-1} & & \\ & R^{\Delta}_y & \\ & & Q^{\Delta} \end{matrix} & 0 \\
\hline
0 & I_{N_2} \otimes \begin{bmatrix} W^{-1} & \\ & Q^{\Delta} \end{bmatrix}
\end{array}\right]
\left[\begin{array}{c}
n_k \\
\xi^y_k \\
\xi^x_k \\
\hline
\tilde{\xi}_{k+1} \\
\xi^x_{k+1} \\
\vdots \\
\xi^x_{k+N_2}
\end{array}\right],
$$

so that the resulting optimization problem (5.8) can be represented as a system of LMIs, one for each vertex of the convex polytope. Interior point methods can then be used to solve this system of LMIs (Boyd et al. 1994), which are known to be computationally very fast. However, even for small values of n*,* N_u*, and* N_2 *this approach often leads to an extremely large number of LMIs that go beyond the capabilities of the existing LMI solvers. It is, in fact, an NP-hard problem in the number of vertex systems (Calafiore and Polyak 2001). The probabilistic approach, on the other hand, is iterative and each iteration is performed for only one randomly generated uncertainty sample. This has the advantage that: (1) at each iteration the problem remains of the size of the original system dimension, (2) no assumption (e.g. affine dependence, etc.) is imposed on the way the system matrices depend on the uncertain parameters as long as they remain bounded, and (3) it allows us to significantly reduce the number of variables by considering the least-squares problem in equation (5.6) instead, which has* $N_2 n$ *variables less.*

We will refer to this control algorithm as the integral predictive control (iPC) approach. It should be noted that equation (5.6) is not convex in the uncertainty Δ, so that one cannot make use of LMI approaches here. Instead, this problem can be addressed in the probabilistic framework of Chapter 2. More specifically, problem (5.8) has the form of the robust linear least squares problem considered in Section 2.4.2 on page 54 (see (2.32) on page 54), where it was demonstrated how the probabilistic approach can be used for solving it. Here we will skip the details to avoid unnecessary repetition. Note, however, that constraints are also considered in the robust LLS problem in Section 2.4.2 that allows us here to introduce constraints on the control action in the optimization problem (5.8) in a straightforward manner.

We also note that due to the fact that the matrix $(L^{\Delta})^{-1}F^{\Delta}$ is full column rank (see Assumption 5.2 and the subsequent discussion), the initial ellipsoid computation falls under Case 2 on page 57 so that no upper and lower bounds on the vector of variables need to be imposed here (as, for instance, in Case 3 on page 57).

In the next section we concentrate on the problem of the computation of the covariance matrix $P_{k+1|k} = S_{k+1|k}S_{k+1|k}^T$ that will be required for the optimization at time instant $(k+1)$.

5.4 Computation of the covariance matrix $P_{k+1|k}$

In this subsection we are concerned with finding the minimum-trace state covariance matrix $P_{k+1|k}$ that is compatible with all possible values of the uncertainty. Its "square root" $S_{k+1|k}$ is then to be used in the optimization problem at the next time instant $(k+1)$. To this end we following the same lines as in Verhaegen and Verdult (2003). Assuming that the state estimate at time instant k is represented as

$$\hat{x}_{k|k-1} = x_k + S_{k|k-1}n_k,$$

the we want to obtain a similar expression for time instant $k+1$

$$\hat{x}_{k+1|k} = x_{k+1} + S_{k+1|k}\tilde{n}_k, \tag{5.9}$$

with $\tilde{n}_k$ zero mean and covariance matrix equal to the identity matrix.

Pre-multiplying the first three lines in Equation (5.6) by the non-singular matrix

$$T_l = \begin{bmatrix} C_y^\Delta & -I & 0 \\ A^\Delta & 0 & -I \\ I & 0 & 0 \end{bmatrix},$$

results in the equation

$$\begin{aligned} &\begin{bmatrix} C_y^\Delta \hat{x}_{k|k-1} + D_{yu}^\Delta u_k - y_k \\ A^\Delta \hat{x}_{k|k-1} + B_u^\Delta u_k \\ \hat{x}_{k|k-1} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k+1} \end{bmatrix} + \begin{bmatrix} C_y^\Delta S_{k|k-1} & R^\Delta & 0 \\ A^\Delta S_{k|k-1} & 0 & Q^\Delta \\ S_{k|k-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} n_k \\ -\xi_k^y \\ -\xi_k^x \end{bmatrix}. \end{aligned} \tag{5.10}$$

Let now T_r be an orthogonal transformation matrix (i.e. $T_rT_r^T = I$) such that

$$\begin{bmatrix} C_y^\Delta S_{k|k-1} & R^\Delta & 0 \\ A^\Delta S_{k|k-1} & 0 & Q^\Delta \\ S_{k|k-1} & 0 & 0 \end{bmatrix} T_rT_r^T \begin{bmatrix} n_k \\ -\xi_k^y \\ -\xi_k^x \end{bmatrix} = \begin{bmatrix} \tilde{R}^\Delta & 0 & 0 \\ \tilde{G}^\Delta & S_{k+1|k}^\Delta & 0 \\ \bullet & \bullet & \bullet \end{bmatrix} \begin{bmatrix} \nu_k \\ \tilde{n}_k \\ \tilde{\xi}_k \end{bmatrix},$$

where the symbols $\bullet$ denote entries of no importance for the sequel. Note that the first row in (5.10) is independent on the variables x_k and x_{k+1} and ν_k can therefore be directly expressed, i.e. $\nu_k = (\tilde{R}^\Delta)^{-1}(C_y^\Delta \hat{x}_{k|k-1} + D_{yu}^\Delta u_k - y_k)$. Substituting this expression in the second row and subsequently moving the term $\tilde{G}^\Delta \nu_k$ to the left side of the equation, one gets an expression of the form (5.9).

Thus $S^{\Delta}_{k+1|k}$ is the square-root covariance matrix which, however, depends on the uncertainty Δ and is therefore unknown. For that reason the covariance matrix will be computed so as to be compatible with all values of the uncertainty. This motivates us to consider the following optimization problem

$$\begin{array}{ll} \underset{\gamma,\ P_{k+1|k}}{\text{minimize}} & \gamma \\ \text{subject to:} & \gamma \geq \mathbf{trace}(P_{k+1|k}) \\ & P_{k+1|k} \geq S^{\Delta}_{k+1|k}(S^{\Delta}_{k+1|k})^T,\ \forall \Delta \in \mathbf{\Delta}. \end{array} \tag{5.11}$$

For simplicity of notations we denote $M(\Delta) = S^{\Delta}_{k+1|k}(S^{\Delta}_{k+1|k})^T$.

To solve the optimization problem (5.11), we will again make use of the probabilistic approach. To this end we consider the feasibility problem, namely for a fixed $\gamma > 0$ we search for a symmetric matrix $P_{k+1|k}$ such that the matrix inequality

$$\begin{bmatrix} \gamma - \mathbf{trace}(P_{k+1|k}) & \\ & P_{k+1|k} - M(\Delta) \end{bmatrix} \geq 0, \tag{5.12}$$

holds for all $\Delta \in \mathbf{\Delta}$. Then Algorithm 2.4 on page 59 can be used to minimize γ.

Let p_{ij} denote the (i,j) entry of the matrix $P_{k+1|k}$. In order to apply the ellipsoid method for probabilistic design we will first need to collect the *free* elements of the matrix variable $P_{k+1|k}$ in the feasibility problem (5.12) in one vector , i.e.

$$\boldsymbol{d} \doteq \begin{bmatrix} p_{11} & \dots & p_{1n} & p_{22} & \dots & p_{2n} & p_{33} & \dots & p_{nn} \end{bmatrix}^T \in \mathbb{R}^{\frac{1}{2}n(n+1)}.$$

The feasibility problem (5.12) can then be equivalently rewritten in the form

$$V^{\Delta}_{\gamma}(\boldsymbol{d}) = V^{\Delta}_{\gamma,0} + \sum_{i=1}^{\frac{1}{2}n(n+1)} V^{\Delta}_{\gamma,i}\boldsymbol{d_i} \geq 0,\ \forall \Delta \in \mathbf{\Delta},$$

where $\boldsymbol{d_i}$ denotes the i-th entry of the vector $\boldsymbol{d}$, and where the matrices $V^{\Delta}_{\gamma,0}$ and $V_{\gamma,i}$ are derived from (5.12). For this problem the feasibility set is defined as

$$\mathcal{S}_{\mathcal{P}} \doteq \{\boldsymbol{d} :\ V^{\Delta}_{\gamma}(\boldsymbol{d}) \geq 0,\ \forall \Delta \in \mathbf{\Delta}\}.$$

The goal here is thus to find a vector $\boldsymbol{d}$ such that $V^{\Delta}_{\gamma}(\boldsymbol{d}) \geq 0$ for all $\Delta \in \mathbf{\Delta}$, on the basis of which an initial ellipsoid that contains $\mathcal{S}_{\mathcal{P}}$ can be formed. This initial ellipsoid would then be used to initialize the probabilistic ellipsoid algorithm, that is now to be applied to the following function

$$w(\boldsymbol{d}, \Delta) \doteq \|\mathbf{\Pi}^{-}[V^{\Delta}_{\gamma}(\boldsymbol{d})]\|^2_F \geq 0, \tag{5.13}$$

that has the property that for any $\boldsymbol{d}^* \in \mathcal{S}_{\mathcal{P}}$, it holds that $w(\boldsymbol{d}^*, \Delta) = 0$ for all $\Delta \in \mathbf{\Delta}$.

Then similarly to Lemma 2.3 we have the following result.

Lemma 5.1 *The function $w(\boldsymbol{d}, \Delta)$, defined in (5.13), is convex and differentiable in $\boldsymbol{d}$, with gradient given by*

$$\nabla w(\boldsymbol{d}, \Delta) = 2 \begin{bmatrix} \mathbf{trace}\left(V^{\Delta}_{\gamma,1}\mathbf{\Pi}^{-}[V^{\Delta}_{\gamma}(\boldsymbol{d})]\right) \\ \mathbf{trace}\left(V^{\Delta}_{\gamma,2}\mathbf{\Pi}^{-}[V^{\Delta}_{\gamma}(\boldsymbol{d})]\right) \\ \vdots \\ \mathbf{trace}\left(V^{\Delta}_{\gamma,\frac{1}{2}n(n+1)}\mathbf{\Pi}^{-}[V^{\Delta}_{\gamma}(\boldsymbol{d})]\right) \end{bmatrix} \tag{5.14}$$

The proof is similar to that of Lemma 2.3 on page 40 and will therefore will be omitted here.

Having an expression of the gradient of the function $w(\boldsymbol{d}, \Delta)$ one can make use of Algorithm 2.2 to find a feasible solution for a fixed γ, provided that an initial ellipsoid containing the feasibility set $\mathcal{S}_{\mathcal{P}}$ is given. One would then only need to make the following substitutions in Algorithm 2.2

$$\begin{array}{rcl} \boldsymbol{d} & \leftarrow & x, \\ \tilde{Q} & \leftarrow & P, \\ w & \leftarrow & v. \end{array} \tag{5.15}$$

Let us now concentrate on the problem of finding the initial ellipsoid

$$E^{(0)} = \{d : \ (d - d^{(0)})^T \tilde{Q}_0^{-1} (d - d^{(0)})\} \supseteq \mathcal{S}_{\mathcal{P}}. \tag{5.16}$$

In order to find $d^{(0)}$ and $\tilde{Q}_0$ we will first try to find a box containing the solution set $\mathcal{S}_{\mathcal{P}}$, after which an ellipsoid that contains this box can easily be formed. Thus we now first consider on the problem of finding (finite) upper $\overline{d}$ and lower $\underline{d}$ bounds on the vector d that guarantee that

$$\{d : \ \underline{d} \leq d \leq \overline{d}\} \supseteq \mathcal{S}_{\mathcal{P}}. \tag{5.17}$$

Note that the matrix inequality problem (5.12) is infeasible if for some $\Delta^* \in \boldsymbol{\Delta}$ it holds that $\gamma < \mathbf{trace}(M(\Delta^*))$, in which case γ needs to be increased. We have the following result.

Theorem 5.1 *Let* m_{ij}, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$, *denote the elements of the matrix* $M(\Delta^*)$ *for some* $\Delta^* \in \boldsymbol{\Delta}$*, and suppose that* $\gamma > \mathbf{trace}(M(\Delta^*))$. *Define the scalars*

$$\begin{aligned} \underline{p}_{ii} &= m_{ii}, \ i = 1, 2, \dots, n, \\ \overline{p}_{ii} &= \gamma - \sum_{j \neq i} m_{jj}, \ \textit{for}\, i = 1, 2, \dots, n, \\ \overline{p}_{ij} &= m_{ij} + \gamma - \mathbf{trace}(M(\Delta^*)), \ i, j = 1, 2, \dots, n, \ j \neq i, \\ \underline{p}_{ij} &= m_{ij} - \gamma + \mathbf{trace}(M(\Delta^*)), \ i, j = 1, 2, \dots, n, \ j \neq i. \end{aligned} \tag{5.18}$$

Let also $P_{k+1|k} = [p_{ij}]$ *be any symmetric matrix for which (5.12) holds for all* $\Delta \in \boldsymbol{\Delta}$. *Then*

$$\underline{p}_{ij} \leq p_{ij} \leq \overline{p}_{ij}, \tag{5.19}$$

for all $i = 1, 2, \dots, n$ *and* $j = 1, 2, \dots, n$.

Proof: From (5.12) it follows that $P_{k+1|k} \geq M(\Delta)$ for all $\Delta \in \boldsymbol{\Delta}$. Therefore it must also hold that $P_{k+1|k} \geq M(\Delta^*) = [m_{ij}]$ for any fixed $\Delta^* \in \boldsymbol{\Delta}$. Therefore

$$p_{ii} \geq m_{ii} = \underline{p}_{ii}, \ i = 1, 2, \dots, n,$$

so that the lower bounds in (5.19) on the diagonal elements p_{ii} of $P_{k+1|k}$ has been shown.

On the other hand, $P_{k+1|k}$ should be such that $\mathbf{trace}(P_{k+1|k}) \leq \gamma$. This implies that

$$\gamma \geq p_{ii} + \sum_{j \neq i} p_{jj} \geq p_{ii} + \sum_{j=1,\ldots,n} m_{jj},\ \forall i = 1, 2, \ldots, n, \tag{5.20}$$

so that

$$p_{ii} \leq \gamma - \sum_{j=1,\ldots,n} m_{jj} = \overline{p}_{ii}, \tag{5.21}$$

that completes the proof for the upper bounds on the diagonal elements p_{ii} of $P_{k+1|k}$.

In order to find lower and upper bounds on the non-diagonal entries we notice that $P_{k+1|k} \geq M(\Delta^*)$ implies

$$\begin{bmatrix} p_{ii} & p_{ij} \\ p_{ji} & p_{jj} \end{bmatrix} \geq \begin{bmatrix} m_{ii} & m_{ij} \\ m_{ji} & m_{jj} \end{bmatrix},\ \forall i \neq j.$$

Using the Schur complement the above inequality is equivalent to

$$\left| \begin{array}{l} p_{jj} - m_{jj} \geq 0, \\ (p_{ii} - m_{ii}) - (p_{ij} - m_{ij})(p_{jj} - m_{jj})^{-1}(p_{ji} - m_{ji}) \geq 0. \end{array} \right.$$

From the second inequality, making use of the symmetry of the matrices $M(\Delta^*)$ and $P_{k+1|k}$ (i.e. $m_{ij} = m_{ji}$ and $p_{ij} = p_{ji}$), it follows that

$$\begin{aligned} |p_{ij} - m_{ij}| &\leq \sqrt{(p_{ii} - m_{ii})(p_{jj} - m_{jj})} \\ &\leq \sqrt{(\overline{p}_{ii} - m_{ii})(\overline{p}_{jj} - m_{jj})}, \end{aligned}$$

Substitution of equation (5.21) then results in

$$\begin{aligned} |p_{ij} - m_{ij}| &\leq \sqrt{(\overline{p}_{ii} - m_{ii})(\overline{p}_{jj} - m_{jj})} \\ &= \sqrt{(\gamma - \mathbf{trace}(M(\Delta^*)))^2} \\ &= |\gamma - \mathbf{trace}(M(\Delta^*))|. \end{aligned}$$

And since $\gamma \geq \mathbf{trace}(M(\Delta^*))$, we have shown that $P_{k+1|k} \geq M(\Delta^*)$ implies

$$\begin{aligned} p_{ij} &\leq m_{ij} + \gamma - \mathbf{trace}(M(\Delta^*)) = \overline{p}_{ij}, \\ p_{ij} &\geq m_{ij} - \gamma + \mathbf{trace}(M(\Delta^*)) = \underline{p}_{ij}, \end{aligned} \tag{5.22}$$

so that also the upper and lower bounds on the non-diagonal elements of $P_{k+1|k}$ have been derived. □

Using the result of Theorem 5.1, the upper and lower bounds on the elements of the vector $\boldsymbol{d}$

$$\begin{aligned} \overline{d} &= \begin{bmatrix} \overline{p}_{11} & \cdots & \overline{p}_{1n} & \overline{p}_{22} & \cdots & \overline{p}_{2n} & \overline{p}_{33} & \cdots & \overline{p}_{nn} \end{bmatrix}^T, \\ \underline{d} &= \begin{bmatrix} \underline{p}_{11} & \cdots & \underline{p}_{1n} & \underline{p}_{22} & \cdots & \underline{p}_{2n} & \underline{p}_{33} & \cdots & \underline{p}_{nn} \end{bmatrix}^T. \end{aligned} \tag{5.23}$$

are such that (5.17) holds, i.e. the solution set $\mathcal{S}_{\mathcal{P}}$ is contained in the box $R = \{d : \underline{d} \leq d \leq \overline{d}\}$. It thus only remains to form an ellipsoid around the box R that

Algorithm 5.1 (Algorithm for computation of $P_{k+1|k}$)

Initialization:

Step 1. Denote $M(\Delta^*) = [m_{ij}]$ for some $\Delta^* \in \mathbf{\Delta}$.

Step 2. Compute $\overline{p}_{ij}$ and $\underline{p}_{ij}$ for $i, j = 1, 2, \ldots, n$ using equations (5.18).

Step 3. Form the vectors $\overline{d}$ and $\underline{d}$ using (5.23)

Step 4. Form the initial ellipsoid (5.16) parametrized by (5.24).

Step 5. Run Algorithm 2.2 with substitutions (5.15). With the obtained solution $\boldsymbol{d}^*$ form

$$P_{k+1|k} = \begin{bmatrix} d_1^* & d_2^* & \cdots & d_n^* \\ \star & d_{n+1}^* & \cdots & d_{2n-1}^* \\ \vdots & \vdots & \ddots & \vdots \\ \star & \star & \star & d_{n(n+1)/2}^* \end{bmatrix}.$$

would then also contain $\mathcal{S}_{\mathcal{P}}$. Using Lemma 2.6 on page 52 the initial ellipsoid can be taken as $E_P^{(0)} = \{d : \ (d - d^{(0)})^T \tilde{Q}_0^{-1} (d - d^{(0)})\}$ with

$$d^{(0)} = \frac{1}{2}(\overline{d} + \underline{d}), \ \tilde{Q}_0 = \frac{\dim d}{4} \left[\mathbf{diag}(\overline{d} - \underline{d})\right]^2, \tag{5.24}$$

with $\underline{d}$ and $\overline{d}$ defined in (5.23).

The complete procedure for obtaining the covariance matrix is summarized in Algorithm 5.1.

A faster algorithm for finding $P_{k+1|k}$

A more conservative, but computationally faster way to compute the covariance matrix $P_{k+1|k}$ so that it is compatible with all possible values of the uncertainty is to try to find it so that

$$P_{k+1|k} \geq M(\Delta), \ \forall \Delta \in \mathbf{\Delta},$$

i.e. without the minimization over the trace of $P_{k+1|k}$ in (5.11). To this end we propose the following algorithm for computation of $P_{k+1|k}$.

The following result sows that by computing $P_{k+1|k}$ using Algorithm 5.2 ensures that $P_{k+1|k} \geq M(\Delta^{(i)})$ (at least) for the generated uncertainty samples $\Delta^{(i)}$.

Lemma 5.2 *Suppose that L iterations of Algorithm 5.1 are performed. Then the matrix $P_{k+1|k} = P_{k+1|k}^{(L)}$ is such that*

Algorithm 5.2 (A faster algorithm for computation of $P_{k+1|k}$)

```
Initialization: small $\varepsilon > 0$, integer $K > 0$.
Step 1. Take $P^{(0)}_{k+1|k} = \varepsilon I$ and set $i = 1$.
Step 2. Set $i \leftarrow i + 1$.
Step 3. Generate a random sample $\Delta^{(i)}$ with probability distribution $f_\Delta$.
Step 4. Compute
    $P^{(i)}_{k+1|k} = P^{(i-1)}_{k+1|k} - \left[P^{(i-1)}_{k+1|k} - M(\Delta^{(i)})\right]^{-}$    (5.26)
Step 5. If $\|P^{(i)}_{k+1|k} - P^{(i-K)}_{k+1|k}\|_F = 0$ then take $P_{k+1|k} = P^{(i)}_{k+1|k}$ Stop else Goto Step 2.
```

(i) $P_{k+1|k} > 0$, *and*

(ii) $P_{k+1|k} \geq M(\Delta^{(i)})$, *for* $i = 1, 2, \ldots, L$.

Proof: (i) Noting that

$$\left[P^{(i-1)}_{k+1|k} - M(\Delta^{(i)})\right]^{-} \leq 0,$$

it follows from equation (5.26) that

$$P^{(i)}_{k+1|k} \geq P^{(i-1)}_{k+1|k} \tag{5.27}$$

for all $i = 1, \ldots, L$, and thus $P_{k+1|k} \geq P^{(0)}_{k+1|k} > 0$.
(ii) Note that

$$\begin{aligned} P^{(i)}_{k+1|k} &= P^{(i-1)}_{k+1|k} - \left[P^{(i-1)}_{k+1|k} - M(\Delta^{(i)})\right]^{-} \\ &= P^{(i-1)}_{k+1|k} - M(\Delta^{(i)}) + M(\Delta^{(i)}) - \left[P^{(i-1)}_{k+1|k} - M(\Delta^{(i)})\right]^{-} \\ &= M^{\Delta^{(i)}} + \left[P^{(i-1)}_{k+1|k} - M(\Delta^{(i)})\right]^{+} \\ &\geq M(\Delta^{(i)}) \end{aligned}$$

which together (5.27) implies (ii). □

5.5 A Case Study

The case study presented has the purpose to outline the capabilities of the newly proposed method to robust output-feedback MPC. As discussed in the intro-

Parameter:	**Symbol:**	**Value:**
gearbox ratio	N	$[-294.478,\ -226.722]$
joint angle of inertial axis	Ω	variable
effective joint input torque	T_j^{eff}	variable
motor torque constant	K_t	0.6
the damping coefficient	β	0.4
deformation torque of the gearbox	T_{def}	variable
inertia of the input axis	I_m	0.0011
inertia of the output system	I_{son}	400
joint angle of the output axis	ϵ	variable
motor current	i_c	variable
spring constant	c	$[11.7 \times 10^4,\ 14.3 \times 10^4]$

Table 5.1: The nominal values of the parameters in the linear model of one joint of the SRM.

duction, at present the available hardware imposes a severe restriction for the real-life applicability of all robust approaches to MPC due to the underlying optimization problem that needs to be solved at each iteration.

The example considered in this section is the linear model of one joint of a real-life space robot manipulator (SRM) system, introduced in Section 3.6 on page 86. The state-space model of the system is given by

$$\begin{aligned}
\dot{x}(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{c}{N^2 I_m} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{\beta}{I_{son}} & -\frac{c}{N^2 I_m} - \frac{c}{I_{son}} & -\frac{\beta}{I_{son}} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{K_t}{N I_m} \\ 0 \\ -\frac{K_t}{N I_m} \end{bmatrix} u(t) \\
y(t) &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & N & 0 & 0 \end{bmatrix} x(t) + R_y \xi^y(t) \\
z(t) &= \begin{bmatrix} 0 & N & 0 & 0 \end{bmatrix} x(t) + R_z \xi^z(t)
\end{aligned} \tag{5.28}$$

This model is discretized with sampling period $T_s = 0.1$, [sec]. The system parameters are given in Table 5.1.

The following choice of the iPC parameters is made: $N_1 = 1$, $N_2 = 8$, $NU = 7$, and $W = 10^3$. It was also selected

$$Q = 10^{-3} I_4,\ R_y = 10^{-3} I_2,\ R_z = 10^{-3},\ P_{0|-1} = I_4.$$

In addition, the parameters c (the spring constant) and N (the gearbox ratio) are considered as uncertain - the true value of these parameters are only known to lie in intervals of respectively 10% and 13% around their nominal values.

As a reference r_k a low-pass filtered step signal from 0 to 1 at time instant $k = 0$, and from 1 to -1 at time instant $k = 30$ is selected. The low-pass filter that was used is as follows

$$W_r(z) = \frac{0.0902z + 0.0646}{z^2 - 1.2131z + 0.3679}. \tag{5.29}$$

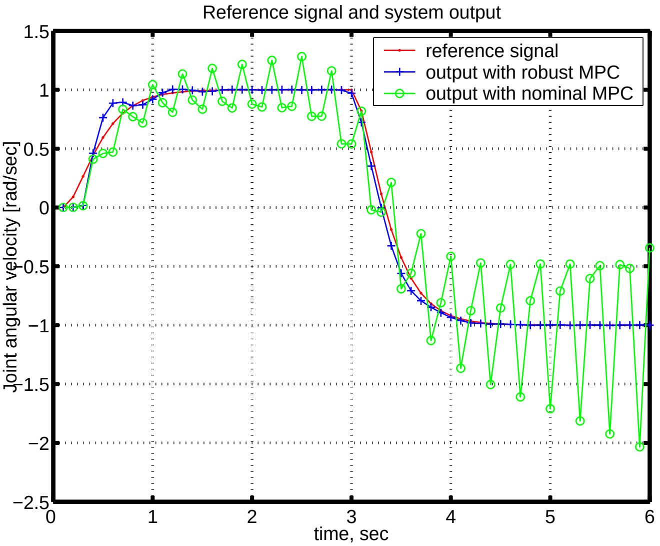


Figure 5.1: Reference trajectory, and the controlled output z_k for the system with the newly proposed robust iPC controller.

A simulation was made in which the true values for the uncertain parameters are selected as $c_{true} = 14.3 \times 10^4$ and $N_{true} = -294.478$. The simulation was performed with two controllers. The first controller is the newly proposed robust iPC controller. The second controller does not take the uncertainty into account; it consists of an interconnection with a Kalman filter and a standard receding horizon MPC both designed based on the assumption that the true values of the two uncertain parameters are $\tilde{c} = 13.585 \times 10^4$ and $\tilde{N} = -245.3549$, for which values the optimization problem (5.8) is solved at each iteration.

Figure 5.1 presents the results from the simulation. The figure depicts the reference trajectory and the controlled output z_k for the system with the two controllers. It can be observed that the newly proposed controller achieves stability and reference trajectory tracking, while the controller that does not take the uncertainty into consideration results in an unstable closed-loop system.

Figures 5.2, 5.3 and 5.4 depict the first output of the system and its measurement, the states and their estimates, and the control action u_k, that result from the simulation with the newly proposed robust iPC controller.

Comparison to other methods

Two other output-feedback methods were tried on this example, the BMI method to passive FTC (Chapter 3) and the probabilistic LPV design method to active FTC (Section 4.3 of Chapter 4). Both methods were unable to find a stabilizing controller. The BMI method failed at the second step of the initial controller computation (see Section 3.4.2). The probabilistic LPV method was used

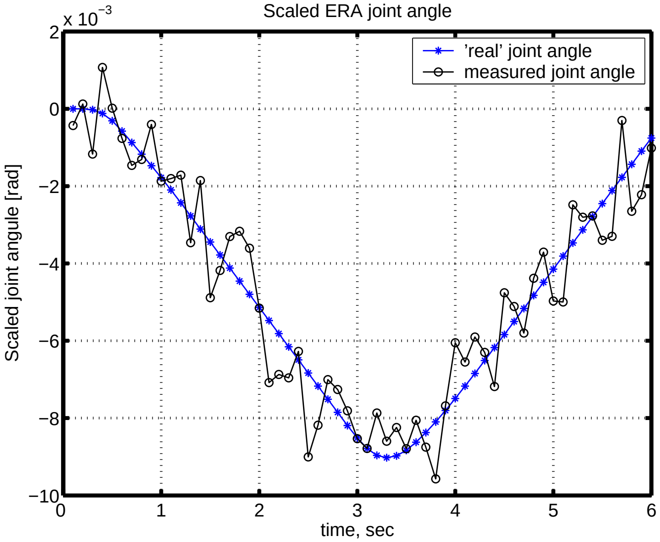


Figure 5.2: The first output of the system with its corrupted by noise measurement.

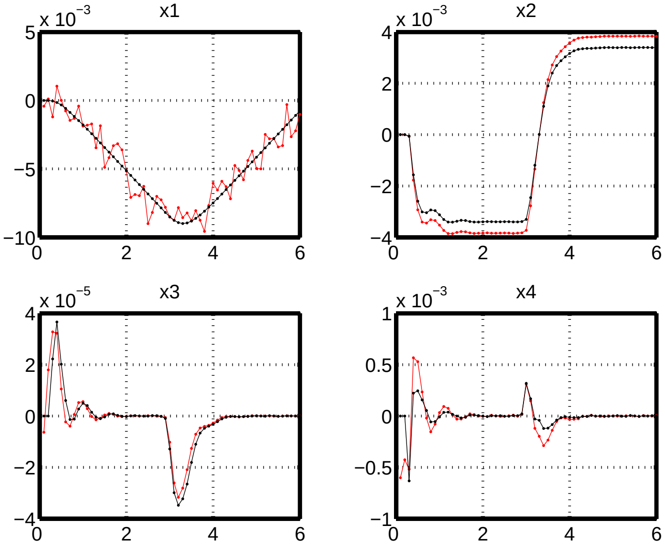


Figure 5.3: The system states and their estimates.

in search for an LPV controller with matrices that are affine in the parameters c and N, i.e. $g_1 = c$ and $g_2 = N$ in (4.37). Different attempts were made to compute a controller achieving closed-loop $\mathcal{H}_\infty$-norm of up to 10^6, but these were all

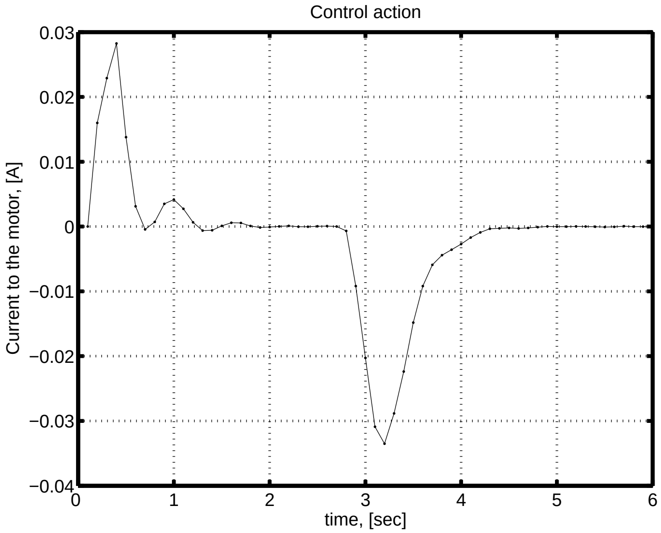


Figure 5.4: The control action generated by the iPC controller.

unsuccessful. It needs to be pointed out here that this output-feedback LPV design problem has 92 optimization variables so that it requires a serious amount of computations. For comparison, the MPC method from this chapter has only 15 optimization variables (8 for the states in two subsequent time instances, and 7 for the control action during the control horizon $N_u = 7$).

5.6 Conclusion

In this chapter a new approach to robust active FTC was presented based on MPC control that does not impose the assumption that the system state is measured or known. The method is based on a combination of a finite-time Kalman filter and a finite horizon MPC into one robust least-squares optimization problem, in which the vector of unknowns consists of the system states and the control action over the selected control horizon. The chapter considers a very general class of uncertainties – the only assumption imposed is that the system state-space matrices remain bounded over all uncertainties. The optimization problem is solved in the probabilistic framework provided that the state covariance matrix is given. Additionally, two methods for finding a covariance matrix that is compatible with all values of the uncertainty were presented. The first one aims at minimization of the trace of the covariance matrix and is computationally more involving, while the second method is more conservative but much faster. The complete MPC approach has been tested on a case study with a model of one joint of a real-life space robotic manipulator.

6

Brushless DC Motor Experimental Setup

In this chapter an experimental setup, consisting of a brushless DC motor (BDCM), is described for demonstrating the capabilities of the probabilistic LPV approach to robust active FTC, developed in Section 4.3 of Chapter 4. To make this active FTC method applicable in real-life, it is extended with a simple fault detection. In this FDD scheme we restrict to estimating the motor parameters and postulating their uncertainty ranges for the sole purpose of demonstrating that the robust active FTC method of Section 4.3 can deal with time-varying fault estimates including the uncertainty ranges of the estimated quantities.

6.1 Introduction

Electrical machines form an important part of the nowadays more and more complex, safe-critical control systems. In such systems simple faults can easily develop into catastrophic failures of important sub-systems leading to unpredictable consequences. Such failures can often be avoided by timely reconfiguration of the control system after a fault has been detected and diagnosed.

In the literature, two main streams of research in this area can be distinguished: the field of *fault detection and diagnosis* (Gertler 1998; Chen and Patton 1999), and the one of *controller reconfiguration* (Astrom et al. 2001; Zhang and Jiang 2003; Patton 1997; Blanke et al. 1997). However, in the field of FDD most of the approaches consider open-loop operation, and in the field of CR the majority of the methods assume perfect estimation of the faults. As a result, the integration of these approaches still remains a challenging problem with both theoretical and practical relevance.

In this chapter an approach to the design of fault-tolerant systems is proposed that consists of two interacting parts, one having the task to detect faults and estimate the parameters of the system (FDD scheme), and the other implementing a CR strategy using the information from the FDD scheme and accounting for uncertainty in this information. The FDD part is based on an RLS scheme with adaptive forgetting factor for on-line motor parameter estimation. Appropriately adapting the forgetting factor of the algorithm makes it possible to track abrupt changes in the parameters while at the same time remaining less sensitive to noise during periods of little or no changes in the parameter estimates. In order to take into consideration the closed-loop operation of the system, the RLS scheme is used to estimate only the parameters of the closed-loop system, from which the real open-loop motor parameters are computed. In addition to that, a CUSUM test (Basseville and Nikiforov 1993) is used to detect changes in the mean values of the open-loop parameter estimates and to trigger fault detection flags, which in turn inform the CR scheme that the estimates might be imprecise. We note that this FDD scheme is only designed for the purpose of demonstrating the on-line capabilities of the robust active FTC method of Section 4.3.

The CR scheme implements the robust active FTC method from Section 4.3.4 of Chapter 4 that is applicable to systems with sensor, actuator and component faults in the presence of model and FDD uncertainty. The estimates of the faults in the system, obtained from the FDD scheme, are also considered uncertain, i.e. are assumed to lie inside some uncertainty intervals with given (but possibly time-varying) sizes. The robust active FTC is scheduled by both the estimates of the faults and the sizes of the corresponding uncertainty intervals. The complete FDD&CR scheme is successfully tested on an experimental setup with a brushless DC motor (BDCM).

Some existing works focused on the problem of FDD and CR of BDCM's are Moseler and Isermann (2000); Liu (1996); Bolognani et al. (2000). In Moseler and Isermann (2000) a discrete square-root implementation of the RLS algorithm is used to directly estimate some of the physical parameters of a continuous-time BDCM model. In Liu (1996) the authors propose an algorithm to FDD in BDCM based on combined parameter estimation and multi-layer perception

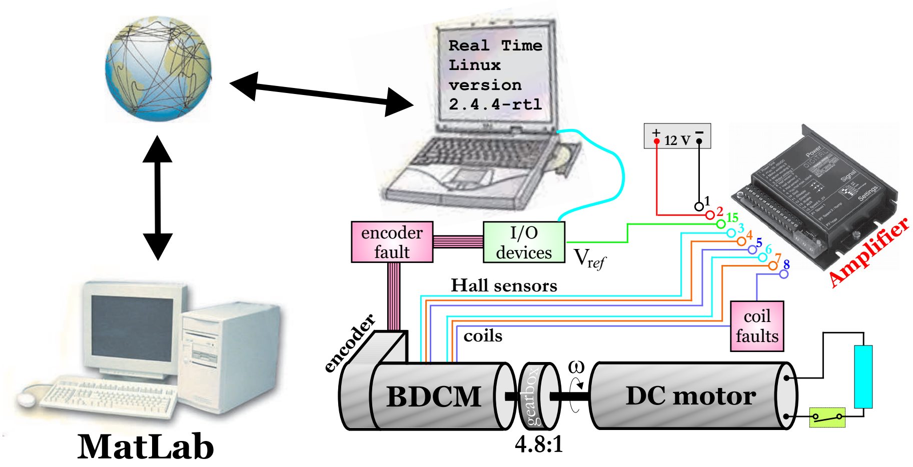


Figure 6.1: Schematic representation of the complete BDCM experimental setup.

neural network. In Bolognani et al. (2000) fault accommodation strategies are discussed for some inverter faults like open or short circuit of one power devise.

The chapter is organized as follows. In Section 6.2 the experimental setup is described and a model of a BDCM is given. The problem is next formulated in Section 6.3, where also the requirements for the FDD and the CR schemes are stated. In Section 6.4 the proposed algorithms are developed, which are subsequently tested in Section 6.5 on the experimental setup. The chapter is concluded with some final remarks in Section 6.6.

6.2 Model of a Brushless DC Motor

A brushless DC motor experimental setup has been recently developed in the Delft Center for Systems and Control group at the Delft University of Technology for the purpose of testing and demonstrating the performances of different FDD and CR approaches. The setup consists of a MAXON 3-phase BDCM (type EC-23), a MAXON 1-Q-EC Amplifier DEC 50-5, an Encoder HEDL 55, a gearbox with ratio $N_{GB} = 1/4.8$, and a DC motor acting as a load (see Figures 6.1 and 6.2). The BDCM has a permanent-magnet rotor, and the three stator windings are implemented in such a way that a trapezoidal back EMF is obtained (see Figure 6.3). Rectangular stator currents are needed to produce a static electric torque. The rotor position is detected by means of three Hall sensors, mounted at 120^o, and the position is used to determine the switching sequence of the six transistors in the Amplifier (see Figure 6.2). The Amplifier implements open-loop control of the motor by adjusting the amplitude of the rectangular stator currents proportionally to the applied reference voltage V_{ref}. For an overview of the literature on analysis and modelling of brushless DC motors the reader is referred to Pillay

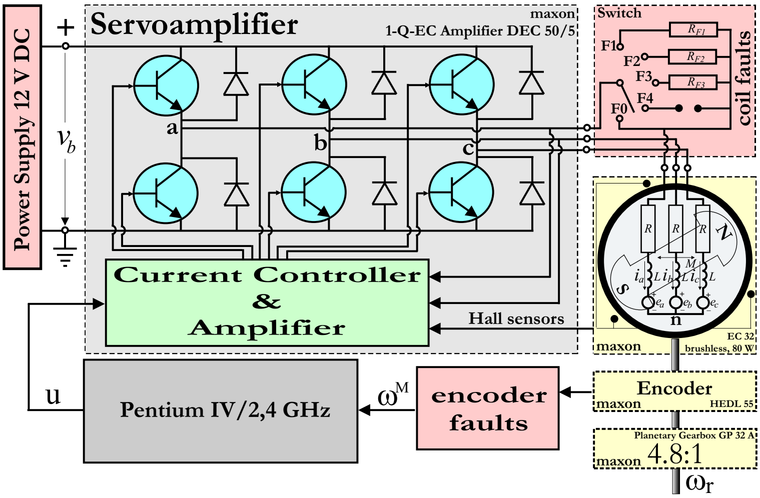


Figure 6.2: Schematic representation of the electrical part of the setup.

and Krishnan (1989); Lee and Efsani (2003) and the references therein.

The goal is to control the angular velocity of ω after the gearbox, both in fault-free and in faulty situations. The experimental setup works under Real-Time Linux 2.4.4-rtl operating system running on a ASUS L3500 computer with a Pentium IV/2.4 GHz processor. The developed software has the tasks to control the setup and to enable the user to monitor and change some of the variables (e.g. the controller parameters) of the algorithm. The later is achieved via a so-called Remote Data Access (RDA) interface that makes it possible to monitor the process via mathematical software packages like MATLAB ® (see Figure 6.1). All algorithms presented in the chapter were implemented in C.

As depicted in Figure 6.1, additional hardware has been developed for the purposes of introducing faults in the system. The additional hardware makes it possible to introduce both coil faults and faults in the encoder. The coil faults that can be introduced consist of increasing the resistance of one coil by 1Ω, 3Ω or 5Ω by means of including resistors in the loop, and broken transistor in the Amplifier (see Figure 6.2). The last fault is made by connecting a diode to one coil that simulates the effect of a broken (not conducting) transistor. The fault in the encoder that can be introduced represent a partial 50% sensor fault (this will be explained in more detail in Section 6.4.1).

For the purposes of deriving a dynamic model of the BDCM the assumption is imposed that the power semiconductor devices in the inverter are ideal. The inductances in the electrical part of the motor are also neglected. In the fault-free case it is assumed also that the resistances of the three phases are the same, i.e. $R_a = R_b = R_c = R$. Under these simplifying assumptions, the electrical part

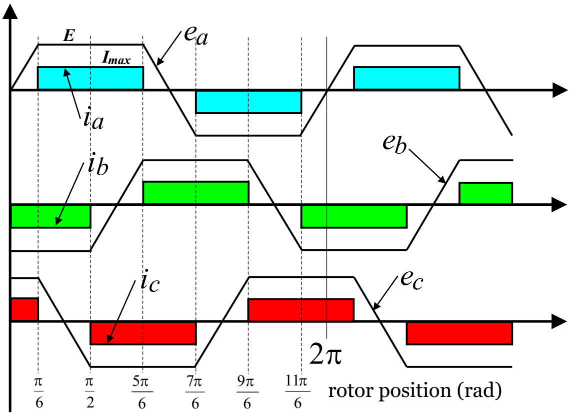


Figure 6.3: Stator currents and the back EMF's in the BDCM.

of the BDCM can be represented as (Pillay and Krishnan 1989)

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & & \\ & R & \\ & & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}, \tag{6.1}$$

where the physical meanings of the parameters and signals in the model, together with their dimensions, are summarized in Table 6.1.

The electromagnetic torque is given by

$$T_e = \frac{e_a i_a + e_b i_b + e_c i_c}{\omega_r}, \tag{6.2}$$

where ω_r is the angular velocity of the rotor, and where the back EMF's $e_a,\ e_b,\ e_c$ of the three phases have trapezoidal shape as shown in Figure 6.3 with amplitude (Lee and Efsani 2003)

$$E = k_E \omega_r. \tag{6.3}$$

Therefore, the electromagnetic torque is independent on the rotor position and can be written in the form

$$T_e = 2k_E I_{max} = k_T I_{max}. \tag{6.4}$$

The equation of motion of the motor is given by

$$J\dot{\omega}_r = T_e - T_{load} - T_{losses}. \tag{6.5}$$

The losses result from Coulomb and viscous friction, so that the following model for T_{losses} can be used (Moseler and Isermann 2000)

$$T_{losses} = \underbrace{c_c sign(\omega_r)}_{\text{Coulomb}} + \underbrace{c_v \omega_r}_{\text{viscous}}. \tag{6.6}$$

The block Current Controller & Amplifier in Figure 6.2 implements open-loop control of the motor. Its input is a reference input voltage V_{ref} from which the desired magnitude of the stator currents I_{max} is computed (Figure 6.3). Then by using either Hysteresis control or PWM control (Pillay and Krishnan 1989), the transistors are switched in such a way that the real stator currents have (approximately) a rectangular shape with magnitude I_{max}. Thus, the resulting equivalent supply voltage at the phases of the motor becomes $V_D u(t)$, where $u(t) = \frac{V_{ref}}{\max(V_{ref})} \in [0,\ 1]$ is a modulating signal. Assume now, without loss of generality, that the rotor position is in the interval $[\frac{\pi}{6}, \frac{\pi}{2}]$ rad (see Figure 6.3), so that coils "a" and "b" are conducting with

$$\begin{array}{lll} v_a = V_D u(t), & i_a = I_{max}, & e_a = E \\ v_b = 0, & i_b = -I_{max}, & e_b = -E. \end{array}$$

Then, using equations (6.1) and (6.3), it can be written

$$V_D u(t) = v_a - v_b = 2RI_{max} + 2k_E\omega_r,$$

so that

$$I_{max} = -\frac{k_E}{R}\omega_r + \frac{V_D}{2R}u(t). \tag{6.7}$$

Combining equations (6.4)-(6.7), the following continuous-time model of the dynamics of the BDCM is derived

$$\dot{\omega}(t) = -\left(\frac{k_E k_T}{RJ} + \frac{c_v}{J}\right)\omega(t) + \frac{K_T V_D N_{GB}}{2RJ}u(t) - \frac{T^L N_{GB}}{J}, \tag{6.8}$$

where $\omega = N_{GB}\omega_r$ is the angular velocity after the gearbox, and where T_L $[N.m]$ as torque resulting from load and Coulomb friction. ω is calculated by using the reading from an encoder that measures the position of the rotor. The setup works at a sampling time of $T_s = 0.01$ $[sec]$, and the discrete-time equivalent to model (6.8) is

$$\omega_{k+1} = a^{nom}\omega_k + b^{nom}u_k + b^{nom}_{off}. \tag{6.9}$$

where (assuming zero order hold discretization)

$$\begin{aligned} a^{nom} &= \exp^{-\left(\frac{k_E k_T}{RJ} + \frac{c_v}{J}\right)T_s}, \\ b^{nom} &= \left(\exp^{-\left(\frac{k_E k_T}{RJ} + \frac{c_v}{J}\right)T_s} - 1\right)\frac{K_T V_D N_{GB}}{2k_E k_T + 2c_v R}, \\ b^{nom}_{off} &= \left(1 - \exp^{-\left(\frac{k_E k_T}{RJ} + \frac{c_v}{J}\right)T_s}\right)\frac{T^L N_{GB} R}{k_E k_T + c_v R}. \end{aligned} \tag{6.10}$$

When sensor, actuator and/or component faults occur in the system, the nominal model changes to the following faulty model

$$\omega_{k+1} = a_f\omega_k + b_f u_k + b_{f,off}, \tag{6.11}$$

$$\omega_k^M = \sigma\omega_k, \tag{6.12}$$

where the faulty scalars a_f, b_f, and b_{off} result from parameter and actuator faults, and where $\sigma \in (0,\ 1]$ represents sensor fault (i.e. fault in the encoder). After the occurrence of a sensor fault the angular velocity ω_k differs from the measured value ω_k^M. The goal in the case of an encoder faults is, of course, to control the true velocity ω_k and not its faulty measurement ω_k^M.

Parameter:	**Symbol:**	**Dimension:**
$v_a,\ v_b,\ v_c$	*voltages at the three phases*	V
$i_a,\ i_b,\ i_c$	*phase currents*	A
$e_a,\ e_b,\ e_c$	*back EMF of the three phases*	V
$R_a,\ R_b,\ R_c$	*stator resistances*	Ω
ω_r	*angular velocity of the rotor*	rad/sec
ω	*angular velocity after the gearbox*	rad/sec
ω_{ref}	*angular velocity reference*	rad/sec
N_{GB}	*gearbox ratio*	$-$
J	*inertia of the rotor*	$kg.m^2$
V_D	*supply voltage*	V
V_{ref}	*reference voltage*	V
T_e	*electromechanical torque*	$N.m$
T_{load}	*load torque*	$N.m$
T_{losses}	*torque due to friction*	$N.m$
k_E	*back EMF constant*	$V/(rad/sec)$
k_T	*torque constant*	Nm
c_c	*Coulomb friction constant*	$N.m$
c_v	*viscous friction constant*	$N.m/(rad/sec)$

Table 6.1: Parameters and signals used in the model of the BDCM.

6.3 Problem Formulation

This chapter aims at the development of an algorithm that can achieve both robust stability and reference trajectory tracking in the presence of faults, provided that the faulty system remains controllable/observable. To this end we will develop two schemes, one having the task to detect and diagnose the faults in the system, and another to reconfigure the controller on the basis of the obtained fault estimates and their uncertainty intervals. The requirements for these two schemes are discussed in the following two subsections.

6.3.1 Fault Detection and Diagnosis Problem

The purpose of the FDD algorithm is on the basis of input-output measurements $\{\omega_k^M, u_k\}$ to produce estimates of the motor parameters that can be used for online controller reconfiguration in closed-loop. To this end, a parameter estimation algorithm is needed that makes a tradeoff between accuracy and tracking speed, and at the same time deals with closed-loop data and control action saturation. A recursive least squares algorithm (RLS) that is capable of dealing with these demands is briefly summarized below.

Tracking changes in the parameters to be estimated is achieved by introducing a forgetting factor $0 < \lambda < 1$ (typically $0.95 < \lambda < 1$) in the RLS algorithm (also known as the exponentially weighted RLS). However, using a forgetting factor makes the estimates much more sensitive to noise. Therefore in the algorithm proposed here an *adaptive forgetting factor* is used, so that $\lambda = \lambda_{min}$ whenever the current parameter estimates do not "explain" the data accurately enough (i.e. whenever an estimation error signal, to be defined later on, is large), and $\lambda = \lambda_{max}$ otherwise. Other forgetting variants, e.g. (Kulhavy and Kraus 1996; So and Leung 2001; Toplis and Pasupathy 1988; Park and Jun 1992), can also be used; in this chapter, however, we perform experiments with this simplified scheme that proves to be sufficient for our purposes.

The operation in closed-loop is made possible by using a-priori knowledge about the structure and the parameters of the controller. The idea is to estimate the parameters of the closed-loop system, from which one can then compute the open-loop parameters. The controller parameters, however, need to be modified appropriately in order to take the saturation effect into consideration. Sections 6.4.1 and 6.4.1 deal with these issues.

Finally, a CUSUM test (Basseville and Nikiforov 1993) will be used to detect changes in the mean of the parameter estimates, which will trigger fault detection flags. When a detection flag is raised, the fault estimates can initially be expected to be rather inaccurate until they converge to their new values. Therefore, immediately after a fault alarm the uncertainty in the estimates should be abruptly increased, and then gradually decreased as the estimates converge to their new values. This is discussed in section 6.4.1.

As mentioned earlier, besides faults in the a and b parameters of the system, sensor faults (faults in the encoder) are also considered. The same RLS scheme is, unfortunately, not applicable to the detection of this fault in combination with an actuator fault. The reason for that is that in SISO systems multiplicative (scaling) faults on the input and output result in the same transfer function, and therefore in the same input-output behavior (Verdult, Kanev, Breeman, and Verhaegen 2003). In order to make it possible to diagnose also this sensor faults we will make use of an additional measurement, namely the signal of one of the three Hall sensors in the motor. This is explained in more detail in section 6.4.1.

6.3.2 Robust Active Controller Reconfiguration Problem

The goal here will be to demonstrate the LPV approach to robust active FTC from Section 4.3. To this end we would like to design an LPV controller that

(i) provides guaranteed robust closed-loop stability and performance with respect to uncertainties in the estimates of the motor parameters provided by the FDD scheme,

(ii) is scheduled both by the estimates of the motor parameters as well as by the sizes of their uncertainty intervals,

(iii) achieves reference trajectory tracking.

As the third demand cannot directly be addressed by the method of Section 4.3, this is achieved here by introducing integral action in the control signal. To this end first an integrator is connected to the measured angular velocity of the motor and the LPV design is subsequently performed for the resulting augmented system. Once the LPV controller is designed, the integrator is included into it resulting into an LPV controller with integrating action. More details about this will follow in Section 6.4.2. Note that introducing integral action in the feedback is a standard strategy for achieving (piecewise) constant trajectory tracking (see, for instance, Goodwin et al. (2001)).

The next section discusses the design of the FDD and FTC schemes.

6.4 Combined FDD and robust active FTC

6.4.1 Algorithm for FDD

As discussed in Section 6.3.1, the FDD scheme is required to be capable to track abrupt changes in the parameters of the system on the one hand, and not to be too sensitive to process noise on the other. In addition, it needs to deal with control action saturation and closed-loop operation. To achieve this an algorithm for FDD is presented in this section that consists of a set of subroutines dealing with these issues. This scheme is based on a exponentially weighted RLS algorithm with adaptive forgetting factor.

Control action saturation

As mentioned in Section 6.2, the control action is saturated below 0 and above 1. As a result, the control action computed using (6.47) may differ from the real control signal applied to the motor at time instant k, which leads to poor parameter estimates.

In order to account for the control action saturation (6.48) in the RLS scheme discussed below we define, for $i = 1, 2$,

$$\bar{F}_i(k) = \begin{cases} \dfrac{F_i(k)}{u_k^R}, & u_k^R > 1 \\[2ex] F_i(k), & 0 \leq u_k^R \leq 1 \\[2ex] 0, & u_k^R < 0, \end{cases}$$

so that

$$SAT(u_k^R) = \begin{bmatrix} \bar{F}_1(k) & \bar{F}_2(k) \end{bmatrix} \begin{bmatrix} \hat{\omega}_k \\ x_k^I \end{bmatrix}.$$

In this way the effect of the saturation is easily accounted for in the controller parameters, and the signals $\hat{\omega}_k$ and x_k^I remain unaltered.

Dealing with closed-loop data

Before developing the RLS scheme the BDCM model in equation (6.51) needs first to be equivalently rewritten in the regression form

$$\hat{\omega}_{k+1} = \xi_k^T \theta_k,$$

where $\theta_k = [a,\ b,\ b_{off}]^T$ represents the parameters that have to be estimated, and $\xi_k = [\hat{\omega}_k,\ u_k,\ 1]$ is the vector of regressors. However, due to the closed-loop configuration, using this form may lead to incorrect estimates $\hat{\theta}_k$ in the case that the control action is dependent on the current estimates $\hat{\theta}_k$ (as is the case with the robust FTC developed later), which makes ξ_k a function of θ_k. To deal with this problem we will directly estimate the parameters of the closed-loop system, from which the open-loop parameters can be computed.

Applying the saturated control u_k to the BDCM (6.51) results in the closed-loop system

$$\begin{bmatrix} \hat{\omega}_{k+1} \\ x^I_{k+1} \end{bmatrix} = \begin{bmatrix} a + b\bar{F}_1(k) & b\bar{F}_2(k) \\ T_s & 1 \end{bmatrix} \begin{bmatrix} \hat{\omega}_k \\ x^I_k \end{bmatrix} - \begin{bmatrix} 0 \\ T_s \end{bmatrix} \omega^{ref}_k + \begin{bmatrix} b_{off} \\ 0 \end{bmatrix}. \tag{6.13}$$

Noting that the parameters that need to be estimated appear only in the first row in (6.13), and that the state x^I_k of the integrator is known, we can write

$$\hat{\omega}_{k+1} = (\xi^{cl}_k)^T \theta^{cl}_k, \tag{6.14}$$

with

$$\xi^{cl}_k = \begin{bmatrix} \hat{\omega}_k \\ \bar{F}_2(k) x^I_k \\ 1 \end{bmatrix},\ \theta^{cl}_k = \begin{bmatrix} a + b\bar{F}_1(k) \\ b \\ b_{off} \end{bmatrix},$$

so that now ξ^{cl}_k is independent on θ^{cl}_k.

Algorithm 6.1 can then be used for estimating the parameters θ^{cl}_k of the closed-loop system using the model (6.14).

Algorithm 6.1 makes use of an exponentially weighted RLS algorithm where the forgetting factor λ_k at each time instant may take one of two values, λ_{min} or λ_{max}. In equation (6.27) it is checked whether the estimates are accurate enough, and if so λ_k is taken equal to λ_{max}. In this way more data from the past will be used in forming the estimate $\hat{\theta}^{cl}_k$, which will become less sensitive to noise. On the other hand, if the estimates do not explain the data accurately, λ_k is set to λ_{min} which makes the algorithm alert to parameter changes. A reasonable way to check whether the estimates are accurate or not is by looking at the estimation error (6.26). Once the squared estimation error becomes higher than a predefined threshold e^2_{max}, λ_i is set to λ_{min} for $i = k, k+1, \ldots, k+t_\lambda$, where t_λ is a given integer that defines the duration time of λ_{min}. In addition, a weighting matrix $W = W^T > 0$ is used to put additional weighting on the estimated parameters $\hat{\theta}^{cl}_k$.

Algorithm 6.1 produces at each time instant estimate of the closed-loop parameters $\hat{\theta}^{cl}_k$ (6.30), which are subsequently filtered in (6.31) for some $0 < \alpha < 1$. This is done to smooth the estimates that, due to the forgetting factor, may be peaky during some time intervals. Then the estimates $\hat{\theta}_k$ of the open-loop system are produced in (6.35) from the filtered closed-loop estimates $\hat{\theta}^{cl,f}_k$. The open-loop parameter estimates obtained in this way can now be used for fault detection and controller reconfiguration. In the next subsection it is explained how a two-sided CUSUM test can be used to detect changes in the mean value θ^{mean}_k of the parameter estimates $\hat{\theta}_k$.

Algorithm 6.1 (RLS with adaptive λ_k)

PARAMETERS: $W = W^T > 0$, $e_{max} > 0$, $t_\lambda > 0$, $0 < \lambda_{min} \leq \lambda_{max} \leq 1$, $0 < \alpha < 1$.

INITIALIZATION: $k = 0$, $t_{end} = 0$, $P_0 = P_0^T > 0$, $x_0^I = 0$, $\xi_0^{cl} = 0$, AND $\hat{\theta}_0^{cl} = [a^{nom},\ b^{nom},\ b_{off}^{nom}]^T$.

Step 1. SET $k \leftarrow k + 1$.

Step 2. GET NEW DATA $\hat{\omega}_k$, ω_k^{ref}, $F_1(k)$, $F_2(k)$, AND COMPUTE

$$e_k = \hat{\omega}_k - (\xi_{k-1}^{cl})^T \hat{\theta}_{k-1}^{cl}, \tag{6.26}$$

$$\lambda_k = \begin{cases} \lambda_{min}, & k > t_{end} \text{ AND } e_k^2 > e_{max}^2, \\ \lambda_{max}, & \text{OTHERWISE.} \end{cases} \tag{6.27}$$

$$\text{IF } (\lambda_k - \lambda_{k-1}) = (\lambda_{max} - \lambda_{min}) \text{ THEN SET } t_{end} \leftarrow k + t_\lambda \tag{6.28}$$

$$\gamma_{k-1} = \frac{P_{k-1} W \xi_{k-1}^{cl}}{(\xi_{k-1}^{cl})^T W P_{k-1} W \xi_{k-1}^{cl} + \lambda_k} \tag{6.29}$$

$$\hat{\theta}_k^{cl} = \hat{\theta}_{k-1}^{cl} + W \gamma_{k-1} e_k, \tag{6.30}$$

$$\hat{\theta}_k^{cl,f} = \alpha \hat{\theta}_{k-1}^{cl,f} + (1 - \alpha) \hat{\theta}_{k-1}^{cl}, \tag{6.31}$$

$$P_k = \frac{1}{\lambda_k} (1 - \gamma_{k-1} (\xi_{k-1}^{cl})^T W) P_{k-1}, \tag{6.32}$$

$$u_k^R = F_1(k) \hat{\omega}_k + F_2(k) x_k^I, \tag{6.33}$$

$$\bar{F}_i(k) = \begin{cases} \frac{F_i(k)}{u_k^R}, & u_k^R > 1 \\ F_i(k), & 0 \leq u_k^R \leq 1 \\ 0, & u_k^R < 0, \end{cases} \tag{6.34}$$

$$\hat{\theta}_k = \begin{bmatrix} 1 & -\bar{F}_1(k) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\theta}_k^{cl,f}. \tag{6.35}$$

$$\xi_k^{cl} = \begin{bmatrix} \hat{\omega}_k, & \bar{F}_2(k) x_k^I, & 1 \end{bmatrix}^T. \tag{6.36}$$

Step 3. GO TO **Step 1.**

Fault detection via CUSUM test

Abrupt faults in the BDCM result in abrupt changes in the motor parameters. These can be detected by means of looking for changes in the mean values of the estimated parameters $\hat{\theta}_k$, for which purpose a two-sided CUSUM test (Basseville and Nikiforov 1993) can be used. Both increases and decreases in the mean can be detected by the two-sided CUSUM test that is summarized in Algorithm 6.2.

Algorithm 6.2 (two-sided CUSUM test)

PARAMETERS: $W_\theta > 0$, ν^U, ν^D, AND h.

INITIALIZATION: $k = 0$, $\theta_0^{mean} = \hat{\theta}_0$, $\theta_0^U = \theta_0^D = 0$.

Step 1. SET $k \leftarrow k + 1$.

Step 2. GET $\hat{\theta}_k$ FROM ALGORITHM 6.1 AND COMPUTE

$$\theta_k^{mean} = \frac{(W_\theta - 1)\theta_{k-1}^{mean} + \hat{\theta}_k}{W_\theta} \tag{6.41}$$

$$\theta_k^U = \left[\theta_{k-1}^U + \hat{\theta}_k - \theta_k^{mean} - \nu^U\right]^+ \tag{6.42}$$

$$\theta_k^D = \left[\theta_{k-1}^D - \hat{\theta}_k + \theta_k^{mean} - \nu^D\right]^+ \tag{6.43}$$

$$\text{IF } \left(\left\|\theta_k^U\right\|_2 + \left\|\theta_k^D\right\|_2 \geq h\right) \text{ AND } \left(\left\|\theta_{k-1}^U\right\|_2 + \left\|\theta_{k-1}^D\right\|_2 < h\right)$$
$$\text{THEN FAULT IS DETECTED AT } t_F = k. \tag{6.44}$$

Step 3. GO TO **Step 1**.

Note that a proper selection of the threshold h in the CUSUM algorithm would, in general, be dictated by the covariance matrix of the parameter estimates $\hat{\theta}_k$ as well as by the magnitude of the jump (Basseville and Nikiforov 1993; Blanke et al. 2003). It is usually suggested to choose h such as to achieve a certain tradeoff between minimal detection delay and maximal time between false alarms: increasing h increases the detection delay but reduces the chance for false alarms. In the experimental results presented in this Chapter, however, the parameter h has been experimentally selected.

Whenever a fault is detected by the CUSUM test a fault detection flag is raised to inform the reconfiguration scheme that the fault estimates might be imprecise due to the absence of enough input-output measurement data immediately after the fault occurrence/detection. The idea pursued in this chapter is, based on a-priori information, to appropriately select an upper δ_{max} and a lower bound δ_{min} on the possible size of the uncertainty in the fault estimates. The size of the uncertainty at each time instant is then computed as a function of the current time instant and the last detection of a fault (see Figure 6.4). The uncertainty size is in this way increased to its maximum δ_{max} after the detection of a fault, and subsequently gradually decreased as time elapses. To achieve this effect, the following function can be used

$$\delta(t) = \delta_{min} + \frac{t_A(\delta_{max} - \delta_{min})}{t_A + 9(t - t_F)}, \tag{6.45}$$

where t is the current time instant, t_F is the time instant of the last fault detec-

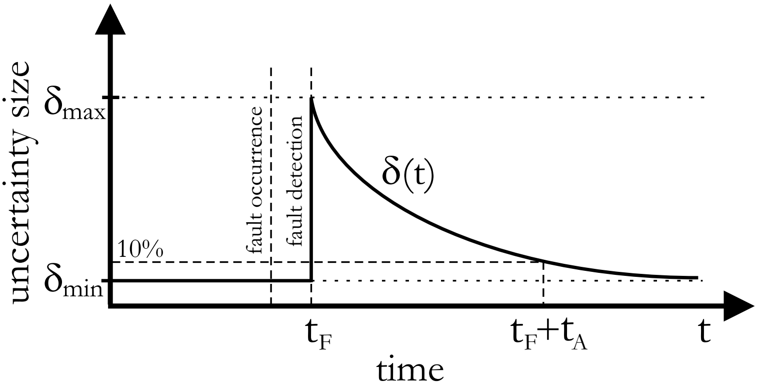


Figure 6.4: The size of the uncertainty in the fault estimates is increased after the detection of a fault by the CUSUM test, and is then gradually decreased with time.

tion, and t_A is the time needed for $\delta(t)$ to decrease from its maximum value δ_{max} to $\delta_{min} + 0.1(\delta_{max} - \delta_{min})$.

Simulation and Detection of Faults in the Encoder

The encoder that is used in the experimental setup is MAXON HEDL 55 and generates 500 pulses per rotation which are used to determine the position of the rotor (see Figure 6.5). It is a so-called incremental encoder as it outputs a number that is increased by 1 each time a pulse has been generated. The value of this number at time instant k is denoted by n_k. On the basis of this reading the angular velocity of the motor after the gearbox is computed as

$$\omega_k = \frac{N_{GB}(n_k - n_{k-1})\pi}{250T_s},$$

where T_s is the sampling time. Possible fault in the encoder is the obstruction of a number holes so that less than the nominal 500 pulses per rotation are obtained. The same effect results also from slip in the link between the rotor and the encoder. Such fault is multiplicative by nature and can be analytically represented as

$$\omega_k^M = \sigma\omega_k = \sigma\frac{N_{GB}(n_k - n_{k-1})\pi}{250T_s}$$

where $0 < \sigma < 1$ represents the gain of the multiplicative sensor fault.

This fault, however, cannot be estimated directly by making use of the parameter estimation algorithm presented above. This is due to the fact that in a SISO system one cannot differentiate between multiplicative sensor fault and multiplicative actuator fault only on the bases of the input and the output signals (see for instance (Verdult et al. 2003)). Indeed, multiplication of the output with σ results in the same input-output behavior of the system as multiplying the input with σ. For this reason in order to be able to detect also sensor faults we will

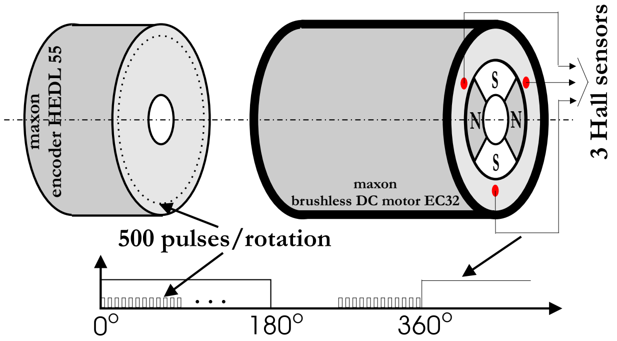


Figure 6.5: The Hall sensors in the BDCM and the encoder.

make use of an additional measurement. The idea exploited here is to estimate the scaling factor σ by estimating the number of pulses we obtain from the encoder per one rotation. This can be achieved by measuring the signal from one of the three Hall sensors mounted on the motor (see Figure 6.5), and counting the number of pulses between two fronts of the Hall sensor signal. On the basis of this we can produce an estimate $\hat{\sigma}$ of the scaling σ. Due to some synchronization issues it turns out the this estimate may deviate a bit from the real fault magnitude σ, which deviation we model as uncertainty (see equation (6.50)).

6.4.2 Algorithm for Controller Reconfiguration

Aiming to achieve demands (i)-(iii) for the controller reconfiguration posed at the beginning of Section (6.3.2) on page 152, a state-feedback controller with integral action of the form

$$\begin{aligned} u_k^{nom} &= \left[\begin{array}{cc} F_1(k) & F_2(k) \end{array} \right] \left[\begin{array}{c} \omega_k \\ x_k^I \end{array} \right], \\ x_k^I &= x_{k-1}^I + T_s(\omega_{k-1} - \omega_{k-1}^{ref}), \end{aligned} \tag{6.46}$$

will be used. In (6.46) ω_k^{ref} is the reference velocity that needs to be followed. However, while this structure is appropriate in the case of no sensor fault – whenever a fault in the encoder is present ω_k is not measured directly. Instead, only the faulty measurement ω_k^M is available together with an *estimate* of the sensor fault magnitude $\hat{\sigma}$. In the case of a partial sensor fault ($\hat{\sigma} > 0$) the following substitute for (6.46) is more appropriate

$$\begin{aligned} u_k^R &= \left[\begin{array}{cc} F_1(k) & F_2(k) \end{array} \right] \left[\begin{array}{c} \frac{\omega_k^M}{\hat{\sigma}} \\ x_k^I \end{array} \right], \\ x_k^I &= x_{k-1}^I + T_s \left(\frac{\omega_{k-1}^M}{\hat{\sigma}} - \omega_{k-1}^{ref} \right). \end{aligned} \tag{6.47}$$

Note, that a total sensor fault ($\sigma = 0$) results in an unobservable system, a situation that falls outside this chapter. Note also, that due to the saturation, the control action that is actually applied is given by

$$u_k = SAT(u_k^R) = \begin{cases} 1, & u_k^R > 1 \\ u_k^R, & 0 \le u_k^R \le 1 \\ 0, & u_k^R < 0. \end{cases} \tag{6.48}$$

Due to the fact that we use the reconstructed velocity, defined as

$$\hat{\omega}_k = \frac{\omega_k^M}{\hat{\sigma}}, \tag{6.49}$$

instead of the true velocity in the control law (6.47), it is more convenient to rewrite the model for $\hat{\omega}_k$ instead of ω_k. To this end we note that from equations (6.11), (6.12), (6.50), and (6.49) we can write

$$\begin{aligned} \hat{\omega}_{k+1} = \frac{\omega_{k+1}^M}{\hat{\sigma}} &= (1+\gamma_{f,\sigma}\Delta_\sigma)\omega_{k+1} \\ &= (1+\gamma_{f,\sigma}\Delta_\sigma)(a_f\omega_k + b_f u_k + b_{f,off}) \\ &= a_f\frac{\omega_k^M}{\hat{\sigma}} + b_f(1+\gamma_{f,\sigma}\Delta_\sigma)u_k + b_{f,off}(1+\gamma_{f,\sigma}\Delta_\sigma) \\ &= a_f\hat{\omega}_k + b_f(1+\gamma_{f,\sigma}\Delta_\sigma)u_k + b_{f,off}(1+\gamma_{f,\sigma}\Delta_\sigma). \end{aligned}$$

For simplicity of notation, the last equation will be written as

$$\hat{\omega}_{k+1} = a\hat{\omega}_k + bu_k + b_{off}.$$

It is assumed here that estimates $(\hat{a}, \hat{b}, \hat{b}_{off}, \hat{\sigma})$ of (a, b, b_{off}, σ) are provided by the FDD scheme together with the sizes $(\gamma_{f,a}, \gamma_{f,b}, \gamma_{f,off}, \delta_\sigma)$ of their corresponding uncertainty intervals, so that

$$\begin{aligned} a &= \hat{a}(1+\gamma_{f,a}\Delta_a), && \text{with } |\Delta_a| \le 1, \\ b &= \hat{b}(1+\gamma_{f,b}\Delta_b), && \text{with } |\Delta_b| \le 1, \\ b_{off} &= \hat{b}_{off}(1+\gamma_{f,off}\Delta_{off}), && \text{with } |\Delta_{off}| \le 1, \\ \sigma &= \hat{\sigma}(1+\gamma_{f,\sigma}\Delta_\sigma), && \text{with } |\Delta_\sigma| \le 1. \end{aligned} \tag{6.50}$$

It needs to be pointed out, however, that in the current implementation the uncertainty intervals $(\gamma_{f,a}, \gamma_{f,b}, \gamma_{f,off}, \gamma_{f,\sigma})$ are not provided by the current implementation of the FDD scheme. Instead, they are artificially formed using some a-priori knowledge about the tracking capabilities of the developed RLS scheme. This is further discussed in section 6.4.1.

Thus, as depicted in Figure 6.6, the design of the controller with integral action (6.47) can be achieved by means of state-feedback design for the following augmented system,

$$\begin{bmatrix} \hat{\omega}_{k+1} \\ x_{k+1}^I \end{bmatrix} = \begin{bmatrix} a & 0 \\ T_s & 1 \end{bmatrix} \begin{bmatrix} \hat{\omega}_k \\ x_k^I \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u_k + \begin{bmatrix} 0 & b_{off} \\ -T_s & 0 \end{bmatrix} \begin{bmatrix} \omega_k^{ref} \\ 1 \end{bmatrix} \tag{6.51}$$

In addition, for controller design purposes we define the controlled outputs

$$z_k = \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} = \begin{bmatrix} W_e x_k^I \\ W_u u_k \end{bmatrix}, \tag{6.52}$$

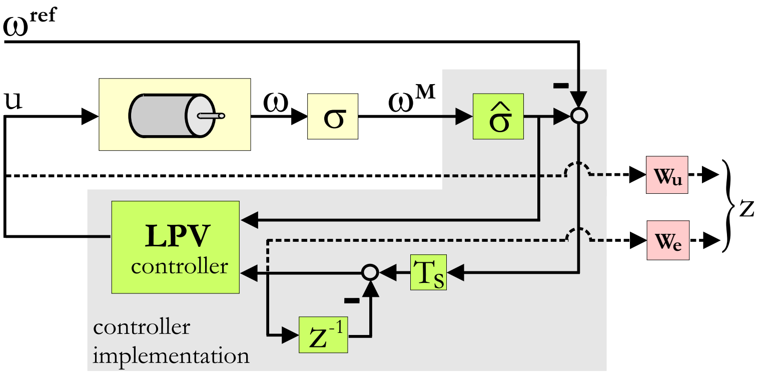


Figure 6.6: Controller implementation.

where $W_e, W_u > 0$ are scaling factors. As discussed later on in more detail, the goal is to achieve some desired (weighted) $\mathcal{H}_\infty$-norm of the closed-loop system with output z_k. By appropriately selecting the input weighting W_u one can make sure that the control action does not become too active and remains inside the saturation bounds, so that the design can be carried out neglecting the saturation (see equations (6.51) and (6.48)).

The system described by equations (6.51) and (6.52) is of the form (4.34) on page 111 with matrices

$$A^{\Delta}(f) = \begin{bmatrix} a(\hat{a}, \Delta_a) & 0 \\ T_s & 1 \end{bmatrix}, \; B_u^{\Delta}(f) = \begin{bmatrix} b(\hat{b}, \Delta_b, \Delta_\sigma) \\ 0 \end{bmatrix},$$

$$B_\xi^{\Delta}(f) = \begin{bmatrix} 0 & b_{off}(\hat{b}_{off}, \Delta_{off}) \\ -T_s & 0 \end{bmatrix},$$

$$C_z^{\Delta}(f) = \begin{bmatrix} 0 & W_e \\ 0 & 0 \end{bmatrix}, \; D_{zu}^{\Delta}(f) = \begin{bmatrix} 0 \\ W_u \end{bmatrix}, \; D_{z\xi}^{\Delta}(f) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

where the matrices a, b, and b_{off} are defined in equation (6.50). The remaining matrices $C_y^{\Delta}(f)$, $D_{yu}^{\Delta}(f)$, and $C_{y\xi}^{\Delta}(f)$ are of no importance in the sequel since only the state-feedback case is considered here.

The fault signal f can be taken of the form

$$f = \begin{bmatrix} a \\ b \\ b_{off} \end{bmatrix}$$

so that its estimate $\hat{f} = \begin{bmatrix} \hat{a} & \hat{b} & \hat{b}_{off} \end{bmatrix}$ provided by the FDD part of the algorithm is such that in view of (6.50) the following holds

$$f = (I + \mathbf{diag}(\gamma_f(k))\hat{\Delta})\hat{f},$$

where the FDD uncertainty is denoted as

$$\hat{\Delta} \doteq \mathbf{diag}(\Delta_a, \Delta_b, \Delta_{off}),$$

and its “size” as

$$\gamma_f \doteq [\gamma_{f,a},\ \gamma_{f,b},\ \gamma_{f,off}]^T.$$

Note that the sensor fault signal σ is not included in the fault vector f. The reason for that the estimate $\hat{\sigma}$ will not be a scheduling parameter for the LPV controller designed later on; instead, $\hat{\sigma}$ is used to compute the reconstructed velocity $\hat{\omega}_k$, i.e. the first state of the considered system (6.51). Note that in the final implementation of the controller, as depicted on Figure 6.6, it will depend directly on $\hat{\sigma}$ although the LPV controller itself does not depend on it.

In this way a state-feedback LPV controller is to be designed by solving the optimization problem (4.39) on page 112 (or, equivalently, (4.39)) where the splitting of the controlled output z_k in equation (4.36) on page 112 is defined for the BDCM in equation 6.52. In this way the integrated tracking error x_k^I can be minimized subject to a constraint on the control action u_k that aims to reduce the effect of the saturation on the closed-loop system.

In this way the results from Section 4.3 can directly be used for the design of the LPV controller. The final FTC controller can then be implemented as shown in Figure 6.6.

6.5 Experimental results

This section presents some results obtained on the BDCM experimental setup explained in Section 6.2.

Nominal model and fault scenario

For the fault-free system a linear discrete-time state-space model has been identified from input/output data using the Subspace Model Identification technique (Verhaegen 1994). This model has the form

$$\omega_{k+1} = 0.9644\omega_k + 1.265u_k - 0.0891, \tag{6.53}$$

and is only used to initialize the parameter estimate procedure summarized in Algorithm 6.1.

In the experiment presented here two hardware faults are introduced:

- increased resistance of one coil by 3Ω (coil fault) is introduced at time instant $t = 12.19,\ [sec]$.
- a 50% encoder fault is introduced at time instant $t = 16.15,\ [sec]$.

Both faults remain active until the end of the experiment.

The effect of the encoder fault has been discussed in more detail in section 6.4.1. The effect of the coil fault can be seen as reduced input voltage V_D during $2/3$ of each turn of the rotor, namely during the interval of time in which the faulty coil is conducting. This, in turn, leads to a decreased angular velocity during these periods so that as a result fluctuations in the angular velocity of the motor are observed. The amplitude of these fluctuations increases with the increase of the resistance of the faulty coil. The frequency of the fluctuations is proportional to the angular velocity of the rotor.

Robust Active FTC

The goal here is to control the angular velocity in such a way that (both in the fault-free and in the faulty case) it can track a piecewise-constant reference trajectory with a settling time of less than $2\ sec$. To this end, integral control law is implemented as described in Section 6.3.2, where the design of the controller gains $F_1(k)$ and $F_2(k)$ is performed by making use of the probabilistic approach proposed in Section 6.4.2. In order to meet the design specifications, in the optimization problem (4.40) on page 113 the controlled output z_k is selected as in equation (6.52) with $W_e = 1$ and $W_u = 5$.

After performing some initial experiments it became clear that the third motor parameter, b_{off}, remains very small and does not undergo significant changes after faults. Indeed, this parameter is related to the load on the motor which is not affected by faults[1]. For that reason the LPV controller is made independent on $\hat{b}_{off}$ by selecting its matrices $F_1(k)$ and $F_2(k)$ (see equation (6.47) on page 158) with the following structure

$$\begin{aligned}
F_1(\hat{f}(k), \gamma_f(k)) &= F_{1,0} + F_{1,1}\hat{a}(k) + F_{1,2}\hat{b}(k) + F_{1,3}\gamma_{f,a}(k) + F_{1,4}\gamma_{f,4}(k), \\
F_2(\hat{f}(k), \gamma_f(k)) &= F_{2,0} + F_{2,1}\hat{a}(k) + F_{2,2}\hat{b}(k) + F_{2,3}\gamma_{f,a}(k) + F_{2,4}\gamma_{f,4}(k).
\end{aligned}$$

This is in the form (4.42) on page 114 where the functions $g_i(\cdot)$ are selected to be affinely dependent on the fault estimates (namely the estimates of the state-space matrices a and b of the BDCM) and the size of the uncertainty of these estimates.

Therefore, for the LPV design the fault signal f in the generalized faulty system representation (4.34) is taken as

$$f = \begin{bmatrix} a \\ b \end{bmatrix},$$

and it is assumed that $f \in \mathcal{F}$ where

$$\mathcal{F} = \left\{ f : \begin{bmatrix} 0.95 \\ 0.1 \end{bmatrix} \leq f \leq \begin{bmatrix} 0.99 \\ 2.5 \end{bmatrix} \right\}.$$

These are realistic bounds since faults in the BDCM result on the one hand in slower dynamics $(a > a^{nom})$, and on the other hands they cannot destabilize the (open-loop) system $(1 > a > a^{nom})$. Additionally it can be seen from equation (6.10) that b decreases when R increases. Faults resulting in $b < 0.1$ are considered as total actuator faults, i.e. the system becomes practically uncontrollable. Again, such faults fall outside the scope of this chapter.

The parameter-varying FTC controller computed with the approach in Section 4.3 is as follows

$$\begin{aligned}
F_1(k) &= 0.1077 - 0.1670\hat{a}(k) + 0.0014\hat{b}(k) - 0.6778\gamma_{f,a}(k) - 0.1089\gamma_{f,b}(k) \\
F_2(k) &= -0.0192 - 0.2626\hat{a}(k) + 0.0091\hat{b}(k) - 2.4604\gamma_{f,a}(k) - 0.4145\gamma_{f,b}(k)
\end{aligned}$$

---

[1] We note here that if the load to the motor changes for some reason then the integrating action will take care to compensate for it.

Algorithm 6.1						
P_0	W	e_{max}	t_λ	λ_{min}	λ_{max}	α
I_3	diag$[1,\ 1,\ 10^{-8}]$	2	100	0.98	0.9999	0.97

Algorithm 6.2				Uncertainty computation		
W_θ	ν^U	ν^D	h	δ_{min}	δ_{max}	t_A
50	$\begin{bmatrix} 1 \\ 0.2 \\ 10 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0.2 \\ 10 \end{bmatrix}$	1	$\begin{bmatrix} 0.001 \\ 0.01 \end{bmatrix}$	$\begin{bmatrix} 0.01 \\ 0.1 \end{bmatrix}$	100

Table 6.2: Parameters in the algorithms used in the experiment.

For the purposes of comparison, a nominal PI controller was designed that achieves the control objective for the fault-free system. This PI controller has the structure given in Equation (6.46) on page 158 with

$$F_1^{PI} = -0.0214,\ F_2^{PI} = -0.0223.$$

FDD algorithm

The implemented algorithm for FDD is explained in detail in Section 6.4.1. Table 6.2 shows the values of the parameters of the complete FDD algorithm that consist of Algorithms 6.1 and 6.2 and the uncertainty size computation in equation (6.45) on page 156.

Experimental results

The results from the experiment are summarized on Figures 6.7-6.8. Figure 6.7 depicts the reference trajectory (the dotted curve), angular velocity after the gearbox with the PI controller (the dash-dotted curve) and with the robust active FTC (the solid curve). It can be seen that both controllers achieve the performance specifications in the fault-free case, i.e. up until time instant $t = 12.19$ $[sec]$ when the coil fault occurs. After this fault it can be seen that the nominal PI controller no longer satisfies the specifications as it is not capable of tracking the reference trajectory with a settling time of less than 2 sec. It should be noted, however, that due to the integral action in the PI controller the tracking error would also eventually go to zero, should the reference trajectory be left constant for a longer period of time. On the other hand the closed-loop system with the FTC controller satisfies the performance specifications after the coil fault.

It should be pointed out that after the occurrence of this first fault, the angular velocity becomes rather fluctuating. As discussed at the beginning of this section, this is not due to increased measurement noise as it looks, but as a result of the coil fault.

At time instant $t = 16.15$ sec a second fault occurs, namely a partial 50% fault in the encoder. After this fault the measured angular velocity ω_k^M is two times smaller than the real velocity that needs to be controlled ω_k. Since the nominal

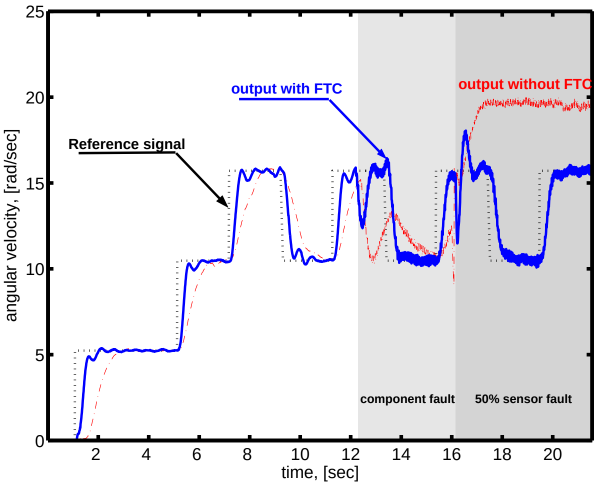


Figure 6.7: Reference trajectory (dotted), angular velocity with nominal PI controller (dash-dotted), and angular velocity with FTC (solid).

PI controller does not possess reconfiguration capabilities it tries to bring the measured angular velocity ω_k^M to the reference signal, trying to make in this way the true velocity ω_k two times higher. This, on its turn, is not possible as the maximal velocity in the presence of the coil fault is around $20\ rad/sec$. As a result the control action hits the saturation and brings the motor to its maximal speed. In comparison, the FTC is capable to reconfigure so that the true angular velocity ω_k continues to track the reference signal satisfactorily.

Figure 6.8 depicts the parameter estimates obtained by the RLS scheme in Algorithm 6.1. The parameter vector $\theta = [a,\ b,\ b_{off}]^T$ consists of the motor parameters. After the occurrence of the coil fault at time instant $t = 12.19\ [sec]$ it can be observed that the first two parameter estimates, i.e. $\hat{a}$ and $\hat{b}$, converge to their new values. The third parameter estimate is kept constant by making its corresponding weight in the RLS algorithm very small (see Table 6.2). While this is not very restricting (as the offset term is only slightly dependant on coil faults), it makes the estimates of the other parameters to converge smoothly to their new post-fault values. Note also that, as expected, the second fault (i.e. the encoder fault) does not affect the parameter estimates. Its detection and diagnosis was discussed in Section 6.4.1.

It should be pointed out that the CUSUM test detected the coil fault at time instant $t = 13.23\ sec$, thus with a detection delay of $1.04\ sec$. No false alarms were observed. Once the fault was detected, the uncertainty in the parameter estimates was increased to its assumed maximum value and then gradually decreased as explained in Section 6.4.1.

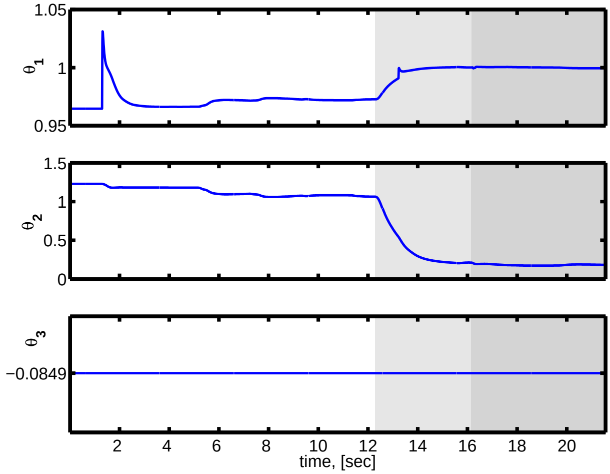


Figure 6.8: BDCM parameter estimates.

6.6 Conclusions

In this chapter a combined fault detection, diagnosis and controller reconfiguration approach was proposed for a brushless DC motor. The approach consists of two interacting schemes, an FDD and a CR scheme. To make the interconnection possible, the FDD scheme is developed to achieve fast fault detection and diagnosis and takes into consideration important issues as closed-loop operation and control action saturation. It is based on a modified weighted-RLS algorithm with adaptive forgetting factor that makes it alert for abrupt parameter changes on the one hand, and not too sensitive to process noise on the other. The CR scheme, on its turn, is designed to deal with time-varying uncertainty in the fault estimates, provided by the FDD scheme. It is based on a parameter-varying controller that is scheduled by both the fault estimates and the sizes of their uncertainties. As the later cannot be directly estimated by the FDD schemes, they are formed heuristically, leaving room for further improvement of the algorithm. The complete approach was successfully tested on an experimental setup consisting of a brushless DC motor.

7

Multiple-Model Approach to Fault-Tolerant Control

So far the attention was paid on linear systems with uncertainties, for which both passive and active approaches to FTC have been developed. In this chapter an attempt is made towards the active FTC design for nonlinear systems. The philosophy used in the method of this chapter is to represent a nonlinear system or a system with faults by means of a set of local linear models where each model corresponds to a particular operating condition of the system. The interacting multiple model estimator is utilized for reconstructing both the state of the nonlinear (faulty) system and the mode probabilities for the local models. Based on this information, a standard cost function in predictive control is optimized under the assumption that the mode probabilities remain constant in the future. The method of this chapter does not consider model uncertainties which remains a topic for future research. The algorithm is illustrated in two different case studies – one with a linear model of one joint of a space robot manipulator, subjected to faults, and one with a nonlinear model of the inverted pendulum on a cart.

7.1 Introduction

Modern control systems are becoming increasingly complex with more and more demanding performance goals. These complex systems must have the capability for fault accommodation in order to operate successfully over long periods of time. Such systems require fault detection, isolation and controller reconfiguration so as to maintain adequate levels of performance with one or more sensor, actuator, and/or component faults, or a combination of these events. The controller reconfiguration technique that is developed in this chapter, though applicable to a general nonlinear system, is very suitable for the control of systems subject to faults, since such systems are (naturally) represented by a set of models (Athans et al. 1977; Maybeck and Stevens 1991; Griffin and Maybeck 1997; Zhang and Li 1998).

When dealing with sensor, actuator and component faults, a *hybrid dynamic model* can be used. The hybrid system is also known as jump linear system: it is linear given the system mode; however it may jump from one such system mode to another at a random time. Such systems can be used to model situations where the system behavior undergoes abrupt changes, such as system faults (Zhang and Li 1998). The hybrid dynamic model (Griffin and Maybeck 1997; Zhang and Li 1998) consists of a set of discrete-time linear models and a switching logic, determining the switching between these models. The switching between models in the hybrid systems is a consequence of factors, such as faults in its sensors, actuators and components. Different methods for the control of hybrid systems have been proposed in the literature. In (Zhang and Jiang 1999b; Campo et al. 1996) an Interacting Multiple Model based control was utilized, a neural adaptive controller is presented in (McDowell et al. 1997), Multiple-Model Adaptive Control (MMAC) is also an important class of control methods with application to the control of jump linear systems (Athans et al. 1977; Griffin and Maybeck 1997; Naredra and Balakrishnan 1997), an algorithm based on the Generalized Pseudo-Bayessian method is given in (Watanabe and Tzafestas 1989). The optimal control of hybrid systems have also been addressed in the literature (Griffiths and Loparo 1995).

A similar model representation might also be used to represent a nonlinear dynamic system when approximating it by a *piecewise linear system* (PWL system). In the piecewise linear system description it is assumed that the state-space $\mathcal{X}$ is divided into regions $\mathcal{X}_i$. Linear dynamics are associated with each such region of the state space. Thus, the PWL system is again described by a set of local models. The reader, who is interested in PWL systems is referred to (Johansson 1999; Johansson and Rantzer 1998; Rantzer and Johansson 1997). In this chapter the hybrid systems and the piecewise linear systems will be treated in a unified manner, as in (Athans et al. 1977; Fabri and Kadirkamanathan 1998b; Johansson 1999; Fabri and Kadirkamanathan 1998a).

The switching logic of the hybrid dynamic system representation in this chapter is determined by the Interacting Multiple Model (IMM) estimator, adopted from e.g. (Zhang and Li 1998; Griffin and Maybeck 1997; Blom and Bar-Shalom 1988; Li 1996), and extended to the case of systems with offset in the state and output. In this approach, the switching logic corresponds to a set of real numbers that determine the *convex* combination of the models in the model set that is valid at a particular time instant, i.e. the convex combination of the states of the local models that represents the system state.

In the classical approach of gain scheduling using local models (Fabri and Kadirkamanathan 1998a; Hunt and Johansen 1997) pre-designed controllers are activated when a detection mechanism detects that one (and only one) model from the model set is active at a particular time instant. When making use of this approach in fault tolerant control, each fault condition should be represented by a single model. Thus, the set of models will very quickly grow unboundedly as a consequence of having to model every (partial or total) faulty condition. This problem is avoided in this chapter by letting the state of the actual system be approximated by a convex combination of the states of the local models. In (Griffin and Maybeck 1997) the authors propose a moving bank of filters to reduce the number of active models and the computational burden, i.e. only the models in close proximity to the model of the "real" system are activated.

In this chapter it is assumed that the nonlinear system is represented by a convex combination of a set of linear discrete-time models $\mathcal{M} = \{M_1, \ldots, M_N\}$. This set will be called *the model set.* The decision about which convex combination of these models is in effect at the current moment of time is made by the Interacting Multiple Model estimator. It runs a bank of Kalman filters in parallel, each based on a particular local model from the model set $\mathcal{M}$. It calculates the probability of each mode to be in effect. The overall state estimate is then computed as a convex combination of the state estimates obtained from the different Kalman filters. A model of the system that is assumed to be currently in effect is also constructed as a convex combination of the local models in the model set. A bank of Generalized Predictive controllers (GPC) is designed, each corresponding to one of these local models. The optimal GPC control law for the model that is assumed to be in effect is calculated at each sample to minimize a standard cost function. This optimal control is *not* a convex combination of the local GPC control laws, optimized for each individual model in the model set.

In (Maybeck and Stevens 1991) the authors have applied a multiple model adaptive control to a STOL F-15 aircraft. They combine a non-interacting MM algorithm with a bank of LQG controllers, each designed for one particular model. The overall control action is computed as a convex combination of the outputs of the different controllers, i.e.

$$u(k) = \sum_{i=1}^{N} u_i(\hat{x}_i(k))\mu_i(k)$$

where $u(k)$ is the overall control action, $u_i(\hat{x}_i(k))$ is the output of the i-th LQG controller (dependent on the state estimate of the i-th Kalman filter, $\hat{x}_i(k)$) and $\mu_i(k)$ is the probability that model M_i is in effect at the moment of time k. However, although such a mixing of the controller outputs seems reasonable and intuitive, it does not guarantee optimality of the performance objective used in the design of the local controllers *when the model in effect is not contained in the model set.* A similar approach was followed in (Athans et al. 1977). An illustration will be presented to show that in the case of unanticipated faults the closed-loop stability can no longer be guaranteed by such a control action.

The remaining part of this chapter is organized as follows. In Section 7.2 the general descriptions of the hybrid dynamic system and the multiple-model representation of a nonlinear system are summarized. Some issues are also given on the model set design. Section 7.3 gives an overview of the interacting multiple model estimator, extended to the case when the model set consists of systems with offset, and makes some comments on the design of the transition probability matrix. In Section 7.4 the controller reconfiguration scheme is outlined by presenting a predictive controller strategy for a set of models. An illustration of this approach is made in Section 7.5 by means of two realistic simulation studies, one with a linear model of one joint of a space robot manipulator (SRM), and one with a nonlinear model of the inverted pendulum on a cart. Finally, Section 7.6 is dedicated to some concluding remarks.

7.2 The model set

In this section two classes of models will be presented that can be treated using the approach developed in what follows. These classes are the hybrid dynamic system and the piecewise linear system. Both systems are based on a set of local linear models and as such can be treated in a unified framework. Attention will also be paid on the problem of selecting the set of local models.

7.2.1 Hybrid dynamic model

A hybrid dynamic system can be described as one with both a continuously-valued base state and discretely-valued structural (parametric) uncertainty (Li 1996). A typical example of such a system is one subject to faults since fault modes are structurally different from each other and from the nominal mode. By *mode* a structure or behavior pattern of the system is meant.

Assume that the actual system at any time can be modelled sufficiently accurately by a stochastic hybrid system (Griffin and Maybeck 1997):

$$H : \left\{ \begin{array}{rcl} x(k+1) & = & A(k,m(k+1))x(k) + B(k,m(k+1))u(k) + \\ & & T(k,m(k+1))\xi(k,m(k+1)), \\ y(k) & = & C_y(k,m(k))x(k) + \eta(k,m(k)), \end{array} \right. \tag{7.1}$$

with the system mode sequence $m(k)$ assumed to be a first order Markov chain with transition probabilities

$$P\{m_j(k+1)|m_i(k)\} = \pi_{ij}(k), \qquad \forall m_i, m_j \in \mathcal{I}$$

where $x \in \mathbb{R}^n$ is the state vector; $y \in \mathbb{R}^p$ is the measured output of the system; $u \in \mathbb{R}^m$ is the control input; $\xi \in \mathbb{R}^{n_\xi}$ and $\eta \in \mathbb{R}^p$ are independent identically distributed discrete-time process and measurement noises with means $\bar{\xi}(k)$ and $\bar{\eta}(k)$, and covariances $Q(k)$ and $R(k)$; $m(k)$ is a discrete-valued *modal state,* i.e. the index of the normal or fault mode, at time k, which denotes the mode in effect during the sampling period ending at k. $\mathcal{I} = \{m_1, m_2, \ldots, m_N\}$ is the set of all possible system modes. $\pi_{ij}(k)$ is the *transition probability* from mode m_i

to mode m_j, i.e. the probability that the system will jump to mode m_j at time instant $(k+1)$ provided that it is in mode m_i at time instant k. Obviously, the following relation must hold for any $m_i \in \mathcal{M}$

$$\sum_{j=1}^{N} \pi_{ij}(k) = \sum_{j=1}^{N} P\{m_j(k+1)|m_i(k)\} = 1.$$

This means that the probability that the system will remain in its current mode of operation plus the probability that it will jump to another mode must be equal to one.

It can be seen from (7.1) that the mode information is embedded (i.e., not directly measured) in the measurement sequence $y(k)$.

The hybrid dynamic model (7.1) is very useful for representation of systems in which a certain (predefined) set of anticipated faults is assumed possible to occur. For such systems one may design a local model for each anticipated faulty mode of operation of the system.

7.2.2 Nonlinear system

Consider the nonlinear system

$$\begin{cases} x(k+1) &= f(x,u,k) \\ y(k) &= g(x,u,k) \end{cases}$$

One way of dealing with nonlinear systems is by means of approximating them with local linear models, derived through linearization of the nonlinear system around different operating points. The dynamics within each local region $\mathcal{X}_i$ is affine in the state vector x, i.e.

$$H_i : \begin{cases} x(k+1) &= A_i x(k) + a_i + B_i u(k), \\ y(k) &= C_{y,i} x(k) + c_{y,i} + D_{y,i} u(k), \end{cases} \quad \text{for} \quad x(k) \in \mathcal{X}_i. \tag{7.2}$$

The idea exploited in the chapter is then to represent the output of the nonlinear systems outside of these pre-specified regions as a weighted combination of the outputs of the local models.

In this chapter, the models in *the model set* $\mathcal{M} = \{M_1, \ldots, M_N\}$ will be described by a collection of N local models

$$M_i : \begin{cases} x_i(k+1) &= A_i(k)x_i(k) + a_i(k) + B_i(k)u(k) + T_i(k)\xi_i(k), \\ y(k) &= C_{y,i}(k)x_i(k) + c_{y,i}(k) + \eta_i(k), \end{cases} \tag{7.3}$$

so that both the hybrid model in equation (7.1) on page 170 and the local linear models (7.2) are covered by (7.3). The process and measurement noise are normally distributed random processes

$$\xi_i(k) \sim \mathcal{N}(\hat{\xi}_i, Q_i(k)),$$
$$\eta_i(k) \sim \mathcal{N}(\hat{\eta}_i, R_i(k)).$$

The matrices A_i, B_i, T_i and $C_{y,i}$ may all be different for different i.

7.2.3 The model set design

The model set design is highly dependent on the particular application considered. However, there are some common features that have to be taken into account. For example, there should be enough separation (distance) between models so that they are identifiable by the IMM estimator. This separation should exhibit itself well in the measurement residuals. Otherwise the IMM estimator will not be very selective in terms of correct fault detection since it is the measurement residuals that have the most dominant effect on the mode probability computation which in turn affects the accuracy of the overall state estimates. On the other hand, if the separation is too large, numerical problems may occur (Zhang and Li 1998). The distances between the models should be measured in closed-loop because it is in closed-loop that the IMM estimator will be used. For example, one possible measure for the separation between two models, $M_1(z)$ and $M_2(z)$, is the $\mathcal{H}_\infty$ norm of the discrepancy between the corresponding closed-loop systems, $M_{1,CL}(z)$ and $M_{2,CL}(z)$, i.e. $\|M_{1,CL}(z) - M_{2,CL}(z)\|_\infty$. Another possible way to define distances between models is the gap-metric (Vinnicombe 1999). Still, the question of how to select the models in the model set remains unanswered.

If systems subject to faults are considered, total actuator faults may be modelled by making zero(s) the appropriate column(s) of the B matrix. For total sensor faults one needs to annihilate the appropriate row(s) of the C_y and c_y matrices. Partial actuator or sensor faults are modelled by multiplying the appropriate column (row) of the B (or C_y and c_y) matrix by a scaling factor. For example, a partial 40% sensor fault is modelled by multiplying the corresponding row of the C_y and c_y matrices by 0.4. To prevent ambiguity, note that in this way 100% fault means no fault at all, and that a 0% fault is a total fault. Note also, that sensor faults affect the offset c_y on the output equation, while actuator faults do not affect the offset a on the state equation.

However, although sensor and actuator faults can be represented in this manner, the problem of which particular fault conditions should be selected to form a “good” model set still stands. Often in practise it turns out reasonable to select the models in the model set to correspond to *total* faults, or to 5-15% partial faults, since in this case the convex combination of the models would cover a greater set of possible faulty models. If, for example, one wants to be able to represent all sensor (actuator) faults in the interval $[10\%, 100\%]$, one should build up a model that describes the system with the 10% sensor (actuator) fault (in addition to the nominal model). Also such a selection has the potential to make the distance between the models not too small.

Since there currently exists no systematic procedure for the choice of $\mathcal{M}$, in this chapter it will be assumed that the model set is given.

7.3 The IMM estimator for systems with offset

This section will briefly summarize the IMM estimator (Zhang and Li 1998; Griffin and Maybeck 1997) that basically consists of a set of Kalman filter, the i-th Kalman filter designed for the i-th local model M_i represented by equation (7.3) on page 171. The offsets $a_i(k)$ and $c_i(k)$ do not change the Kalman filters (as well as the IMM estimator) since they are additive to $T_i(k)\bar{\xi}_i(k)$ and $\bar{\eta}_i(k)$, respectively. Thus, in the original setting of the IMM estimator (Zhang and Li 1998), the offsets in the state $a_i(k)$ and in the output $c_i(k)$ should be added to the offsets resulting from the mean values of the process and measurement noises, $T_i(k)\bar{\xi}_i(k)$ and $\bar{\eta}_i(k)$, in order to take them into account.

The better performance of the IMM estimator over other multiple-model estimators is mostly due to the way the local Kalman filters are re-initialized at each time instant. In the first step of the IMM estimator a model-conditional re-initialization of the filters is performed. At time instant k the initial state estimate $\hat{x}_j^0(k-1|k-1)$ and covariance $P_j^0(k-1|k-1)$ of the j-th filter are computed using the estimates of all filters at the previous time instant $k-1$ under the assumption that mode j is in effect at time instant j, i.e. (Zhang and Li 1998)

$$\begin{aligned} \hat{x}_j^0(k-1|k-1) &= E\{x(k-1)|\, \{y_t\}_0^{k-1},\, m_j(k)\} \\ P_j^0(k-1|k-1) &= E\{\hat{x}_j^0(k-1|k-1)\}. \end{aligned}$$

In this way the Kalman filters are interacting with each other and are not running individually (as is the case with the noninteracting MM estimators).

At the second step the individual Kalman filters are run in parallel. The mode probability is subsequently updated in the third step using model-conditional likelihood functions. Finally, in the fourth step the overall state estimate and its covariance are computed by means of a probabilistically weighted sum of the local state estimates and covariances of the Kalman filters. Note that the inherent parallel structure of the IMM estimator makes it very attractive for parallel processing. For more details on the IMM estimator the reader is referred to (Zhang and Li 1998) and the references therein.

Table 7.1 presents a complete cycle of the IMM estimator with Kalman filters.

The design parameters of the IMM algorithm are the transition probability matrix and the model set. Note that the performance of the IMM estimator depends also on the type and magnitude of control input excitation used. However, the design of the transition probability matrix π is very important since the sensitivity of the mode probabilities $\mu_i(k)$ with respect to π is very high.

A recommended choice of the diagonal entries in the transition probability matrix is to match roughly the mean sojourn time of each mode:

$$\pi_{ii} = max\left\{l_i, 1 - \frac{T}{\tau_i}\right\}$$

where τ_i is the expected sojourn time of the i-th mode; T is the sampling interval; l_i is a designed limit of the transition probability of the i-th mode to itself. For example, the "normal-to-normal" transition probability can be obtained by $\pi_{11} = 1 - T/\tau_1$, where τ_1 denotes the mean time between faults, which in practice, is significantly greater than T.

7.4 The MM-based GPC

In this section it will be shown how the GPC, adopted from (Kinnaert 1989), can be extended to the case of systems with offset and combined with the IMM esti-

1. Mixing of the Estimates (for $j = 1, \ldots, N$):	
predicted mode prob.:	$\mu_j(k\|k-1) = \sum_{i=1}^{N} \pi_{ij}\mu_i(k-1)$
mixing probability:	$\mu_{i\|j}(k-1) = \pi_{ij}\mu_i(k-1)/\mu_j(k\|k-1)$
mixing estimate:	$\hat{x}_j^0(k-1\|k-1) = \sum_{i=1}^{N} \mu_{i\|j}(k-1)\hat{x}_i(k-1\|k-1)$
predicted errors:	$e_{i\|j}(k-1) = \hat{x}_j^0(k-1\|k-1) - \hat{x}_i(k-1\|k-1)$
mixing covariance:	$P_j^0(k-1\|k-1) = \sum_{i=1}^{N} \mu_{i\|j}(k-1)\left(P_i(k-1\|k-1) + e_{i\|j}(k-1)e_{i\|j}^T(k-1)\right)$
2. Model-Conditional Filtering (for $j = 1, \ldots, N$):	
predicted state:	$\hat{x}_j(k\|k-1) = A_j(k-1)\hat{x}_j^0(k-1\|k-1) + a_i(k-1) + B_j(k-1)u(k-1) + T_j(k-1)\bar{\xi}_j(k-1)$
predicted covariance:	$P_j(k\|k-1) = A_j(k-1)P_j^0(k-1\|k-1)A_j(k-1)^T + T_j(k-1)Q_j(k-1)T_j(k-1)^T$
measurement residual:	$\nu_j(k) = y(k) - C_{y,j}(k)\hat{x}_j(k\|k-1) - c_{y,j}(k) - \bar{\eta}_j(k)$
residual covariance:	$S_j(k) = C_{y,j}(k)P_j(k\|k-1)C_{y,j}^T(k) + R_j(k)$
filter gain:	$K_j(k) = P_j(k\|k-1)C_{y,j}^T(k)S_j^{-1}(k)$
updated state:	$\hat{x}_j(k\|k) = \hat{x}_j(k\|k-1) + K_j(k)\nu_j(k)$
updated covariance:	$P_j(k\|k) = P_j(k\|k-1) - K_j(k)S_j(k)K_j^T(k)$
3. Mode Probability Update (for $j = 1, \ldots, N$):	
likelihood function:	$L_j(k) = \frac{1}{\sqrt{\|2\pi S_j(k)\|}} exp[-\frac{1}{2}\nu_j^T(k)S_j^{-1}(k)\nu_j(k)]$
mode probability:	$\mu_j(k) = \frac{\mu_j(k\|k-1)L_j(k)}{\sum_{i=1}^{N} \mu_i(k\|k-1)L_i(k)}$
4. Combination of Estimates:	
overall state estimate:	$\hat{x}(k\|k) = \sum_{i=1}^{N} \mu_i(k)\hat{x}_i(k\|k)$
local estimation errors:	$e_i(k) = \hat{x}(k\|k) - \hat{x}_i(k\|k)$
overall covariance:	$P(k\|k) = \sum_{i=1}^{N} \mu_i(k)(P_i(k\|k) + e_i(k)e_i^T(k))$

Table 7.1: One cycle of the IMM estimator for systems with offset.

mator to yield a technique for control of nonlinear systems.

7.4.1 The GPC for systems with offset

First, the optimal GPC for systems with offset will be derived. In this section, in order to significantly simplify the expressions that will follow, the state space matrices in the local models of the model set $\mathcal{M}$ will be considered as time-invariant. Similar results can be derived for the general case when the matrices are (known) functions of k.

Consider a state-space model in the innovation form:

$$\tilde{M}: \left\{ \begin{array}{rcl} \tilde{x}(k+1) & = & \tilde{A}\tilde{x}(k) + \tilde{a} + \tilde{B}u(k) + \tilde{K}e(k), \\ y(k) & = & \tilde{C}_y\tilde{x}(k) + \tilde{c}_y + e(k), \end{array} \right. \quad \tilde{M} \in \mathcal{M}. \tag{7.4}$$

where $e(k) = y(k) - \tilde{C}_y\tilde{x}(k) - \tilde{c}_y$ is the innovation sequence, and $\tilde{K}$ is the gain of the Kalman filter that corresponds to model $\tilde{M}$.

Consider the filter F given in state-space by

$$F: \left\{ \begin{array}{rcl} x_F(k+1) & = & A_F x_F(k) + B_F y(k) \\ z(k) & = & C_F x_F(k) + D_F y(k) \end{array} \right. \tag{7.5}$$

where $z(k) \in \mathbb{R}^{n_z}$ is a vector of (filtered) output signals that will be referred to as *the controlled outputs.*

The augmented system is then obtained by combining (7.4) and (7.5):

$$\tilde{S}: \left\{ \begin{array}{rcl} \hat{x}^a(k+1) & = & A\hat{x}^a(k) + a + Bu(k) + Ke(k) \\ z(k) & = & C_z\hat{x}^a(k) + c_z + D_z e(k) \end{array} \right. \tag{7.6}$$

where $\hat{x}^a(k) = \left[\begin{array}{cc} \tilde{x}^T(k) & x_F^T(k) \end{array} \right]^T$ is the augmented state, and

$$A = \left[\begin{array}{cc} \tilde{A} & 0 \\ B_F\tilde{C}_y & A_F \end{array} \right], \quad a = \left[\begin{array}{c} \tilde{a} \\ 0 \end{array} \right], \quad B = \left[\begin{array}{c} \tilde{B} \\ 0 \end{array} \right], \quad K = \left[\begin{array}{c} \tilde{K} \\ 0 \end{array} \right]$$

$$C_z = \left[\begin{array}{cc} D_F\tilde{C}_y & C_F \end{array} \right], \quad c_z = D_F\tilde{c}_y, \quad D_z = D_F$$

Now, let $\hat{x}^a(k+j|k)$ be defined as the estimate of $\hat{x}^a(k+j)$ made at time instant k, i.e. given the input/output data up to time instant k. Then following result holds.

Theorem 7.1 (j-step ahead predictor for systems with offset) *Consider the augmented system with offset (7.6). An unbiased prediction given the input/output measurements up to time instant* k *is*

$$\hat{x}^a(k+j|k) = A^j\hat{x}^a(k) + \sum_{i=0}^{j-1} A^{j-1-i}(Bu(k+i) + a) + A^{j-1}Ke(k) \tag{7.7}$$

An unbiased prediction of the filtered output signal is

$$\begin{array}{rcl} \hat{z}(k+j|k) & = & C_zA^j\hat{x}^a(k) + \\ & & \sum_{i=0}^{j-1} C_zA^{j-1-i}(Bu(k+i) + a) + C_zA^{j-1}Ke(k) + c_z. \end{array} \tag{7.8}$$

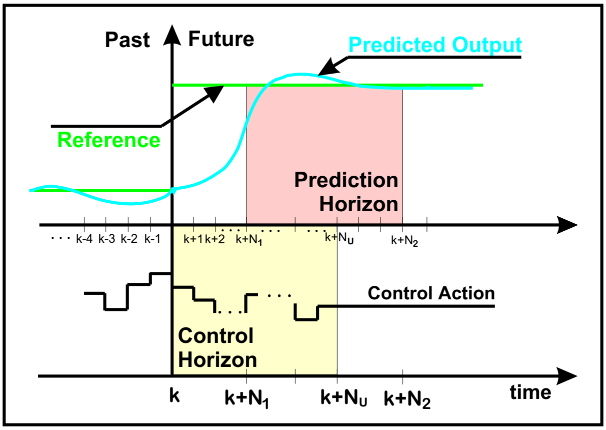


Figure 7.1: The GPC tries to match the predicted output in the prediction horizon to the reference trajectory signal while at the same time minimizing the "energy" (i.e. the control action) during the control horizon.

Proof: It can be written that

$$\begin{aligned}
\hat{x}^a(k+j|k) &= E\{\hat{x}^a(k+j)\} \\
&= E\{A\hat{x}^a(k+j-1)+a+Bu(k+j-1)+Ke(k+j-1)\} \\
&= E\{A^2\hat{x}^a(k+j-2)+Aa+ABu(k+j-2)+ \\
&\quad AKe(k+j-2)+a+Bu(k+j-1)+Ke(k+j-1)\} \\
&= E\left\{A^j\hat{x}^a(k)+\sum_{i=0}^{j-1}A^{j-1-i}\left(Bu(k+i)+Ke(k+i)+a\right)\right\}. \\
&= A^j\hat{x}^a(k)+\sum_{i=0}^{j-1}A^{j-1-i}\left(Bu(k+i)+KE\{e(k+i)\}+a\right).
\end{aligned}$$

Since the innovation $e(k+i)$ is not known for $i \geq 1$, but is white noise, $E\{e(k+i)\} = 0$ for $i \geq 1$, so that equation (7.7) follows.

Equation (7.8) then follows directly by observing that (for $j \geq 1$)

$$\begin{aligned}
\hat{z}(k+j|k) &= E\{C_z\hat{x}^a(k+j)+c_z+D_ze(k+j)\} \\
&= C_z\hat{x}^a(k+j|k)+c_z.
\end{aligned}$$

where substitution of equation (7.7) completes the proof. □

The basic idea behind the Predictive Control is visualized on Figure 7.1 and can be summarized as follows. Given a desired reference trajectory signal $\omega(k)$ the GPC tries to minimize a weighted norm of the difference between the *predicted* output and the reference trajectory in a future interval of time called the *prediction horizon*, while at the same time trying to minimize the control action in another future interval of time called *control horizon*. To achieve this the following cost function is defined

$$J(k) \doteq \sum_{j=N_1}^{N_2} \|\hat{z}(k+j|k) - \omega(k+j)\|^2 + \sum_{j=1}^{N_u} \|u(k+j-1)\|_r^2, \tag{7.9}$$

where the integers N_1, and N_2 ($N_2 > N_1$) define the prediction horizon, and $N_u \leq N_2$ - the control horizon, and where it is denoted $\|x\|_Q = x^T Q x$, I_{n_z} is the $(n_z \times n_z)$ identity matrix, r is an $(m \times m)$ diagonal matrix $r = diag\{r_i\}$ weighting the inputs, and $\omega(k) \in \mathbb{R}^{n_z}$ is the vector of the references for each controlled (filtered) output $z(k)$. The standard assumption is imposed that that the control action remains constant after the control horizon (Clarke and Mohtadi 1989), i.e.

Assumption 7.1 $u_{k+i} = u_{k+N_u}$ *for* $i \geq N_u$.

The cost function (7.9) is to be minimized over the control signal in the control horizon. To this end the matrices are now formed

$$U(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_u-1) \end{bmatrix}, \quad \hat{Z}(k) = \begin{bmatrix} \hat{z}(k+N_1|k) \\ \hat{z}(k+N_1+1|k) \\ \vdots \\ \hat{z}(k+N_2|k) \end{bmatrix}$$

the predictive model for the filtered output for $(N_2 - N_1 + 1)$ future time instants can be written as

$$\hat{Z}(k) = \Gamma \hat{x}^a(k) + HU(k) + We(k) + O.$$

with

$$H \doteq \begin{bmatrix} C_z A^{N_1-1} B & \dots & 0 & \dots & 0 \\ C_z A^{N_1} B & \dots & C_z B & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ C_z A^{N_2-1} B & \cdots & C_z A^{N_2-N_1-1} B & \dots & C_z \sum\limits_{i=N_u}^{N_2-1} A^{N_2-i-1} B_u \end{bmatrix},$$

$$\Gamma \doteq \begin{bmatrix} C_z A^{N_1} \\ C_z A^{N_1+1} \\ \vdots \\ C_z A^{N_2} \end{bmatrix}, \; W \doteq \begin{bmatrix} C_z A^{N_1-1} K \\ C_z A^{N_1} K \\ \vdots \\ C_z A^{N_2-1} K \end{bmatrix}, \; O \doteq \begin{bmatrix} c_z + \sum\limits_{i=0}^{N_1-1} C_z A^i a \\ c_z + \sum\limits_{i=0}^{N_1} C_z A^i a \\ \vdots \\ c_z + \sum\limits_{i=0}^{N_2-1} C_z A^i a \end{bmatrix}. \tag{7.10}$$

Theorem 7.2 (The GPC control law for systems with offset) *Consider the system (7.6) and the cost function (7.9) on page 177. The optimal control law that minimizes the cost function (7.9) is given by*

$$U(k) = -(H^T H + R)^{-1} H^T (\Gamma \hat{x}^a(k) + We(k) + O - \Omega(k), \tag{7.11}$$

where the matrices H, Γ, W and O are defined in equation (7.10), and where

$$R \doteq r \otimes I_{N_u}, \quad \Omega(k) \doteq \begin{bmatrix} \omega(k+N_1) \\ \vdots \\ \omega(k+N_2) \end{bmatrix}. \tag{7.12}$$

Proof: Using the notation in equations (7.10) and (7.12), the cost function (7.9) can be rewritten in the following way

$$\begin{aligned} J &= \|\hat{Z}(k) - \Omega(k)\|^2 + \|U(k)\|_R^2 \\ &= (HU(k) + \Gamma \hat{x}^a(k) + We(k) + O - \Omega(k))^T \times \\ &\quad (HU(k) + \Gamma \hat{x}^a(k) + We(k) + O - \Omega(k)) + U(k)^T R U(k). \end{aligned}$$

Denote, for simplicity of notations,

$$Q(k) = \Gamma \hat{x}^a(k) + We(k) + O - \Omega(k), \tag{7.13}$$

so that $Q(k)$ contains signals that are independent on $U(k)$. Therefore

$$\begin{aligned} J &= (HU(k) + Q(k))^T (HU(k) + Q(k)) + U(k)^T R U(k) \\ &= U(k)^T H^T H U(k) + U(k)^T H^T Q(k) + \\ &\quad Q^T(k) H U(k) + Q^T(k) Q(k) + U(k)^T R U(k) \\ &= U(k)^T (H^T H + R) U(k) + U(k)^T H^T Q(k) + \\ &\quad Q^T(k) H U(k) + Q^T(k) Q(k). \end{aligned}$$

Taking the partial derivative of J with respect to $U(k)$ yields

$$\frac{\partial J}{\partial U(k)} = 2\left\{ (H^T H + R) U(k) + H^T Q(k) \right\}.$$

Setting the righthand side of the above equation equal to zero, solving subsequently with respect to $U(k)$, and then substituting the expression for $Q(k)$ in equation (7.13) results in equation (7.11). □

Although the control action is computed for N_u time instances ahead in the future, only the control action at the current time instant is implemented

$$u(k) = \begin{bmatrix} I_m, & 0 \end{bmatrix} U(k). \tag{7.14}$$

7.4.2 The combination of the GPC with the IMM estimator

Next, the MM-based GPC for systems with offset will be derived. It is based on a combination of the IMM estimator and the GPC, both for systems with offset.

Consider the model set of augmented systems $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$, where each S_i results after taking $\tilde{M} = M_i \in \mathcal{M}$ in equation (7.4) on page 175 and augmenting the resulting model with the filter F in equation (7.5) on page 175. The state of the i-th augmented model S_i is denoted as $\hat{x}_i^a(k) = [\hat{x}_i^T(k|k),\ x_F^T(k)]^T$. Let also

$$\hat{Z}_i(k) = H_i U(k) + \Gamma_i \hat{x}_i^a(k) + W_i e_i(k) + O_i$$

be the corresponding predictive model of the filtered output. The matrices H_i, Γ_i, W_i, and O_i can then be obtained from equation (7.10) on page 177 after making the substitution

$$(A, a, B, K, C_z, c_z) \longleftarrow (A_i, a_i, B_i, K_i, C_{z,i}, c_{z,i}).$$

The innovation sequences are here similarly defined as

$$e_i(k) = z(k) - C_{y,i}\hat{x}_i(k|k) - c_{y,i}.$$

We remind that, for the sake of simplifying the expressions that follow, it is assumed that the matrices $(A_i, a_i, B_i, K_i, C_{z,i}, c_{z,i})$ time-invariant. Still, the results can easily be extended for the case when the matrices are *known* functions of the time instant k as in (7.3) on page 171.

Assumed that the "true system" can be represented accurately enough as a convex combination of the models in the model set, i.e.

$$S = \sum_{i=1}^{N} \mu_i S_i,\ \mu_i \in \mathbb{R} \tag{7.15}$$

with

$$\sum_{i=1}^{N} \mu_i = 1,\ \mu_i \geq 0.$$

Thus, we define the extended system (in innovation form)

$$\begin{aligned} x^e(k+1) &= Ax^e(k) + a + Bu(k) + Ke^e(k), \\ z(k) &= Cx^e(k) + c + De^e(k), \end{aligned}$$

where it is denoted

$$A = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_N \end{bmatrix},\ a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix},\ B = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix}, \tag{7.16}$$

$$C = \begin{bmatrix} \mu_1 C_{z,1}, & \mu_2 C_{z,2}, & \ldots & \mu_N C_{z,N} \end{bmatrix},\ c = \sum_{i=1}^{N} \mu_i c_{z,i},$$

$$D = \begin{bmatrix} \mu_1 D_{z,1}, & \mu_2 D_{z,2}, \ldots \mu_N D_{z,N} \end{bmatrix},\ K = \mathbf{diag}(K_1, \ldots, K_N),$$

and where the state and the innovation vectors for the system S are given by

$$\begin{aligned} x^e(k) &= [(\hat{x}_1^a(k))^T, \ldots, (\hat{x}_N^a(k))^T]^T, \\ e^e(k) &= [e_1^T(k), \ldots, e_N^T(k)]^T. \end{aligned}$$

In this way the *outputs* of the local models are blended. One may prefer to form a mixing of both the states and the outputs instead. One could then follow similar lines of reasoning to derive the corresponding control action.

Note that this model is “fictitious” – it is not used for estimation but only as a way to represent the real system using the local state estimates and innovations from the IMM algorithm.

Since the GPC is based on a prediction of the controlled output in the prediction horizon, the assumption is imposed that the weights $\mu_i(k)$ in the convex combination (7.15) remain unchanged during the prediction horizon.

Assumption 7.2 *the mode probabilities do not change over the maximum costing horizon, i.e.* $\mu_i(k+j) = \mu_i(k)$, $\forall j \leq N_2$.

When dealing with faults this assumption is in practise not very restrictive since faults are events that occur rarely, so that the weights can be expected to change only once in large intervals of time.

Lemma 7.1 (The MM-based GPC for systems with offset) *Consider the system S with state space matrices given by (7.16), and with state reconstructed by the IMM estimator (given in Table 7.1). Then under Assumption 7.2 the predictive model for the augmented system (7.15) is*

$$\hat{Z}(k) = \sum_{i=1}^{N} \mu_i(k) \hat{Z}_i(k),$$

and the cost function (7.9) on page 177 for the augmented system S (7.16) is minimized by

$$\begin{aligned} U(k,\mu) &= -(H^T(\mu)H(\mu) + R)^{-1} H(\mu)^T \times \\ &\quad (\Gamma(\mu)x^e(k) + W(\mu)e^e(k) + O(\mu) - \Omega(k)), \end{aligned} \tag{7.17}$$

where

$$\begin{aligned} H(\mu) &= \sum_{i=1}^{N} \mu_i H_i, \qquad \Gamma(\mu) = [\mu_1 \Gamma_1, \ldots, \mu_N \Gamma_N], \\ W(\mu) &= [\mu_1 W_1, \ldots, \mu_N W_N], \qquad O(\mu) = \sum_{i=1}^{N} \mu_i O_i. \end{aligned} \tag{7.18}$$

Proof: The prediction of the filtered output for the system S can be written as

(see Theorem 7.1)

$$
\begin{aligned}
\hat{z}(k+j|k) &= CA^j x^e(k) + \sum_{p=0}^{j-1} CA^{j-1-p} a + \\
&\quad \sum_{p=0}^{j-1} CA^{j-1-p} Bu(k+p) + CA^{j-1} K e^e(k) + c \\
&= \sum_{i=1}^{N} \mu_i C_{z,i} A_i^j \hat{x}_i^a(k) + \sum_{i=1}^{N} \sum_{p=0}^{j-1} \mu_i C_{z,i} A_i^{j-1-p} a_i + \sum_{i=1}^{N} \mu_i c_{z,i} + \\
&\quad \sum_{i=1}^{N} \sum_{p=0}^{j-1} \mu_i C_{z,i} A_i^{j-1-p} B_i u(k+p) + \sum_{i=1}^{N} \mu_i C_{z,i} A_i^{j-1} K_i e_i^e(k) \\
&= \sum_{i=1}^{N} \mu_i \Bigg\{ C_{z,i} A_i^j \hat{x}_i^a(k) + \sum_{p=0}^{j-1} c_{z,i} A_i^{j-1-p} a_i + \\
&\quad \sum_{p=0}^{j-1} C_{z,i} A_i^{j-1-p} B_i u(k+p) + C_{z,i} A_i^{j-1} K_i e_i^e(k) + c_{z,i} \Bigg\} \\
&= \sum_{i=1}^{N} \mu_i \hat{z}(k+j|k).
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\hat{Z}(k) &= \sum_{i=1}^{N} \mu_i(k) \hat{Z}_i(k) \\
&= \sum_{i=1}^{N} \mu_i(k) [H_i U(k) + \Gamma_i x_i^e(k) + W_i e_i^e(k) + O_i] \\
&= \left(\sum_{i=1}^{N} \mu_i(k) H_i \right) U(k) + \sum_{i=1}^{N} \mu_i(k) \Gamma_i x_i^e(k) + \\
&\quad \sum_{i=1}^{N} \mu_i(k) W_i e_i^e(k) + \sum_{i=1}^{N} \mu_i(k) O_i \\
&= H(\mu) U(k) + \Gamma(\mu) x^e(k) + W(\mu) e^e(k) + O(\mu),
\end{aligned}
$$

where the matrices $H(\mu)$, $\Gamma(\mu)$, $W(\mu)$, and $O(\mu)$ are defined in equation (7.18). Application of Theorem 7.2 to this predictive model yields the optimal control law (7.17). □

Remark 7.1 *Notice that although the global predictive model $\hat{Z}(k)$ for the augmented system S (7.15) is a convex combination of the local predictive models $\hat{Z}_i(k)$ for the local systems S_i, this is not the case with the optimal control law $U(k, \mu)$, i.e. it cannot be represented as a convex combination of the optimal control laws $U_i(k, \mu)$ obtained by the GPC controllers corresponding to the systems S_i.*

The control action that is actually implemented at time instant k is then

$$u(k, \mu) = \begin{bmatrix} I_m, & 0 \end{bmatrix} U(k, \mu).$$

Algorithm 7.1 (Controller Reconfiguration)

```
GIVEN $\hat{x}_i(k|k)$, $\Omega(k)$, AND $z(k)$.

Step 1. INITIALIZATION:
    $H = \mu_1(k)H_1,\ \Gamma = \mu_1(k)\Gamma_1,\ W = \mu_1(k)W_1,\ O = \mu_1(k)O_1,$
    $x^e(k) = [\hat{x}_1^T(k|k)^T,\ x_F^T(k)]^T,$
    $e^e(k) = (z(k) - C_{z,1}\hat{x}_1(k|k) - c_{z,1})^T.$

Step 2. FORM THE NECESSARY MATRICES:
    FOR $i = 2 : N$
        $H \leftarrow H + \mu_i(k)H_i$
        $\Gamma \leftarrow [\Gamma,\ \mu_i(k)\Gamma_i]$
        $W \leftarrow [W,\ \mu_i(k)W_i]$
        $O \leftarrow O + \mu_i(k)O_i$
        $x^e(k) \leftarrow [(\hat{x}^e(k))^T,\ \hat{x}_i^T(k|k)^T,\ x_F^T(k)]^T$
        $e^e(k) \leftarrow [(e^e(k))^T,\ (z(k) - C_{z,i}\hat{x}_i(k|k) - c_{z,i})^T]^T$
    END

Step 3. COMPUTATION OF THE CONTROL ACTION:
    $L = [I_m,\ 0](H^TH + R)^{-1}$
    $u(k) = -LH^T(\Gamma\hat{x}^e(k) + We^e(k) + T - \Omega(k))$
```

Algorithm 7.1 provides a summary of the reconfiguration algorithm.

7.5 Simulation results

In this section two case studies will be presented. In the first one a linear model of one joint of a space robot manipulator is used and the case is considered when sensor faults occur. The second example deals with a nonlinear model of the inverted pendulum on a cart. The model set in this experiment is obtained by performing linearization of the nonlinear system around five different operating points.

7.5.1 Experiment with the SRM

The case study that will be presented in this section is simple but also a very illustrative one. It considers the linear model of one joint of a space robot manipulator system, described in Section 3.6 on page 86. A schematic representation of the SRM is given in Figure 3.2 on page 86.

Parameter:	Symbol:	Value:
gearbox ratio	N	-260.6
joint angle of inertial axis	Ω	variable
effective joint input torque	T_j^{eff}	variable
motor torque constant	K_t	0.6
the damping coefficient	β	0.4
deformation torque of the gearbox	T_{def}	variable
inertia of the input axis	I_m	0.0011
inertia of the output system	I_{son}	400
joint angle of the output axis	ϵ	variable
motor current	i_c	variable
spring constant	c	130000

Table 7.2: The relevant values of the parameters in the linear model of one joint of the SRM.

In this experiment the case when a fault occurs which is not in the model set is considered. This faulty model will be represented as a convex combination of the models in $\mathcal{M}$.

The state-space model of the system is given by

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{c}{N^2 I_m} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{\beta}{I_{son}} & -\frac{c}{N^2 I_m} - \frac{c}{I_{son}} & -\frac{\beta}{I_{son}} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{K_t}{N I_m} \\ 0 \\ -\frac{K_t}{N I_m} \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & N & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \xi(t) \end{aligned} \quad (7.19)$$

This model is discretized with sampling period $T_s = 0.1$, [sec]. The system parameters are given in Table 7.2.

The following two models comprise the model set in this experiment:

- M_1: the nominal (no faults) model (see equation (7.19) and Table 7.2).
- M_2: a faulty model, representing 10% (partial) fault of sensor No.1.

Note, that each model representing a partial fault of sensor No.1 in the interval $[10\%,\ 100\%]$ can be written as a convex combination of the two models in $\mathcal{M}$, i.e. $\overline{co}\{M_1, M_2\}$.

The scenario for this experiment is the following:

- The system is in its normal mode of operation (model M_1 is active) in the time interval $[0,\ 99]$.
- At k=100 a 75% (partial) fault of sensor No.1 occurs. It corresponds to the following convex combination of the models in the model set:

$$M_{REAL} = 0.7222 M_1 + 0.2778 M_2.$$

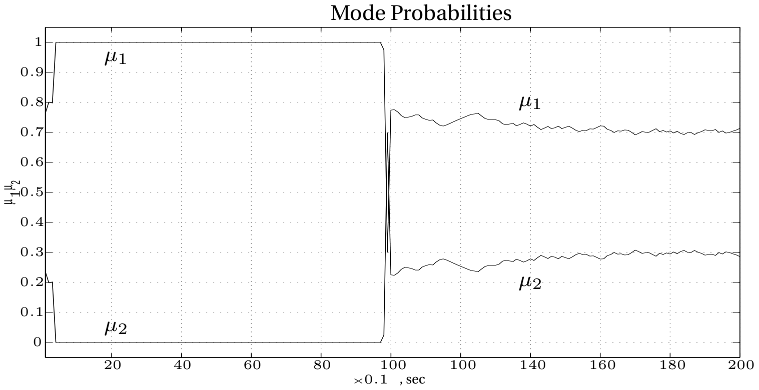


Figure 7.2: The mode probabilities for the experiment with the SRM.

The following choice of the predictive control parameters is made:

- Minimum costing horizon: $N_1 = 1$.
- Maximum costing horizon: $N_2 = 15$.
- Control horizon: $N_u = 8$.
- Weights on the control action: $R = 0.02I_8$.
- Reference signal: as a reference $\omega(k)$ a (low-pass) filtered step signal $\bar{\omega}(k)$ from 0 to 1 at $k = 0$, and from 1 to 0 at $k = 80$ is selected. The filter used is the following:
$$W_F(z) = \frac{\omega(z)}{\bar{\omega}(z)} = \frac{0.1813}{z - 0.8187}. \tag{7.20}$$
- Transition probability matrix:
$$\pi = \begin{bmatrix} 0.55 & 0.45 \\ 0.55 & 0.45 \end{bmatrix}.$$

Figures 7.2 and 7.3 present the results from this experiment. Since in the initial experiments there were very big fluctuations in the mode probabilities, the following low-pass filter was introduced

$$\tilde{\mu}_i(k) = 0.98\tilde{\mu}_i(k-1) + 0.02\mu_i(k-1).$$

It can be seen from Figure 7.2 that during the normal operation of the system ($k < 100$), the probability that corresponds to model M_1 is equal to 1. Then, when at $k = 100$ a 75% fault occurs, the two mode probabilities change accordingly. To be more precise, their means during the second half of the simulation time are,

- $\bar{\mu}_1(k) = 0.7213$, for $k = 100, \ldots, 200$, and

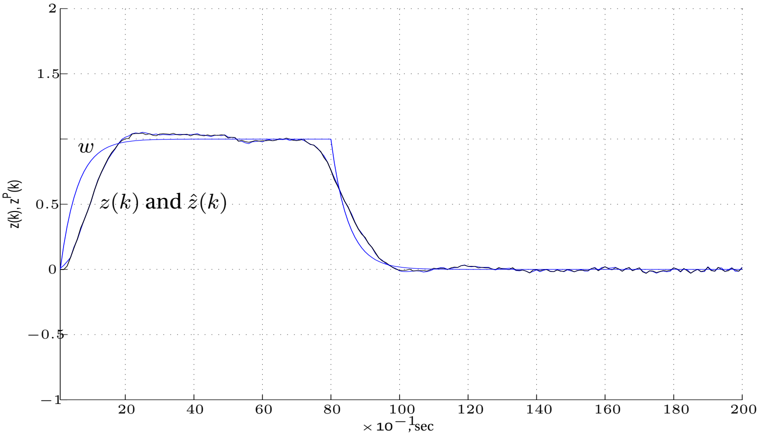


Figure 7.3: The output $z_1(k)$, its prediction $\hat{z}_1(k)$, and the reference $w(k)$, for the experiment with the SRM.

- $\bar{\mu}_2(k) = 0.2787$, for $k = 100, \ldots, 200$.

and as a result a model representing 74.92% fault of sensor No.1 is detected.

Figure 7.3 gives a plot of the system output, its prediction and the (filtered) reference signal.

As it was argued in the introduction, the existing MMAC algorithms (Maybeck and Stevens 1991; Athans et al. 1977), based on a bank of LQG controllers, can be inefficient when an unanticipated fault occurs. This is because the optimal LQG controller for the model of such a fault is *not* a convex combination of the optimal LQG controllers, each based on a given model from the model set. To illustrate this an additional experiment is presented, in which two optimal LQG controllers were designed: one for the nominal system (model M_1), and one for the faulty model M_2, representing a 10% fault of sensor No. 1. In this scenario the 75% unanticipated fault of sensor No. 1 is in effect throughout the whole simulation. The reference signal was selected in the same way as in the simulation with the MM-based GPC (see equation (7.20)). Figure 7.4 depicts the results from this experiment. It can be seen that in this experiment such a convex combination of the control actions may lead to an unstable closed-loop system for some unanticipated faults.

7.5.2 Experiment with the inverted pendulum on a cart

The next experiment illustrates the application of the MM-based GPC to the control of nonlinear systems. For this purpose a nonlinear model of the inverted pendulum on a cart is considered (Khalil 1996) as a control system. First, a linearization of the nonlinear model around five different operating points is performed, leading to a model set of five models. Afterwards the MM-based GPC is applied to the nonlinear model of the pendulum. The dynamic equations of the

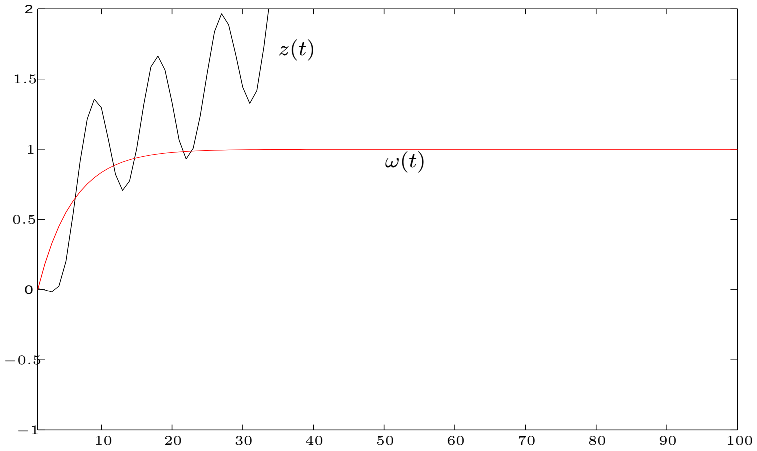


Figure 7.4: The output $z(t)$ and the reference signal $\omega(t)$ of system with an unanticipated 75% sensor fault. The control action now is calculated as a convex combination of the control actions of the two optimal LQG controllers.

system (see Figure 7.5) are

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \dfrac{g}{l^2} sin(x_1) - \dfrac{a}{l^2} cos(x_1) \end{cases}$$

where $x_1 = \Theta, [rad]$ is the angle between the pendulum and the vertical axis, $x_2 = \dot{\Theta}, [rad/sec]$ is the angular velocity of the pendulum, $g = 9.81, [m/sec^2]$ is gravity acceleration, $l, [m]$ is the length of the pendulum, and $a, [m/sec^2]$ is the acceleration of the cart.

Remark 7.2 *A similar problem is encountered during a rocket launch, when the rocket boosters have to be fired in a controlled manner so as to maintain the upright position of the rocket.*

Let the system be linearized around the point $x^* = [x_1^*,\ 0]^T$. Defining

$$f_2(x_1, x_2, a) = \frac{g}{l^2} sin(x_1) - \frac{a}{l^2} cos(x_1)$$

the linearized systems can be derived the following way (the first equation is linear and will not be considered):

$$\begin{aligned} \dot{x}_2 &= f_2(x_1^*, x_2^*, 0) + \left.\tfrac{\partial f_2}{\partial x_1}\right|_{x=x^*} (x_1 - x_1^*) + \left.\tfrac{\partial f_2}{\partial x_2}\right|_{x=x^*} (x_2 - x_2^*) + \left.\tfrac{\partial f_2}{\partial a}\right|_{x=x^*} a \\ &= \tfrac{g}{l^2} sin(x_1^*) + \tfrac{g}{l^2} \cos(x_1^*)(x_1 - x_1^*) + \tfrac{1}{l^2} \cos(x_1^*) a \\ &= \left(\tfrac{g}{l^2} \cos(x_1^*)\right) x_1 - \tfrac{g}{l^2} \left(sin(x_1^*) + \cos(x_1^*)\right) x_1^* + \left(\tfrac{1}{l^2} \cos(x_1^*)\right) a \end{aligned}$$

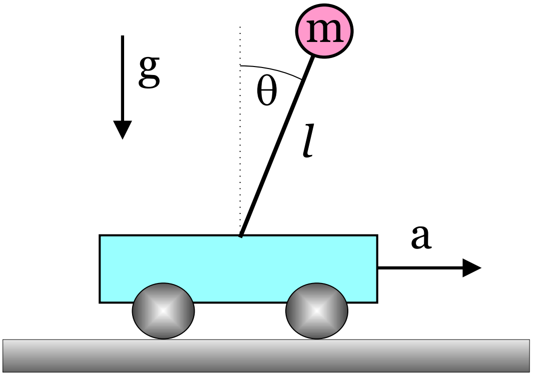


Figure 7.5: The Inverted Pendulum on a Cart.

Therefore the system can be written in the form (7.2) with

$$A_i = \begin{bmatrix} 0 & 1 \\ \frac{g}{l^2}\cos(x_1^i) & 0 \end{bmatrix}, \; a_i = \begin{bmatrix} 0 \\ -\frac{g}{l^2}\left(sin(x_1^i) + \cos(x_1^i)\right) x_1^i \end{bmatrix},$$
$$B_i = \begin{bmatrix} 0 \\ \left(\frac{1}{l^2}\cos(x_1^i)\right) \end{bmatrix}, \; C_i = \begin{bmatrix} 1 & 0 \end{bmatrix}, \; c_i = 0, \; D_i = 0$$

where x_1^i is the i-th linearization point. The model set considered for this simulation corresponds to the linearization points $x_1^i = \{0, \pm 10, \pm 20\}, [deg]$.

All these models are discretized with sampling time $T_s = 0.1, [sec]$ It was decided that models corresponding to angles $|\Theta| > 20, [deg]$ are unnecessary since the problem will be to maintain an angle of $10, [deg]$ between the pendulum and its upright position.

The following parameters were chosen:

- Length of the pendulum: $l = 1m$.
- Minimum costing horizon: $N_1 = 1$.
- Maximum costing horizon: $N_2 = 5$.
- Control horizon: $N_u = 4$.
- Weights on the control action: $R = \mathbf{diag}\{0, 0, 0, 0\}$.
- Reference signal $w(k) = \frac{\pi}{18}[1,\ 1,\ 1,\ 1]^T, [rad]$. This corresponds to a set-point of $10, [deg]$.
- Transition probability matrix:

$$\pi = \begin{bmatrix} 0.95 & 0.0125 & 0.0125 & 0.0125 & 0.0125 \\ 0.0125 & 0.95 & 0.0125 & 0.0125 & 0.0125 \\ 0.0125 & 0.0125 & 0.95 & 0.0125 & 0.0125 \\ 0.0125 & 0.0125 & 0.0125 & 0.95 & 0.0125 \\ 0.0125 & 0.0125 & 0.0125 & 0.0125 & 0.95 \end{bmatrix}$$

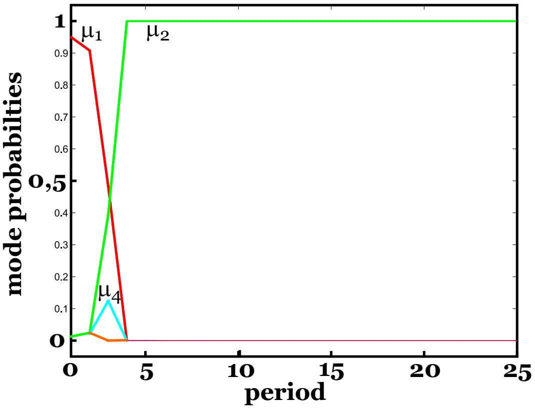


Figure 7.6: The Mode Probabilities.

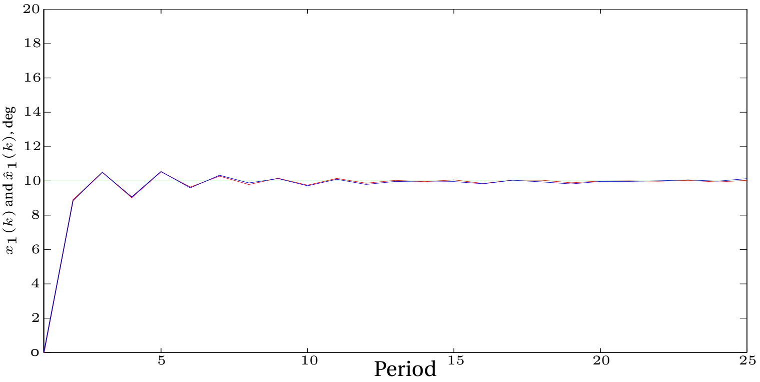


Figure 7.7: The Output Signal $z = x_1$ and its prediction $z^p = \hat{x}_1$ from the IMM estimator.

- Means of the noises: $\bar{\xi} = \bar{\eta} = 0$.

- Covariances of the noises: $Q = 10^{-6} I_2$ and $R = 10^{-6}$.

The results of this simulation are depicted on Figures 7.6 and 7.7. The simulations are made with the nonlinear model of the inverted pendulum. Figure 7.7 shows both the angle Θ and its prediction $\hat{\Theta}$ by the IMM estimator, which seem to overlap. A deviation between the two curves, of course, exists. Note, that when the system output gets close 10^o, which corresponds to model M_2 from the model set, the corresponding to this model mode probability μ_2 (see Figure 7.6) gets close to one, i.e. $\mu_2 \approx 1$. Since the pendulum never goes to negative degrees, the mode probabilities μ_3 and μ_5 stay at zero during the simulation.

7.6 Conclusion

This chapter presented an algorithm for the control of nonlinear systems represented by hybrid dynamic models or nonlinear systems. The method consists of two parts: identification and controller reconfiguration. The first part is essentially the IMM estimator whose purpose is to give a state estimate and a mode probability for each model in the model set. The controller reconfiguration part utilizes this information to derive a GPC action, assuming that the mode probabilities are constant over the maximum costing horizon.

The performance of the IMM estimator is strongly dependent on the choice of the transition probability matrix, as well as on the models in the model set $\mathcal{M}$. A model set consisting of models close to one another results in a deterioration of the performance of the IMM estimator, which in turn affects the performance of the MM-based GPC.

Another very important issue is the choice of the transition probability matrix π. It should be pointed out that serious difficulties regarding the selection of this matrix were encountered, since the IMM estimator turns out to be extremely sensitive to this design parameter. In addition, the entries in the transition probability matrix represent the probabilities for switches from one expected mode to another expected mode. However, they do not reflect the probabilities for jumps to *unexpected* modes.

8 Conclusions and Recommendations

In this thesis some novel approaches were presented to both passive and active fault-tolerant control with the main focus on *dealing with model and FDD uncertainty.* This final chapter discusses on the results presented in this thesis and gives suggestions for future research.

8.1 Conclusions

In the field of fault-tolerant systems design there are two main research streams, the one (called FDD) dealing with the detection and diagnosis of faults that occur in the controlled system, and the other (FTC) looking at the problem of achieving fault-tolerance by means of designing passive or active fault-tolerant control methods. Unfortunately, although there is a large amount of publications in the field, the interaction between these two research lines is still rather weak. The FTC methods that rely on fault diagnosis, for instance, often assume that the perfect FDD scheme is present that provides precise fault estimates with no delay. The FDD approaches, on the other hand, do not take into consideration the presence and the needs of the FTC part. As a result, the integration of these schemes becomes very difficult in practice. This thesis considers the FTC problem with a clear view on the presence of an imperfect FDD scheme by means of considering uncertainty in the fault estimates provided by the FDD. Moreover, FDD uncertainty with time-varying size can also be dealt with. This allows us to consider an even more realistic operation of the FDD scheme that would produce fault estimates with bigger uncertainty in the first moments after the occurrence of a fault, and would subsequently provide more and more accurate fault estimates (with little uncertainty) as more measurement data becomes available from the system. Furthermore, in addition to the FDD uncertainty, the developed methods make it possible to treat also model uncertainty which allows to reduce the gap between the real-life system and its (linear) model. In this way an attempt is made in this thesis to develop methods that could more easily be combined with existing FDD approaches in a real-life application.

The main assumption imposed in this thesis is that both the fault-free system and the faulty system are stabilizable and detectable[1]. In this way, the de-

---

[1] Stabilizability (detectability) is weaker than controllability (observability) as in the former some of the states corresponding to stable dynamics are allowed to be uncontrollable (unobservable).

veloped methods are applicable only to faults that do not affect these two basic properties that practically ensure that a stabilizing controller exists, so that controller reconfiguration still makes sense. If a fault occurs that results in an un-stabilizable system then there exists no controller that can yield the closed-loop system stable. In such cases other measures than controller reconfiguration have to be taken as, for instance, safe termination of the operation of the system. Such cases fall outside the scope of this thesis.

Both passive and active methods for FTC are developed in this thesis. A short description of them is provided below.

The *passive approaches* to FTC, discussed in Chapters 2 and 3, do not require the availability of a FDD scheme. Instead, these methods aim at designing one robust controller that achieves reduced closed-loop system sensitivity to a certain set of anticipated faults. To this end, besides the parametric uncertainty of the model, the set of possible faults are also viewed as uncertainty in the model of the system. In this way the goal is to design a controller that provides guaranteed robust stability and performance in the presence of faults and model uncertainty. The method of Chapter 2 can be used for solving this problem in the state-feedback case where it is representable as a robust LMI problem. The standard methods for solving robust LMIs usually assume convenient structure of the model uncertainty, e.g. polytopic, affine or in LFT form, and as such impose a restriction on the class of faults that can be addressed. To circumvent this, the method in Chapter 2 is developed in a probabilistic framework that makes it possible to consider a very general dependance of the system matrices on the faults and the uncertainties. However, Chapter 2 is much more than just a method for passive FTC design; the probabilistic approach presented there can be viewed as a central tool that is utilized throughout some of the other chapters of the thesis where robust LMIs appear.

Contrary to the state-feedback case, in the output-feedback case most robust controller design methods are not representable as robust LMI problems. Even worse, they often take the form of robust BMI optimization problem that have been shown to be NP-hard problems. Chapter 3 is focused on finding locally optimal solutions to such BMI problems by means of performing a local BMI optimization. To initialize the optimization, a method is proposed for finding an initially feasible robust output-feedback controller based on LMIs. This LMI method might be conservative but can be used as a good initial guess for the BMI optimization. The complete method developed in this chapter allows to design robust output-feedback FTC in the presence of polytopic uncertainty, i.e. unlike the probabilistic method this approach cannot deal with a general uncertainty representation. For that reason this method may be very conservative if applied to systems in which the state-space matrices do not depend in an affine way on the faults and/or on the uncertain parameters.

Although passive FTC methods possess a number of advantages (e.g. no FDD is necessary, simple for implementation, etc.), a serious drawback is they can only consider a restricted set of anticipated faults that do not have “big” effect on the model of the system – searching for only one controller that provides acceptable performance in the face of a wide variety of possible (combinations of) faults is in practice too optimistic. For that reason the active approaches have attracted much more attention by the FTC community. *Active FTC methods* are developed in Chapters 4-7 of this thesis. These methods assume the presence of an FDD scheme that can provide estimates of the faults. A wide class of sensor, actuator and component faults can be treated by these active methods for FTC. Moreover, FDD uncertainty with time-varying size can also be addressed by the methods in Chapters 4-5. This is achieved in Chapter 4 by means of *LPV controller design* in which the controller can be scheduled by both the fault estimates as well as by other quantities related to the size of the FDD uncertainty. By making use of the probabilistic approach, this method can deal with a wide class of faults in the presence of both model and FDD uncertainties. Both the state-feedback and the output-feedback cases are considered. The state-feedback LPV-based FTC has also been demonstrated on a real-life experimental setup in Chapter 6 consisting of a brushless DC motor. The output-feedback LPV-based FTC is much more computationally involving and can be somewhat conservative as it relies on the LMI-based initialization of the BMI optimization in Chapter 3. It basically consists of first designing a state-feedback gain matrix, which is next kept fixed during the design of the remaining controller matrices. Furthermore, the second step involves an additional structural constraint on the Lyapunov matrix that adds up to the conservatism. Conservatism is, however, unavoidable in deriving any numerically tractable method for an NP-hard problem.

In order to avoid solving nonconvex, NP-hard BMI optimization problems in the output-feedback case, a method for robust output-feedback MPC was developed in Chapter 5. Unlike most of the existing state-space approaches to MPC, the method from this chapter does not assume that the state vector is available for measurement – a rather restrictive assumption in practice. When the state is not measured, the standard approach is to make use of a state observer to reconstruct the missing state information and to use the state estimate, instead of the true state, in the computation of the control law. Moreover, the well-known separation principle allows us to design the observer and the controller separately from each other. This separation principle is, however, not valid when uncertainty is considered in the model. For that reason, the approach followed in Chapter 5 combines the design of a Kalman filter over a finite time window with the design of a finite-horizon MPC into one worst-case robust linear least squares problem. In this way the robust output-feedback MPC problem takes a very attractive form for the probabilistic setting of Chapter 2. Among the useful properties of this method are

- the self-reconfigurability property of the MPC controller, i.e. reconfiguration can be achieved by changing the internal model after a fault has been diagnosed by the FDD scheme,
- control action saturation can explicitly be treated by adding up constraints to the robust LLS problem,
- a very wide class of faults, model and FDD uncertainties can be considered due to the probabilistic setting of the solution.

An obvious drawback is the computational complexity and the fact that, in its basic form, the stability of the closed-loop system with an MPC controller cannot be theoretically guaranteed.

Finally, Chapter 7 represents an attempt in the direction of developing robust active FTC for nonlinear systems in the presence of model uncertainties. The chapter presents a multiple-model method where the starting point is the design of a set of models that can accurately enough represent the behavior of a nonlinear (or faulty) system. At each time instant the current mode of operation of the system is represented as a convex combination of the local models in the model set. The weights in this convex combination, called the mode probabilities, are provided by the interacting multiple model (IMM) method based on a bank of Kalman filters, one for each model. Then under the assumption that the mode probabilities do not change in some future interval of time, a generalized predictive controller is reconfigured by the estimated mode probabilities. The main difficulty in this approach is the choice of the local models in the model set as well as the transition probability matrix that assigns probabilities for jumps from one mode to the other. The IMM algorithm turns out to be very sensitive to these quantities when the current mode of operation of the system cannot be represented accurately by any individual local model. Another disadvantage of this approach is that no uncertainty can be considered.

8.2 Recommendations

In view of the shortcomings of the methods presented in this thesis, the following suggestions for future research are made:

- The passive and active methods for FTC, presented in Chapters 2-5, can only deal with linear systems in the presence of model and FDD uncertainty. The multiple model (MM) method from Chapter 7, on the other hand, is applicable to nonlinear systems but cannot deal with uncertainty. It would therefore be useful to investigate the possibility of combining the MM method with one of the active FTC methods with the aim to develop a robust active FTC approach for uncertain nonlinear systems. To this end one may, for instance, think about replacing the MPC controller in the MM method of Chapter 7 with the robust MPC controller from Chapter 5 (called iPC). The bank of Kalman filters, used in the IMM method, would then no longer be needed as the local state estimates will be provided by the local iPC controllers. Although this approach looks attractive, a strong disadvantage would be the increased computational burden since the iPC controllers are based on a time-consuming optimization.

- The main disadvantages of the iPC method from Chapter 5 are the computational complexity and the lack of theoretical guarantees for closed-loop stability. To enforce stability, one may think of including end-point constraints into the optimization which, however, increases even more the computational burden. From practical viewpoint it is therefore important that a computationally faster implementation of the scheme is developed. To this end one could, for instance, analyze the possibility of using the optimal ellipsoid computed at each time-iteration at the next iteration. This is expected to significantly reduce the computational effort of the algorithm.

- It was discussed in Chapter 2 that the ellipsoid algorithm (EA), developed there, has an advantage over the existing subgradient iteration algorithm (SIA) method that in its original form the latter requires a-priori information about the solution set, namely it assumes that a radius of ball contained into the solution set is given. Although some modifications were subsequently proposed that overcome this difficulty, these modifications result in an increased number of iterations before convergence to a feasible solution. For both the SIA and the EA method upper bounds on the maximum number of correction steps that can be executed before a feasible solution is found have been derived and, as demonstrated in Example 2.2 on page 48, the upper bound for the EA method is much lower than that of SIA. By only comparing these upper bounds, however, one cannot draw the conclusion that EA method will actually converge faster than SIA. Moreover, for some problems the solution set is unbounded so that any radius would do for the SIA method, while the EA method might face difficulties in the computation of the initial ellipsoid. On the other hand, as shown in Section 2.4.2 on page 54, for some robust least squares problems the initial ellipsoid can be computed in a straightforward way, while the SIA algorithm can be expected to face difficulties due to the fact that the solution set may have a very small volume or may even be empty. Hence, the one algorithm may be more suitable for certain problems than the other, and vice versa. A more thorough evaluation and analysis of these two methods is therefore necessary. A user-friendly implementation of the probabilistic methods in the form of a toolbox for MATLAB® might also be useful.

Summary

Faults in controlled systems represent malfunctioning of the system that results in performance degradation or even instability of the system. When untreated, small faults in some subsystems of the controlled system can easily develop into serious overall system failures. In safe-critical systems such failures can have serious consequences ranging from economical losses to human deaths. It is therefore important that such safe-critical systems possess the properties of increased reliability and safety.

A standard approach in practice to improve these properties is to introduce *hardware redundancy* in the system through duplicating or triplicating some critical hardware components. This approach, however, has limited applicability due to the increased weights and costs. For that reason most of the research being conducted in this field is focused on exploiting the existing in the system *analytical redundancy* by means of using model-based fault-tolerant control (FTC).

There are two main approaches to fault-tolerant control: passive and active. The *passive FTC* methods are based on robust controller design methods and aim at achieving increased insensitivity of the closed-loop system with respect to a certain class of anticipated system faults. The main disadvantages of this approach is that only a very restricted set of faults can be treated (usually faults that do not significantly affect the dynamic behavior of the system) and that it results in a decreased system performance. The latter is explained by the worst-case optimization approach that is inherent to robust control so that as a result one and the same performance is achieved for all considered faulty modes of operation of the system. The *active FTC* approach, on the other hand, is either based on a complete controller redesign (reconfiguration) or on a selection of a controller from a set of pre-designed controllers. This method often requires the presence of a fault detection and diagnosis (FDD) scheme that has the task to detect and localize the faults that occur in the system. The structure of a complete active FTC system based on FDD is depicted in Figure 8.1.

This thesis presents both methods for passive and for active FTC. It does not consider the FDD part in Figure 8.1, which is a separate line of research in this area. However, striving to arrive at a more realistic problem formulation the thesis is focused on the problem of dealing with inaccurate information coming from the FDD part (i.e. FDD uncertainty) as well as with uncertainty in the model of the controlled system.

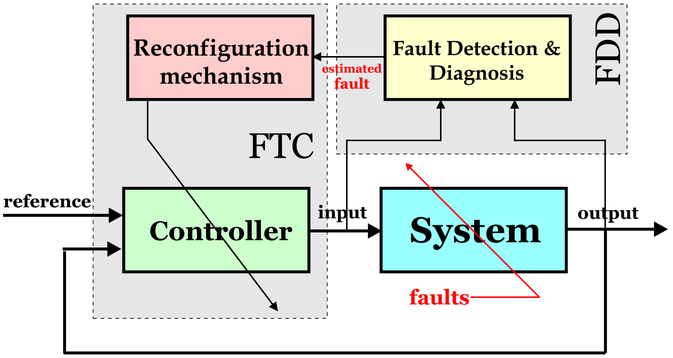


Figure 8.1: Main components of an active FDD-based FTC system.

Passive FTC Approaches

In Chapters 2 and 3 passive methods for FTC design were presented where the aim is to achieve closed-loop system robustness with respect to model uncertainties and system faults at the same time. To this end, the problem of passive FTC has been defined in the thesis as the problem of computing a robust controller K by solving the following optimization problem

$$K_{PFTC} = \arg\min_{K} \sup_{\substack{\delta \in \boldsymbol{\Delta} \\ f \in \mathcal{F}}} J(\mathcal{F}_L(M(\delta, f), K)),$$

where $M(\delta, f)$ denotes the controlled system that depends on some vector $\delta \in \boldsymbol{\Delta}$ that represents the model uncertainty as well as the considered class of anticipated system faults represented by the vector $f \in \mathcal{F}$. The lower linear fractional transformation $\mathcal{F}_L(M, K)$ is used here to denote the closed-loop system, and the continuous map $J(\cdot)$, called the performance index, is a measure of the closed-loop system performance, i.e. the smaller the value of $J(\cdot)$, the better the performance of the system. The optimization problem above is known as a *worst-case optimization* problem since a controller is sought that achieves the best performance for the worst-case uncertainty and faults. The reasoning behind that is that the achieved worst-case performance is also guaranteed for any fault from $\mathcal{F}$ and uncertainty from $\boldsymbol{\Delta}$.

There are two main difficulties with this optimization problem, both related to convexity. In the state-feedback case, provided that the set $\{M(\delta, f) : \delta \in \boldsymbol{\Delta}, f \in \mathcal{F}\}$ is a convex polytope, the optimization problem is convex in the controller parameters for most standard performance indexes, including the $\mathcal{H}_2$-norm and the $\mathcal{H}_\infty$-norm. In this case the worst-case optimization problem above can be represented as a linear matrix inequality (LMI) optimization problem, that can be numerically solved in a very efficient way by the existing LMI solvers. However, when $\{M(\delta, f)\}$ is not a convex set the original optimization problem is also nonconvex and the LMI solvers cannot be used. A "brute force" method to circumvent this problem is to over-bound the set $\{M(\delta, f)\}$ by a convex polytope, and again make use of the LMI solvers. This approach, however, introduces unnecessary, and sometimes unacceptable, conservatism in the solution.

In order to be able to deal with a more general fault and uncertainty representations, for which the set $\{M(\delta, f)\}$ is possibly not nonconvex, an approach is proposed in Chapter 2 in a probabilistic framework, inspired by Polyak and Tempo (2001) and Calafiore and Polyak (2001). This approach makes it possible to consider a very general dependence of the system matrices on the uncertain parameters and on the faults; in fact, they are only assumed to remain bounded for all faults and uncertainties. This method is applicable to controller design problems that are representable as *robust* LMIs (as in the state-feedback case). It is basically an iterative algorithm where the starting point is the computation of an ellipsoid that contains the set of all solutions to the problem. Then, at each iteration it generates a random uncertainty sample for which the ellipsoid is computed such that it also contains the solution set and that it has a smaller volume than the ellipsoid at the previous iteration. The approach is proved to converge to the solution set in a finite number of iterations with probability one. This method, however, is used in this thesis not only as an approach to passive FTC, but additionally as a very useful tool, exploited in some of the other chapters, to parameter-dependent LMIs problems.

The probabilistic method is, unfortunately, not applicable to the output feedback case in the presence of uncertainty, when the optimization problem can no longer be represented as a convex LMI problem, but are represented by *bilinear matrix inequalities* (BMIs). Even worse, it has been shown in the literature that the BMI problem is a nonconvex, NP-hard problem. A local BMI optimization approach is developed in Chapter 3 that can be used to tackle such BMI problems. The method has guaranteed convergence to a local optimum of the performance index $J(\cdot)$.

Active FTC Approaches

The active methods for FTC, developed in Chapters 4 and 5, and tested in Chapter 6, assume that an estimate $\hat{f}$ of the fault vector f is provided by some FDD part. Unlike most of the other methods for FTC, this estimate is furthermore considered in this thesis as imperfect in an attempt to arrive at a more realistic assumption about the FDD process, which is expected to eventually facilitate the interconnection between the developed active FTC methods with some existing FDD methods. In order to represent this imprecision in the fault estimate, uncertainty is added to it so that the true fault vector is assumed to be given by $f = (I + \Delta_f)\hat{f}$ for some $\Delta_f \in \mathbf{\Delta}_f$. In this way the FDD uncertainty is represented by the uncertainty set $\mathbf{\Delta}_f$. This FDD uncertainty, however, usually increases in size immediately after the occurrence of a fault due to the absence of enough measurements from the system for precise diagnosis. Later on, as more input-output data becomes available from the system, the faults estimates are refined by the FDD scheme and the uncertainty in them decreases. Hence, the performance of the overall FTC system can be improved by allowing the controller to be able to deal with such time-varying size of the FDD uncertainty. To this end, however, the FDD scheme should be capable of providing not only an estimate of the fault but also an upper bound on the size of the uncertainty in this estimate.

The size of the FDD uncertainty is represented in the thesis by a vector $\gamma_f(k)$ such that $f_k = (I + \mathbf{diag}(\gamma_f(k))\bar{\Delta}_f)\hat{f}_k$ with $\|\bar{\Delta}_f\|_2 \leq 1$. In this way the different elements of the vector $\gamma_f(k)$ assign different uncertainty sizes on the different entries of the fault vector f_k. Therefore, provided that the FDD scheme produces $(\hat{f}_k, \gamma_f(k))$ at each time instant, the active approaches in this thesis aim at fining a controller by solving the following optimization problem

$$K_{AFTC}(\hat{f}, \gamma_f) = \arg \min_{K(\hat{f},\gamma_f)} \sup_{\substack{\delta \in \mathbf{\Delta} \\ \bar{\Delta}_f \in \bar{\mathbf{\Delta}}_f \\ \underline{\gamma_f} \leq \gamma_f \leq \bar{\gamma}_f}} J(\mathcal{F}_L(M(\delta, f), K(\hat{f}, \gamma_f)))$$

where $\bar{\mathbf{\Delta}}_f = \{\Delta \in \mathbf{\Delta}_f : \|\Delta\| \leq 1\}$, and where the vectors $\{\underline{\gamma_f}, \bar{\gamma}_f\}$ define a lower and an upper bound on the possible uncertainty sizes.

The focus of Chapter 4 is on the design of linear parameter-varying (LPV) controllers for robust active FTC. Two approaches are proposed. Section 4.2 deals with sensor and actuator faults only. It presents a deterministic approach that consists of the off-line design of a set of parameter-varying robust output-feedback controllers, in which the only scheduling parameter is the size $\gamma_f(k)$ of the FDI uncertainty. The set of such LPV controllers is built up in such a way that each controller corresponding to a suitably defined fault scenario. After a fault has been diagnosed by the FDD scheme, the reconfigured controller is taken as a scaled version of one of the predesigned controllers. Although a finite set of controllers are initially designed, the reconfiguration scheme deals with an arbitrary combination of multiplicative sensor and actuator faults as long as the system remains stabilizable and detectable. This approach is based on LMIs that are derived by neglecting the structure of the uncertainty, and is therefore conservative.

In order to circumvent the conservatism of the deterministic LPV approach, another approach is proposed in Section 4.3 in the probabilistic framework of Chapter 2. This approach to robust output-feedback FTC has the following advantages:

- the controller is scheduled by both the fault estimates $\hat{f}_k$ and the size $\gamma_f(k)$ of its uncertainty,
- it deals with structured uncertainty,
- it is applicable to not only sensor and actuator faults, but also to component faults.

In order to be applicable in the output-feedback case this method makes use of the two-step procedure from Chapter 3 that was used there to initialize the BMI optimization.

Clearly, the LPV methods for robust active FTC are very suitable for online implementation due to the fact that the design is performed completely off-line. This results in limited on-line computations for controller re-configuration after the occurrence of a fault. The LPV approach from Section 4.3 is tested in Chapter 6 on a real-life experimental setup consisting of a brushless DC motor. The FTC approach is combined there with an FDD scheme for the detection and estimation of parameter and sensor faults.

Chapter 5 presents a finite-horizon output-feedback MPC design approach is that is robust with respect to model and FDD uncertainties. The approach consists in a combination of a Kalman filter and a finite-horizon MPC into one min-max (worst-case) optimization problem, that is solved at each iteration by making use of the probabilistic method of Chapter 2 provided that the state covariance matrix is given. In order to obtain it, two methods are presented. In the first method the aim is to find the covariance matrix by minimizing its trace under the constraint that it is compatible with all values of the uncertainty. This method also makes use of the probabilistic ellipsoid algorithm from Chapter 2, and is therefore computationally expensive. To reduce the computations, the second method is proposed that is much faster but also more conservative. The complete MPC algorithm has the advantage that it deals with the robust output-feedback problem directly without having to solve BMI optimization problems. A disadvantage is its computational demand and the lack of guaranteed closed-loop stability.

The passive and active approaches to FTC discussed above are focused on only one local linear model in the presence of uncertainty in both the model description and in the FDD scheme. The question of how to extend them to deal with the complete multiple model representation of a nonlinear system is much more difficult and is in this thesis only partially addressed. Specifically, in the case of no uncertainty a method is developed in Chapter 7 that can be used for control of nonlinear systems represented by multiple local models. The starting point is the construction of a model set $\mathcal{M}$ that contains either local linear approximations of a nonlinear system or models representing faulty modes of operation of a (linear) system. The nonlinear system is then at each time instant represented by a convex combination (with weights μ_i) of the local linear models. The method consists of a multiple model estimator that provides local and global state-estimates as well as estimates of the weights $\hat{\mu}_i$. They are then used to parametrize an MPC. The multiple model estimator consists of a bank of Kalman filters, one for each local model. The Kalman filters are independently designed from the MPC. In the case when uncertainty is present in the system (and, therefore, also in the local models), however, the design of the state observer and the controller can no longer be executed separately due to the fact that the well known separation principle no longer holds. It therefore remains for future research to investigate how to deal with uncertainties in the elements of the model set $\mathcal{M}$.

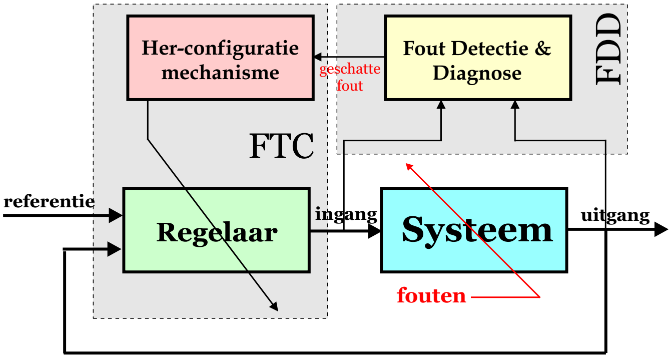
Samenvatting

Fouten in regelsystemen zijn het gevolg van storingen in systemen en leiden tot een achteruitgang van de prestaties of zelfs tot instabiliteit van het systeem. Wanneer bepaalde fouten in sommige onderdelen van een systeem niet op tijd worden aangepakt kan dit makkelijk leiden tot een complete uitval van het systeem. Bij veiligheidskritische systemen kunnen zulke *failures* ernstige consequenties hebben variërend tussen grote economische verliezen en dodelijke ongelukken. Er zijn talrijke voorbeelden van voorvallen met dergelijke tragische uitkomst. Het is dus belangrijk dat zulke veiligheidskritische systemen beschikken over eigenschappen als verhoogde bedrijfszekerheid en veiligheid.

Een in de industrie vaak gebruikte aanpak om deze eigenschappen te verbeteren is door gebruik te maken van hardware redundantie in het systeem, dat wil zeggen door middel van het verdubbelen of verdrievoudigen van de meest kritische hardware componenten. Een nadeel dat de toepassing van deze aanpak vaak verhindert, is de toename van het gewicht en een verhoging van de kosten. Daarom is het grootste gedeelte van het huidige onderzoek op het gebied van *fault-tolerant control* (FTC) gefocusseerd op het ontwikkelen van modelgebaseerde technieken die gebruik maken van de al in het systeem bestaande analytische redundantie.

In het algemeen zijn er twee benaderingen tot fout-tolerant regelen: passieve en actieve. De *passieve* FTC benadering is gebaseerd op robuust regelaarontwerp en probeert een verhoogde ongevoeligheid voor bepaalde geanticipeerde systeemfouten in het gesloten-lus systeem te bewerkstelligen. Het grootste nadeel van deze aanpak is dat op deze manier alleen een beperkte verzameling van fouten kan worden behandeld (met name fouten die geen grote invloed hebben op het dynamische gedrag van het systeem) en dat deze benadering tot lagere prestatie van het systeem leidt. Het laatstgenoemde kan worden verklaard met de *worst-case* optimalisatie die inherent is aan het robuuste regelaarontwerp. Als gevolg hiervan krijgt men voor het nominale systeem dezelfde prestatie als voor alle verwachtte foutieve regimes van het systeem. De *actieve* benadering tot FTC is daarentegen gebaseerd op of een compleet her-ontwerp (herconfiguratie) van de regelaar, of de selectie van een regelaar uit een verzameling van off-line ontworpen regelaars. Deze methode vereist vaak de aanwezigheid van een fout-detectie en diagnose (FDD) onderdeel dat belast is met de taak de fouten die in het systeem optreden te detecteren en lokaliseren. Figuur 8.2 geeft de structuur van een compleet actief FDD-gebaseerd FTC systeem weer.

In dit proefschrift worden zowel passieve als actieve FTC technieken voor-



Figuur 8.2: Hoofd componenten van een actief FDD-gebaseerd FTC systeem.

gesteld. Het beschouwt niet het FDD deel in Figuur 8.2; dit is een apart onderzoeksgebied op zich. In een poging tot een meer realistische probleemformulering te komen wordt er in dit proefschrift veel aandacht besteden aan de vraag hoe men moet omgaat met inaccurate informatie van het FDD onderdeel (FDD onzekerheid) alsmede met onzekerheid in het model van het regelsysteem.

Passieve FTC Benadering

In Hoofdstukken 2 en 3 worden passieve FTC methodes voorgesteld waarin het doel is robuustheid van het gesloten-lus systeem te realiseren te aanzien van zowel onzekerheid in het model als systeemfouten. In overeenstemming met dit doel wordt in het proefschrift het probleem van het bepalen van een robuuste regelaar K gedefinieerd als de oplossing van het volgende optimalisatie probleem

$$K_{PFTC} = \arg\min_{K} \sup_{\substack{\delta \in \boldsymbol{\Delta} \\ f \in \mathcal{F}}} J(\mathcal{F}_L(M(\delta, f), K)),$$

waarin $M(\delta, f)$ het regelsysteem representeert als een functie van een bepaalde vector $\delta \in \boldsymbol{\Delta}$ die hier gebruikt wordt om model onzekerheid alsmede de beschouwde groep van geanticipeerde systeemfouten $f \in \mathcal{F}$ voor te stellen. De *lower linear fractional* transformatie $\mathcal{F}_L(M, K)$ geeft het gesloten-lus systeem aan, en de continu afbeelding $J(\cdot)$, die prestatie index wordt genoemd, wordt gebruikt als een maatstaf voor de prestatie van het gesloten-lus systeem. Des te kleiner de waarde van $J(\cdot)$ des te beter de prestatie van het systeem. Het bovengenoemde optimalisatie probleem staat bekend als het *worst-case* optimalisatie probleem omdat er een regelaar wordt gezocht dat de beste prestatie garandeert in het geval van de slechtste mogelijke onzekerheid en fouten. De redenering hierachter is dat de *worst-case* prestatie die op deze manier wordt bereikt, gegarandeerd is voor iedere fout uit $\mathcal{F}$ en onzekerheid uit $\boldsymbol{\Delta}$.

Er zijn twee complicaties met dit optimalisatie probleem, die allebei verband houden met convexiteit. In het geval van toestandsterugkoppeling, en wanneer de set $\{M(\delta, f) : \delta \in \mathbf{\Delta},\ f \in \mathcal{F}\}$ een *convex polytope* is, is het optimalisatie probleem convex in de parameters van de regelaar voor de meeste prestatie indexes, inclusief de $\mathcal{H}_2$-norm en de $\mathcal{H}_\infty$-norm. In dit geval kan het bovenstaande optimalisatie probleem makkelijk worden omgezet in een *linear matrix inequality* (LMI) optimalisatie probleem. Zulk LMI optimalisatie probleem kan heel efficient numeriek worden opgelost door de bestande *LMI solvers*. Aan de andere kant, wanneer $\{M(\delta, f)\}$ geen convex set is, blijft het originele optimalisatie probleem nog altijd *nonconvex* en zo kunnen de *LMI solvers* hierop niet toegepast worden. De meest voor de hand liggende oplossing is eerst de set $\{M(\delta, f)\}$ te begrenzen door een *convex polytope* en vervolgens gebruik te maken van de *LMI solvers*. Deze benadering produceert echter een heel conservatieve oplossing dat hetgeen onacceptabel is.

Om met wat meer algemenere representaties van fouten en onzekerheden om te kunnen gaan, waarbij de set $\{M(\delta, f)\}$ mogelijk *nonconvex* is, een probabilistische benadering voorgesteld in Hoofdstuk 2 is. Deze aanpak is geïnspireerd door het werk van Polyak and Tempo (2001) en Calafiore and Polyak (2001). Deze benadering maakt het mogelijk dat de systeemmatrices op een heel algemene manier afhankelijk zijn van de onzekere parameters en van de fouten; in feite wordt er alleen aangenomen dat de systeemmatrices begrensd blijven voor alle fouten en mogelijke waarden van de onzekerheid. Deze aanpak is toepasbaar op regelaarontwerp problemen die in de vorm van robuuste LMIs kunnen worden voorgesteld. Het is in principe een iteratief algoritme dat begint met het berekenen van een initiële ellipsoïde die de set van alle oplossingen van het probleem bevat. Gedurende elke iteratie genereert het algoritme een willekeurige monster van de onzekerheid en op basis daarvan wordt de volgende ellipsoïde berekend zodanig dat zij ook de set van alle oplossingen van het probleem bevat en haar volume kleiner is dan dat van de ellipsoïde op de vorige iteratie. Er wordt bewezen dat dit algoritme met een waarschijnlijkheid van één in een eindig aantal stappen convergeert naar de set van oplossingen. Deze aanpak wordt in het proefschrift gebruikt niet alleen als een benadering voor passive FTC, maar ook als een hele nuttige tool voor LMI problemen die op een algemene manier van parameters afhangen. Daar wordt ook nog in andere hoofdstukken gebruik van gemaakt.

Helaas is de probabilistische methode niet direct toepasbaar op uitgangsterugkoppeling problemen in de aanwezigheid van onzekerheid. Voor deze problemen kan het optimalisatieprobleem niet in convex LMI probleem worden omgezet, maar wordt beschreven door *bilinear matrix inequalities* (BMIs). Het is aangetoond in de literatuur dat het BMI probleem een *nonconvex*, NP-hard probleem is. In Hoofdstuk 3 wordt een lokale BMI optimalisatie benadering ontwikkeld die toegepast kan worden op zulke BMI optimalisatieproblemen. Deze methode heeft een gegarandeerde convergentie naar een lokaal optimum van de prestatie index $J(\cdot)$.

Actieve FTC Benadering

In de actieve methodes voor FTC, beschreven in Hoofdstukken 4 en 5, en getest in Hoofdstuk 6, wordt er aangenomen dat er een schatting $\hat{f}$ van de vector

f bestaat verkregen uit het FDD schema. In tegenstelling tot de meeste andere methodes voor FTC wordt deze schatting in dit proefschrift als onnauwkeurig beschouwd teneinde een meer realistischer veronderstelling over de FDD proces te verkrijgen. Dit zou dan de combinatie moeten vergemakkelijken tussen de in dit proefschrift ontwikkeld actieve FTC methodes en de al bestande FDD algoritmen. Om deze onnauwkeurigheid in de schatting van de fout wiskundig te kunnen voorstellen wordt onzekerheid geïntrodoceerd zodat de echte fout vector beschreven wordt als $f = (I + \Delta_f)\hat{f}$ voor $\Delta_f \in \mathbf{\Delta}_f$. Op deze manier wordt de FDD onzekerheid voorgesteld door de onzekerheid set $\mathbf{\Delta}_f$. De grootte van deze FDD onzekerheid neemt echter na het optreden van een fout vaak toe omdat er dan niet genoeg ingang-uitgang metingen beschikbaar zijn voor een nauwkeurige detectie. Later, als meer informatie van het systeem beschikbaar is worden de fout schattingen verfijnd door het FDD schema en zal de onzekerheid afnemen. De prestatie van het FTC systeem kan dus worden verbeterd door de regelaar zodanig te ontwerpen dat hij dan met tijdsvariërende FDD onzekerheid om kan gaan. Daarvoor moet natuurlijk het FDD schema in staat zijn om niet alleen de fouten te schatten maar ook de grootte van de onzekerheid in deze schattingen.

In dit proefschrift wordt de grootte van de FDD onzekerheid voorgesteld door een vector $\gamma_f(k)$ zodanig dat $f_k = (I + \mathbf{diag}(\gamma_f(k))\bar{\Delta}_f)\hat{f}_k$ met $\|\bar{\Delta}_f\|_2 \leq 1$. Op deze manier worden door γ_k verschillende wegingsfactoren toegewezen aan de elementen van de vector f_k. Onder de veronderstelling dat het FDD schema op elk tijdstip $(\hat{f}_k, \gamma_f(k))$ produceert, proberen de actieve FTC methodes in dit proefschrift een regelaar te vinden door het volgende optimalisatie probleem op te lossen:

$$K_{AFTC}(\hat{f}, \gamma_f) = \arg \min_{K(\hat{f},\gamma_f)} \sup_{\substack{\delta \in \mathbf{\Delta} \\ \bar{\Delta}_f \in \bar{\mathbf{\Delta}}_f \\ \underline{\gamma_f} \leq \gamma_f \leq \bar{\gamma}_f}} J(\mathcal{F}_L(M(\delta, f), K(\hat{f}, \gamma_f)))$$

waarin $\bar{\mathbf{\Delta}}_f = \{\Delta \in \mathbf{\Delta}_f : \|\Delta\| \leq 1\}$, en de vectoren $\{\underline{\gamma_f}, \bar{\gamma}_f\}$ een onder- een een bovengrens definiëren op de grootte van de onzekerheid.

Hoofdstuk 4 concentreert zich op het ontwerp van lineair parameter-variërende (LPV) regelaars voor robuust actief FTC. Twee benaderingen worden voorgesteld. In Sectie 4.2 worden alleen sensor en actuator fouten beschouwd. Daarin wordt een deterministische benadering ontwikkeld die bestaat uit het off-line ontwerp van een verzameling van robuust uitgangsterugkoppeling LPV regelaars, waarin de enige *scheduling* parameter $\gamma_f(k)$ is, d.w.z. de grootte van de FDD onzekerheid. Deze verzameling van LPV regelaars wordt zodanig opgebouwd dat elke regelaar met één specifiek fout scenario overeenkomt. Nadat een fout is gediagnosticeerd door de FDD schema, wordt als regelaar een geschaalde versie van een van de LPV regelaars gebruikt. Hoewel er slechts een eindig aantal LPV regelaars off-line is ontworpen, behandelt dit her-configuratie schema iedere willekeurige combinatie van multiplicatieve sensor en actuator fouten zolang het systeem stabiliseerbaar en detecteerbaar blijft. Deze benadering is gebaseerd op LMIs die de structuur van de onzekerheid negeren, en is dus conservatief.

Om het conservatisme van de deterministische LPV benadering te omzeilen wordt er in Sectie 4.3 een probabilistische benadering voorgesteld op basis van de resultaten van Hoofdstuk 2. Deze LPV benadering tot robuust uitgangsterugkoppeling heeft de volgende voordelen:

- de regelaar hangt zowel van de fout schattingen $\hat{f}_k$ als van de grootte van hun onzekerheid $\gamma_f(k)$ af,
- het gaat met gestructureerd onzekerheid om,
- het is niet alleen toepasbaar op sensor en actuator fouten, maar ook op component fouten.

In het uitgangsterugkoppeling geval maakt deze benadering gebruikt van de twee-stappen procedure van Hoofdstuk 3 die daar gebruik werd om de BMI optimalisatie te initialiseren.

De LPV methodes voor robuust actief FTC zijn met name heel geschikt voor online implementatie vanwege het feit dat het regelaarontwerp off-line gebeurt. Dit resulteert in minder online berekeningen voor de her-configuratie van de regelaar na het optreden van een fout. De LPV benadering van Sectie 4.3 wordt later in Hoofdstuk 6 getest op een experimentele opstelling bestaand uit een borsteloze DC motor. Daar wordt de FTC benadering gekoppeld aan een FDD schema voor de detectie en schatting van parameter- en sensor fouten.

In Hoofdstuk 5 wordt een eindige horizon uitgangsterugkoppeling MPC benadering voorgesteld die robuust is voor model- en FDD onzekerheid. Deze aanpak is gebaseerd op een combinatie van een Kalman filter en een eindige horizon MPC in een min-max (*worst-case*) optimalisatieprobleem. Dit wordt dan op ieder tijdstip opgelost met behulp van de probabilistische methode van Hoofdstuk 2 onder de veronderstelling dat de covariantie matrix van de toestand bekend is. Om deze te bepalen zijn er twee methodes voorgesteld. In de eerste methode is de doel om het spoor van de covariantie matrix te minimaliseren onder de beperking dat deze compatibel is voor alle waardes van de onzekerheid. Deze methode maakt ook gebruik van de probabilistische ellipsoïde benadering uit Hoofdstuk 2 en is daardoor veeleisend ten aanzien van de nodige berekeningen. Om de berekeningen te verminderen wordt er een tweede methode voorgesteld die veel sneller is maar ook veel conservatiever. Het complete MPC algoritme heeft als voordeel dat het het robuuste uitgangsterugkoppeling probleem benadert zonder BMI optimalisatie problemen op te hoeven lossen. Nadeel is de rekencomplexiteit en het feit dat de stabiliteit van de gesloten-lus niet gegarandeerd is.

De hierboven besproken passieve en actieve FTC benaderingen richten zich alleen op een local lineair model in de aanwezigheid van onzekerheid in de beschrijving van het model en in de FDD schema. De vraag hoe deze uitgebreid kunnen worden om met een *multiple-model* representatie van een niet-lineair systeem om te kunnen gaat is veel ingewikkelder en is in dit proefschrift slechts gedeeltelijk beschouwd. In het bijzonder, voor het geval wanneer er geen onzekerheid in het model is, is een methode in Hoofdstuk 7 ontwikkeld die gebruikt kan worden voor het regelen van niet-lineaire systemen die door een aantal locale modellen worden beschreven. Het uitgangspunt is het opbouwen van een model set $\mathcal{M}$ die bestaat of uit locale lineaire approximaties van een niet-lineair systeem of uit modellen die verschillende fout regimes van een (lineair) systeem voorstellen. Het niet-lineaire systeem wordt dan op elk tijdstip voorgesteld door een convexe combinatie (met wegingsfactoren μ_i) van de locale lineaire modellen. Deze methode is gebaseerd op een *multiple model estimator* dat de locale toestanden en de globale toestand schat alsmede de wegingen $\hat{\mu}_i$. Deze worden vervolgens gebruikt in een MPC. De *multiple model estimator* bestaat uit een verzameling van Kalman filters, een voor elk model, die onafhankelijk van de MPC regelaar zijn ontworpen. Als er wel onzekerheid is in het systeem (en dus ook in de locale modellen) kan het ontwerp van de toestandsschatter echter niet ontkoppeld worden van het ontwerp van de regelaar als gevolg van het feit dat in dit geval het welbekende *separation principle* niet meer geldig is. Dus de vraag hoe we met onzekerheid in de locale modellen om kunnen gaan blijft open en moet in de toekomst nader worden bekeken.

Notation

$\mathbb{R}$	the space of real numbers
$\mathbb{R}^+$	the set of positive real numbers
$\mathbb{R}^n$	the space of real-valued vectors of dimension n
$\mathbb{R}^{n\times m}$	the space of real-valued $n \times m$ matrices
$\mathbb{C}$	the space of complex numbers
$\mathcal{R}^{n\times m}$	the set of rational transfer matrices
$\mathcal{RH}_\infty$	the set of stable real rational transfer matrices
$\mathcal{L}_2$	the space of square-integrable signals
A^T	the transpose of the matrix A
A^*	the complex conjugate transpose of the matrix A
A^{-1}	the inverse of the matrix A
$A^{\frac{1}{2}}$	the symmetric positive-definite square root of the matrix A
$A^\dagger$	the pseudo-inverse of the matrix A
$A > 0$ $(A \geq 0)$	The symmetric matrix A is positive (semi)definite
$A < 0$ $(A \leq 0)$	The symmetric matrix A is negative (semi)definite
I_n	the identity matrix of dimension $n \times n$
$I_{n\times m}$	the $n \times m$ matrix with ones on the main diagonal
$\doteq$	equal by definition
$\|x\|_2$	the vector/signal 2-norm of x
$\|x\|_W^2$	$(= x^T W x)$ the weighted vector 2-norm
$\|A\|_F$	the Frobenius norm of the matrix A
$\langle A, B\rangle$	$\doteq \mathbf{trace}(A^T B)$
$\lambda(A)$	the eigenvalues of the matrix A

$\sigma(A)$	the singular values of the matrix A
$\mathbf{vol}A$	the volume of the closed set A
$\lceil a \rceil$	the minimum integer number larger than or equal to a
$\star$	entries in LMIs implied by symmetry
$\bullet$	entries in matrices of no importance
$\mathbf{\Pi}^{-}[A]$	the projection onto the cone of symmetric negative definite matrices
$\mathbf{\Pi}^{+}[A]$	the projection onto the cone of symmetric positive definite matrices
$Sym(A)$	$\doteq A + A*$
$\bigoplus_{i=1}^{n} A_i$	the direct sum of the matrices A_i, $i = 1, 2, \ldots, n$
$A \bigotimes B$	the Kronecker product of the matrices A and B
$\mathbf{co}\{\mathcal{S}\}$	the convex hull of the set $\mathcal{S}$
$\mathcal{F}_L(M, K)$	the lower LFT of the transfer matrices M and K
$\mathcal{F}_L(M, \Delta)$	the upper LFT of the transfer matrices M and Δ
$\lVert M \rVert_2$	the $\mathcal{H}_2$-norm of the system M
$\lVert M \rVert_\infty$	the $\mathcal{H}_\infty$-norm of the system M
$\mathbf{diag}\{A_1, \ldots, A_n\}$	the block-diagonal matrix with the matrices A_i on the main diagonal
$\mathbf{diag}\{x\}$	the diagonal matrix with off-diagonal entries equal to zero and the voctor x on its main diagonal
$\mathbf{trace}(A)$	the trace of the matrix A
$\det(A)$	the determinant of the matrix A
$\lim$	the determinant of the matrix A
$\min$	minimum
$\max$	maximum
$\sup$	supremum
$\inf$	infenium
$\in$	belongs to
$\subseteq$	subset of
$\bigcup$	union

$\bigcap$	intersection
$\square$	end of proof
$\mathcal{N}(\bar{x}, S)$	random Gaussian process with mean $\bar{x}$ and covariance matrix S

List of Abbreviations

BDCM	Brushless DC Motor
BMI	Bilibear Matrix Inequality
CR	Controller Reconfiguration,
CRLLS	Constraint Robust Linear Least-Squares Problem
DK	Alternating Coordinate Method
EA	Ellipsoid Algorithm
EMF	Electro-Magnetic Force
EsA	Eigenstructure Aassignment
FDD	Fault Detection and Diagnosis
FTC	Fault-Tolerant Control
FTCS	Fault-Tolerant Control System
GPC	Generalized Predictive Control
IMM	Interacting Multiple Model
iPC	Integral Predictive Control
LFT	Linear Fractional Transformation
LLS	Linear Least Squares
LMI	Linear Matrix Inequality
LPV	Linear Parameter-Varying
LQ	Linear Quadratic
LGG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
MC	Method of Centers

MM	Multiple Model
MMAC	Multiple-Model Adaptive Control
MPC	Model Predictive Control
NMI	Nonlinear Matrix Inequality
PATH	Path-Following Method
PI	Proportional-Integral
PIM	Pseudo-Inverse Method
PMF	Perfect Model Following
PWM	Pulse-Width modulated
PWL	Piecewise Linear
RDA	Remote Data Access
RLS	Recursive Least Squares
RM	Reconfiguration Mechanism
RMA	Rank Minimization Approach
SDP	Semi-Definite Programming
SIA	Subgradient Iteration Algorithm
SISO	Single-Input Single-Output
SRM	Space Robotic Manipulator

References

Ahmed-Zaid, F., Ioannou, P., Gousman, K., Rooney, R., 1991. Accomodation of failures in the F-16 aircraft using adaptive control. IEEE Control Systems Magazine , 73–78.

Apkarian, P., Adams, R., 1998. Advanced gain-scheduling techniques for uncertain systems. IEEE Transactions on Control Systems Technology 6(1), 21–32.

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

Askari, J., Heiming, B., Lunze, J., 1999. Controller reconfiguration based on a qualitative model: A solution of three-tanks benchmark problem. In: Proceedings of the 5th European Control Conference (ECC'99). Karlsruhe, Germany.

Astrom, K., Albertos, P., Blanke, M., Isidori, A., Schaufelberger, W., Sanz, R., 2001. Control of Complex Systems. Springer Verlag.

Athans, M., Castanon, D., Dunn, K., Greene, C., Lee, W., Sandell, N., Willsky, A., 1977. The stochastic control of the F-8C aircraft using multiple model adaptive control (MMAC) method - part 1: Equilibrium flight. IEEE Transations on Automatic Control 22(5), 768–780.

Badgwell, T., 1997. Robust model predictive control of stable linear systems. International Journal of Control 68(4), 797–818.

Balas, G., Doyle, J., Glover, K., Packard, A., Smith, R., 1998. μ-Analysis and Synthesis Toolbox - For Use with Matlab. MUSYN Inc. and MathWorks Inc.

Ballé, P., Fischer, M., Füssel, Nelles, O., Isermann, R., 1998. Integrated control, diagnosis and reconfiguration of a heat exchanger. IEEE Control Systems Magazine 18(3), 52–63.

Basseville, M., 1998. On-board component fault detection and isolation using the statistical local approach. Automatica 34(11), 1391–1415.

Basseville, M., Nikiforov, V., 1993. Detection of Abrupt Changes - Theory and Application. Prentice-Hall, Englewood Cliffs, N.J.

Battaini, M., Dyke, S., 1998. Fault tolerant structural control systems for civil engineering applications. Journal of Structural Control 5(26).

BBC World , 12 December 2001. Chernobyl head sacked over misused funds. `http://news.bbc.co.uk/go/em/fr/-/1/hi/world/europe/1707392.stm`.

Belkharraz, A., Sobel, K., 2000. Fault tolerant flight control of control surface failures. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA.

Bemporad, A., Borrelli, F., Morari, M., 2002. Model predictive control based on linear programming – the explicit solution. IEEE Transactions on Automatic Control 47(12), 1974–1985.

Bemporad, A., Garulli, A., 2000. Output-feedback predictive control of constrained linear systems via set-membership state estimation. International Journal of Control 73(8), 655–665.

Bemporad, A., Morari, M., 1999. Robust Model Predictive Control: A Survey, robustness in identification and control, A. Garulli, A. Tesi, and A. Vicino, Eds., number 245 in lecture notes in control and information sciences Edition. Springer-Verlag, Philadelphia, PA.

Bennani, S., van der Sluis, R., Schram, G., Mulder, J., 1999. Control law reconfiguration using robust linear parameter varying control. In: AIAA Guidance, Navigation, and Control Conference and Exhibit. Portland, pp. 977–987.

Beran, E., Vandenberghe, L., Boyd, S., 1997. A global BMI algorithm based on the generalized benders decomposition. In: Proceedings of the European Control Conference (ECC'97). Brussels, Belgium.

Blanke, M., 1996. Consistent design of dependable control systems. Control Engineering Practice 4(9), 1305–1312.

Blanke, M., Bøgh, S., Jørgenson, R., Patton, R., 1995. Fault detection for a diesel engine actuator – a benchmark for FDI. Control Engineering Practice 3(12), 1731–1740, `http://www.control.auc.dk/ftc/html/actuator_.html`.

Blanke, M., Frei, C., Kraus, F., Patton, R., Staroswiecki, M., July 2000. What is fault tolerant control? In: Proceedings of the 4th Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS'00. Budapest, Hungary, pp. 40–51.

Blanke, M., Izadi-Zamanabadi, R., Bøgh, S., Lunau, C., 1997. Fault-tolerant control systems - a holistic view. Control Engineering Practice 5(5), 693–702.

Blanke, M., Izadi-Zamanabadi, R., Lootsma, T., 1998. Fault monitoring and reconfigurable control for a ship propulsion plant. International Journal of Adaptive Control and Signal Processing 12, 671–688.

Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., 2003. Diagnosis and Fault-Tolerant Control. Springer Verlag, Heidelberg.

Blanke, M., Staroswiecki, M., Wu, N. E., 2001. Concepts and methods in fault-tolerant control. In: Tutorial in American Control Conference. Arlington, VA, USA.

Blom, H., Bar-Shalom, Y., 1988. The interacting multiple model algorithm for systems with markovian switching coefficients. IEEE Transactions on Automatic Control 33(8), 780–783.

Bodson, M., Groszkiewicz, J., 1997. Multivariable adaptive control algorithms for reconfigurable flight control. IEEE Transactions on Control Systems Technology 5(2).

Bolognani, S., Zordan, M., Zigliotto, M., 2000. Experimental fault-tolerant control of a pmsm drive. IEEE Transactions on Industrial Electronics 47(5), 1134–1141.

Bonivento, C., Marconi, L., Paoli, A., Rossi, C., 2003a. A framework for reliability analysis of complex diagnostic systems. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 567–572.

Bonivento, C., Paoli, A., Marconi, L., 2001a. Direct fault-tolerant control approach for a winding machine. In: 9th Mediterranean IEEE Conference on Control and Automation (MED01).

Bonivento, C., Paoli, A., Marconi, L., 2001b. Fault-tolerant control for a ship propulsion systems. In: Proceedings of the 6th European Control Conference (ECC'01). Porto, Portugal, pp. 1964–1969.

Bonivento, C., Paoli, A., Marconi, L., 2003b. Fault-tolerant control of the ship propulsion system benchmark. Control Engineering Practice 11(5), 483–492.

Bošković, J., Li, S., Mehra, R., 1999. Intelligent control of spacecraft in the presence of actuator failures. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA.

Bošković, J., Li, S., Mehra, R., 2000a. A decentralized fault-tolerant scheme for flight control applications. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA.

Bošković, J., Li, S., Mehra, R., 2000b. Fault-tolerant contol of spacecraft in the presence of sensor bias. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA.

Bošković, J., Mehra, R., 1998. A multiple model-based reconfigurable flight control system design. In: Proceedings of the 37th IEEE Conference on Decision and Control (CDC'98). Tampa, Florida, USA, pp. 4503–4508.

Bošković, J., Mehra, R., 2003. Failure detection, identification and reconfiguration system for a redundant actuator assembly. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 429 – 434.

Boyd, S., Ghaoui, L., Feron, E., Balakrishnan, V., 1994. Linear Matrix Inequalities in System and Control Theory. SIAM Studies in Applied Mathematics, volume 15, Philadelphia, PA.

Buffington, J., Chandler, P., Pachter, M., 1999. On-line system identification for aircraft with distributed control effectors. International Journal of Robust and Nonlinear Control 9, 1033–1049.

Burken, J., Lu, P., Wu, Z., 1999. Reconfigurable fligth control designs with application to the X-33 vehicle. In: Proceedings of the AIAA Guidance Navigation and Control Conference. Portland, Oregon, AIAA-99-4134.

Calafiore, G., Dabbene, F., Tempo, R., 1999. Radial and Uniform Distributions in Vector and Matrix Spaces for Probabilistic Robustness, in *topics in control and its applications* (Eds. D.E. Miller and L. Qiu) Edition. Springer-Verlag, London.

Calafiore, G., Dabbene, F., Tempo, R., 2000. Randomized algorithms for probabilistic robustness with real and complex structured uncertainty. IEEE Transactions on Automatic Control 45(12), 2218–2235.

Calafiore, G., Polyak, B., 2001. Stochastic algorithms for exact and approximate feasibility of robust LMIs. IEEE Transactions on Automatic Control 46(11), 1755–1759.

Campo, L., Bar-Shalom, Y., Li, X., 1996. Control of discrete-time hybrid stochastic systems, international series on advances in control and dynamic systems, vol. 76, (c.t. leondes, ed.) Edition. Academic Press.

Casavola, A., Giannelli, M., Mosca, E., 2000. Min-max predictive control strategies for input-saturated polytopic uncertain systems. Automatica 36, 125–133.

Chang, B., Baipai, G., Kwatny, H., 2001. A regulator design to address actuator failures. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA.

Chang, T., 2000. Reliable control for systems with block-diagonal feedback structure. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA, pp. 829–833.

Chen, J., Patton, R., 1999. Robust model-based fault diagnosis for dynamic systems. Kluwer.

Chen, J., Patton, R., 2001. Fault-tolerant control systems design using the linear matrix inequality approach. In: Proceedings of the 6th European Control Conference (ECC'01). Porto, Portugal.

Chen, J., Patton, R., Chen, Z., 1998a. Linear matrix inequality formulation of fault-tolerant control systems design. In: Proceedings of the IFAC Worshop On-line Fault Detection & Supervision in the Chemical Process Industries. Lyon.

Chen, J., Patton, R., Chen, Z., 1998b. An lmi approach to fault-tolerant control of uncertain systems. In: Proceedings of 1998 IEEE ISIC/CIRA/ISAS Joint Conference. Gaithersburg, MD, USA, pp. 14–17.

Chen, L., Narendra, K., 2001. Nonlinear adaptive control using neural networks and multiple models. Automatica 37(8), 1245–1255.

Chen, X., Zhou, K., 1998. Order statistics and probabilistic robust control. Systems & Control Letters 35(3), 175–182.

Chilali, M., Gahinet, P., Apkarian, P., 1999. Robust pole placement in LMI regions. IEEE Transactions on Automatic Control 44(12), 2257–2269.

Cho, Y., Bien, Z., 1989. Reliable control via an additive redundant controller. International Journal of Control 50(1), 385–398.

Clarke, D., Mohtadi, C., 1989. Properties of generalized predictive control. Automatica 25(6), 859–875.

Collins, E., Sadhukhan, D., Watson, L., 1999. Robust controller synthesis via non-linear matrix inequalities. International Journal of Control 72(11), 971–980.

Cuzzola, F., Ferrante, A., 2001. Explicit formulas for LMI-based $\mathcal{H}_2$ filtering and deconvolution. Automatica 37(9), 1443–1449.

Dabbene, F., 1999. Randomized Algorithms for Probabilistic Robustness Analysis and Design, ph.d. thesis Edition. Politecnico di Torino.

Dardinier-Maron, V., Hamelin, F., Noura, H., 1999. A fault-tolerant control design against major actuator failures: Application to a three tank system. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA.

Demetriou, M., 2001a. Adaptive reorganization of switched systems with faulty actuators. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA.

Demetriou, M., 2001b. Utilization of lmi methods for fault tolerant control of a flexible cable with faulty actuators. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA.

Diao, Y., Passino, K., 2001. Fault tolerant stable adaptive fuzzy/neural control for a turbine engine. IEEE Transactions on Control Systems Technology 9(3), 494–509.

Diao, Y., Passino, K., 2002. Intelligent fault-tolerant control using adaptive and learning methods. Control Engineering Practice 10(8), 801–817.

Dionísio, R., Moska, E., Lemos, J., Shirley, P., 2003. Adaptive fault tolerant control with adaptive residual generation. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 283–288.

Doyle, J., 1983. Synthesis of robust controllers and filters. In: Proceedings of the 22th Conference on Decision and Control (CDC'83). San Antonio, Texas, USA.

Eberhardt, R., Ward, D., 1999. Indirect adaptive flight control system interactions. International Journal of Robust and Nonlinear Control 9, 1013–1031.

Fabri, S., Kadirkamanathan, V., 1998a. Adaptive gain-scheduling with modular networks. In: In proceedings of the UKACC International Conference on Control'98, 1. Wales, pp. 44–48.

Fabri, S., Kadirkamanathan, V., 1998b. A self-organized multiple model approach for neural adaptive control of jump non-linear systems. In: In preprints of the IFAC Symposium on Adaptive Systems for Control and Signal Processing. Glasgol, UK.

Fei, J., Chen, S., Tao, G., Joshi, S., 2003. A discrete-tome robust adaptive actuator failure compensation control scheme. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 423–428.

Ferreira, P., 2002. Tracking with sensor failures. Automatica 38(9), 1621–1623.

Forsgren, A., 2000. Optimality conditions for nonconvex semidefinite programming. Mathematical Programming 88(1), 105–128.

Frank, P., Ding, S., Köppen-Seliger, B., July 2000. Current developments in the theory of FDI. In: Proceedings of the 4th Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS'00. Budapest, Hungary, pp. 16–27.

Fray, C., Kuntze, H., Giesen, K., 2003. A neuro-fuzzy based fault tolerant control concept for smart multy-sensory robots. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 573 – 578.

Frei, C., Kraus, F., Blanke, M., 1999. Reconfigurability viewed as a system property. In: Proceedings of the 5th European Control Conference (ECC'99). Karlsruhe, Germany.

Fujisaki, Y., Dabbene, F., Tempo, R., 2001. Probabilistic robust design of lpv control systems. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA, pp. 2019–2024.

Fujisaki, Y., Kozawa, Y., 2003. Probabilistic robust controller design: Probable near minmax value and randomized algorithms. In: Proceedings of the 42th IEEE Conference on Decision and Control (CDC'03). Maui, Hawaii, USA.

Fukuda, M., Kojima, M., 2001. Branch-and-cut algorithms for the bilinear matrix inequality eigenvalue problem. Computational Optimization and Applications 19, 79–105.

Gahinet, P., 1996. Explicit controller formulas for LMI-based $\mathcal{H}_\infty$ synthesis. Automatica 32(7), 1007–1014.

Gahinet, P., Nemirovski, A., Laub, A., Chilali, M., 1995. LMI Control Toolbox User's Guide. MathWorks, Inc.

Ganguli, S., Marcos, A., Balas, G., 2002. Reconfigurable LPV control design for Boeing 747-100/200 longitudinal axis. In: Proceedings of the American Control Conference (ACC'02). Anchorage, AK, USA, pp. 3612–3617.

Gao, Z., Antsaklis, P., 1991. Stability of the pseudo-inverse method for reconfigurable control systems. International Journal of Control 53(3), 717–729.

Gao, Z., Antsaklis, P., 1992. Reconfigurable control system design via perfect model-following. International Journal of Control 56(4), 783–798.

Gaspar, P., Szaszi, I., Bokor, J., 2003. Fault-tolerant control structure to prevent the rollover of heavy vehicles. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 465–470.

Ge, J., Frank, P., 1995. Stochastic stability for discrete time linear active fault tolerant control systems. In: Proceeding of the 8th International Symposium on System Modeling and Control. Zakopane, Poland, pp. 273–278.

Ge, J., Frank, P., Lin, C., 1996. Design of reliable controller for state delayed systems. European Journal of Control 2, 239–248.

Ge, J., Lin, C., 1996. $\mathcal{H}_\infty$ control for discrete-tome active fault tolerant control systems. In: Proceeding of the 13th Triennial World Congress of IFAC. San Fransisco, USA, pp. 439–444.

Gehin, A., Staroswiecki, M., 1999. A formal approach to reconfigurablity analysis application to the three tank benchmark. In: Proceedings of the 5th European Control Conference (ECC'99). Karlsruhe, Germany.

Geromel, J., 1999. Optimal linear filtering under parameter uncertainty. IEEE Transactions on Signal Processing 47(1), 168–175.

Geromel, J., Bernussou, J., Garcia, G., Oliveira, M., 2000. $\mathcal{H}_2$ and $\mathcal{H}_\infty$ robust filtering for discrete-time linear systems. SIAM Journal on Control and Optimization 38(5), 1353–1368.

Geromel, J., Bernussou, J., Oliveira, M., 1999. $\mathcal{H}_2$-norm optimization with constrained dynamic output feedback controllers: Decentralized and reliable control. IEEE Transactions on Automatic Control 44(7), 1449–1454.

Geromel, J., Oliveira, M., 2001. $\mathcal{H}_2$ and $\mathcal{H}_\infty$ robust filtering for convex bounded uncertain systems. IEEE Transactions on Automatic Control 46(1), 100–107.

Gertler, J., 1998. Fault Detection and Diagnosis in Engineering Systems. Marcel Dekker, New Jork.

Gertler, J., 2000. Designing dynamic consistancy relations for fault detection and isolation. International Journal of Control 73(8), 720–732.

Goh, K.-C., Safonov, M., Papavassilopoulos, G., 1994. A global optimization approach for the BMI problem. In: Proceedings of the 33d Conference on Decision and Control (CDC'94). Lake Buena Vista, FL, USA, pp. 2009–2014.

Goodwin, G., Graebe, F., Salgado, M., 2001. Control System Design. Prentice Hall, Upper Saddle River, New Jersey.

Gopinathan, M., Bošković, J., Mehra, R., Rago, C., 1998. A multiple model predictive scheme for fault-tolerant flight control design. In: Proceedings of the 37th IEEE Conference on Decision and Control (CDC'98). Tampa, Florida, USA, pp. 1376–1381.

Griffin, G., Maybeck, P., 1997. MMAE/MMAC control for bending with multiple uncertain parameters. IEEE Transations on Aerospace and Electronic Systems 33(3), 903–911.

Griffiths, B., Loparo, K., 1995. Optimal control of jump linear gaussian systems. International Journal of Control 42(4), 791–819.

Grigoradis, K., Skelton, R., 1996. Low-order control design for LMI problems using alternating projection methods. Automatica 32(8), 1117–1125.

Grötschel, M., Lovász, L., Schrijver, A., 1988. Geometric Algorithms and Combinatorial Optimization. Springer-Verlag, Berlin, Germany.

Hajiyev, C., Caliskan, F., 2003. Fault Diagnosis and Reconfiguration in Flight Control Systems. Kluwer Academic Publishers, Boston.

Hamada, Y., Shin, S., Sebe, N., 1996. A design method for fault-tolerant control systems based on $\mathcal{H}_\infty$ optimization. In: Proceedings of the 35th Conference on Decision and Control (CDC'96). Kobe, Japan, pp. 1918–1919.

Hassibi, A., How, J., Boyd, S., 1999. A path-following method for solving BMI problems in control. In: Proceedings of the American Control Conference (ACC'99). San Diego, California, USA, pp. 1385–1389.

Heiming, B., Lunze, J., 1999. Definition of the three-tank benchmark problem for controller reconfiguration. In: Proceedings of the 5th European Control Conference (ECC'99). Karlsruhe, Germany, `http://www.control.auc.dk/ftc/html/others.html`.

Ho, L., Yen, G., 2001. Reconfigurable control system design for fault diagnosis and reconfiguration. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA.

Hol, C., Scherer, C., der Meché, E. V., Bosgra, O., 2003. A nonlinear SPD approach to fixed order controller synthesis and comparison with two other methods applied to an active suspension system. European Journal of Control 9(1), 11–26.

Hsieh, C., 2002. Performance gain margins of the two-stage LQ reliable control. Automatica 38(11), 1985–1990.

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

Huzmezan, M., Maciejowski, J., 1997. Reconfigurable control methods and related issues - a survey. Tech. Rep. Technical report prepared for the DERA under the Research Agreement no. ASF/3455, Department of Engineering, University of Cambridge.

Huzmezan, M., Maciejowski, J., 1998a. Automatic tuning for model based predictive control during reconfiguration. In: Proceedings of AERO. Seoul, South Korea.

Huzmezan, M., Maciejowski, J., 1998b. A novel strategy for fault tolerant control. In: COSY Workshop. Mulhouse, France.

Huzmezan, M., Maciejowski, J., 1998c. Reconfiguration and scheduling in flight using quasi-LPV high-fidelity models and MBPC control. In: Proceedings of the American Control Conference (ACC'98). Philadelphia, USA.

Huzmezan, M., Maciejowski, J., 1999. Reconfigurable flight control during actuator failures using predictive control. In: Proceedings of the 14th Triennial World Congress of IFAC. Beijing, China.

Ibaraki, S., Tomizuka, M., 2001. Rank minimization approach for solving BMI problems with random search. In: Proceedings of the American Control Conference (ACC'01). Arlington, VA, USA, pp. 1870–1875.

Ikeda, K., Shin, S., 1998. Fault tolerance of autonomous decentralized adaptive control systems. International Journal of Systems Science 29(7), 773–782.

Isermann, R., Ballé, P., 1997. Trends in the application of model-based fault detection and diagnosis of technical processes. Control Engineering Practice 5(5), 709–719.

Iwasaki, T., 1999. The dual iteration for fixed order control. IEEE Transations on Automatic Control 44(4), 783–788.

Iwasaki, T., Rotea, M., 1997. Fixed order scaled $\mathcal{H}_\infty$ synthesis. Optimal Control Applications & Methods 18(6), 381–398.

Iwasaki, T., Skelton, R., 1995. The XY-centering algorithm for the dual LMI problem: A new approach to fixed order control design. International Journal of Control 62(6), 1257–1272.

Izadi-Zamanabadi, R., Blanke, M., 1999. A ship propulsion system model for fault-tolerant control. Control Engineering Practice 7(2), 227–239, `http://www.control.auc.dk/ftc/html/ship_propulsion_.html`.

Izadi-Zamanabadi, R., Staroswiecki, M., 2000. A structural analysis method formulation for fault-tolerant control system design. In: Proceedings of the 39th IEEE Conference on Decision and Control (CDC'00). Sydney, Australia.

Jakubek, S., Jorgl, H., 2000. Fault-diagnosis and fault-compensation for non-linear systems. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA, pp. 3198–3202.

Jiang, B., Staroswiecki, M., Cocquempot, V., 2003. Active fault tolerant control for a class of nonlinear systems. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 127–132.

Jiang, J., Zhang, Y., 2002. Graceful performance degradation in active fault-tolerant control systems. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Johansen, T., 1994. Operating Regime based Process Modeling and Identification, ph.d. thesis, itk-report 94-109-w Edition. The Norwegian Institute of Technology, University of Trondheim, available via `http://www.itk.ntnu.no/ansatte/Johansen_Tor.Arne/public.html`.

Johansen, T., Foss, B., 1995. Identification of non-linear system structure and parameters using regime decomposition. Automatica 31(2), 321–326.

Johansson, M., 1999. Piecewise Linear Control Systems, ph.d. thesis Edition. Lund University, Department of Automatic Control.

Johansson, M., Rantzer, A., 1998. Computation of piecewise quadratic lyapunov functions for hybrid systems. IEEE Transactions on Automatic Control 43(4), 555–559.

Jonckheere, E., Lohsoonthorn, P., 2000. A geometric approach to model matching reconfigurable propulsion control. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA, pp. 2388–2392.

Jones, C., 2002. Reconfigurable flight control first year report. Tech. rep., Control Group, Department of Engineering, University of Cambridge.

Joshi, S., 1997. Design of failure accommodating multiloop LQG-type controllers. IEEE Transations on Automatic Control 32(8), 740–741.

Kanev, S., Scherer, C., Verhaegen, M., Schutter, B. D., 2003a. A BMI optimization approach to robust output-feedback control. In: to appear in Proceedings of the 41th IEEE Conference on Decision and Control (CDC'03). Maui, Hawaii, USA.

Kanev, S., Scherer, C., Verhaegen, M., Schutter, B. D., 2003b. Robust output-feedback controller design via local BMI optimization. accepted in Automatica .

Kanev, S., Schutter, B. D., Verhaegen, M., 2002. The ellipsoid algorithm for probabilistic robust controller design. In: Proceedings of the 41th IEEE Conference on Decision and Control (CDC'02). Las Vegas, Nevada, USA.

Kanev, S., Schutter, B. D., Verhaegen, M., 2003c. An ellipsoid algorithm for probabilistic robust controller design. Systems & Control Letters 49(5), 365–375.

Kanev, S., Verhaegen, M., 2000a. A bank of reconfigurable lqg controllers for linear systems subjected to failures. In: Proceedings of the 39th IEEE Conference on Decision and Control (CDC'00). Sydney, Australia.

Kanev, S., Verhaegen, M., 2000b. Controller reconfiguration for non-linear systems. Control Engineering Practice 8(11), 1223–1235.

Kanev, S., Verhaegen, M., 2001. An approach to the isolation of sensor and actuator faults based on subspace identification. In: ESA Workshop on "On-Board Autonomy". Noordwijk, The Netherlands.

Kanev, S., Verhaegen, M., 2002. Reconfigurable robust fault-tolerant control and state estimation. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Kanev, S., Verhaegen, M., 2003a. Combined FDD and robust active FTC for a brushless DC motor. submitted to Control Engineering Practice .

Kanev, S., Verhaegen, M., 2003b. Controller reconfiguration in the presence of uncertainty in the fdi. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA.

Kanev, S., Verhaegen, M., 2003c. Robust output-feedback integral MPC: A probabilistic approach. submitted to Automatica .

Kanev, S., Verhaegen, M., 2003d. Robust output-feedback integral mpc: A probabilistic approach. In: to appear in Proceedings of the 41th IEEE Conference on Decision and Control (CDC'03). Maui, Hawaii, USA.

Kanev, S., Verhaegen, M., Nijsse, G., 2001. A method for the design of fault-tolerant systems in case of sensor and actuator faults. In: Proceedings of the 6th European Control Conference (ECC'01). Porto, Portugal.

Keating, M., Pachter, M., Houpis, C., 1997. Fault tolerant flight control system: QFT design. International Journal of Robust and Non-Linear Control 7(6), 551–559.

Kececi, E., Tang, X., Tao, G., 2003a. Adaptive actuator failure compensation for concurrently actuated manipulators. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 411–416.

Kececi, E., Tang, X., Tao, G., 2003b. Adaptive actuator failure compensation for cooperating multiple manipulator systems. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 417–422.

Kerrigan, E., Maciejowski, J., 1999. Fault-tolerant control of a ship propulsion system using model predictive control. In: Proceedings of the 5th European Control Conference (ECC'99). Karlsruhe, Germany.

Khalil, H., 1996. Nonlinear Systems. Prentice Hall, Upper Saddle River, NJ, Second edition.

Kim, S., Kim, Y., Kim, H., Nam, C., 2001a. Adaptive reconfigurable flight control system based on recursive system identification. In: Proceedings of the JSASS 15th International Sessions in 39th Aircraft Symposium. Gifu, Japan.

Kim, Y., Rizzoni, G., Utkin, V., 2001b. Developing a fault-tolerant power-train control system by integrating design of control and diagnistics. International Journal of Robust and Non-Linear Control 11, 1095–1114.

Kinnaert, M., 1989. Adaptive generalized predictive controller for mimo systems. International Journal of Control 50(1), 161–172.

Kinnaert, M., 2003. Fault diagnosis based on analytical models for linear and nonlinear systems - a tutorial. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 37–50.

Konstantopoulos, I., Antsaklis, P., 1995. An optimization strategy to reconfigurable control systems. Tech. Rep. Technical Report ISIS-95-006, University of Notre Dame, ISIS group.

Konstantopoulos, I., Antsaklis, P., 1996a. An eigenstructure assignment approach to control reconfiguration. In: Proceedings of 4th IEEE Mediterranean Symposium on Control and Automation. Greece.

Konstantopoulos, I., Antsaklis, P., 1996b. Eigenstructure assignment in reconfigurable control systems. Tech. Rep. Technical Report ISIS-96-001, University of Notre Dame, ISIS group.

Konstantopoulos, I., Antsaklis, P., 1999. An optimization approach to control reconfiguration 9(3), 255–270.

Kose, I., Jabbari, F., 1999a. Control of LPV systems with partly measured parameters. IEEE Transactions on Automatic Control 44(3), 658–663.

Kose, I., Jabbari, F., 1999b. Robust control of linear systems with real parametric uncertainty. Automatica 35(4), 679–687.

Kothare, M., Balakrishnan, V., Morari, M., 1996. Robust constrained model predictive control using linear matrix inequalities. Automatica 32(10), 1361–1379.

Kovácsházy, T., Péceli, G., Simon, G., 2001. Transient reduction in reconfigurable control systems utilizing structure dependence. In: Proceedings of Instrumentation and Measurement Technology Conference. Budapest, Hungary, pp. 1143–1147.

Kulhavy, R., Kraus, F., 1996. On duality of regularized exponential and linear forgetting. Automatica 32(10), 1403–1415.

Kushner, H., Yin, J., 1997. Stochastic Approximation Algorithms and Applications. Springer-Verlag, New York.

Lee, B.-K., Efsani, M., 2003. Advanced simulation model for brushless DC motor drives. Journal of Electric Power Components and Systems 31(9).

Leibfritz, F., 2001. An LMI-based algorithm for designing suboptimal static $\mathcal{H}_2/\mathcal{H}_\infty$ output feedback controllers. SIAM Journal on Control & Optimization 39(6), 1711–1735.

Leibfritz, F., Mostafa, E., 2002. An intereior point constraint trust region method for a special class of nonlinear semidefinite programming problems. SIAM Journal on Optimization 12(4), 1047–1074.

Lemos, J., Rato, L., Marques, J., 1999. Switching reconfigurable control based on hidden markov models. In: Proceedings of the American Control Conference (ACC'99). San Diego, California, USA.

Li, D., Fukushima, M., 2001. On the convergence of the BFGS method for nonconvex unconstrained optimization problems. SIAM Journal on Optimization 11(4), 1054–1064.

Li, X., 1996. Hybrid estimation techniques, c. t. leondes, ed., control and dynamic systems: advances in theory and applications, vol. 76 Edition. Academic Press, San Diego.

Li, X., McInroy, J., Hamann, J., 1999. Optimal fault tolerant control of flexure jointed hexapods for applications requiring less than six degrees of freedom. In: Proceedings of the American Control Conference (ACC'99). San Diego, California, USA.

Liang, Y., Liaw, D. C., Lee, T. C., 2000. Reliable control of non-linear systems. IEEE Transactions on Automatic Control 45(4), 706–710.

Liao, F., Wang, J., Yang, G., 2002. Reliable robust tracking control: An LMI approach. IEEE Transations on Control Systems Technology 10(1), 76–89.

Liaw, D., Y.Liang, 2002. Quadratic polynomial solutions to the hamilton-jacobi inequality in reliable control design. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E81-A(9), 1860–1866.

Liberzon, D., Tempo, R., 2003. Gradient algorithms for finding common Lyapunov functions. In: Proceedings of the 42th IEEE Conference on Decision and Control (CDC'03). Maui, Hawaii, USA.

Liu, G., Patton, R., 1998. Eigenstructure Assignment for Control Systems Design. John Wiley & Sons.

Liu, W., 1996. An on-line expert system-based fault-tolerant control system. Expert Systems with Applications 11(1), 59–64.

Liu, X., Zhang, H., Liu, J., Yang, J., 2000. Fault detection and diagnosis of permanent-magnet dc motor based on parameter estimation and neural network. IEEE Transationa on Industrial Electronics 47(5), 1021–1030.

Looze, D., Weiss, J., Eterno, J., Barrett, N., 1985. An automatic redesign approach for restructurable control systems. IEEE Control Systems Magazine 5(2), 16–22.

Lopez-Toribio, C., Patton, R., Daley, S., 1999. Supervisory Takagi-Sugeno fuzzy fault-tolerant control of a rail traction system. In: Proceedings of the 14th Triennial World Congress of IFAC. Beijing, China.

Maciejowski, J., Jones, C., 2003. MPC fault-tolerant flight control case study: Flight 1862. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 121–126.

Maghami, P., Sparks, D., Lim, K., 1998. Fault accommodation in control of flexible systems. Journal of Guidance, Control, and Dynamics 21(3), 500–507.

Mahmoud, M., Jiang, J., Zhang, Y., 1999. Analysis of the stochastic stability for fault tolerant control systems. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA.

Mahmoud, M., Jiang, J., Zhang, Y., 2000a. Optimal control law for fault tolerant control systems. In: Proceedings of the 39th IEEE Conference on Decision and Control (CDC'00). Sydney, Australia.

Mahmoud, M., Jiang, J., Zhang, Y., 2000b. Stochastic stability of fault tolerant control systems in the presence of noise. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA.

Mahmoud, M., Jiang, J., Zhang, Y., 2000c. Stochastic stability of fault tolerant control systems in the presence of model uncertainties. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA.

Mahmoud, M., Jiang, J., Zhang, Y., 2001. Stochastic stability analysis of fault-tolerant control systems in the presence of noise. IEEE Transactions on Automatic Control 46(11), 1810–1815.

Mahmoud, M., Jiang, J., Zhang, Y., 2002. Stability of fault tolerant control systems driven by actuators with saturation. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Mahmoud, M., Jiang, J., Zhang, Y., 2003. Active Fault Tolerant Control Systems: Stochastic Analysis and Synthesis. Springer-Verlag, Berlin.

Mahmoud, M., Xie, L., 2000. Positive real analysis and synthesis of uncertain discrete-time systems. IEEE Transactions on Circuits and Systems-Part I: Fundamental Theory and Applications 47(3), 403–406.

Maki, M., Jiang, J., Hagino, K., 2001. A stability guaranteed active fault-tolerant control system against actuator failures. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA.

Marcos, A., Ganguli, S., Balas, G., 2003. New strategies for fault tolerant control and fault diagnostic. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 277 – 282.

Marcu, T., Matcovschi, M., Frank, P., 1999. Neural observer-based approach to fault-tolerant control of a three-tank system. In: Proceedings of the 5th European Control Conference (ECC'99). Karlsruhe, Germany.

Masubuchi, I., Ohara, A., Suda, N., 1998. LMI-based controller synthesis: A unified formulation and solution. International Journal of Robust and Nonlinear Control 8(8), 669–686.

Maybeck, P., 1999. Multiple model adaptive algorithms for detecting and compensating sensor and actuator/surface failures in aircraft flight control systems. International Journal of Robust and Non-Linear Control 9, 1051–1070.

Maybeck, P., Stevens, R., 1991. Reconfigurable flight control via multiple model adaptive control methods. IEEE Trans. on Aerospace and Electronic Systems 27(3), 470–479.

McDowell, D., Irwin, G., Lightbody, G., McConnell, G., 1997. Hybrid neural adaptive control for bank-to-turn missiles. IEEE Transactions on Control Systems Technology 5(3), 297–308.

Médar, S., Chabonnaud, P., Noureddine, F., 2002. Active fault accommodation of a three tank system via switching control. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Mesic, S., Verdult, V., Verhaegen, M., Kanev, S., 2003. Estimation and robustness analysis of actuator faults based on kalman filtering. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA.

Mohamed, S., El-shafei, A., Bahgat, A., Hallouda, M., 1997. Adaptive-control and fault-diagnosis implementation using an industrial computer system. In: Proceedings of the American Control Conference (ACC'97). New Mexico, USA, pp. 57–61.

Morse, W., Ossman, K., 1990. Model-following reconfigurable flight control system for the AFTI/F-16. Journal of Guidance, Control, and Dynamics 13(6), 969–976.

Moseler, O., Isermann, R., 2000. Application of model-based fault detection to a brushless DC motor. IEEE Transations on Industrial Electronics 47(5), 1015–1020.

Murad, G., Postlethwaite, I., Gu, D., 1996. A robust design approach to integrated controls and diagnostics. In: Proceeding of the 13th Triennial World Congress of IFAC. San Fransisco, USA, pp. 199–204.

Musgrave, J., Guo, T., Wong, E., Duyar, A., 1997. Real-time accommodation of actuator faults on a reusable rocket engine. IEEE Transactions on Control Systems Technology 5(1), 100–109.

Naredra, K., Balakrishnan, J., 1997. Adaptive control using multiple models. IEEE Transactions on Automatic Control 42(2), 171–187.

Niemann, H., Stoustrup, J., 2002. Reliable control using the primary and dual Youla parameterizations. In: Proceedings of the 41th IEEE Conference on Decision and Control (CDC'02). Las Vegas, Nevada, USA, pp. 4353–4358.

Niemann, H., Stoustrup, J., 2003. Passive fault tolerant control of a double inverted pendulum – a case study example. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 1029–1034.

Nikolaou, M., 2001. Model Predictive Controllers: A Critical Synthesis of Theory and Industrial Needs, in Advances in Chemical Engineering Series Edition. Academic Press.

Niksefat, N., Sepehri, N., 2002. A QFT fault-tolerant control for electrohydraulic positioning systems. IEEE Transactions on Control Systems Technology 10(4), 626–632.

Noura, H., Bastogne, T., Dardinier-Maron, V., 1999. A general fault tolerant control approach: Application to a winding machine. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA.

Noura, H., Sauter, D., Hamelin, F., Theilliol, 2000. Fault-tolerant control in dynamic systems: Application to a winding machine. IEEE Control Systems Magazine 20(1), 33–49.

NTSB, 1979. Aircraft accident report - american airlines, inc. DC-10-10. Tech. Rep. NTSB-AAR-79-17, National Transpotration Safety Board, USA.

Öhrn, K., Ahlén, A., Sternad, M., 1995. Order statistics and probabilistic robust control. IEEE Transactions on Automatic Control 40(3), 405–418.

Oishi, Y., Kimura, H., 2001. Randomized algorithm to solve parameter dependent linear matrix inequalities and their computational complexity. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA, pp. 2025–2030.

Oliveira, M., Bernussou, J., Geromel, J., 1999a. A new discrete-time robust stability condition. Systems & Control Letters 37(4), 261–265.

Oliveira, M., Geromel, J., Bernussou, J., 1999b. An LMI optimization approach to multiobjective controller design for discrete-time systems. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA, pp. 3611–3616.

Palhares, R., de Souza, C., Peres, P., 1999. Robust $\mathcal{H}_\infty$ filter design for uncertain discrete-time state-delayed systems: An LMI approach. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA.

Palhares, R., Ramos, D., Peres, P., 1996. Alternative LMIs characterization of $\mathcal{H}_2$ and central $\mathcal{H}_\infty$ discrete-time controllers. In: Proceedings of the 35th Conference on Decision and Control (CDC'96). Kobe, Japan, pp. 1459–1496.

Palhares, R., Takahashi, R., Peres, P., 1997. $\mathcal{H}_2$ and $\mathcal{H}_\infty$ guaranteed costs computation for uncertain linear systems. International Journal of Systems Science 28(2), 183–188.

Park, D., Jun, B., 1992. Selfperturbing recursive least squares algorithm with fast tracking capability. Electronics Letters 28(6), 558 – 559.

Patton, R., 1997. Fault tolerant control: the 1997 situation. In: Proceedings of the 3th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'97). Hull University, Hull, UK, pp. 1033–1054.

Peres, P., Palhares, R., 1995. Optimal $\mathcal{H}_\infty$ state feedback control for discrete-time linar systems. In: Second Latin American Seminar on Advanced Control, LASAC'95. Chile, pp. 73–78.

Peters, M., Stoorvogel, A., 1994. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control in a stochastic framework. Linear Algebra and its Applications 205-206, 971–996.

Pillay, P., Krishnan, R., 1989. Modeling, simulation, and analysis of permanent-magnet motor drives, part II: The brushless DC motor drive. IEEE Transactions on Industry Applications 25(2), 274–279.

Piug, V., Quevedo, J., 2001. Fault-tolerant PID controllers using a passive robust fault diagnosis approach. Control Engineering Practice 9, 1221–1234.

Podder, T., Surkar, N., 2001. Fault-tolerant control of an autonomous underwater vehicle under thurster redundancy. Robotics and Autonomous Systems 34, 39–52.

Polyak, B., Tempo, R., 2001. Probabilistic robust design with linear quadratic regulators. Systems & Control Letters 43(5), 343–353.

Ponsart, J., Join, C., Theilliol, D., Sauter, D., 2001. Sensor fault diagnosis and accommodation in nonlinear system. In: Proceedings of the 6th European Control Conference (ECC'01). Porto, Portugal.

Qu, Z., Ihlefeld, C., Yufang, J., Saengdeejing, A., 2001. Robust control of a class of nonlinear uncertain systems - fault tolerance against sensor failures and subsequent recovery. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA.

Rantzer, A., Johansson, M., 1997. Piecewise quadratic optimal control. In: Proceedings of the American Control Conference (ACC'97). New Mexico, USA.

Rato, L., Lemos, M., 1999. Multimodel based fault tolerant control of the 3-tank system. In: Proceedings of the 5th European Control Conference (ECC'99). Karlsruhe, Germany.

Rauch, H., 1994. Intelligent fault diagnosis and control reconfiguration. IEEE Control System Magazine 14(3), 6–12.

Rauch, H., 1995. Autonomous control reconfiguration. IEEE Control Systems Magazine 15(6), 37–48.

Safonov, M., Goh, K., Ly, J., 1994. Control system synthesis via bilinear matrix inequalities. In: Proceedings of the American Control Conference (ACC'94). Baltimore, USA, pp. 45–49.

Schdeier, G., Frank, P., 1999. Fault-tolerant ship propulsion control: sensor fault detection using a non-linear observer. In: Proceedings of the 5th European Control Conference (ECC'99). Karlsruhe, Germany.

Scherer, C., 1996. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control for time-varying and linear parametrically-varying systems. International Journal of Robust and Nonlinear Control 6(9-10), 929–952.

Scherer, C., Gahinet, P., Ghilali, M., 1997. Multiobjective output-feedback control via LMI optimization. IEEE Transactions on Automatic Control 42(7), 896–911.

Schram, G., Copinga, G., Bruijn, P., Verbruggen, H., 1998. Failure-tolerant control of aircraft: a fuzzy logic approach. In: Proceedings of the American Control Conference (ACC'98). Philadelphia, USA, pp. 2281–2285.

Scokaert, P., Mayne, D., 1998. Min-max feedback model predictive control for constrained linear systems. IEEE Transactions on Automatic Control 43(8), 1136–1142.

Seo, C., Kim, B., 1996. Design of robust reliable $\mathcal{H}_\infty$ output feedback control for a class of uncertain lineat systems with sensor failure. International Journal of Systems Science 27(10), 963–968.

Seron, M., Goodwin, G., Doná, J., 1996. Eigenstructure assignment in reconfigurable control systems. Tech. Rep. Technical Report ISIS-96-001, University of Notre Dame, ISIS group.

Shin, J., Belcasrto, C., 2003. Analysis of a fault tolerant control system: False fault detection case. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 289 – 294.

Shin, J., Wu, N., Belcasrto, C., 2002. Linear parameter varying control synthesis for actuator failures based on estimated parameters. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference. Monterey, California, USA.

Siwakosit, W., Hess, R., 2001. Multi-input/multi-output reconfigurable flight control design. Journal of Guidance, Control, and Dynamics 24(6), 1079–1088.

So, C., Leung, S., 2001. Variable forgetting factor rls algorithm based on dynamic equation of gradient of mean square error. IEE Electronics Letters 37(3), 202–203.

Somov, Y., Kozlov, A., Rayevsky, V., Anshakov, G., Antonov, Y., 2002. Nonlinear dynamic research of the spacecraft robust fault-tolerant control systems. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Staroswiecki, M., 2002. On reconfigurability with respect to actuator failures. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Staroswiecki, M., Hoblos, G., Aitouche, A., 1999. Fault tolerance analysis of sensor systems. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA.

Stengel, R., 1991. Intelligent failure-tolerant control. IEEE Control Systems Magazine 11(4), 14–23.

Stengel, R., Ray, L., 1991. Stochastic robustness of linear time-invariant control systems. IEEE Transactions on Automatic Control 36(1), 82–87.

Stoustrup, J., Grimble, M. J., Niemann, H., 1997. Design of integrated systems for the control and detection of actuator/sensor faults. Sensor Review 139-149, 138–149.

Stoustrup, J., Niemann, H., 2001. Fault tolerant feedback control using the youla parametrization. In: Proceedings of the 6th European Control Conference (ECC'01). Porto, Portugal.

Suyama, K., 2002. What is reliable control? In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Suyama, K., Zhang, F., 1997. A new type reliable control system using decision by majority. In: Proceedings of the American Control Conference (ACC'97). New Mexico, USA.

Suzuki, T., Tomizuka, M., 1999. Joint synthesis of fault detector and controller based on structure of two degree of freedom control system. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA, pp. 3599–3604.

Tao, G., Chen, S., Joshi, S., 2002a. An adaptive actuator failure compensation controller using output feedback. IEEE Transactions on Automatic Control 47(3), 506–511.

Tao, G., Chen, S., Joshi, S., 2002b. An adaptive control scheme for systems with unknown actuator failures. Automatica 38(6), 1027–1034.

Tao, G., Ma, X., Joshi, S., 2000a. Adaptive output tracking control of systems with actuator failures. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA, pp. 2654–2658.

Tao, G., Ma, X., Joshi, S., 2000b. Adaptive state feedback control of systems with actuator failures. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA.

Tao, G., Ma, X., Joshi, S., 2001. Adaptive state feedback and tracking control of systems with actuator failures. IEEE Transactions on Automatic Control 46(1), 78–95.

Tempo, R., Dabbene, F., 2001. Randomized Algorithms for Analysis and Control of Uncertain Systems: An Overview, in perspectives in robust control - lecture notes in control and information science ,(ed. s.o. moheimani) Edition. Springer-Verlag, London.

Theilliol, D., Noura, H., Sauter, D., 1998. Fault-tolerant control method for actuator and component faults. In: Proceedings of the 37th IEEE Conference on Decision and Control (CDC'98). Tampa, Florida, USA, pp. 604–609.

Theilliol, D., Ponsart, J., Noura, H., Sauter, D., 2001. Sensor fault-tolerant control method based on multiple model approach. In: Proceedings of the 6th European Control Conference (ECC'01). Porto, Portugal, pp. 1981–1986.

Theilliol, D., Sauter, D., Ponsart, J., 2003. A multiple model based approach for fault tolerant control in non linear systems. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 151 – 156.

Toker, O., Özbay, H., 1995. On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In: Proceedings of the American Control Conference (ACC'95). Seattle, WA, USA, pp. 2525–2526.

Toplis, B., Pasupathy, S., 1988. Tracking improvements in fast rls algorithms using a variable forgetting factor. IEEE Transactions on Acoustics, Speech, and Signal Processing 36(2), 206–227.

Tortora, G., Kouvaritakis, B., Clarke, D., 2002. Optimal accomodation of faults in sensors and actuators. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Tuan, H., Apkarian, P., 2000. Low nonconvexity-rank bilinear matrix inequalities: Algorithms and applications in robust controller and structure designs. IEEE Transations on Automatic Control 45(11), 2111–2117.

Tuan, H., Apkarian, P., Hosoe, S., Tuy, H., 2000a. D.C. optimization approach to robust control: Feasibility problems. International Journal of Control 73(2), 89–104.

Tuan, H., Apkarian, P., Nakashima, Y., 2000b. A new Lagrangian dual global optimization algorithm for solving bilinear matrix inequalities. International Journal of Robust and Nonlinear Control 10, 561–578.

Tyler, M., Morari, M., 1994. Optimal and robust design of integrated control and diagnostic modules. In: Proceedings of the American Control Conference (ACC'94). Baltimore, USA, pp. 2060–2064.

Ugrinovskii, V., 2001. Randomized algorithms for robust stability and control of stochastic hybrid systems with uncertain switching policies. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA, pp. 5026–5031.

van Schrik, D., 2002. Fault-tolerant control menagement - a conceptual view -. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

VanAntwerp, J., Braatz, R., 2000. A tutorial on linear and bilinear matrix inequalities. Journal of Process Control 10, 363–385.

VanAntwerp, J., Braatz, R., Sahinidis, N., 1997. Globally optimal robust control for systems with time-varying nonlinear perturbations. Computers and Chemical Engineering 21, S125–S130.

Veillette, R., 1992. Design of reliable control systems. IEEE Transactions on Automatic Control 37(3), 290–304.

Veillette, R., 1995. Reliable linear-quadratic state-feedback control. Automatica 31(1), 137–143.

Verdult, V., Kanev, S., Breeman, J., Verhaegen, M., 2003. Estimating multiple sensor and actuator scaling faults using subspace identification. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA.

Verhaegen, M., 1994. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. Automatica 30(1), 61–74.

Verhaegen, M., Verdult, V., 2003. Filtering and System Identification: An Introduction, lecture notes for the course sc4040 (et4094) Edition. TU-Delft.

Vidyasagar, M., 1998. Statistical learning theory and randomized algorithms for control. IEEE Control Systems Magazine 18(6), 69–85.

Vinnicombe, G., 1999. Measuring the Robustness of Feedback Systems, ph.d. thesis Edition. University of Cambridge, Department of Engineering.

Visinski, M., Cavallaro, J., Walker, I., 1995. A dynamic fault tolerance framework for remote robots. IEEE Transactions on Robotics and Automation 11(4), 477–490.

Watanabe, K., Tzafestas, S., 1989. Multiple model adaptive control for jump linear stochastic system. Internatioal Journal of Control 50(5), 1603–1617.

Wise, K., Brinker, J., Calise, A., Enns, D., Elgersma, M., Vougaris, P., 1999. Direct adaptive reconfigurable flight control for a tailless advanced fighter aircraft. International Journal of Robust and Nonlinear Control 9, 999–1012.

Wu, N., 1993. Reconfigurable control design: Achieving stability robustness and failures tracking. In: Proceedings of the 32th Conference on Decision and Control (CDC'93). San Antonio, Texas, USA.

Wu, N., 1997a. Reliability of reconfigurable control systems: A fuzzy set theoretic approach. In: Proceedings of the 36th Conference on Decision and Control (CDC'97). San Diago, California, USA, pp. 3352–3356.

Wu, N., 1997b. Robust feedback design with optimized diagnostic performance. IEEE Transactions on Automatic Control 42(9), 1264–1268.

Wu, N., 2001a. Reliability of fault tolerant control systems: Part I. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA.

Wu, N., 2001b. Reliability of fault tolerant control systems: Part II. In: Proceedings of the 40th IEEE Conference on Decision and Control (CDC'01). Orlando, Florida, USA.

Wu, N., Patton, R., 2003. Reliability and supervisory control. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 139–144.

Wu, N., Thavimani, S., Zhang, Y., Blanke, M., 2003. Sensor fault masting of a ship propulsion system. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 435–440.

Wu, N., Zhang, Y., Zhou, K., 2000a. Detection, estimation, and accommodation of loss of control effectiveness. International Journal of Adaptive Control and Signal Processing 14, 775–795.

Wu, N., Zhou, K., Salomon, G., 2000b. Control reconfigurability of linear time-invariant systems. Automatica 36(11), 1767–1771.

Wu, N., Zhou, K., Salomon, G., July 2000c. On reconfigurability. In: Proceedings of the 4th Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS'00. Budapest, Hungary, pp. 846–851.

Xie, L., Fu, M., Souza, C., 1992. $\mathcal{H}_\infty$ control and quadratic stabilization of systems with parameter uncertainty via output feedback. IEEE Transactions on Automatic Control 37(8), 1253–1256.

Yamada, Y., Hara, S., 1998. Global optimization for the $\mathcal{H}_\infty$ control with constant diagonal scaling. IEEE Transactions on Automatic Control 43(2), 191–203.

Yamada, Y., Hara, S., Fujioka, H., 2001. ϵ-feasibility for $\mathcal{H}_\infty$ control problems with constant diagonal scalings. Transactions of the Society of Instrument and Control Engineers E-1(1), 1–8.

Yamé, J., Kinnaert, M., 2003. Performance-based switching for fault-tolerant control. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 555–560.

Yang, G., Wang, J., Soh, Y., 1999. Reliable LQG control with sensor failures. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA.

Yang, Z., Blanke, M., 2000a. Adaptive control mixer method for nonlinear control reconfiguration: A case study. In: Proceedings of the IFAC Symposium on System Identification (SYSID'00). Santa Barbara, California, USA.

Yang, Z., Blanke, M., 2000b. The robust control mixer module method for control reconfiguration. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA.

Yang, Z., Hicks, D., 2002. Reconfigurability of fault-tolerant hybrid control systems. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Yang, Z., Stoustrup, J., 2000. Robust reconfigurable control for parametric and additive faults with FDI uncertainties. In: Proceedings of the 39th IEEE Conference on Decision and Control (CDC'00). Sydney, Australia.

Yen, G., 1994. Reconfigurable learning control in large space structures. IEEE Transactions on Control Systems Technology 2(4), 362–370.

Yen, G., Ho, L., 2000. Fault tolerant control: An intelligent sliding mode control. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA, pp. 4204–4308.

Zhang, P., Ding, S., Wang, G., Jeinsch, T., Zhou, D., 2002. Application of robust observer-based FDI systems to fault-tolerant control. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Zhang, Y., Jiang, J., 1999a. Design of integrated fault detection, diagnosis and reconfigurable control systems. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA.

Zhang, Y., Jiang, J., 1999b. An interacting multiple-model based fault detection diagnosis and fault-tolerant control approach. In: Proceedings of the 38th IEEE Conference on Decision and Control (CDC'99). Phoenix, Arizona, USA.

Zhang, Y., Jiang, J., 2000. Design of proportional-integral regonfigurable control systems via eigenstructure assignment. In: Proceedings of the American Control Conference (ACC'00). Chicago, Illinois, USA.

Zhang, Y., Jiang, J., 2001. Integrated active fault-tolerant control using IMM approach. IEEE Transactions on Aerospace and Electronic Systems 37(4).

Zhang, Y., Jiang, J., 2002. Design of restructurable active fault-tolerant control systems. In: Proceedings of the 15th Triennial World Congress of IFAC (b'02). Barcelona, Spain.

Zhang, Y., Jiang, J., 2003. Bibliographical review on reconfigurable fault-tolerant control system. In: Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'2003). Washington D.C., USA, pp. 265–276.

Zhang, Y., Li, X., 1998. Detection and diagnosis of sensor and actuator failures using IMM estimator. IEEE Transactions on Aerospace and Electronic Systems 34(4).

Zhao, G., Jiang, J., 1998. Reliable state feedback control system desgin agains actuator failures. Automatica 34(10), 1267–1272.

Zhou, D., Frank, P., 1998. Fault diagnosis and fault tolerant control. IEEE Transactions on Aerospace and Electronic Systems 34(2), 420–427.

Zhou, K., 2000. A new controller architecture for high performance, robust, and fault tolerant control. In: Proceedings of the 39th IEEE Conference on Decision and Control (CDC'00). Sydney, Australia.

Zhou, K., Doyle, J., 1998. Essentials of Robust Control. Prentice-Hall.

Zhou, K., Khargonekar, P., Stoustrup, J., Niemann, H., 1995. Robust performance of systems with structured uncertainties in state-space. Automatica 31(2), 249–255.

Zhou, K., Ren, Z., 2001. A new controller architecture for high performance, robust, and fault-tolerant control. IEEE Transactions on Automatic Control 46(10), 1613–1618.

Curriculum Vitae

Stoyan Kanev was born on the 26th of June 1975 in the city of Sofia. He obtained his secondary school diploma from the High School of Mathematics and Informatics, Sofia, in 1993. In 1998 he received his M.Sc. degree in Control and Systems Engineering from the Technical University of Sofia, Bulgaria. The work on his master's thesis and the defense took place at the Control Lab of the Delft University of Technology, the Netherlands. In the period October 1999 until December 2003 he was a Ph.D. student at the University of Twente, Netherlands. As of January 2004 he works as a postdoctoral researcher in the Delft Center for Systems and Control at the Delft University of Technology. His research interests include fault-tolerant control, randomized algorithms, linear and bilinear matrix inequalities.