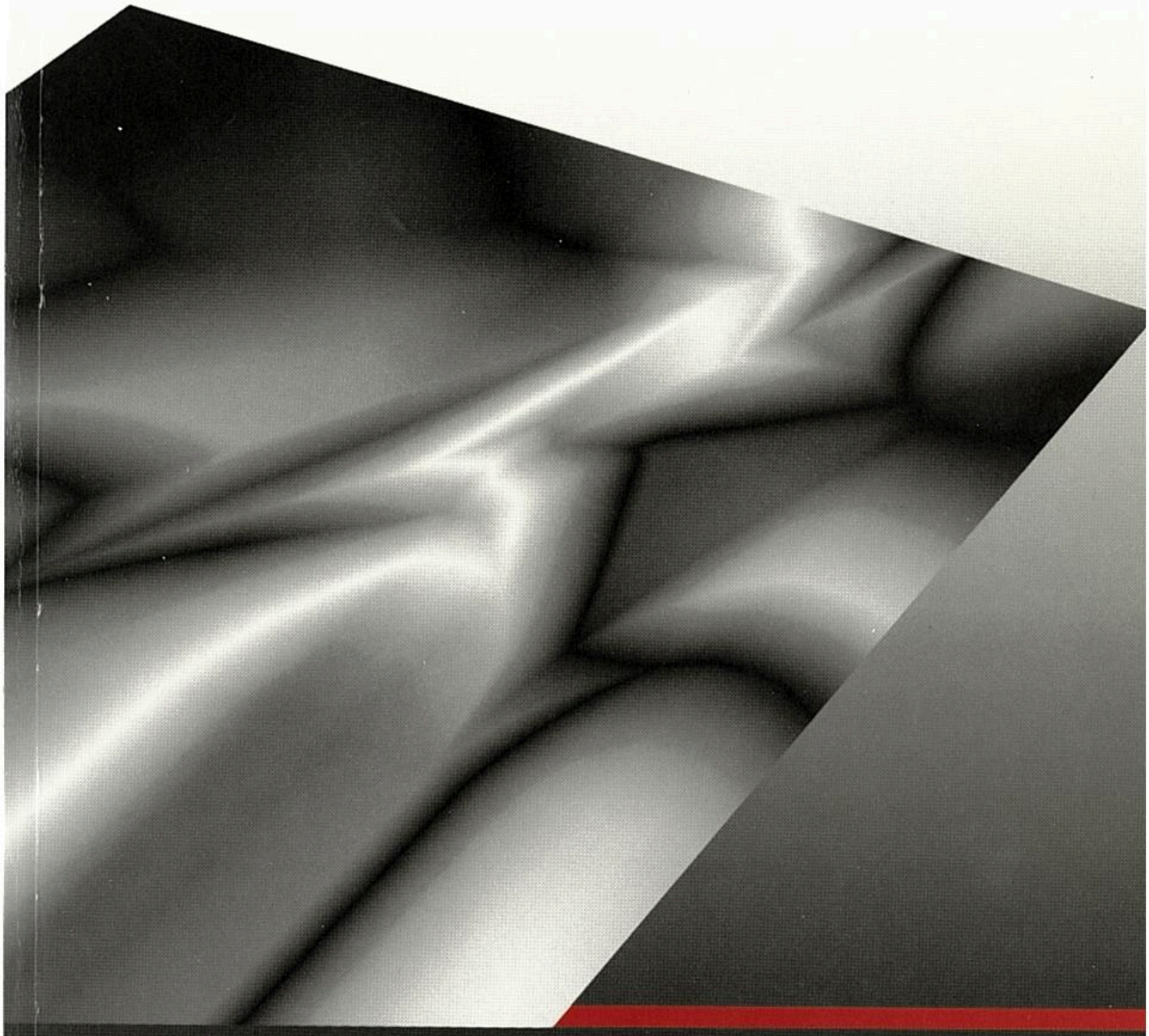
Algebraic Optimization with Applications to System Theory



Dorina Jibetean

VRIJE UNIVERSITEIT

Algebraic Optimization with Applications to System Theory

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. T. Sminia,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op woensdag 11 juni 2003 om 13.45 uur
in het auditorium van de universiteit,
De Boelelaan 1105

door

Dorina Jibetean

geboren te Alba Iulia, Roemenië

promotor: prof.dr.ir. J.H. van Schuppen
copromotor: dr. B. Hanzon

Algebraic Optimization with Applications to System Theory

Dorina Jibetean

Cover design by Simona Orzan
Printed by Ponsen & Looijen BV
ISBN 90 6196 521 7

THOMAS STIELTJES INSTITUTE
FOR MATHEMATICS

The research reported in this thesis has been carried out at the Centre for Mathematics and Computer Science (CWI) Amsterdam. It has been financially supported by the Netherlands Organisation for Scientific Research (NWO) and partly by the Thomas Stieltjes Institute for Mathematics.

Acknowledgments

This thesis is the result of a four year period that I spent at the Centre for Mathematics and Computer Science (CWI) in Amsterdam. There are many people at CWI who deserve to be acknowledged. Above all, I would like to thank my close collaborators, Jan van Schuppen, the promoter of the thesis, and Bernard Hanzon, the co-promoter of the thesis and my main advisor. The principal line of investigation in the thesis reflects our common research interests.

Occasional discussions led sometimes to new sections in the thesis. For that, I would like to mention Ralf Peeters, my main collaborator outside CWI, who has been involved in the project at all times, and Luc Habets, Anton Stoorvogel and Michiel Hazewinkel for the discussions we had.

The thesis has been greatly improved due to the very careful reading and detailed comments of Jan van Schuppen and Bernard Hanzon and, at a later stage, by comments of the reading committee consisting of Laurent Baratchart, Michael Möller, André Ran and Anton Stoorvogel.

Finally, I wish to thank all my friends at CWI and elsewhere for turning these four years into a very enjoyable experience and for offering their help and support whenever needed.

Amsterdam, April 2003

Contents

Acknowledgements **i**

1 Introduction **1**

2 Background material **7**

2.1 Solving polynomial equations 7

2.1.1 Ideals and varieties 7

2.1.2 Gröbner bases 8

2.1.3 Stetter-Möller method 10

2.2 Counting solutions in $\mathbf{R}^n$ 12

2.2.1 Sylvester and Sylvester-Habicht sequences 13

2.2.2 A few results of real algebraic geometry 16

2.3 Hilbert's 17th problem 17

3 Global optimization of polynomial functions **19**

3.1 Numerical algorithms based on LMI relaxations 20

3.1.1 The method of Shor-Parrilo 20

3.1.2 The method of Lasserre 24

3.2 An exact algebraic method 24

3.2.1 Construction of an auxiliary polynomial 25

3.2.2 Computing the minimum 30

3.2.3 Case: the polynomial p has a minimum 35

3.2.4 Case: the polynomial p does not have a minimum 41

3.2.5 Examples 42

3.2.6 Conclusions 44

3.3 Families of polynomials 45

3.4 Conclusions 51

4 Global optimization of rational functions **53**

4.1 Unconstrained optimization of rational functions 53

4.1.1 An equivalent formulation 53

4.1.2 An exact solution 56

4.1.3 A numerical solution using LMI's 60

4.2 Constrained optimization of rational functions 63

4.2.1 A particular case 64

4.2.2 Numerical methods for the general case 68

4.3 Conclusions 68

5 The H_2 model reduction problem 71

5.1 Computing the H_2 norm 72

5.1.1 Continuous-time systems 72

5.1.2 Discrete-time systems 72

5.2 The H_2 distance and the optimization problem 72

5.2.1 Continuous-time systems 73

5.2.2 Discrete-time systems 73

5.3 H_2 model reduction: continuous-time SISO case 75

5.4 H_2 model reduction: continuous-time MIMO case 81

5.5 Analysis on a lower bound for the H_2 model reduction 83

5.5.1 An equivalent formulation of the H_2 criterion 84

5.5.2 A relaxation of the H_2 criterion 87

5.5.3 A comparison of the exact H_2 criterion with its relaxation 89

5.6 Conclusions 92

6 Other applications of optimization algorithms 93

6.1 Model Reduction w.r.t. Divergence Rate Criterion 93

6.1.1 Introduction 94

6.1.2 Problem formulation 95

6.1.3 Procedure for infimization of divergence rate 96

6.1.4 Algebraic method 97

6.1.5 Examples 99

6.1.6 Conclusions 101

6.1.7 Appendix 102

6.2 Systems with uncertainties 105

6.2.1 Problem formulation 106

6.2.2 Computing the worst case H_2 norm 109

6.2.3 Example 112

6.2.4 Conclusions 114

6.3 Global identifiability analysis 115

6.3.1 Local identifiability 116

6.3.2 Global identifiability 117

6.3.3 Algorithms for obtaining partial solutions of systems of polynomial equations and inequalities 118

6.3.4 Conclusions 122

7 Conclusions and directions for further research 123

Samenvatting 125

References 127

Chapter 1

Introduction

A large collection of problems in system identification and control theory can be reformulated as optimization problems. That is, given an *objective* (or *criterion*) function, one needs to compute its optimal value (minimum or maximum) attained on a given domain (the *feasibility domain*).

In this thesis, we are interested in algorithms for *global* optimization. It is well known that, except rather specific classes of problems, global optimization is still widely open as far as algorithms are concerned. In fact, "in practical problems, even if the problem which is posed is of a global character, we often are content with using results and algorithms from local optimization ... for lack of something else. " ([33]). The theory of global optimization has developed in two main directions. On the one hand, attempts were made for designing algorithms which approximate the optimum of a general, arbitrary function. Since such algorithms use no information on the criterion to be optimized, they are also called black-box optimization algorithms. This is the case of probabilistic methods, interval methods, homotopy methods, etcetera. On the other hand, there are the algorithms 'specialized' to particular classes of problems. The specific properties of the functions in such a class are exploited in order to derive theoretical results and better computational time.

Consider for example the so-called *convex optimization problems.* These are optimization problems where both the feasibility set and the criterion (objective) function are convex. Convex problems have important theoretical properties; in particular, any local minimum is a global minimum . This means that any algorithm that can compute a local minimum for a convex optimization problem will compute in fact a global minimum. For convex optimization problems, powerful algorithms exist, which are guaranteed to converge to a global minimum efficiently. See for example, the *interior point methods* designed by Karmarkar (1984) for linear programming and generalized by Nesterov and Nemirovsky (1989) to a variety of convex non-linear optimization problems. In system and control theory, a particular class of convex non-linear optimization problems finds a large number of applications (see [7]). These are optimization problems over the cone of positive semidefinite symmetric matrices, more commonly known in system theory as *linear matrix inequalities* problems.

Problems which do not fall into the convex class are called *non-convex optimization* problems. In general, they are much harder than the convex optimization problems, both theoretically and practically. As far as we are concerned, there are two main issues to be considered when comparing algorithms for global optimization. Firstly, the algorithm should *guarantee* finding the *global* optimum. There exist a number of special classes of non-convex optimization (see [34]) for which algorithms, *guaranteed* to compute the global optimum, can be designed. Secondly, we are interested in the computational complexity of such algorithms. Unfortunately, except the convex optimization problems, there are very few problems which have efficient algorithms. A vast majority of problems are shown to be NP-hard, see e.g. [69]. However, for practical purposes, it is believed that design of problem-tailored algorithms might improve on the computational complexity, even if the algorithm remains impractical for large problems.

In this spirit, we have considered here two classes of functions for optimization: multivariate polynomial and rational functions. We are therefore in the area of non-convex optimization. As for the computational complexity, optimization of a multivariate polynomial is known to be an NP-hard problem. And since polynomial functions are a particular case of a rational function (where the denominator equals the constant polynomial 1), optimization of a multivariate rational function is NP-hard as well. That limits from the start our hopes for having algorithms able to deal with large problems. Nevertheless, it is important to have a procedure which computes, in principle, the global optimum. Our interest is in designing algorithms which *guarantee* finding the global solution.

Let us now state the problem more precisely for the polynomial case. Let $p \in \mathbf{R}[x_1, x_2, ..., x_n]$ be a polynomial of total degree larger than 1. Then we want to compute

$$\inf_{(x_1,\ldots,x_n)\in\mathbf{R}^n} p(x_1,\ldots,x_n).$$

Regarding the terminology, we use *infimum* (inf) instead of the more common *minimum* (min), or, *supremum* (sup) instead of *maximum* (max), simply to stress that the optimal value may not be attained in $\mathbf{R}^n$ but only approached asymptotically.

The problem of finding the infimum of a polynomial is an old one. In fact, one related question, namely deciding whether a given multivariate polynomial is nonnegative everywhere, was studied by David Hilbert more than one hundred years ago. His investigations and hypothesis led to the development of the field called *real algebraic geometry.* Various tools and results form this field, together with results from the so-called *complex algebraic geometry* were employed in this thesis for deriving our guaranteed algorithms. A brief introduction into these fields is given in Chapter 2. There we restrict ourselves to notions and results that will be later used in the thesis.

Chapter 3 starts with a review of two recently developed methods for polynomial optimization, based on convex relaxations of the problem (Sections 3.1.1, 3.1.2). One main contribution of the thesis is an algorithm for global optimization of polynomial functions which can be found in Section 3.2. When the polynomial has a minimum the algorithm returns the global minimal value and finds at least one point in every connected component of the set of minimizers. A characterization of such points is given. When the polynomial does not have a minimum the algorithm computes its infimum. No assumption is made on the polynomial. As an extra, the algorithm turns out to be *exact.* In fact, the algorithm is suited for *symbolic computations*, that is, suited for instances which are not fully specified numerically and, therefore, may depend on some parameters. The advantages of exact, symbolic computations, are not fully developed in the thesis. A few steps are made in this direction when discussing families of polynomials (Section 3.3) or exact methods for optimization of rational functions (Section 4.1.2).

In Chapter 4, several algorithms for global optimization of polynomial functions are extended in order to deal with rational functions. The extensions are based on a known result from real algebraic geometry ([9]), which is independently proved here. The problem of computing the global optimum of rational functions is certainly not new. Only [58] contains more than 1000 references concerning the so-called fractional programming, presenting, as well, several applications of such problems in economy, information theory, operation research, etcetera. Also [49] considers the rational optimization problem in a manner similar to the one of Section 4.1.3. However, all methods referred above make an assumption on the sign of the denominator of the rational function, namely that the denominator is either strictly positive or strictly negative. In this thesis, using the real algebraic geometry result, we prove that the assumption on the sign of the denominator, although relaxed to non-strict inequalities, is in fact a *necessary* condition for the rational function to have a finite optimum. We believe the connection is rather important, since it enables one to treat the general case, without making any assumptions.

The rest of the thesis discusses several applications of the algorithms developed in Chapters 3 and 4 to problems of system theory. The objects of study in system identification are approximate, simplified models of real phenomena. The process of matching a model to the observed reality bears the name of *system identification.* The models one deals with in practice are simplified, for complexity reasons. For example, one simplifying assumption is that the system is constant in time, leading to the class of time-invariant systems. Or, one could describe the dependency of the output of the system on the input of the system as being linear, leading to the class of linear systems. In this thesis we consider mainly linear and time-invariant systems.

Some immediate questions arise in the system identifiability context. The first main question is how to chose a model? In this situation, one needs to compare different models, according to a specified criterion, and choose the model that satisfies *best* the selected criterion. Typically, the criterion relates to the error that the model gives, compared to the observed data. In this case, one attempts to chose a model having *minimal* error in the chosen class of models. Thus, the need for optimization results and algorithms becomes clear. The literature on system identification is quite broad, see for example the textbooks [46], [60] and it is an active research area. A model, once it is chosen, may be used to derive information about the behavior of the real system in order to predict or control the system.

In the thesis we present a few problems for linear time-invariant systems, show how they translate into optimization of polynomial or rational functions and analyze how the algorithms of Chapters 3 and 4 perform on them. The fact that the algorithms are able to find the global optimum is rather important in these examples.

An important problem in system identification is the approximation problem which can be formulated in the following way: given a system (or model) find a less complex model which has approximately the same input-output behavior. Since in this thesis we are concerned with time-invariant, linear systems, the reduction in complexity refers to reduction of the order of the system, and this is called the *model order reduction* problem. Two linear systems of finite order have 'close' input-output behavior if the difference between their transfer functions, evaluated in some norm (or semi-norm), is small. In Chapter 5 we consider the model order reduction in H_2 norm. This problem was originally introduced in 1967 in [47] and subsequently analyzed in [2], where several results, regarding the existence and generic properties of a solution, are deduced. Since an analytic solution to the problem is not known, other papers, like [61], [35], [3] to cite just a few, are concerned with designing algorithmic solutions. These however return, in the best case, local optima. An exact method, which under certain assumptions computes the global optimum, is presented in [26]. The method is based on computing all critical points of the criterion function, therefore it is rather expensive. Since the H_2 model reduction can be rewritten as an optimization problem whose criterion is a rational function, we apply the algorithms of previous chapters for computing a *global* optimum, hence for computing the *best* approximant of a given order. This can be done in an *exact* manner, at a certain computational cost.

Chapter 6 is divided into three sections, each treating a different problem. Each section of the chapter can be read independently from the others. In Section 6.1 we treat the optimal model order reduction for stationary Gaussian systems with respect to the *divergence rate* criterion. It is shown that essentially, this reduces to an optimization of rational functions.

A completely different problem is treated in Section 6.2. When working with approximate models of the reality, one needs to take possible errors into consideration. Since the reality is in general too complex to be described by a single model, one may work with a class of models in which a *representation of uncertainty* is included. Uncertain models are the subject of the so-called *robustness analysis*. There, results on stability or performance are derived for the whole class of models (robust stability, respectively robust performance). In Section 6.2, we compute the H_2-norm (which is a performance criterion) of an uncertain model with structured uncertainty. In fact, we perform a worst-case analysis by computing the largest H_2-norm of a model in that class. The procedure can be adapted in order to derive a robust stability test.

Section 6.3 discusses the *global identifiability* of a given parameterization. This problem is closely related to a basic question in mathematics, that is, establishing the injectivity of a multivariate polynomial function. Although for the latter problem several necessary conditions are known, the famous conjecture of Jacobi is still open (for an overview, see [43] and [14], [54]). In applications, one is interested in the injectivity of an n-variate polynomial function in a given domain $\Omega \subseteq \mathbf{R}^n$. We argue that this question can be answered algorithmically, as discussed in Section 6.3.

Part of the results of this thesis have appeared as journal papers, conference papers or research reports. This is the case of Section 3.2 which has appeared as [25]. The extension to families of polynomials of Section 3.3 is new. Sections 4.1.1 and 4.1.3 formed [37], but the exact method described in 4.1.2, as well as the methods for constrained rational optimization of Section 4.2 have not been submitted for publication. In Chapter 5, the Sections 5.3 and 5.5 have appeared as [39], respectively have been submitted as [53]. In Chapter 6, Sections 6.1 and 6.3 have appeared as [40], respectively [38], but the results of Section 6.2 have not been submitted for publication.

Chapter 2

Background material

The thesis is concerned with several problems in system theory like optimal model reduction, global identifiability, systems with uncertainties. The system theoretical problems themselves may not have that much in common. However, in their mathematical formulations one will see the same questions coming up. These questions relate to polynomial or rational functions, whether that is finding the global optimum of such a function or deciding whether a certain system of polynomial equations and inequalities has solutions and if so, how many. We start therefore by presenting in this chapter known results concerning (systems of) polynomials, which will be later used in the thesis.

2.1 Solving polynomial equations

Let K be a field, in our case K is either $\mathbf{R}$, the field of real numbers, or $\mathbf{C}$, the field of complex numbers. We work with the field K whenever a certain definition can be stated or a result is valid in both fields $\mathbf{R}$ and $\mathbf{C}$, otherwise we specify K.

Studying the set of solutions of polynomial equations with n variables and coefficients in the field K, as a subset of K^n is, at a very basic level, the object of algebraic geometry. Let us introduce now a few basic notions and notations.

2.1.1 Ideals and varieties

To begin, we recall some definitions and results regarding the solution set of a system of polynomial equations. Let us denote by $K[x_1, \ldots, x_n]$ the set of all polynomials in variables $x_1, \ldots, x_n$ with coefficients in the field K.

Definition 2.1.1 *Let us consider the polynomials $f_1, \ldots, f_s \in K[x_1, \ldots, x_n]$. The set of all simultaneous solutions in K^n of a system of polynomial equations*

$$\{(x_1, \ldots, x_n) \in K^n \mid f_1(x_1, \ldots, x_n) = 0, \ldots, \; f_s(x_1, \ldots, x_n) = 0 \}$$

is called the affine variety *(or* algebraic set*) defined by $f_1, \ldots, f_s$ and it is denoted by $V(f_1, \ldots, f_s)$.*

Definition 2.1.2 *The set*

$$I = \{p_1 f_1 + \ldots + p_s f_s \ : \ p_i \in K[x_1, \ldots, x_n], \ i = 1, \ldots, s\}$$

is called the (polynomial) ideal generated *by $f_1, \ldots, f_s$. The set $\{f_1, \ldots, f_s\}$ is called a* set of generators *(or a* basis*) of I, with the notation $I =< f_1, \ldots, f_s >$.*

It is easy to see that $I =< f_1, \ldots, f_s >$ is indeed an *ideal* in $K[x_1, \ldots, x_n]$, that is, for all $g_1, g_2 \in I$ and $q_1, q_2 \in K[x_1, \ldots, x_n]$ we have $q_1 g_1 + q_2 g_2 \in I$.

In algebraic geometry one exploits the duality existing between the notion of variety, which is a geometric notion, and that of ideal, which is an algebraic notion. Given an affine variety $V \subseteq \mathbf{K}^n$, one can define the associated ideal

$$I(V) = \{f \in K[x_1, \ldots, x_n] \mid f(x) = 0, \ \forall x \in V\}.$$

Conversely, given a polynomial ideal I one can define the associated affine variety

$$V(I) = \{(x_1, \ldots, x_n) \in K^n |\ f(x_1, \ldots, x_n) = 0, \ \forall f \in I\}.$$

Note that if $I =< f_1, \ldots, f_s >$, then $V(I)$ equals $V(f_1, \ldots, f_s)$. Note that the set of generators of an ideal is in general not unique.

Proposition 2.1.3 *If $f_1, \ldots, f_s$ and $g_1, \ldots, g_t$ are bases of the same ideal in $K[x_1, \ldots, x_n]$, then $V(f_1, \ldots, f_s) = V(g_1, \ldots, g_t)$.*

This proposition shows that one has a certain freedom in describing a certain variety, i.e. the zeros of a polynomial system, and one may wonder how different bases of a given ideal compare and whether some particular basis presents advantages over another basis of the same ideal. This will be treated in the next section, but before that let us state a last important result. We have seen so far that any finite set of polynomials generates a polynomial ideal. The converse is also true, and it is one of the most important results in algebraic geometry.

Theorem 2.1.4 (Hilbert Basis Theorem) *Every ideal $I \subseteq K[x_1, \ldots, x_n]$ has a finite generating set. That is, there exists an $s \in \mathbf{N}^*$ and the polynomials $f_1, \ldots, f_s \in I$ such that $I = \langle f_1, \ldots, f_s \rangle$.*

2.1.2 Gröbner bases

Recall that our main interest here is finding the solution set of a system of polynomial equations, i.e. an affine variety. Hence our emphasis is on *computational* algebraic geometry, for which, undoubtedly, the most important tools are the Gröbner bases. Let us first recall a few definitions. This section is based on [11], Chapter 2, §2, §5, §7, and Chapter 4, §1.

Definition 2.1.5 *A* monomial ordering *on $K[x_1, \ldots, x_n]$ is any relation $>$ on $\mathbf{Z}^n_{\geq 0}$, or equivalently, any relation on the set of monomials $x^\alpha = x_1^{\alpha_1} \ldots x_n^{\alpha_n}$, $\alpha \in \mathbf{Z}^n_{\geq 0}$, satisfying:*

- *$>$ is a total ordering on $\mathbf{Z}^n_{\geq 0}$.*
- *If $\alpha > \beta$ and $\gamma \in \mathbf{Z}^n_{\geq 0}$, then $\alpha + \gamma > \beta + \gamma$.*
- *$>$ is a well-ordering in $\mathbf{Z}^n_{\geq 0}$, that is, every nonempty subset of $\mathbf{Z}^n_{\geq 0}$ has a smallest element under $>$.*

We present here two of the most used monomial orderings. Let $\alpha, \beta \in \mathbf{Z}^n_{\geq 0}$.

- **Lexicographic order** $\alpha >_{lex} \beta$ (and also $x^\alpha >_{lex} x^\beta$) if in the vector $\alpha - \beta$, the left-most nonzero entry is positive.
- **Total degree lexicographic** $\alpha >_{tdeg} \beta$ (and also $x^\alpha >_{tdeg} x^\beta$) if

$$\Big(|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i\Big) \text{ or } \Big(|\alpha| = |\beta| \text{ and } \alpha >_{lex} \beta\Big).$$

Example 2.1.6 $x_1^3 x_2^1 x_3 >_{lex} x_2^8 x_3^2$ *but* $x_1^3 x_2^1 x_3 <_{tdeg} x_2^8 x_3^2$.

Definition 2.1.7 *Let $f = \sum_\alpha a_\alpha x^\alpha$ be a nonzero polynomial in $K[x_1, \ldots, x_n]$ and let $>$ be a monomial order.*

- *The* degree *of f is* $\deg(f) = \max\{\alpha \in \mathbf{Z}^{\mathbf{n}}_{\geq \mathbf{0}} \mid a_\alpha \neq 0\}$.
- *The* leading coefficient *of f is* $\text{lc}(f) = a_{\deg(f)} \in K$.
- *The* leading monomial *of f is* $\text{lm}(f) = x^{\deg(f)}$.
- *The* leading term *of f is* $\text{lt}(f) = \text{lc}(f) \cdot \text{lm}(f)$.

Remark 2.1.8 *Occasionally we also speak about the* total degree of a polynomial, *that is* $\text{tdeg}(f) = \max\{|\alpha| = \sum_{i=1}^n \alpha_i \mid a_\alpha \neq 0\}$.

Definition 2.1.9 *A polynomial $f \in K[x_1, \ldots, x_n]$ is* homogeneous *of degree k, if, for every $\alpha \in K$,*

$$f(\alpha x_1, \ldots, \alpha x_n) = \alpha^k f(x_1, \ldots, x_n).$$

Definition 2.1.10 *Fix a monomial ordering and let $\langle \text{lt}(I) \rangle$ denote the ideal generated by the leading terms of elements of I. The set*

$$B = \{x^\alpha \mid x^\alpha \notin \langle \text{lt}(I) \rangle\}$$

is called the normal set *of I.*

Note that for an ideal $I \neq K[x_1, \ldots, x_n]$, the normal set B contains the constant monomial 1, obtained for $\alpha = (0, \ldots, 0)$. See also Example 2.1.12.

Definition 2.1.11 *Fix a monomial order. A finite subset $G = \{g_1, \ldots, g_s\}$ of an ideal I is said to be a* Gröbner basis *if $\langle \text{lt}(g_1), \ldots, \text{lt}(g_s) \rangle = \langle \text{lt}(I) \rangle$, where $\text{lt}(I)$ is the set of leading terms of elements of I.*

Example 2.1.12 *[12] Let* $G = \{x_1^2 + \frac{3}{2}x_1x_2 + \frac{1}{2}x_2^2 - \frac{3}{2}x_1 - \frac{3}{2}x_2, x_1x_2^2 - x_1, x_2^3 - x_2\}$. *It can be shown that* G *is a Gröbner basis for the ideal* $I = \langle G \rangle$ *generated by* G *with respect to the total degree lexicographical order* $x_1 > x_2$. *By examining the leading monomials of* G, *we see that* $\langle \mathrm{lt}(I) \rangle = \langle x_1^2, x_1x_2^2, x_2^3 \rangle$ *and the only monomials not lying in this ideal are those in* $B = \{1, x_1, x_2, x_1x_2, x_2^2\}$.

Proposition 2.1.13 *Fix a monomial order. Then every ideal* $I \subseteq K[x_1, \ldots, x_n]$, $I \neq \{0\}$ *has a Gröbner basis. Furthermore, any Gröbner basis for an ideal* I *is a basis for* I.

It is well known that if G is a Gröbner basis, then the remainder of the division of a polynomial f (also called the *normal form* of f) by G, obtained from the division algorithm is independent of the order of the elements in G. That is in general not true for an arbitrary basis of an ideal and this is the main reason for which Gröbner bases are so important for computational purposes.

Definition 2.1.14 *A* monic reduced Gröbner basis *for a polynomial ideal* I *is a Gröbner basis* G *for* I *such that:*

- $\mathrm{lc}(p) = 1$ *for all* $p \in G$. *(monic)*
- *For all* $p \in G$, *no monomial of* p *lies in* $\langle \mathrm{lt}(G - \{p\}) \rangle$. *(reduced)*

For monic reduced Gröbner basis the following holds.

Proposition 2.1.15 *Let* $I \neq \{0\}$ *be a polynomial ideal. Then, for a given monomial ordering,* I *has a unique monic reduced Gröbner basis.*

Buchberger algorithm and its variants are used for computing monic reduced Gröbner bases (see [18]). One can always compute in principle a monic reduced Gröbner basis using the Buchberger algorithm, however it can be extremely demanding from a computational point of view.

Next we give an immediate application of Gröbner bases to solving polynomial equations in $\mathbf{C}^n$. On an algebraically closed field, in particular on $\mathbf{C}$, the following consequence of the Weak Hilbert Nullstellensatz holds.

Theorem 2.1.16 *Let* $I \subset \mathbf{C}[x_1, \ldots, x_n]$. *Then* $V(I) = \emptyset$ *if and only if the monic reduced Gröbner basis of* I, *with respect to any monomial ordering, is* $G = \{1\}$.

Remark that on $\mathbf{R}^n$, which is not algebraically closed, this is not true. For example $I = \langle 1 + x^2 \rangle$ has an empty variety (over $\mathbf{R}$).

2.1.3 Stetter-Möller method

This section is based on [12], [27], [62]. Given a polynomial ideal I one can define the quotient space $K[x_1, \ldots, x_n]/I$. This set together with an internal addition operation and a scalar multiplication operation has a vector space structure.

The elements of this space are classes of polynomials of the form $[f] = \hat{f} + I$. If $G = \{g_1, \ldots, g_n\}$ is a Gröbner basis for I, then for every $f \in K[x_1, \ldots, x_n]$ there exists a unique $\hat{f} \in K[x_1, \ldots, x_n]$ such that $f = f_1 g_1 + \ldots + f_n g_n + \hat{f}$ and no term of $\hat{f}$ is divisible by any of the leading terms of the elements in G. $\hat{f}$ is called the remainder of the division of f by G. Obviously, the remainder is zero if and only if $f \in I$ and polynomials in the same class have the same remainder. The following theorem (Finiteness Theorem of [12]), characterizing the finite dimensional quotient spaces, is of importance for us.

Theorem 2.1.17 *Let $K \subseteq \mathbf{C}$ be a field and $I \subseteq K[x_1, \ldots, x_n]$ be an ideal. The following conditions are equivalent:*

a. The vector space $K[x_1, \ldots, x_n]/I$ is finite dimensional over K.

b. The associated variety $V(I) \subseteq \mathbf{C}^n$ is a finite set.

c. If G is a Gröbner basis for I, then for each i, $1 \leq i \leq n$, there is an $m_i \geq 0$ such that $x_i^{m_i}$ is the leading monomial of g for some $g \in G$.

The following theorem (Theorem 2.10 of [12]) gives a bound on the cardinality of $V(I)$.

Theorem 2.1.18 *Let I be an ideal in $\mathbf{C}[x_1, \ldots, x_n]$ such that $\mathbf{C}[x_1, \ldots, x_n]/I$ is finite dimensional over $\mathbf{C}$. Then the dimension of $\mathbf{C}[x_1, \ldots, x_n]/I$ is greater than or equal to the number of points in $V(I)$.*

Next we recall the Stetter-Möller method for solving a system of polynomial equations or, in other words, for calculating the points of the variety associated to the generated ideal. Throughout the remainder of this section we take the field K to be equal to the field of complex numbers $\mathbf{C}$. When the system of equations has finitely many solutions, that is when $\mathbf{C}[x_1, \ldots, x_n]/I$ is a finite dimensional vector space over $\mathbf{C}$, the method evaluates an arbitrary polynomial at the points of $V(I)$. In particular, considering f equal to x_i, $i = 1, \ldots, n$, the method gives the $i-$th coordinate of each point in $V(I)$.

Let $f \in \mathbf{C}[x_1, \ldots, x_n]$ be an arbitrary polynomial. Define

$$A_f : \mathbf{C}[x_1, \ldots, x_n]/I \to \mathbf{C}[x_1, \ldots, x_n]/I, \quad A_f([g]) = [f][g] = [fg].$$

Note that the multiplication is well defined on $\mathbf{C}[x_1, \ldots, x_n]/I$ due to the fact that I is an ideal. As A_f is a linear mapping from a finite dimensional space to itself, there exists a matrix representation of it with respect to a basis of $\mathbf{C}[x_1, \ldots, x_n]/I$. The normal set of I (see Definition 2.1.10) constitutes a basis for the linear vector space $\mathbf{C}[x_1, \ldots, x_n]/I$ ([12], Chapter 5§3, Proposition 1) and it will be referred to as the *normal basis* of $\mathbf{C}[x_1, \ldots, x_n]/I$. Let B denote the normal set of I and N denote the cardinality of B. In the following we use the same notation for the linear mapping A_f as well as for the matrix associated to it. The following properties hold for the $N \times N$ matrices A_f (see [12]).

Proposition 2.1.19 *Let $f, g \in \mathbf{C}[x_1, \ldots, x_n]$. Then:*

a. $A_f = 0$ *if and only if* $f \in I$.

b. $A_{f+g} = A_f + A_g$.

c. $A_{fg} = A_f A_g$.

d. Given a polynomial $h \in \mathbf{C}[t]$ *we have* $A_{h(f)} = h(A_f)$

Consider the particular matrices A_{x_i}, $i = 1, \ldots, n$. Using the properties above it is not difficult to see that $(A_{x_1}, \ldots, A_{x_n})$ is in fact a matrix element in the variety $V(I)$, that is $\forall f \in I, f(A_{x_1}, \ldots, A_{x_n}) = 0$. Here 0 denotes the zero matrix and $f(A_{x_1}, \ldots, A_{x_n})$ is well-defined due to the commutativity of the matrices.

Since matrices $A_{x_1}, \ldots, A_{x_n}$ are pairwise commutative, they have common eigenvectors and the n-tuple $(\xi_1, \ldots, \xi_n)$ of eigenvalues of $A_{x_1}, \ldots, A_{x_n}$ respectively, corresponding to the same common eigenvector will be an element of $V(I)$. Moreover, *all* the points in $V(I)$ are found as n-tuples of eigenvalues corresponding to the same common eigenvector (see for example, [27]). For a general polynomial f we have:

Theorem 2.1.20 *Let $I \subseteq \mathbf{C}[x_1, \ldots, x_n]$ be an ideal with the associated variety being zero-dimensional, $f \in \mathbf{C}[x_1, \ldots, x_n]$, and A_f the associated matrix. Then z is an eigenvalue of A_f if and only if z is a value of the function f on $V(I)$.*

In their papers, [48], [62], Stetter and Möller use instead of A_f the so-called *multiplication table* which is in fact the transpose of our matrix. By looking at the eigenvectors of the multiplication table (which in our case become the left eigenvectors) Stetter makes the interesting remark that if the eigenspace associated to a certain eigenvalue of A_f is 1-dimensional, then the vector $(\xi^{\alpha(1)}, \ldots, \xi^{\alpha(N)})$, where ξ is a solution of the system, is an eigenvector. In that case we call an eigenvector a Stetter vector. Hence, if $x_1, \ldots, x_n \in B$, the solutions of the system can be retrieved from the (left) eigenvectors of A_f.

2.2 Counting solutions in $\mathbf{R}^n$

One is often interested only in the *real* solutions of a system of polynomial equations. Moreover, when working on the reals, one might need to know only the solutions satisfying certain constraints, expressed in general as polynomial inequalities.

In Sections 2.1.2, 2.1.3 we have seen how Gröbner bases and Stetter-Möller methods can be used for solving systems of polynomial equations in $\mathbf{C}^n$. For counting solutions in $\mathbf{R}^n$, one might use the previous methods and simply discard the strictly complex solutions (or the real ones which do not satisfy all the constraints). However simple, this is not always possible, especially when the number of solutions in $\mathbf{C}^n$ is not finite.

Particular methods exist for dealing with the $\mathbf{R}^n$ case (see [8]). We discuss here only the well-known algorithm of Sturm (1835), for counting the real roots of a univariate polynomial, situated in a subinterval of $(-\infty, \infty)$. By repeated bisection of the interval one can obtain an approximation, as good as desired, of every root. Hence the algorithm can also be used for approximating with arbitrary precision *all* real roots of a polynomial. Section 2.2.1 reviews this method for counting/approximating with arbitrary precision the real solutions of a univariate polynomial. The method can be extended as in [21], Chapter 6, to deal with systems of multivariate polynomials having a *finite* number of complex solutions.

In Section 2.2.2 we review some results related to the general case, that is the case of a system of multivariate polynomials having an infinite number of (real) solutions.

2.2.1 Sylvester and Sylvester-Habicht sequences

We present here two important tools for counting the number of real roots of a polynomial. This section is based on Chapter 6 of [21]. We only discuss the univariate polynomial case.

We start by presenting the algorithm for computing the Sylvester sequence associated to two univariate polynomials $A, B \in \mathbf{R}[x]$. The computation of a Sylvester sequence is extremely simple, however it has certain short-comings which are corrected in a variant of it called the Sylvester-Habicht sequence. Notice that when $B = A'$, the Sylvester sequence is also called the Sturm sequence (or Sturm chain) of A.

Next we show how either sequence can be used for counting real solutions of polynomial equations, possibly satisfying some extra sign requirements.

Algorithm 2.2.1 *The following algorithm computes the Sylvester sequence* $\{S_j \mid j = 0, \ldots, l\}$ *of two polynomials* A, B.

```
1. Input: The polynomials $A, B$.
2. $S_0 \leftarrow A,\ S_1 \leftarrow B,\ i \leftarrow 1$.
3. While $S_i \neq 0$, compute $S_{i+1} = -\mathrm{Rem}(S_{i-1}, S_i)$; $i \leftarrow i + 1$.
4. Output: $\{S_j, j = 0, \ldots, l\}$ (where $S_{l+1} = 0$).
```

Since the only operation of the algorithm is polynomial division (for computing the remainder Rem at step 3), the algorithm is suited for symbolic computation. It means that one can compute the Sylvester sequence of two polynomials whose coefficients may be denoted by symbols. In such a case the Sylvester sequence may be a sequence of rational functions in the unknown coefficients of A, B (see

Example 2.7 of [21]). In general one can compute the Sylvester sequence of A, B for specific values of the coefficients from the generic Sylvester sequence. However, since this is a sequence of rational functions, it may happen that for particular values, a denominator of such a rational function vanishes. In such cases one speaks about *specialization problems.*

Specialization problems can be avoided by modifying the algorithm such that all elements in the sequence remain polynomials. In this case we call it a Sylvester-Habicht sequence. Let us define polynomials

$$\begin{aligned} A = &\ a_d X^d + a_{d-1}X^{d-1} + \ldots + a_0, \\ B = &\ \qquad\qquad b_q X^q + \ldots + b_0, \end{aligned}$$

where $a_d \neq 0$. The Sylvester-Habicht sequence associated to polynomials A and B consists, for $0 \leq j \leq d$, of polynomials $SH_j(A, B)$ of respective degrees $\leq j$. The j-th principal Sylvester-Habicht coefficient, which is the coefficient of degree j of SH_j, will be denoted h_j.

Algorithm 2.2.2 *The following algorithm computes the Sylvester-Habicht sequence* $\{SH_j \mid j = 0, \ldots, d\}$ *of two polynomials A and B, where* $\deg(A) = d$.

1. *Input: The polynomials A, B.*
2. $SH_d \leftarrow A$, $\bar{h}_d = a_d^{-1}$, *and*

   $$SH_{d-1} \leftarrow \begin{cases} B & \text{, } \textit{if } q = \deg(B) < d \\ \mathrm{Rem}(a_d^{2e}B, A) & \text{, } \textit{otherwise, where } e = \lceil \frac{q-d+1}{2} \rceil \end{cases}$$

   $j \leftarrow d.$
3. *While $j > 1$ execute:*
   Let $k = \deg(SH_{j-1})$. The lacking SH_l and $\bar{h}_l$ are computed up to SH_{k-1} respectively $\bar{h}_k$ as follows

   (i) **Computation of SH_l for $k < l < j-1$:** *If $k < j-2$, then $SH_l \leftarrow 0$.*

   (ii) **Computation of $\bar{h}_l$ for $k < l < j-1$:** *Let c_{j-1} denote the leading coefficient of SH_{j-1}. If $k = j-1$, then go to (iv), else ($k < j-1$) compute $\bar{h}_l$, l decreasing from $j-1$ to k by*

   $$\begin{aligned} \bar{h}_{j-1} &\leftarrow c_{j-1} \\ \bar{h}_l &\leftarrow (-1)^{j-l-1}\frac{\bar{h}_{l+1}c_{j-1}}{\bar{h}_j} \end{aligned}$$

   (iii) **Computation of SH_k**

   $$SH_k \leftarrow \frac{\bar{h}_k SH_{j-1}}{c_{j-1}}, \quad h_k \leftarrow \bar{h}_k.$$

(iv) **Computation of SH_{k-1}**

$$SH_{k-1} \leftarrow -\frac{1}{h_j \bar{h}_j}\mathrm{Rem}(c_{j-1}h_k SH_j, SH_{j-1});$$

$j \leftarrow k.$

4. Output: $\{SH_j, j = 0, \ldots, d\}$.

Let us see how we can use these sequences.

Definition 2.2.3 *For two polynomials A, B, let us denote by*

$$c_+(A,B) = \mathrm{card}(\{x \in \mathbf{R} \mid A(x) = 0,\ B(x) > 0\}),$$

$$c_-(A,B) = \mathrm{card}(\{x \in \mathbf{R} \mid A(x) = 0,\ B(x) < 0\}),$$

$$c_0(A,B) = \mathrm{card}(\{x \in \mathbf{R} \mid A(x) = 0,\ B(x) = 0\}).$$

Definition 2.2.4 *Let $V_S(A,B,a)$ (respectively $V_{SH}(A,B,a)$) denote the* number of sign changes of the Sylvester (respectively Sylvester-Habicht) sequence *of A and B evaluated at $a \in [-\infty, \infty]$, that is the number of sign changes in the ordered sequence $S_j(a)$, $j = 0, \ldots, l$ (respectively $SH_j(a)$, $j = 0, \ldots, d$). Denote*

$$V_S(A,B) = V_S(A,B,-\infty) - V_S(A,B,\infty),$$

$$\textit{respectively } V_{SH}(A,B) = V_{SH}(A,B,-\infty) - V_{SH}(A,B,\infty).$$

See also [21], Chapter 6, §7.3. Let $SQ(M,N)$ denote $c_+(M,N) - c_-(M,N)$. Then the following holds

Theorem 2.2.5 *For M, N polynomials, where $'$ denotes the derivative, we have*

$$V_S(M, M'N) = V_{SH}(M, M'N) = SQ(M,N)$$

Proof See [21], Corollaries 2.9 and 2.18 of Chapter 6. □

According to [21], Proposition 4.1. of Chapter 6 we have:

Theorem 2.2.6 *The following holds:*

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_0(A,B) \\ c_+(A,B) \\ c_-(A,B) \end{bmatrix} = \begin{bmatrix} SQ(A,1) \\ SQ(A,B) \\ SQ(A,B^2) \end{bmatrix}.$$

Hence any of the numbers $c_0(A,B)$, $c_+(A,B)$, $c_-(A,B)$ can be computed using either the Sylvester or the Sylvester-Habicht sequences, as in Theorem 2.2.6. In conclusion, counting/approximating with arbitrary precision the roots of a univariate polynomial can be done using either the Sylvester or the Sylvester-Habicht sequences.

2.2.2 A few results of real algebraic geometry

In the previous section we discussed an approach for counting the real solutions of a given univariate polynomial and we have mentioned that there exists an extension of the method to the case of a system of polynomial equations having a finite number of solutions on $\mathbf{C}^n$. The question now is how should one proceed in the general case, that is when the system of polynomial equations does not necessarily have a finite number of solutions in $\mathbf{C}^n$. Note that, in case there is an infinite number of solutions in $\mathbf{C}^n$, the number of solutions in $\mathbf{R}^n$ may be either finite or infinite. The main purpose of this section is to present the following result concerning the set of *real* solutions of a system of polynomial equations (and inequalities). Even when there is an infinite number of (real) solutions, there is still a *finite* number of connected components which make up the solution set.

The main object of study here are the semi-algebraic sets. That is, the sets defined by a boolean combination of polynomial equations and inequalities. As a formal definition we use the following one.

Definition 2.2.7 *A* semi-algebraic set *in* $\mathbf{R}^n$ *is a finite union of sets of the form:*

$$\{x \in \mathbf{R}^n \mid f_1(x) = \ldots = f_l(x) = 0, g_1(x) > 0, \ldots, g_m(x) > 0\},$$

where l, $m \in \mathbf{N}^*$, $f_1, \ldots, f_l, g_1, \ldots, g_m \in \mathbf{R}[x_1, \ldots, x_n]$.

Definition 2.2.8 *A semi-algebraic set* $A \subseteq \mathbf{R}^n$ *is called semi-algebraically connected if for every two semi-algebraic sets* F_1 *and* F_2*, closed in* A *(with respect to the Euclidean topology), such that* $F_1 \bigcap F_2 = \emptyset$ *and* $F_1 \bigcup F_2 = A$*, we have* $F_1 = A$ *or* $F_2 = A$.

The following theorem (Theorem 2.4.5 of [6]) is important when studying the number of elements in a semi-algebraic set.

Theorem 2.2.9 *A semi-algebraic set* $A \subseteq \mathbf{R}^n$ *is semi-algebraically connected if and only if it is connected. Any semi-algebraic set (and in particular any semi-algebraic subset of* $\mathbf{R}^n$*) has a finite number of connected components, which are semi-algebraic.*

The problem of counting the number of solutions of a system of polynomial equations (and inequalities) may be reformulated now as how many connected components has the solution set and are they zero-dimensional? Moreover, can one find at least one point in each one of the connected components? Extensive research has been done in this direction and methods like QE (quantifier elimination) and CAD (cylindrical algebraic decomposition) have been developed for it. In Section 3.2 we give an algorithmic solution to this problem as well.

Real algebraic sets enjoy special properties, not valid in general for the complex varieties. The following is noted in [6], Proposition 2.1.3.

Proposition 2.2.10 *Let $V = \{x \in \mathbf{R}^n \mid f_1(x) = \ldots = f_s(x) = 0\}$ then there exists an $f \in \mathbf{R}[x_1, \ldots, x_n]$ such that $V = \{x \in \mathbf{R}^n \mid f(x) = 0\}$.*

Proof Take $f = \sum_{i=1}^{s} f_i^2$. □

Hence, a system of polynomial equations in $\mathbf{R}^n$ can always be rewritten as a single equation. This result is important since it shows that checking the existence of a solution of a system of polynomial equations in $\mathbf{R}^n$ can be treated as a polynomial optimization problem. More precisely, $f_i(x) = 0,\ i = 1, \ldots, s$ has a solution in $\mathbf{R}^n$ if and only if $\min_{x \in \mathbf{R}^n} f = 0$, where $f = \sum_{i=1}^{s} f_i^2$.

2.3 Hilbert's 17th problem

This section is based on [6], Chapter 6. Our presentation is specialized to the real field $\mathbf{R}$ of real numbers. Searching for a characterization of nonnegative polynomials, Hilbert investigated whether any nonnegative polynomial $f \in \mathbf{R}[x_1, \ldots, x_n]$ can be written as a sum of squares of polynomials in $\mathbf{R}[x_1, \ldots, x_n]$. The question has a negative answer in general. A very famous example was given by Motzkin (1965). He proved that, for every $n \geq 3$, the polynomial

$$(x_1^2 + \ldots + x_{n-1}^2 - nx_n^2)x_1^2 \ldots x_{n-1}^2 + x_n^{2n}$$

is nonnegative on $\mathbf{R}^n$ but cannot be written as a sum of squares of polynomials (see [55]).

There exist a few particular cases in which a nonnegative polynomial can always be written as a sum of squares of polynomials. More precisely, for homogeneous polynomials (see Definition 2.1.9), the following holds.

Theorem 2.3.1 *Let $P_{n,m}$ denote the set of nonzero homogeneous polynomials in n variables of (total) degree m with coefficients in $\mathbf{R}$, that are nonnegative in $\mathbf{R}^n$ and let $\Sigma_{n,m}$ denote the subset of polynomials of $P_{n,m}$ which are sums of squares of polynomials. The following hold:*

- *If $n \leq 2$ or $m = 2$, then $P_{n,m} = \Sigma_{n,m}$.*
- *$P_{3,4} = \Sigma_{3,4}$.*
- *If $n \geq 3,\ m \geq 4$ and $(n, m) \neq (3, 4)$, then $P_{n,m} \neq \Sigma_{n,m}$.*

The results above can be immediately translated for non-homogeneous polynomials. A (non-homogeneous) nonnegative polynomial can be written as a sum of squares in the cases $n = 1,\ m = 2$ and $n = 2,\ m = 4$. It is however possible to write every polynomial, nonnegative on $\mathbf{R}^n$, as a sum of squares of rational functions, as it was shown by Emil Artin (1927).

Theorem 2.3.2 *Let $f \in \mathbf{R}[x_1, \ldots, x_n]$. If f is nonnegative on $\mathbf{R}^n$, then f is a sum of squares in the field of rational functions $\mathbf{R}(x_1, \ldots, x_n)$.*

Artin also characterized polynomials, nonnegative on a restricted semi-algebraic set in $\mathbf{R}^n$, using sums of squares of rational functions. More precisely let

$$W = \{x \in \mathbf{R}^n \mid g_1(x) \geq 0, \ldots, g_k(x) \geq 0\},$$

with $g_1, \ldots, g_k \in \mathbf{R}[x_1 \ldots, x_n]$, then there exist $f_1, \ldots, f_r \in \mathbf{R}[x_1 \ldots, x_n]$, all nonnegative on W such that every polynomial $f \in \mathbf{R}[x_1 \ldots, x_n]$, nonnegative on W can be represented as $f = \sum_{i=1}^{r} s_i f_i$, with s_i sums of squares of rational functions. Explicitly, we can take for f_i all products $\prod_{j=1}^{k} g_j^{\epsilon_j}$, where $\epsilon_j \in \{0, 1\}$, $j = 1, \ldots, k$. This characterization of polynomials which are nonnegative on a restricted subset of $\mathbf{R}^n$ stands at the basis of some known algorithms for global optimization of polynomial functions with constraints (see [42], [50]). We also discuss a certain application of these results in Section 4.2.

Chapter 3

Global optimization of polynomial functions

The chapter deals with methods for *global* polynomial optimization. The hope is that, by looking at specific classes of functions and exploiting their particularities, one can design more efficient algorithms for global optimization. The class of polynomial functions is a good candidate for such an investigation for at least two reasons. Firstly, they have been studied extensively, as illustrated in Chapter 2, and their behavior is quite well understood. Secondly, as we shall argue in the latter chapters, they are quite important in several applications coming from system theory. Nevertheless, even for this particular class, global optimization proves to be a rather difficult task. In this chapter, in Section 3.2, we provide an algebraic, and therefore *exact*, method for this problem. The method makes no assumptions on the polynomial to be optimized. We believe it is the first exact method dealing with polynomial global optimization in this generality.

The algorithm for solving this problem has a high computational complexity (increasing exponentially with the number of variables). However, this is to be expected since, according to [49], '*it is well-known that some NP-hard combinatorial problems can be rewritten as a problem of minimizing a multivariate polynomial*'. Therefore, minimizing a multivariate polynomial is itself an NP-hard problem.

We discuss in this chapter several methods for constrained and unconstrained optimization of polynomial functions. We present the methods of [50] and [42] based on relaxations of the polynomial optimization problem in Sections 3.1.1, respectively 3.1.2. The main contributions of this chapter can be found in Sections 3.2 and 3.3. In Section 3.2 we discuss an original, *exact* method for unconstrained optimization. Section 3.3 takes advantage of the algebraic nature of the method of Section 3.2 and employs it for discussing families of polynomials.

3.1 Numerical algorithms based on LMI relaxations

We start by describing some of the recently developed numerical methods for minimization of polynomial functions. In these algorithms, the polynomial structure of the problem is exploited. The algorithm of [50] is designed for unconstrained optimization and solves in fact an LMI relaxation of the polynomial optimization problem. The algorithm of [42] is designed for polynomial constraint optimization and constructs, although in a different manner, an LMI relaxation of the original problem which returns in general a bound on the sought optimum. If, for some reason, the bound is not satisfactory, it can be improved as much as desired using a sequence of LMI relaxations. In principle, this gives a method for computing the optimum of the initial problem. However, the more steps one takes in this sequence, the more demanding the computations are.

There are several reasons for considering LMI relaxations for polynomial minimization. First of all, LMI problems are convex, therefore any local minimum is a global minimum as well. This gives, for the polynomial minimization problem, a *guaranteed* lower bound. The sharpness of the bound can be checked under certain conditions. Optimization of polynomial functions by using LMI relaxations is currently an active research area. Moreover, efficient polynomial-time algorithms, as for example interior point algorithms, can be employed for solving LMI problems. It should be mentioned though that, for relatively low degree polynomials with a small number of variables, the size of the corresponding LMI relaxation can be quite large (see the examples in Chapter 5).

3.1.1 The method of Shor-Parrilo

Let us consider the problem of minimizing a multivariate polynomial with real coefficients

$$\inf_{x\in\mathbf{R}^n} p(x), \quad p \in \mathbf{R}[x]. \tag{3.1}$$

and present the approach described in [50]. There, one actually wants to find the largest real number α such that the polynomial $p(x) - \alpha$ is nonnegative everywhere on $\mathbf{R}^n$, i.e. solve

$$\begin{array}{ll} \sup & \alpha \\ \text{s.t.} & p(x) - \alpha \geq 0, \quad \forall x \in \mathbf{R}^n. \end{array} \tag{3.2}$$

The two problems are obviously equivalent. Note that when the solution to (3.1) is $-\infty$, problem (3.2) is infeasible, meaning that its solution is $-\infty$ as well.

For this method, the formulation (3.2) is preferred due to a particular feature of polynomials which are nonnegative on $\mathbf{R}^n$. Obviously, in order to prove that a polynomial is nonnegative on $\mathbf{R}^n$ it is sufficient to show that the respective polynomial can be written as a sum of squares of other polynomials. The converse is true in certain particular cases, see Theorem 2.3.1. However in general the converse does not hold. It is known that any polynomial nonnegative on

$\mathbf{R}^n$ can be written as a sum of squares of rational functions, see Theorem 2.3.2.

From the computational point of view it is still more advantageous to consider writing a polynomial as a sum of squares of polynomials, rather than as a sum of squares of rational functions. That is because in the first case we would have an upper bound on the total degree of the polynomials appearing in the sum, given by the total degree of the original polynomial. However in the general case, no upper bound is known for the total degree of the rational functions involved. Moreover, it turns out that *if* a polynomial can be written as a sum of squares of polynomials, writing it effectively is equivalent to checking the feasibility of a linear matrix inequality (LMI) problem. This is a linear semidefinite programming problem and there exist several numerical algorithms for it (see [66] and the references contained therein). Unfortunately, one does not know in advance whether a given polynomial can be written as a sum of squares of polynomials, except for some very special cases (see Section 2.3).

Let us describe in more detail how one can obtain a relaxation of the problem (3.2), following [50]. Let us consider a polynomial $F(x)$, $x \in \mathbf{R}^n$. If the total degree of F is odd, then its infimum will be $-\infty$. Hence, in this case the polynomial cannot be nonnegative everywhere. We can therefore restrict ourselves, without loss of generality, to the case of even degree polynomials. We follow closely [50] and produce a relaxation of the problem (3.2). Let F have a total degree $2d$. We want to find a matrix Q, positive semi-definite, such that, for every $x \in \mathbf{R}^n$, the following holds

$$F(x) = z^T Q z, \quad z^T = [1, x_1, x_2, \ldots, x_n, x_1x_2, \ldots, x_n^d]. \tag{3.3}$$

z contains all monomials in the variables $x_1, \ldots, x_n$ of degree less than or equal to d. A matrix Q satisfying (3.3) is called a *Gram matrix* associated to F (see [55]).

For that, we first compute the set of *all symmetric* matrices Q, satisfying (3.3). Note that this is in general a subset of the set of *all* matrices Q, satisfying (3.3). Let Q denote an arbitrary symmetric matrix, whose size equals the dimension of z. We compute the matrix Q by making the computation on the right-hand side of $F(x) = z^T Q z$, where z depends entirely on x's, and equalize the coefficients of the corresponding monomials. Note that Q is computed by solving a linear system of $N_1 = \binom{n+2d}{2d}$ equations (this is the number of monomials of degree less than or equal to $2d$ in n variables) with $N_2(N_2+1)/2$ unknowns, where $N_2 = \binom{n+d}{d}$ (the number of monomials of degree less than or equal to d in n variables).

To show that there exists at least one such Q, it suffices to remark that any monomial (of F) of degree less than or equal to $2d$ can be written as a product of two elements of z. By writing this in a matrix form and adding up we obtain a matrix Z, not necessarily symmetric, which satisfies (3.3). A symmetric matrix Q which satisfies (3.3) is obtained by $Q = (Z + Z^T)/2$. Since a monomial's decomposition into a product of monomials is in general non-unique, Q is not uniquely determined.

In fact all symmetric matrices Q satisfying (3.3) determine an affine space (see, e.g. [50]). Let us denote by $Q(\lambda)$ the generic matrix of the affine space, where $\lambda \in \mathbf{R}^{\kappa}$. More precisely

$$Q(\lambda) = Q_0 + \sum_{i=1}^{\kappa} Q_i \lambda_i. \tag{3.4}$$

It is known that the matrices Q_i, $i = 1, \dots, \kappa$ are completely determined by the vector z. Hence, if we consider a different polynomial $\tilde{F}$ in the variables $x_1, \dots, x_n$ of degree smaller than or equal to $2d$ and compute its matrix representation $\tilde{Q}$ as before, we obtain that, up to ordering, $\tilde{Q}_i = Q_i$, $i = 1, \dots, \kappa$. What differs of course, is the *free term*, i.e. $\tilde{Q}_0 \neq Q_0$.

We have shown so far how $Q(\lambda)$ can be constructed. The following property however is very important for our problem. According to [55], $F(x)$ is a sum of squares of polynomials if and only if there exists a λ_* such that $Q(\lambda_*) \succeq 0$ (that is $Q(\lambda_*)$ is positive semidefinite). In consequence, if there exists a λ_* such that $Q(\lambda_*) \succeq 0$, then $F(x) \geq 0$, $\forall x \in \mathbf{R}^n$. Deciding whether such λ_* exists reduces to checking the feasibility of an LMI problem and for that there exist quite efficient algorithms.

Suppose now that one is interested in actually writing F as a sum of squares of polynomials after a $Q(\lambda_*) \succeq 0$ was computed. Then it is known that there exists a matrix R such that $Q(\lambda_*) = R^T R$. Hence $F(x) = (Rz)^T Rz$ which is a sum of squares of polynomials.

To resume, if there exists a λ_* such that $Q(\lambda_*) \succeq 0$, then $F(x) \geq 0$, $\forall x \in \mathbf{R}^n$. The converse is not always true. There exist examples of polynomials which are nonnegative on $\mathbf{R}^n$ and for which no matrix Q, constructed as described above, is positive semi-definite (see Motzkin's example in Section 2.3). However, a nonnegative polynomial can always be written as a sum of squares of rational functions (Theorem 2.3.2). This can be reformulated as follows. There exists a polynomial $G(x)$ such that $G^2(x)F(x)$ is a sum of squares of polynomials. It is not clear however how to choose the polynomial $G(x)$.

For a good exposition on Hilbert's 17th problem see [6] and [55]. More remarks on this method will be made in the examples of Section 4.1.3 and Chapter 5.

Here we are interested in applying this method to the polynomial $F(x) = p(x) - \alpha$. We construct as above a generic symmetric matrix Q, which now depends on α as well. In fact, this matrix, denoted $Q(\alpha, \lambda)$, is affine in (α, λ) (see [50]), that is $Q(\alpha, \lambda) = Q_0 + \sum_{i=1}^{\kappa} Q_j \lambda_j + Q_{\kappa+1}\alpha$. Then the problem

$$\begin{array}{l} \sup \ \alpha \\ \text{s.t. } Q_0 + \sum_{i=1}^{\kappa} Q_j \lambda_j + Q_{\kappa+1}\alpha \succeq 0 \end{array} . \tag{3.5}$$

constitutes an LMI relaxation of (3.2), i.e. (3.5) returns in general a lower bound of (3.1). The lower bound is tight if and only if $p-\alpha$ can be expressed as a sum of squares of polynomials. Note however that any method which can provide a lower bound on (3.1) would be very interesting. Upper bounds are of course easy to obtain since every value of the function is an upper bound for its infimum.

In practical situation, one is mostly interested to know whether, for the given instance, by solving the relaxation one obtains the global infimum or just a lower bound of it. The following checking procedure, which makes use of the dual formulation of the LMI problem, is indicated in [50]. Let us recall first the (primal (P) and dual (D)) problems in semi-definite programming.

$$\begin{array}{ll} (P)\max c^T w & (D)\min \mathrm{trace}(A_0 Y) \\ \quad \text{s.t. } A_0 + \sum_{j=1}^{J} A_j w_j \succeq 0 & \quad \text{s.t. } \mathrm{trace}(A_j Y) = -c_j, \quad j = 1, \ldots, J \\ & \qquad\quad Y \succeq 0. \end{array} \tag{3.6}$$

Here Y and A_j, $j = 0, \ldots, J$ are symmetric matrices in $\mathbf{R}^{M \times M}$ and c, w are vectors in $\mathbf{R}^J$. It is known that, with this formulation, the value of the primal is smaller than or equal to the value of the dual (*weak duality* property). Moreover, under certain conditions, e.g., existence of a strictly feasible solution for one of the two problems, the value of the primal equals the value of the dual (*strong duality* property).

Let us return to the polynomial optimization problem (3.1) and its relaxation (3.5). By comparing (3.5) with (P), we have that, in this case, $J = \kappa + 1$, $A_j = Q_j$, $j = 0, \ldots, J$, $w^T = (\lambda^T, \alpha)$ and $c^T = \begin{pmatrix} 0 & \ldots & 0 & 1 \end{pmatrix}$. In order to prove that the relaxation is tight, i.e. that the lower bound equals the global optimum, it is sufficient to determine a point x^* such that $p(x^*)$ equals the lower bound. If such a point exists, then let us denote by z^* the vector z evaluated at x^* using (3.3). It is not difficult to show that the matrix $Y^* = z^* z^{*T}$ is an optimal solution of (D). Conversely, if Y^*, the computed solution of (D), has rank 1, then there exists a vector z^* such that $Y^* = z^* z^{*T}$. Hence, from the solution Y^* of (D), one could compute the vector z^* and further, the point x^* where the optimum is attained. It is argued in [50] that, under no degeneracies, the solution Y^* of the dual problem is a matrix of rank 1. Then x^* is found from z^* using (3.3). The fact that Y^* is solution of (D) of rank 1, implies that z^* has the structure given by (3.3). Notice also that practical implementations of semi-definite programming compute both the optimal vector solution w of the primal and the optimal matrix solution Y of the dual.

Since in general the polynomial $p(x) - \alpha$ need not be a sum of squares of polynomials, (3.5) is just a relaxation and not equivalent to the initial polynomial optimization problem (3.1). We do not intend here to discuss further the method. Both its advantages and disadvantages are well explained by their authors (see [59], [50]).

3.1.2 The method of Lasserre

We sketch briefly the method of [42]. The method is based on the strong relation between the theory of *moments*, the theory of nonnegative polynomials, and Hilbert's 17th problem on the representation of nonnegative polynomials. The polynomial minimization problem, of finding

$$\inf_{x \in \mathbf{R}^n} p(x), \quad \text{(respectively } \min_{x \in K} p(x)) \tag{3.7}$$

where K is a (not necessarily convex) compact set defined by polynomial inequalities, has an equivalent moment problem

$$\inf_{\mu \in \mathcal{P}(\mathbf{R}^n)} \int p(x)\mu(dx), \quad \text{(respectively } \min_{\mu \in \mathcal{P}(K)} \int p(x)\mu(dx)) \tag{3.8}$$

where $\mathcal{P}(\mathbf{R}^n)$ (respectively $\mathcal{P}(K)$) is the space of probability measures with support contained in $\mathbf{R}^n$ (respectively in K). A proof of the equivalence of these problems can be found in [42].

Notice now that the problem (3.8) is linear with respect to $\{\int x^\alpha d\mu\}$, the moments of the monomials of p with respect to the probability measure μ. This linearity is exploited by using semi-definite programming algorithms for solving efficiently the moment problem (3.8). This direct approach may also fail to return the global minimum when $p(x) - p^*$, where $p^* = \inf_{x \in \mathbf{R}^n} p(x)$, cannot be written as a sum of squares of polynomials. In general it returns a lower bound. However the approach can be extended to return an increasing sequence of lower bounds, convergent to the global minimum of the problem (3.7). Unfortunately, to do that, one is required to constrain the feasibility domain, modifying in this way the problem. Moreover, as one advances with computing elements in the sequence of lower bounds, the complexity of the computations increases extremely rapidly. For details on this interesting method, please see [42].

3.2 An exact algebraic method

This section introduces an exact method for solving (3.1). It represents one of the main theoretical contributions of the thesis. Further developments of this method will be presented in Section 3.3.

Other exact methods can be found in [22] and [68]. The first paper looks at the first order conditions. They form a system of polynomial equations that can be solved for example by using Gröbner basis techniques. However, in the case of an infinite number of critical points, even when a Gröbner basis is known, its elements may describe very complicated sets of points. It is not explained in the paper how one would proceed from there. The second paper mentioned makes some assumptions on the given polynomial, restricting in this way its applicability. The algorithms mentioned above work when the given polynomial has a minimum, without considering an approach for finding the infimum.

The method that we present in this section makes no assumptions on the polynomial p. Note also that we do not include in this setting any domain constraints.

The remainder of the section is organized as follows. In Section 3.2.1 we propose a certain perturbation on the given problem which allows us to treat the general case using the method of Stetter-Möller, and we give some theoretical results. Sections 3.2.2, 3.2.3, 3.2.4 deal with the actual computations, describing in more detail the output of the algorithm. In the end, in Sections 3.2.5 and 3.2.6, we discuss the algorithm in three particular examples and draw the conclusions.

3.2.1 Construction of an auxiliary polynomial

Consider a family of polynomials, depending on the *real positive* parameter $\lambda \in (0, \infty)$, given by

$$p_\lambda(x_1, x_2, ..., x_n) = p(x_1, x_2, ..., x_n) + \lambda(x_1^{2m} + x_2^{2m} + \ldots + x_n^{2m}),$$

where m is a fixed positive integer such that $m > \text{tdeg}(p)/2$ and $\text{tdeg}(p)$ stands for the total degree of p. One can rewrite $p_\lambda(x) = p(x) + \lambda\|x\|^{2m}$, where $\|x\|$ denotes the Minkowski $2m$ norm of $x = (x_1, x_2, ..., x_n)$, namely $\|x\| = (\sum_{i=1}^n x_i^{2m})^{1/2m}$.

If $\lambda > 0$ is fixed, the problem

$$\min_{x \in \mathbf{R}^n} p_\lambda(x_1, \ldots, x_n)$$

has two major advantages over the problem of finding $\inf_{x\in\mathbf{R}^n} p(x_1, \ldots, x_n)$. Firstly, the minimum of p_λ is always attained, hence the global minimum equals the smallest critical value of p_λ. Secondly, the first order conditions, used to compute the critical points and critical values, form a reduced Gröbner basis with respect to any total degree ordering (irrespective of the ordering of the variables). Hence a Gröbner basis is available by construction.

When λ goes to zero, from the family of polynomials p_λ we obtain again the polynomial p. We will study the relation between the minima of the polynomials p_λ and the infimum of p. Actually, we shall prove that $\inf_{x\in R^n} p(x_1, x_2, ..., x_n) = \lim_{\lambda\downarrow 0} \min_{x\in R^n} p_\lambda(x_1, x_2, ..., x_n)$. Therefore, we can concentrate on solving the new problem $\min_{x\in R^n} p_\lambda(x_1, x_2, ..., x_n)$, from which we deduce the answer for the original one. But let us first discuss in detail the relation between the two problems.

Let us denote by $\mathcal{I}$ the ideal generated by the first order derivatives of p_λ.

Proposition 3.2.1 *The first order derivatives of the polynomial p_λ form a reduced Gröbner basis for the ideal $\mathcal{I}$ generated by themselves.*

Proof The partial derivatives of p_λ are $\partial p_\lambda(x)/\partial x_i = 2m\lambda x_i^{2m-1} + \partial p(x)/\partial x_i$, $\forall i = 1, \dots, n$. With our choice of m, we have $2m > \text{tdeg}(p)$ hence $2m - 1 > \text{tdeg}(\partial p(x)/\partial x_i)$, $\forall i = 1, \dots, n$. In other words, the leading term of $\partial p_\lambda(x)/\partial x_i$ is $2m\lambda x_i^{2m-1}$ and it depends on x_i only. According to [11], Ch. 2, § 9, Theorem 3 and Proposition 4, the set $\{\partial p_\lambda(x)/\partial x_i \mid i = 1, \dots, n\}$ is a Gröbner basis of $\mathcal{I}$ (with respect to any total degree ordering). It is obvious that $\{\partial p_\lambda(x)/\partial x_i \mid i = 1, \dots, n\}$ is a reduced Gröbner basis. □

Throughout the rest of the section we use as a Gröbner basis, the set consisting of the derivatives of p_λ with respect to the variables $x_1, \dots, x_n$. The associated normal set B, as defined in Section 2.1.2, contains all monomials $\prod_{j=1}^{n} x_j^{\alpha_j}$ with $\alpha_j \in \{0, 1, \dots, 2m-2\}$, $j = 1, \dots, n$. Therefore, the cardinality of B is $N = (2m-1)^n$.

In the following we discuss the relation between the infimum of the polynomial p and the minima of the polynomials p_λ.

Lemma 3.2.2 *For every positive λ, the polynomial p_λ has a minimum.*

Proof We want to show that for every $\lambda > 0$ there exists an r_λ such that the minimum of p_λ is reached inside the Minkowski ball $B(0, r_\lambda)$.

Let $x \in \mathbf{R}^n$ with the Minkowski norm $\| x\| = r$. Then for every component of x we have $-r \le x_i \le r$, $i = 1, \dots, n$ and

$$p_\lambda(x) = \|x\|^{2m}(\lambda + p(x)/\|x\|^{2m}).$$

But $-p_{abs}(r) \le p(x) \le p_{abs}(r)$ for all x with $\|x\| = r$ implies

$$r^{2m}(\lambda - p_{abs}(r)/r^{2m}) \quad \le \quad p_\lambda(x).$$

Here p_{abs} is the polynomial obtained from p by replacing all its coefficients by their absolute value and taking all its variables equal. By construction we have that $2m$ is strictly larger than the total degree of the polynomial p (and also of p_{abs}), therefore $p_{abs}(r)/r^{2m}$ is a rational function in the variable r having the degree of the numerator strictly smaller than the degree of the denominator. Hence $\lim_{r\to\infty} p_{abs}(r)/r^{2m} = 0$ and so there exists an $r_\lambda^1 > 0$ such that for every $r \ge r_\lambda^1$ we have $\lambda > p_{abs}(r)/r^{2m}$. That means that for every x with $\|x\| = r \ge r_\lambda^1$ we have

$$0 \ < \ r^{2m}(\lambda - p_{abs}(r)/r^{2m}) \le p_\lambda(x). \tag{3.9}$$

From (3.9) we see that $p_\lambda(x)$ goes to infinity for $r \to \infty$, $r = \|x\|$. Hence $\exists r_\lambda \ge r_\lambda^1$ such that $\forall r \ge r_\lambda$ and x, $\|x\| = r$, we have $p_\lambda(x) > p_\lambda(0)$, where $p_\lambda(0) = p(0)$ is a fixed number. Hence $\forall x$, $\|x\| \ge r_\lambda$ we have $p_\lambda(x) > p_\lambda(0)$ which implies that the minimum value of p_λ must be attained inside the Minkowski ball $B(0, r_\lambda)$. This completes our proof. □

Denote by X_λ the set of real points where the minimum of p_λ is attained

$$X_\lambda = \{x_\lambda \in \mathbf{R}^n \mid p_\lambda(x_\lambda) = \min_{x \in \mathbf{R}^n} p_\lambda(x)\}.$$

From Lemma 3.2.2 we know that X_λ is nonempty for every λ positive. Also X_λ is a finite set for every positive value of λ. That can be seen from Theorem 2.1.17, part c, applied to the ideal $\mathcal{I}$, generated by the first order derivatives of p_λ. In the following we will use the notion of limit set as defined below. The set L given by

$$L \;=\; \{x \in \mathbf{R}^n \mid \forall \varepsilon > 0 \;\exists\; \lambda_\varepsilon \; s.t. \; \forall \lambda, \; 0 < \lambda < \lambda_\varepsilon, \; X_\lambda \cap B(x, \varepsilon) \neq \emptyset\}$$

is called the limit set of X_λ. For a multi-valued function with branches, by definition, the limit set will be simply the set of limits on the branches, assuming they exist.

Theorem 3.2.3 *The following statements are true:*

(i) $\lim_{\lambda \downarrow 0} \min_{x \in \mathbf{R}^n} p_\lambda(x) = \inf_{x \in \mathbf{R}^n} p(x)$.

(ii) $\lim_{\lambda \downarrow 0} p(x_\lambda) = \inf_{x \in \mathbf{R}^n} p(x), \; \forall x_\lambda \in X_\lambda$.

(iii) If p has a minimum then $L \subseteq \{\underline{x} \in \mathbf{R}^n \mid p(\underline{x}) = \min_{x \in \mathbf{R}^n} p(x)\}$.

Proof *(i)* We consider two cases. First, we treat the case when p has a minimum attained at some point $\underline{x}$. Then

$$p(\underline{x}) = \inf_{x \in \mathbf{R}^n} p(x) \leq \inf_{x \in \mathbf{R}^n} (p(x) + \lambda \|x\|^{2m}) \leq p(\underline{x}) + \lambda \|\underline{x}\|^{2m}.$$

The above relation holds for every $\lambda > 0$, hence the relation is also valid at the limit $\lambda \downarrow 0$:

$$p(\underline{x}) \leq \lim_{\lambda \downarrow 0} \inf_{x \in \mathbf{R}^n} p_\lambda(x) \leq p(\underline{x})$$

which proves our statement.

Suppose now that $\inf_{x \in \mathbf{R}^n} p(x) = p_{inf}$ and $\forall x \in \mathbf{R}^n, \; p(x) > p_{inf}$. Here, p_{inf} may be finite or infinite. Let M be a real number $M > p_{inf}$, arbitrarily close to p_{inf}. Although p does not reach p_{inf}, there exists an $\underline{x} \neq 0$ such that $p(\underline{x}) < M$; then there is an $\underline{\varepsilon} > 0$ such that $p(\underline{x}) + \underline{\varepsilon} < M$. Define $\lambda_{\underline{\varepsilon}} = \underline{\varepsilon}/\|\underline{x}\|^{2m}$, where $\|x\|$ is the Minkowski norm. Then we have that for every $\lambda < \lambda_{\underline{\varepsilon}}$

$$\min_{x \in \mathbf{R}^n} [p(x) + \lambda \|x\|^{2m}] \leq p(\underline{x}) + \lambda \|\underline{x}\|^{2m} < M.$$

Since for every positive λ_1, λ_2 with $\lambda_1 < \lambda_2$ we have $p_{\lambda_1}(x) \leq p_{\lambda_2}(x), \; \forall x \in \mathbf{R}^n$, the limit exists and

$$\inf_{x \in R^n} p(x) \leq \lim_{\lambda \downarrow 0} [\min_{x \in R^n} [p(x) + \lambda \|x\|^{2m}]] \leq M$$

As M is arbitrarily close to p_{inf},

$$\lim_{\lambda \downarrow 0}[\min_{x \in R^n}[p(x) + \lambda\|x\|^{2m}]] = p_{inf}$$

(ii) It follows immediately from *(i)* since $\inf_{x \in \mathbf{R}^n} p(x) \leq p(x_\lambda) \leq p_\lambda(x_\lambda)$, $\forall x_\lambda \in X_\lambda$.

(iii) Define $S = \{\underline{x} \in \mathbf{R}^n \mid p(\underline{x}) = \min_{x \in \mathbf{R}^n} p(x)\}$. By hypothesis, $S \neq \emptyset$. We want to show $L \subseteq S$. If $L = \emptyset$ then the claim is obviously true. Let us consider the case $L \neq \emptyset$. Suppose that $\exists x_0 \in L$. From the definition of the limit set L, we can construct a function which associates to every $\lambda > 0$ an $x_\lambda \in X_\lambda$ such that

$$\forall \varepsilon > 0 \; \exists \lambda_\varepsilon > 0, \; s.t. \; \forall \lambda, \; 0 < \lambda < \lambda_\varepsilon \quad x_\lambda \in B(x_0, \varepsilon)$$

But this says exactly that $\lim_{\lambda \downarrow 0} x_\lambda = x_0$. As p is a continuous function we have that $\lim_{\lambda \downarrow 0} p(x_\lambda) = p(x_0)$. From part (ii) we have $\lim_{\lambda \downarrow 0} p(x_\lambda) = \min_{x \in \mathbf{R}^n} p(x)$, hence $x_0 \in S$. □

According to the theorem, one can obtain the infimum of p from the minima of the family of polynomials p_λ and, in case the minimum exists, one can also obtain a set of points at which the minimum is attained, the limit set denoted here by L. To complete the discussion, we need to prove that L is a nonempty set, whenever the minimum of p is attained, and moreover is finite.

Theorem 3.2.4 *The set L is finite.*

Proof According to Theorem 2.1.18, the number of critical points of p_λ is bounded by $N = \dim(\mathbf{C}[x_1, \ldots, x_n]/\mathcal{I})$ for every positive λ, where $\mathcal{I}$ is the ideal generated by the first order derivatives of p_λ. It follows that the cardinality of X_λ is also bounded by N for every positive λ, since every point in X_λ is a critical point of p_λ. We will show that L has at most N points. Suppose that L has more than N distinct points and consider $N+1$ of them $l_1, \ldots, l_{N+1}$. Let $\delta > 0$ denote the smallest distance between any two of these points. For every $i = 1, \ldots, N+1$ construct the pairwise disjoint balls $B(l_i, \delta/2)$. By definition of L we have that there exists a $\lambda_{\delta/2} > 0$ such that every $B(l_i, \delta/2)$ has a nonempty intersection with X_λ, for each $\lambda \in (0, \lambda_{\delta/2})$. But for every $\lambda > 0$, X_λ has at most N elements, hence for each $\lambda \in (0, \lambda_{\delta/2})$, each of the $N+1$ disjoint balls should contain at least one of the N elements, which is impossible. Therefore L has at most N points. □

For our purposes, the non-emptiness is the most interesting part. In this way we have a guarantee that at least one point of global minimum is *always* obtained with our procedure, that is, when the minimum is attained.

Theorem 3.2.5 *If the polynomial p has a minimum, then L is nonempty.*

The proof of this theorem is given in Section 3.2.2.

So far we have shown that with this method we can find the minimum value of every polynomial and some of the points in which the minimum is attained. When the number of points of global minimum is infinite, we do not find all such points (see Example 3.2.20). One may wonder then which points we *do* find and the answer is partially given in the next proposition.

Theorem 3.2.6 *If p has a minimum, then the set L contains only points of minimum of p which have minimal Minkowski norm. In other words,*

$$L \subseteq \{x_0 \in \mathbf{R}^n \mid \|x_0\| = \min_{\{x \mid p(x)=p_{min}\}} \|x\|\},$$

where p_{min} denotes the minimal value of p.

Proof i) Let x_0 be a point where the minimum of p is attained, of minimal Minkowski norm. We prove that

$$\|x_\lambda\| \leq \|x_0\|, \ \forall \lambda > 0, \ \forall \ x_\lambda \in X_\lambda. \tag{3.10}$$

From

$$p_\lambda(x_\lambda) = p(x_\lambda) + \lambda \|x_\lambda\|^{2m}, \ p_\lambda(x_0) = p(x_0) + \lambda \ \|x_0\|^{2m}$$

and $p_\lambda(x_\lambda) \leq p_\lambda(x_0)$ (by definition of x_λ) we have

$$\lambda \ [\|x_\lambda\|^{2m} - \|x_0\|^{2m}] \ \leq \ p(x_0) - p(x_\lambda) \ \leq \ 0$$

(using the definition of x_0) and therefore $\|x_\lambda\| \ \leq \ \|x_0\|, \ \forall \lambda > 0$.

ii) Since p has a minimum, by Theorem 3.2.5 L is non-empty. As the norm is a continuous function, using the result of part i) we have

$$\forall x \in L \ \ , \ \ \|x\| \ = \ \|\lim_{\lambda \downarrow 0} x_\lambda\| \leq \|x_0\|$$

But $\forall x \in L$ we have from Theorem 3.2.3, part (iii), and from the fact that x_0 is a point of minimum of p of minimal Minkowski norm that $\|x\| \geq \|x_0\|$. Hence $\|x\| \ = \ \|x_0\|$ which implies $x \in \{x_0 \mid \|x_0\| = \min_{\{x \ \mid \ p(x)=p_{min}\}} \|x\|\}$ for every $x \in L$, so $L \subseteq \{x_0 \mid \|x_0\| = \min_{\{x \ \mid \ p(x)=p_{min}\}} \|x\|\}$. □

Denote by X the multi-valued function defined on $(0, \lambda_1)$ which associates to each $\lambda \in (0, \lambda_1)$ the set X_λ. To give more insight into the properties of the branches of X, we prove their monotonicity. However, this result will not be used in the remainder of the thesis.

Proposition 3.2.7 *The multi-valued function X satisfies:*
for any λ_1, λ_2 with $0 < \lambda_1 < \lambda_2$ and any $x_{\lambda_1} \in X_{\lambda_1}, \tilde{x}_{\lambda_2} \in X_{\lambda_2}$ we have

$$\|x_{\lambda_1}\| \geq \|\tilde{x}_{\lambda_2}\|.$$

In particular, for one branch ($x = \tilde{x}$) the proposition states that the branch is non-increasing with respect to λ in Minkowski norm.

Proof Given $\lambda_1 < \lambda_2$ we have

$$\begin{cases} p_{\lambda_1}(x_{\lambda_1}) \leq p_{\lambda_1}(\tilde{x}_{\lambda_2}) \\ p_{\lambda_2}(\tilde{x}_{\lambda_2}) \leq p_{\lambda_2}(x_{\lambda_1}) \end{cases}$$

or equivalently

$$\begin{cases} p(x_{\lambda_1}) + \lambda_1 \|x_{\lambda_1}\|^{2m} - p(\tilde{x}_{\lambda_2}) - \lambda_1 \|\tilde{x}_{\lambda_2}\|^{2m} \leq 0 \\ p(\tilde{x}_{\lambda_2}) + \lambda_2 \|\tilde{x}_{\lambda_2}\|^{2m} - p(x_{\lambda_1}) - \lambda_2 \|x_{\lambda_1}\|^{2m} \leq 0 \end{cases}$$

By adding the two inequalities we obtain

$$(\lambda_1 - \lambda_2)(\|x_{\lambda_1}\|^{2m} - \|\tilde{x}_{\lambda_2}\|^{2m}) \leq 0$$

which implies $\|x_{\lambda_1}\| \geq \|\tilde{x}_{\lambda_2}\|$. □

To summarize, we have constructed a family of polynomials p_λ, such that the infimum of our initial polynomial p can be obtained from the minima of the polynomials in the family, by letting the parameter λ decrease to 0. If the original polynomial has a minimum, the method will find at least one point at which the minimum is attained. We also have the Stetter-Möller method for solving the system of first order conditions which is by construction a reduced Gröbner basis. Hence, we need to compute the limits of the eigenvalues of a matrix A_{p_λ} associated to the polynomial p_λ for λ going to 0.

In the following section, we propose a method for computing these limits.

3.2.2 Computing the minimum

From the previous section we know that we can find the minimum of the original polynomial p by computing the limits, when λ goes to 0, of the eigenvalues of the matrix A_{p_λ}.

Proposition 3.2.8 *For each polynomial $g \in \mathbf{C}[x_1, \ldots, x_n]$, the associated matrix A_g is a polynomial matrix in $1/\lambda$. In particular, for each $i = 1, \ldots, n$, A_{x_i} is polynomial in $1/\lambda$ and A_{p_λ} is polynomial in $1/\lambda$.*

Proof The proof goes by induction on the number of reduction steps, that is polynomial reduction modulo the ideal $\mathcal{I}$. Recall that our Gröbner basis has a particular form in which the leading monomials are pure powers of the variables and λ appears only in the leading coefficient. Hence we start with constant entries but, due to the particular form of the Gröbner basis, whenever we make a reduction step (see for example [12]), we introduce a $1/\lambda$ or a power of it in some entries. Therefore, the entries of the final matrix will be polynomials in $1/\lambda$. □

In order to underline the dependency on λ, we denote $A_g = A_g(\lambda)$, where g is an arbitrary polynomial. The size of $A_g(\lambda)$ is given by the dimension of the basis B which is $N = (2m-1)^n$.

Recall the interpretation of the eigenvalues in the Stetter-Möller method. The eigenvalues of $A_g(\lambda)$ are the values of the polynomial g evaluated at the critical points of p_λ. In particular, when $g = p_\lambda$, these eigenvalues are the critical values of p_λ. The global minimal value of p_λ is among them and it converges to the infimum of p when $\lambda \downarrow 0$. The eigenvectors of A_{p_λ} will give the corresponding points and their limits for $\lambda \downarrow 0$ will allow us to read off a critical point of p where the minimum is attained. However if the set of critical points of p is not finite we are not able in general to find the whole set, but we find a finite subset of it.

For this reason, we study in the following the limits for λ decreasing to 0 of the eigenvalues of a matrix $A_g(\lambda)$. The equation

$$\det(A_g(\lambda) - zI) = 0 \quad , \quad \lambda > 0 \quad , \quad z \in \mathbf{C}$$

is satisfied if and only if

$$\lambda^k \det(A_g(\lambda) - zI) = 0 \quad , \quad \lambda > 0 \quad , \quad z \in \mathbf{C} \tag{3.11}$$

where k is the highest power of $1/\lambda$ appearing in the determinant. The second equation, polynomial in both z and λ, was studied extensively in the literature. Its solutions $z(\lambda)$ which satisfy the equation for every positive λ are known as *algebraic functions* (see [5]). An algebraic function is a multi-valued function having a finite number of branches $\zeta_i(\lambda)$, $i = 1, \ldots N$. The values of each branch around an arbitrary $\lambda_0 \geq 0$ are given by a Puiseux expansion in rational powers of $\lambda - \lambda_0$. To be more precise, the following proposition holds ([5], Theorem 13.1).

Proposition 3.2.9 *In a neighborhood V of every finite point $\lambda = \lambda_0$ (λ, $\lambda_0 \in \mathbf{C}$) all (complex) values of an algebraic function $z(\lambda)$ are determined by branches of the form*

$$\lambda = \lambda_0 + t^r \quad , \qquad z = z_{-\kappa}t^{-\kappa} + z_{-\kappa+1}t^{-\kappa+1} + \ldots + z_0 + z_1 t + \ldots \tag{3.12}$$

in which r is a positive integer ($r \in \mathbf{N}^*$), the coefficients $z_{-\kappa}, z_{-\kappa+1}, \ldots$ indicated are complex, possibly zero ($z_{-\kappa}, \ldots \in \mathbf{C}$), and κ is a non-negative integer ($\kappa \in \mathbf{N}$). For a value $\lambda \neq \lambda_0$ in V, (3.12) determines r distinct values of $z(\lambda)$ when the r values of the root $t = (\lambda - \lambda_0)^{1/r}$ are substituted in the series for z.

In [5], $\lambda_0 \in \mathbf{C}$, but obviously the proposition above holds for $\lambda_0 \in \mathbf{R}$ as well. We are now able to give the proof of a previously stated proposition.

Proof of Theorem 3.2.5.

In the definition of L, X_λ denotes the set of real points where the minimum of p_λ is attained. To show that L is nonempty we first prove that the coordinates of X_λ are algebraic functions, and therefore are continuous on branches on an interval $(0, \bar{\lambda})$ for $\bar{\lambda}$ sufficiently small. For that, we refer to Stetter-Möller theory. From Theorem 2.1.20 it follows that the coordinates of the point in $\mathbf{R}^n$ where the minimum of p_λ is attained, i.e. the coordinates of X_λ, can be obtained as the eigenvalues of the matrices $A_{x_i}(\lambda)$ for $i = 1, \ldots, n$, where $A_{x_i}(\lambda)$ denotes the linear mapping associated to the polynomial x_i.

From Proposition 3.2.8 we have that the matrices $A_{x_i}(\lambda)$ are polynomial matrices in $1/\lambda$. So, the eigenvalues of $A_{x_i}(\lambda)$ are the solutions of the equation in z, $\det(A_{x_i}(\lambda) - zI) = 0$ or equivalently, $\lambda^{k_i} \det(A_{x_i}(\lambda) - zI) = 0$ where k_i is the highest power of $1/\lambda$ appearing in the determinant. As the equation is polynomial in z and λ, the solutions $X_i(\lambda)$ are algebraic functions, for every $i = 1, \ldots, n$. Let $\lambda_0 = 0$ in Proposition 3.2.9. Then the algebraic functions $X_i(\lambda)$, $i = 1, \ldots, n$, admit in a neighborhood of $\lambda_0 = 0$ an expansion in which radicals or (a finite number of) terms with negative exponent may be involved (see Proposition 3.2.9). This implies in particular that the branches of X_i as functions of λ are continuous in an open right neighborhood of 0, say $(0, \bar{\lambda}_i)$ for $\bar{\lambda}_i, i = 1, \ldots, n$ sufficiently small. Since $X_i(\lambda)$ are coordinates of X_λ, then also X_λ is continuous in an open right neighborhood of 0, namely $(0, \bar{\lambda})$, where $\bar{\lambda} = \min\{\bar{\lambda}_i \mid i = 1, \ldots, n\}$.

Next we argue that, when p has a minimum, there will be a branch of X_λ which does not contain negative powers of λ in its expansion around 0. As p has a minimum, there exists a point in which the minimum is attained. We know that the branches of X_λ are bounded in the Minkowski norm by such a point (see equation (3.10)). Hence X_λ will have finite limits on the branches when $\lambda \downarrow 0$ and all these limits belong to the limit set L which is therefore nonempty. □

Recall that we want to compute the limits of the branches when $\lambda \downarrow 0$ so in our case $\lambda_0 = 0$ and V is a neighborhood of 0. The expansion of a branch of an algebraic function may have a finite number of terms containing negative powers of λ. We say that a branch has an *infinite limit* when $\lambda \downarrow 0$ if its expansion contains negative powers of λ. Otherwise we say that it has *finite limit.* The branches that have finite limits will tend, when $\lambda \downarrow 0$, to z_0, the term of the expansion which does not depend on λ, see equation (3.12).

Let

$$\det(A_g(\lambda) - zI) = f(\lambda, z) = 1/\lambda^k f_0(z) + 1/\lambda^{k-1} f_1(z) + \ldots + f_k(z).$$

where $f_0, f_1, \ldots, f_k$ are polynomials in z. Then equation (3.11) becomes

$$f_0(z) + \lambda f_1(z) + \ldots + \lambda^k f_k(z) = 0$$

We can easily see from Proposition 3.2.9 that the finite limits for $\lambda \downarrow 0$, provided they exist, are solutions of the equation $f_0(z) = 0$. In fact one can show a bit more.

Proposition 3.2.10 *The values of an arbitrary polynomial g evaluated at the critical points of p_λ define a finite number of branches having, when $\lambda \downarrow 0$, finite or infinite limits. The set of finite limits on these branches coincides with the set of solutions of $f_0(z) = 0$. In particular, for $g = p_\lambda$, the set of finite limits (on branches) of the critical values of p_λ coincides with the set of roots of a univariate polynomial.*

Proof The first part of the theorem was already discussed. For the last part, consider $\zeta(\lambda)$ a branch having a finite limit. By replacing $\zeta(\lambda)$ by its expansion, one can easily see that the lambda-free term in the expansion, is a solution of $f_0(z) = 0$. Hence the number of branches having a finite limit (multiplicities included) is at most equal to the degree of f_0, denoted by d. We will show that in fact the equality holds, hence the two sets must be equal. For this purpose we consider next the branches having infinite limits, i.e. their expansion contains negative powers of λ. Let $\zeta(\lambda)$ be a solution of (3.11), with a certain multiplicity, whose expansion around 0 contains negative powers of λ. Then $\omega(\lambda) = 1/\zeta(\lambda)$ is a solution of the equation $f(\lambda, 1/w) = 0$, with the same multiplicity, or equivalently, a solution of

$$w^N f(\lambda, 1/w) = 0. \tag{3.13}$$

Note that equation (3.13) was obtained by bringing the terms in $f(\lambda, 1/w)$ to the common denominator w^N and taking afterwards the numerator equal to 0. Remark that $\lim_{\lambda \downarrow 0} \omega(\lambda) = 0$ as can be seen for example from the expansion of $\zeta(\lambda)$. Hence $\omega(\lambda)$ is solution of the polynomial equation (3.13) and, having limit 0, is a finite solution of the equation. Rewriting the equation (3.13) we have

$$w^N[f_0(1/w) + \lambda f_1(1/w) + \ldots + \lambda^k f_k(1/w)] = 0$$

and we need to compute the number of branches $\omega(\lambda)$ (multiplicities included) that tend to 0 when $\lambda \downarrow 0$. But as we have argued before, every 0 limit of a branch $\omega(\lambda)$ is a root of the λ-free term, $w^N f_0(1/w)$. But $w^N f_0(1/w)$ has exactly $N - d$ zero roots, where d was the degree of f_0. Hence the number of branches $\omega(\lambda)$ (multiplicities included) having the limit 0, which equals the number of branches $\zeta(\lambda)$ (multiplicities included) having infinite limits, is at most $N - d$. To conclude, we have exactly N branches (multiplicities included) having either finite or infinite limit and we have shown that among them at most d have finite limits and at most $N - d$ have infinite limits. Hence there must be *exactly* d branches having finite limits and *exactly* $N - d$ having infinite limits (multiplicities included). □

So far we have considered the sets X_λ containing all global minimizers of p_λ. However, more information can be obtained by looking at the set of all *local* minimizers of p_λ, which includes X_λ, as the following results show.

Proposition 3.2.11 *Suppose that p has a minimum and $\underline{x}$ is an isolated point of minimum of the polynomial p. There exists a branch of local minima of p_λ convergent to $\underline{x}$ for $\lambda \downarrow 0$.*

Proof See the proof of Theorem 3.2.13 where a more general result which implies the one stated above is shown. □

Corollary 3.2.12 *If p has a minimum, then for each isolated point of global minimum of the polynomial p there exists a branch of local minima of p_λ converging to it for $\lambda \downarrow 0$. In particular, if p has a finite number of points of minimum, then they are all limits of branches of local minima of p_λ.*

Proof For each isolated point of minimum of p we apply Proposition 3.2.11. For the second part, remark that if p has a finite number of points of minimum, then they are all isolated. □

Theorem 3.2.13 *If p has a minimum then the set $p^{-1}(\{\min_{x\in\mathbf{R}^n} p(x)\})$ consists of one or more connected components. In each component there exists at least one point which is the limit of a branch of local minima of p_λ when $\lambda \downarrow 0$. Moreover, these points have minimal Minkowski norm inside the component.*

Proof Note that the number of connected components of $p^{-1}(\{p_{min}\})$ is finite (see Theorem 2.2.9 or [6], Th 2.4.5), where $p_{min} = \min_{x\in\mathbf{R}^n} p(x)$. Pick a point, say $x(j)$, in each component C_j, where

$$C = \bigcup_{j\in J} C_j = \{x \in \mathbf{R}^n \mid p(x) = p_{min}\}.$$

Let $M_j = \|x(j)\|$ and $M > \max_{j\in J} M_j$.

We want to show that for every $j \in J$, there will be a local minimum of p_λ whose points of minimum are in the Minkowski ball $B(0, M)$ and converge to an element of C_j. If this holds then, from the local minima of p_λ, we obtain at least one point in each component C_j.

Note that in each component C_j there is a point, namely $x(j)$, such that

$$p_\lambda(x(j)) < p_{min} + \lambda M^{2m} \le p_\lambda(x), \ \forall x \notin B(0, M).$$

Hence

$$p_\lambda(x(j)) < p_\lambda(x), \ \forall x \notin B(0, M)$$

and the minima of p_λ are in the Minkowski ball $B(0, M)$.

Consider $p_\lambda\big|_{\overline{B(0,M)}}$. The number of connected components of $C \bigcap \overline{B(0, M)}$ is still finite since the set $\{x \in \mathbf{R}^n \mid \|x\|^{2m} \le M^{2m}, p(x) = p_{min}\}$ is a semi-algebraic set (Theorem 2.2.9). Denote these connected components by D_l. Since $\overline{B(0, M)}$ is a compact set and the sets D_l are closed and disjoint, it follows that

$\exists\varepsilon_0 > 0$ such that $\forall l_1 \neq l_2$, $d(D_{l_1}, D_{l_2}) > \varepsilon_0$, where d denotes the Minkowski distance between sets.

Define the neighborhood of a component D_l as

$$N_{\varepsilon_0/3}(D_l) = \{x \in B(0,M) |\ d(x, D_l) < \varepsilon_0/3\}.$$

We want to show that the minimum of $p_\lambda \Big|_{\overline{N_{\varepsilon_0/3}(D_l)}}$ is not attained on the boundary of $\overline{N_{\varepsilon_0/3}(D_l)}$ for all λ small enough. Note that any point on the boundary satisfies either $\|x\| = M$ or $d(x, D_l) = \varepsilon_0/3$. We already know that the points on the boundary of $B(0,M)$ are not minima.

Let $\bar{p} = \min_{\bigcup_l (\partial N_{\varepsilon_0/3}(D_l) \bigcap B(0,M))} p(x)$. Then $\bar{p} > p_{min}$. For any l, we have $p_\lambda \Big|_{\partial N_{\varepsilon_0/3}(D_l) \cap B(0,M)} \geq \bar{p}$.

On the other hand, for any $x \in D_l$ we have $p_\lambda(x) = p_{min} + \lambda\|x\|^{2m} \leq p_{min} + \lambda M^{2m} < \bar{p}$ for λ sufficiently small, namely $\lambda < (\bar{p} - p_{min})/M^{2m}$. Therefore, if $\lambda < (\bar{p} - p_{min})/M^{2m}$ then $\min_{x \in \overline{N_{\varepsilon_0/3}(D_l)}} p_\lambda(x)$ is attained in the open set, not on the boundary.

We have proved that for λ smaller than a certain value, for every component D_l there exists an open neighborhood of it containing points of local minimum of $p_\lambda \Big|_{\overline{B(0,M)}}$.

Let x_λ^l be a global minimizer of $p_\lambda \Big|_{\overline{N_{\varepsilon_0/3}(D_l)}}$. Then x_λ^l is a local minimizer of p_λ (on R^n). Since x_λ^l is local minimizer, it is convergent when $\lambda \downarrow 0$ as in the proof of Theorem 3.2.5 to a point, say $x_* \in \overline{N_{\varepsilon_0/3}(D_l)}$.

We want to show that $x_* \in D_l$. We have $p(x) \leq p_\lambda(x)$ and $\lim_{\lambda\downarrow 0} p_\lambda(x) = p(x)$, $\forall x \in \mathbf{R}^n$. Hence $p(x_\lambda^l) \leq p_\lambda(x_\lambda^l) \leq p_\lambda(x_*)$. When $\lambda \downarrow 0$ we obtain $\lim_{\lambda\downarrow 0} p_\lambda(x_\lambda^l) = p(x_*)$.

Take $x_0 \in D_l$. We have $p_\lambda(x_\lambda^l) \leq p_\lambda(x_0)$ and at the limit it becomes $p(x_*) \leq p_{min}$ or in fact $p(x_*) = p_{min}$. This implies that $x_* \in D_l$. □

We have proved here (Theorem 3.2.13) that, if p has a minimum, any algorithm which is able to compute all the limits of the branches of local minima of p_λ, computes in fact at least one point in each connected component of the set of minimal values of the polynomial p. Such an algorithm is described in the following section (Algorithm 3.2.18).

3.2.3 Case: the polynomial p has a minimum

From Theorem 3.2.3 we know that $\min_{x \in R^n} p_\lambda(x) = p_\lambda(x_\lambda)$ converges to $\min_{x \in R^n} p(x)$. But $p_\lambda(x_\lambda)$ satisfies the equation (3.11) where $g = p_\lambda$, so $(p_\lambda(x_\lambda))_{\lambda > 0}$ is a branch of the algebraic function associated to the equation (3.11) for $g = p_\lambda$. Moreover, we know it has a finite limit. Hence $\lim_{\lambda\downarrow 0} p_\lambda(x_\lambda)$ will be a root of f_0. The smallest real root is our candidate for the minimum of p. Note that we have been working over the field of complex numbers and it is possible that the smallest real root is a value of p attained in a complex point. Hence, before deciding that the smallest real root is the minimum of p, we need to do a check at the point where the minimum is attained. We will discuss this issue later, but until then, in order to make the discussion easier, we will assume that the smallest real eigenvalue is indeed the minimum.

The way to compute $\min_{x \in R^n} p(x)$ becomes more clear now. Having constructed the matrix A_{p_λ}, one can calculate $\det(A_{p_\lambda} - zI)$, polynomial in $1/\lambda$ and z, then isolate the coefficient of the largest power of $1/\lambda$. This is a polynomial in z whose smallest real root gives us the minimum of p.

We have now a straightforward way to compute the minimum of our polynomial p. However, the drawback of using the determinant is that, besides the high computational complexity, it will not tell us anything about the corresponding eigenvectors. As we already remarked, knowing the eigenvectors may be helpful in finding not only the minimum but also (at least) a point in which the minimum is attained. Hence we need a more refined method for the actual calculations.

We describe here a method for computing the finite limits of the eigenvalues, without actually computing the determinant. It will be clear that with this new method, we can not only find the corresponding eigenvectors but also we do less calculations, as we only need one term of the determinant.

The method is a special case of the well-known algorithm of [16] for minimizing the sum of the row degrees of a polynomial matrix over an equivalence class of polynomial matrices. With this method we obtain the coefficient of the highest power of $1/\lambda$ in the expression of the determinant $\det(A_g(\lambda) - zI)$ as the determinant of a polynomial matrix in z. After applying linearization techniques to this polynomial matrix in z (see [20], § 7.2) we reduce the problem to finding the eigenvalues of a pencil. Since the original matrix is nonsingular over $\mathbf{R}[z]$ and the linearization procedure leaves the determinant unchanged, the generalized eigenvalue problem obtained is always nonsingular.

Remark that the problem of finding the minimum of a polynomial and some point where this is attained is reduced to solving a generalized eigenvalue problem. For this new problem, a large variety of algorithms exists and they can handle quite large matrices.

Let us describe now in more detail how to find the coefficient of the highest power of $1/\lambda$ in the expression of the determinant $\det(A_g(\lambda) - zI)$. The procedure is quite general and can be applied to an arbitrary polynomial matrix. Let $B(\mu)$ be a polynomial matrix in μ, $B(\mu) \in \mathbf{R}^{b \times b}[\mu]$. The degree of the i-th row, denoted d_i, is the highest degree in μ of all its entries. The total row degree of the matrix is the sum of its row degrees, $d = \sum_{i=1}^{b} d_i$. The associated high-order coefficient matrix, denoted HOCM, is constructed by retaining from each entry of the i-th row, the coefficient of μ^{d_i} (see also Example 3.2.16 for further clarifications). The algorithm for finding the leading term of $\det(B(\mu))$, i.e. the term containing the highest power of μ in the expression of the determinant $\det(B(\mu))$, is based on the following:

Proposition 3.2.14 *Let $B(\mu)$ be a polynomial matrix in μ and let d denote its total row degree. The leading term of the polynomial $\det(B(\mu))$ in μ is $\det(\mathrm{HOCM}(B(\mu)))\mu^d$ if and only if $\mathrm{HOCM}(B(\mu))$ is nonsingular.*

Proof It follows immediately from the well-known formula for computing determinants. □

If we apply the procedure for $\mu = 1/\lambda$, we can find the leading coefficient of $\det(A_g(\lambda) - zI)$, polynomial matrix in $1/\lambda$, for any polynomial g. Note that by construction of $A_g(\lambda)$, each column of the matrix corresponds to multiplication of g by an element in the normal basis. Consequently, the entries of each column, which are polynomials in $1/\lambda$, have similar degrees in μ. Hence, the total row degree of $(A_g(\mu) - zI)^T$ is in general much smaller than the total row degree of $A_g(\mu) - zI$. Therefore, for computational reasons, we work with $(A_g(\mu) - zI)^T$.

Algorithm 3.2.15 *The following procedure returns a matrix, polynomial in μ and rational in z, of minimal total row degree in μ, equivalent to the input matrix $(A_g(\mu) - zI)^T$.*

```
1. Input: $B(\mu) \leftarrow (A_g(\mu) - zI)^T, \Delta \leftarrow 1$.
2. Compute $d_i, i = 1, \ldots, N$ and $\mathrm{HOCM}(B(\mu))$. If $\mathrm{HOCM}(B(\mu))$ is nonsingular, then go to step 7.
3. Else compute a nonzero vector $v = (v_1, \ldots, v_N)$ in the left kernel of $\mathrm{HOCM}(B(\mu))$. The vector can be chosen polynomial in $z$.
4. Construct the vector $\tilde{v} = (v_1\mu^{d_*-d_1}, \ldots, v_N\mu^{d_*-d_N})$, where $d_* = \max_{\{i=1,\ldots,N \mid v_i \neq 0\}} d_i$.
5. Construct a matrix $L(\mu, z)$ from the identity matrix by replacing its i-th row by $\tilde{v}$, where $i$ is chosen such that $d_i = d_*$.
6. $B(\mu) \leftarrow L(\mu, z)B(\mu)$, $\Delta \leftarrow \Delta \cdot \det(L(\mu, z))$. Go to Step 2.
7. Output: $\bar{A}_g(\mu, z) \leftarrow B(\mu)$, with $\det((A_g(\mu) - zI)^T) = \frac{1}{\Delta} \cdot \det(B(\mu))$ and $\mathrm{HOCM}(B(\mu))$ nonsingular.
```

As $A_g(\mu) - zI$ is nonsingular, i.e. its determinant is non-identically zero, the degree in μ in the expression $\det(A_g(\mu) - zI)$ is a positive natural number $\tilde{d}$. As we run the algorithm, the total row degree of the matrix is decreased by 1, at least, every time we execute step *6*. Hence the algorithm stops after a finite number of steps, when the total row degree of $B(\mu)$ reaches the value $\tilde{d}$.

Remark that HOCM($B(\mu)$) is polynomial matrix in z hence a vector as in step *3* always exists. Remark also that the determinant of $L(\mu, z)$ from step *5* does not depend on μ. It may depend on z, therefore we need the corrections Δ. Matrices like $L(\mu, z)$ depending on a parameter μ, whose determinant does not depend on μ are called z-modular or unimodular over $\mathbf{R}[z]$.

Since at step *6* we multiply with z-modular matrices, our HOCM may become polynomial, not linear, in z. The nonsingular polynomial matrix in z can be brought by a linearization procedure (see [20], § 7.2) into an equivalent matrix, linear in z of a larger dimension. Note however that in the reduction process while multiplying on the left with z-modular matrices we introduce some new solutions. Hence we must keep track of the solutions we introduce and subtract them in the end.

To be more precise, after running the algorithm we have

$$\bar{A}_g(\mu, z) = \bar{L}(\mu, z)(A_g(\mu) - zI)^T ,$$

where $\bar{A}_g(\mu, z)$ has a nonsingular HOCM and $\bar{L}(\mu, z)$ is z-modular. Here, $\bar{L}(\mu, z)$ denotes the product of all matrices $L(\mu, z)$ constructed at step *5* during the execution of the algorithm. For the determinants, the following holds:

$$\det(\bar{A}_g(\mu, z)) = \det(\bar{L}(\mu, z)) \det(A_g(\mu) - zI)$$

and using Proposition 3.2.14 and the fact that $\det(\bar{L}(\mu, z))$, which equals our final value of Δ in the algorithm, does not depend on μ it follows that the leading term in μ of $\det(A_g(\mu) - zI)$ satisfies

$$\mathrm{lt}(\det(A_g(\mu) - zI)) = (\det(\bar{L}(\mu, z)))^{-1} \det(\mathrm{HOCM}(\bar{A}_g(\mu, z))).$$

The roots of $\det(\bar{L}(\mu, z))$ are artificially introduced so we must eliminate them.

The algorithm can be applied in general for finding a (left-)equivalent representation of a matrix of minimal total row degree . In the following we give a small example to illustrate how the algorithm works.

Example 3.2.16 *Consider a matrix $M(\mu)$, polynomial in μ, of non-minimal total row degree. $M(\mu)$ plays the role of $A_g(\mu)$, the difference being that $M(\mu)$ is not associated to a polynomial. Let*

$$M(\mu) = \begin{pmatrix} \mu^2 & 0 & \mu \\ 1 & 0 & -2 \\ \mu^3 & \mu & \mu^2 \end{pmatrix} .$$

The matrix $B(\mu) = (M(\mu) - zI)^T$ becomes

$$B(\mu) = \begin{pmatrix} \mu^2 - z & 1 & \mu^3 \\ 0 & -z & \mu \\ \mu & -2 & \mu^2 - z \end{pmatrix}$$

with the row degree vector $(3,1,2)$*, hence the total row degree* 6*. However its* HOCM *is singular,*

$$\mathrm{HOCM}(B(\mu)) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

hence its total row degree is not minimal. Pick up a vector in the left kernel of HOCM$(B(\mu))$*, say* $v = (-1,1,0)$ *and construct* $\tilde{v} = (-1,\mu^2,0)$*. The matrix* $L(\mu,z)$ *becomes*

$$L(\mu,z) \leftarrow \begin{pmatrix} -1 & \mu^2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and by multiplication on the right with $B(\mu)$*,*

$$B(\mu) \leftarrow \begin{pmatrix} -\mu^2+z & -1-z\mu^2 & 0 \\ 0 & -z & \mu \\ \mu & -2 & \mu^2-z \end{pmatrix} \quad \text{and} \quad \Delta \leftarrow -1.$$

Since the new matrix has a singular HOCM*, we return to step 2 and continue the reduction procedure. Hence*

$$B(\mu) \leftarrow \begin{pmatrix} -\mu^2+z & -1-z\mu^2 & 0 \\ 0 & -z & \mu \\ \mu & \mu z-2 & -z \end{pmatrix} \quad \text{and} \quad \Delta \leftarrow -1.$$

Remark that another reduction step is necessary and finally we obtain

$$B(\mu) = \begin{pmatrix} z & -2\mu-1 & -\mu z \\ 0 & -z & \mu \\ \mu & \mu z-2 & -z \end{pmatrix}$$

whose high-order coefficient matrix

$$\mathrm{HOCM}(B(\mu)) \leftarrow \begin{pmatrix} 0 & -2 & -z \\ 0 & 0 & 1 \\ 1 & z & 0 \end{pmatrix}$$

is nonsingular. Remark that the determinant of $B(\mu)$ *is* $-2\mu^2z^2+z^3+2z\mu-2\mu^3-\mu^2$ *and it is equal to* $\Delta\det(M(\mu)-zI)$*. In this example, the total row degree was reduced from* 6 *to the minimal row degree which is* 3*.*

In general, when Δ depends on z we introduce false solutions during the reduction procedure, that is we introduce the roots of $\det(\bar{L}(\mu,z))$. An improvement on the algorithm would be to avoid introducing such solutions or if we do, to eliminate them in a smarter way. The problem reduces basically to the following one: Having a polynomial matrix $\tilde{M}(z)$ and a polynomial in $\tilde{m}(z)$ which divides its determinant, find a polynomial matrix whose determinant is $\det\tilde{M}(z)/\tilde{m}(z)$.

Obviously, such a matrix exists as well as an algorithm to compute it. The question is whether we can compute such a matrix in an efficient way.

Note that the eigenvectors of the matrix $\text{HOCM}(\bar{A}_g(1/\lambda, z))$, polynomial in z, preserve the property of the Stetter vectors. Namely, when the eigenspace is 1-dimensional, the vector generating it is an eigenvector evaluated at the critical point. More precisely, let $\bar{A}_g(1/\lambda, z)$ denote the output of the Algorithm 3.2.15 for the input matrix $(A_g(\lambda) - zI)^T$. Let $H(z)$ denote the $\text{HOCM}(\bar{A}_g(1/\lambda, z))$ and recall that $H(z)$ is polynomial in z.

Proposition 3.2.17 *If $\hat{z}$ is an eigenvalue of $H(z)$, that is $H(\hat{z})$ is singular, and its corresponding eigenspace is 1-dimensional, then there exists $\xi \in \mathbf{R}^n$ such that $(\xi^{\alpha(1)}, \ldots, \xi^{\alpha(N)})^T$ is a generating eigenvector and $g(\xi) = \hat{z}$, where $(x^{\alpha(1)}, \ldots, x^{\alpha(N)})^T$ is the normal basis vector.*

Proof For every $\lambda > 0$ let $\hat{z}(\lambda)$ be an eigenvalue of $A_g(\lambda)$. Since the matrix $(A_g(\lambda) - \hat{z}(\lambda)I)$ is singular, there exists a nonzero vector $v(\lambda)$ such that $(A_g(\lambda) - \hat{z}(\lambda))^T v(\lambda) = 0$. If the eigenspace corresponding to $\hat{z}(\lambda)$ is 1-dimensional, then $v(\lambda)$ is a Stetter vector. That is, $\exists \xi(\lambda) \in \mathbf{R}^n$ such that
$v(\lambda) = (\xi(\lambda)^{\alpha(1)}, \ldots, \xi(\lambda)^{\alpha(N)})^T$ with $g(\xi(\lambda)) = \hat{z}(\lambda)$.

Let $\xi(\lambda)$ and $\hat{z}(\lambda)$ have finite limits for $\lambda \downarrow 0$ and denote $\lim_{\lambda \downarrow 0} \xi(\lambda) = \xi$. As discussed earlier in this section, we know that $\lim_{\lambda \downarrow 0} \hat{z}(\lambda)$ is an eigenvalue of $H(z) = \text{HOCM}(\bar{A}_g(1/\lambda, z))$.

When running the Algorithm 3.2.15 we multiply the matrix $(A_g(\lambda) - zI)^T$ only on the left-hand side, hence its right-eigenvectors are preserved. In the end we obtain,

$$\bar{L}(1/\lambda, z)(A_g(\lambda) - zI)^T v(\lambda) = 0, \ \forall \lambda > 0.$$

By premultiplying with $\text{diag}(\lambda^{d_1}, \ldots, \lambda^{d_N})$, where d_j is the (minimal) row degree of row j we obtain an N-dimensional equation in λ, valid for every $\lambda > 0$ and well-defined in $\lambda = 0$. Then the equation must hold also for $\lambda = 0$, but that is exactly $\text{HOCM}(\bar{A}_g(1/\lambda, z)) \cdot \lim_{\lambda \downarrow 0} v(\lambda) = 0$, where $\lim_{\lambda \downarrow 0} v(\lambda) = (\xi^{\alpha(1)}, \ldots, \xi^{\alpha(N)})^T$. That insures us that the eigenvector of $H(z) = \text{HOCM}(\bar{A}_g(1/\lambda, z))$ will indeed correspond to an eigenvalue $\hat{z}$ of $H(z)$.

□

In [12] a method is proposed for choosing the polynomial g such that the left-eigenspaces of A_g are 1-dimensional, so that one can 'read' immediately not only the values of g on $V(I)$ but also the points where the value is obtained. As suggested there, g can be an arbitrary linear combination of the variables, i.e. $g = c_1x_1 + \ldots + c_nx_n$ where $c_1, \ldots, c_n$ are complex constants. Such choice may be important if one wants to use the properties of the Stetter vectors.

To resume, the computational procedure we suggest is:

Algorithm 3.2.18 *The following procedure can be used for computing the minimum of p.*

1. *Select a polynomial g and construct the corresponding matrix $A_g(\lambda)$.*
2. *Compute the* HOCM$(\bar{A}_g(1/\lambda, z))$ *by running the Algorithm 3.2.15.*
3. *Compute those values of z for which* HOCM$(\bar{A}_g(1/\lambda, z))$ *(polynomial matrix in z) becomes singular. Compute the corresponding eigenvectors.*
4. *Read off the values of x corresponding to each z computed at step 3 from the eigenvectors by using the Stetter interpretation.*
5. *Evaluate the polynomial p at all these critical points and identify the global minimum as the smallest value.*

The choice of the polynomial g at step *1* is left to the user. It may equal p or p_λ, or a linear combination of the variables which (ideally) leads to 1-dimensional eigenspaces and therefore allows an immediate reading of the critical points. Note however that for the latter choice of g, the assumption that the polynomial p has a minimum is essential! In case this is not true, one can find the value of the finite infimum (if this exists) only in a direct way, by choosing g equal to p or p_λ.

Remark 3.2.19 *The global minimum and the point where this is attained can be computed with arbitrary accuracy, therefore we call the method which employs Algorithm 3.2.18 an* exact *method. Note that the global minimum is a root in z of a univariate polynomial, namely* $\det(HOCM(\bar{A}_p(1/\lambda, z)))$*, and can be computed with arbitrary accuracy using, for example, Sylvester or Sylvester-Habicht sequences (see Section 2.2.1). Similarly, for every $i = 1, \ldots, n$, the $i-$th coordinate of a point where the global minimum is attained is a root in z of a polynomial, namely* $\det(HOCM(\bar{A}_{x_i}(1/\lambda, z)))$.

3.2.4 Case: the polynomial p does not have a minimum

In the previous section we have described an algorithm for computing the global minimum of a polynomial, in case it exists. When the same procedure is applied for g equal p or p_λ, the algorithm actually computes the value of the (finite) infimum, if that exists. We believe this is one of the very important features of the algorithm.

At this point we do not have a direct way of deciding whether the infimum is finite or not. However, the following procedure can in principle be used to decide this. Compute the candidate for the finite infimum by running the Algorithm 3.2.18 . Let us denote the obtained value by c. Then form the polynomial $(p - c + \alpha)^2$, α being a positive constant, and run the algorithm again. If c was indeed the infimum of p, then the new polynomial must have infimum α^2. If there are values of p strictly smaller than c, then due to the continuity of p there must exist a point x such that $p(x) = c - \alpha$. Hence the new polynomial will have the minimum equal to 0.

Further research will be useful into finding a direct way to decide upon this matter.

3.2.5 Examples

We consider here rather small examples. There are a few reasons for our choices. The first one is that the method we have proposed requires a number of calculations that increases rapidly with the degree of the polynomial and the number of variables. The second, and more important reason, is that in these cases we already know the minimum and the set of points where it is attained, therefore it is possible to analyze the algorithm in these specific examples. We considered interesting the case of an infinite number of critical points. In the finite case we know from the theory that the algorithm finds all the points.

Example 3.2.20 *Let $p(x_1, x_2) = (x_1^2 + x_2^2 - 1)^2$. The minimum is obviously 0 and the set of points where it is attained is the circle of radius 1, centered in $(0, 0)$. We apply the algorithm by first constructing the family of polynomials*

$$p_\lambda(x_1, x_2) = (x_1^2 + x_2^2 - 1)^2 + \lambda(x_1^6 + x_2^6).$$

The power in the extra-term was chosen to be an even number strictly larger than 4, the total degree of p. Next we construct our matrices using the Stetter-Möller method.

We follow the Algorithm 3.2.18. As g polynomial we choose the following linear combination of variables $g = x_1 + 3x_2$. We construct the associated matrix $A_g(\lambda)$, polynomial in $1/\lambda$, of size $(6-1)^2 \times (6-1)^2 = 25 \times 25$. The total row degree of $(A_g(\lambda) - zI)^T$ is 12. However it is not minimal, i.e. the highest power of $1/\lambda$ appearing in the determinant of $A_g(1/\lambda, z) = (A_g(\lambda) - zI)^T$ is actually 6 as results by running the total row degree reduction algorithm of Forney (Algorithm 3.2.15) on $A_g(1/\lambda, z)$ which will return the matrix $\bar{A}_g(1/\lambda, z)$. At this point we have also obtained the coefficient of the highest power of $1/\lambda$ in the expression $\det(A_g(1/\lambda, z))$. This is the determinant of the HOCM of $\bar{A}_g(1/\lambda, z)$. Computing the eigenvalues of HOCM, i.e. the zeroes of the determinant of HOCM, we obtain (by Maple)

$$\{0, 1, -1, 3, -3, 2\sqrt{2}, -2\sqrt{2}, \sqrt{2}, -\sqrt{2}\}.$$

All eigenvalues have multiplicity 1, therefore from the corresponding eigenvectors we read off the following corresponding points:

$$\{(0, 0), (\pm 1, 0), (0, \pm 1), (\pm\frac{\sqrt{2}}{2}, \pm\frac{\sqrt{2}}{2}).$$

Evaluating the polynomial p at these points, we conclude that the candidate for the minimum is 0 and it is attained at all points above except $(0, 0)$. To be completely safe, we should check that p has indeed a minimum. It is easy to check that p does not have a finite infimum and we do that by rerunning the algorithm *for $g = p$. The value returned by the algorithm equals 0, the minimal value we have found already.*

In order to check that the polynomial does not have an infinite infimum, we need to apply the trick described in Section 3.2.4. Therefore we run again the algorithm for $g = (p+1)^2$ and obtain that the minimum of the new polynomial equals 1. The critical points of the new polynomial coincide with the critical points of p. If p had an infinite infimum, $(p+1)^2$ should have had a minimum at 0. Therefore we conclude that the minimum of p is indeed 0.

Remark that the values $(\pm\sqrt{2}/2, \pm\sqrt{2}/2)$ are points where the minimum of p is attained, of minimal Minkowski norm. This was predicted in Theorem 3.2.6. However we obtain some extra points which in this case are points of maximal Minkowski norm. It is an open question whether we find points of maximal Minkowski norm in every connected component whenever the component is bounded.

The running time of Algorithm 3.2.18 depends mainly on the size of the matrices involved and the number of iterations in the Forney reduction procedure at step 2. Let us now give information about the running time of the algorithm in this particular example. The computations were done on a SUNW Ultra-4 Sparc station with 2048 Mb. We give here both the CPU-time, that is the amount of time the Central Processing Unit is actually executing instructions, and the approximate total execution time, that is CPU-time plus the time while the computer fetches data from the keyboard or disk, or sends data to an output device, etcetera.

The algorithm applied to minimizing p produces matrices of size 25. When $g = x_1 + 3x_2$, the CPU-time is $6.83s$ while the total execution time is $15s$ and the Forney procedure reduces the total row degree from 12 to 6. When $g = p$, the CPU-time is $13.97s$ while the total execution time is $22s$ and the Forney procedure reduces the total row degree from 44 to 16. When the algorithm is applied to minimizing $(p+1)^2$, it produces matrices of size 81. For $g = x_1$ the CPU-time is $141.79s$ while the total execution time is $145s$ and the Forney procedure reduces the total row degree from 21 to 9. When $g = x_2$, the CPU-time is $171s$ while the total execution time is $184s$ and the Forney procedure reduces the total row degree from 21 to 9.

Example 3.2.21 *Let us consider now a polynomial having a finite infimum, as in [68]:*

$$p(x_1, x_2) = x_1^2x_2^4 + x_1x_2^2 + x_1^2.$$

We run the algorithm for $g = p$ and obtain the results 0 (with multiplicity 3) and $-1/4$ (with multiplicity 12). Obviously, the candidate for the infimum is $-1/4$, being the smallest among the two. If $-1/4$ were a minimum of the polynomial, then we should be able to find out the coordinates of the respective point by rerunning the algorithm for $g = x_1$ and for $g = x_2$. But by doing so, we *only obtain the point* $(0,0)$*, hence we conclude that* $-1/4$ *is not attained. We should still check whether the infimum is not* $-\infty$ *as we did in the first example, however we do not make the computations here.*

The algorithm applied to minimizing p produces matrices of size 49*. When* $g = p$*, the CPU-time is* 37.97s *while the total execution time is* 50s *and the Forney procedure reduces the total row degree from* 72 *to* 54*. When* $g = x_1$ *the CPU-time is* 13.18s *while the total execution time is* 23s *and the Forney algorithm produces no reduction. When* $g = x_2$ *the CPU-time is* 14.89s *while the total execution time is* 25s *and the Forney procedure reduces the total row degree from* 11 *to* 9*.*

Example 3.2.22 *The last example illustrates a technicality related to the method of Algorithm 3.2.18. At step* 1*, Algorithm 3.2.18 requires the choice of a polynomial g. Since we are interested in the infimum of a given polynomial p, we might choose* $g = p$ *and compute the smallest real eigenvalue z of the matrix HOCM*$((A_p(\lambda) - zI)^T)$*, which is polynomial in z. However we cannot conclude immediately that this is the infimum of p. It might happen, as we illustrate in this example, that the the smallest real eigenvalue is attained at a complex value of x. Therefore it is always necessary to check the so-called* admissibility *of the smallest real eigenvalue. Let us consider*

$$p(x_1, x_2) = (x_1^2 + x_2^2 + 1)^2.$$

The minimum of p equals 1 *and is obtained at* $(x_1, x_2) = (0,0)$*. However, there exist complex critical points of p, satisfying* $x_1^2 + x_2^2 + 1 = 0$*, for which the corresponding critical value is* 0*. This will also be reflected in the results of the algorithm, when we run it for* $g = p$*. Namely, we obtain the eigenvalues* 0*, with multiplicity* 8*, and* 1 *with multiplicity* 9*. However, at this point we cannot conclude anything. We need to rerun the algorithm for a different choice of g, say* $g = x_1 + 2x_2$*. This actually produces different eigenvalues, each having* 1*-dimensional eigenspaces form which we can read off the critical points, as in step* 4 *of Algorithm 3.2.18. Since the only admissible (real) value for* (x_1, x_2) *is* $(0,0)$*, we conclude that the minimum of p is* 1*.*

The algorithm applied to minimizing p produces matrices of size 25*. When* $g = p$*, the CPU-time is* 13.11s *while the total execution time is* 22s *and the Forney procedure reduces the total row degree from* 44 *to* 16*. When* $g = x_1 + 2x_2$ *the CPU-time is* 6.65s *while the total execution time is* 16s *and the Forney algorithm reduces the total row degree from* 12 *to* 6*.*

We refer to Example 3.2.22 in Section 3.3 where the admissibility issue comes into play.

3.2.6 Conclusions

The proposed method is guaranteed to find the global minimum of a general polynomial, whenever the minimum exists. Moreover, if the minimum does not exist, we can decide if the infimum is finite or not, and give its value in the first case. The method avoids Buchberger algorithm, which is known to be computationally very demanding.

Another very important feature of the algorithm is that it returns a point in every connected component of the set of global minimizers. Using the algorithm we can in fact answer a different problem as well. Given a set of polynomial equations $f_i(x_1, \ldots, x_n) = 0$, $i = 1, \ldots s$, we can find a point in every connected component of the solution set, simply by minimizing $f = \sum_{i=1}^{s} f_i^2$. Such problems received much attention (see for example [56] and the references contained therein).

At last, as we shall argue in the next section, due to the fact that this is an algebraic, therefore exact method, we can treat with it a larger class of problems, namely optimization of polynomials depending on some parameters.

3.3 Families of polynomials

We argued in the Section 3.2 that using exact algebraic methods it is possible, at least in principle, to obtain the optimum in (3.1) with any desired accuracy. In this section, we emphasize another main difference (and advantage) of the exact methods over the numerical ones. Imagine a situation in which the polynomial to be optimized is not known exactly, but depends on the parameter (vector) $x \in \mathbf{R}^n$. There are various ways in which one can end up with such a problem. One application can be found in Chapter 4 where we extend the methods for optimization of polynomial functions to the class of rational functions.

Let us formulate the problem. Let $p \in D[y]$, where D denotes the field of real, n-variate functions in x and $D[y]$ is the set of polynomials in variable y with coefficients in D. We are interested in computing

$$\inf_{y \in \mathbf{R}^m} p(x, y), \tag{3.14}$$

where $x \in \mathbf{R}^n$ is considered as a parameter. The infimum in (3.14) depends obviously on the parameter x. We intend to prove that a slight variation of Algorithm 3.2.18, namely Algorithm 3.3.1, provides the answer to (3.14) in an implicit form, namely as a root z of $P(z, x)$, where P is polynomial in z. That is, of course when the infimum is finite.

Algorithm 3.3.1 *The following procedure computes $P(z, x)$, a polynomial in z, which defines in an implicit form the value of (3.14).*

1. *Construct the matrix $A_p(\lambda)$.*
2. *Compute the* $\mathrm{HOCM}(\bar{A}_p(1/\lambda, z))$ *by running the Algorithm 3.2.15.*
3. *Compute* $P(z, x) = \det(\mathrm{HOCM}(\bar{A}_p(1/\lambda, z)))/\Delta$*, polynomial in z.*

Since we are interested in the value of the infimum, we have modified the Algorithm 3.2.18 accordingly and have chosen at step *1*, as polynomial $g(y) = p(x, y)$. Let us go through the steps of Algorithm 3.3.1. The matrix $A_p(\lambda)$ constructed at step *1* is polynomial in $1/\lambda$. At step *2*, we apply the procedure for reducing the total row degree in $1/\lambda$ of $A_p(\lambda) - zI$ (Algorithm 3.2.15). However, all operations performed there are multiplications and additions, hence the resulting HOCM matrix will be polynomial in z. Its determinant, as well as $P(z, x)$, computed at step *3*, are also polynomial in z. From Section 3.2 we know that the infimum of (3.14), if it is finite, is a root of $P(z, x)$ in z. Hence we have obtained the infimum (in case it is finite) of a family of polynomials in an implicit form, that is as a root of a particular polynomial equation. In fact, the candidate for the infimum is the *smallest* real root of P. However, as discussed in Example 3.2.22, it might happen that the smallest real root is attained at a complex value, and therefore it is not admissible as a solution. Although it is not very likely to have real critical values for complex critical points, one should be aware of this possibility. To conclude,

$$\inf_{y \in \mathbf{R}^m} p(x, y) = \begin{cases} -\infty, & x \in \mathcal{K} \subseteq \mathbf{R}^n \\ \underline{z}, \ \underline{z} \text{ is the smallest admissible} \\ \quad \text{real root of } P(z, x), \ x \in \mathbf{R}^n \setminus \mathcal{K}. \end{cases} \tag{3.15}$$

In general we do not have a description of the set $\mathcal{K}$, however for each particular value of x we can decide, as described in Section 3.2.4, whether x belongs to $\mathcal{K}$ or not.

In the sequel, we restrict ourselves to the case of $p \in \mathbf{R}[x][y]$ in (3.14), that is p is a polynomial in both x and y. This implies that $A_p(\lambda)$ is a polynomial matrix in both $1/\lambda$ and x, HOCM is polynomial in both z and x, and consequently, $P(z, x)$ is polynomial in both x and z.

Let us consider now the relation (3.15). Aside from the technical problems, related to cases where either the infimum is not finite or if it is finite, the smallest real root is not admissible, the infimum of a family of polynomial functions is obtained as the smallest real root of a certain polynomial. In the sequel, we give an algebraic description of the smallest real root in z of a polynomial $P(z, x)$, where $x \in \mathbf{R}^n$. It turns out that the conditions for a $\underline{z}$ to be the smallest real root of $P(z, x)$ describe a semi-algebraic set. The explicit description of such an algebraic set is obtained using Sylvester-Habicht sequences.

Assumption 1 *$\mathcal{K} = \emptyset$, i.e. $\forall x \in \mathbf{R}^n$, $\inf_{y \in \mathbf{R}^m} p(x, y)$ is bounded.*

Notice that in general the assumption may be difficult to check. However in certain applications it is automatically satisfied.

For the rest of this section we consider that Assumption 1 is satisfied. We do not deal with the admissibility of the real smallest root here. Notice however that the smallest real root of $P(z, x)$, even if it is not admissible, returns a lower bound on (3.15). We do not expect the admissibility problem to show up very often and therefore, believe that in general (3.16) equals (3.15.)

$$\underline{z}, \text{ such that } \underline{z} \text{ is the smallest real root of } P(z, x), \tag{3.16}$$

where $P(z, x)$ is the polynomial obtained by applying the Algorithm 3.3.1.

Next we give an approach for solving (3.16). More formal, the problem could be written as

$$\underline{z}, \text{ such that } P(\underline{z}, x) = 0 \text{ and } \{z \in \mathbf{R} \mid z < \underline{z},\ P(z, x) = 0\} = \emptyset \tag{3.17}$$

This problem is equivalent to (3.16). Notice that the difficult part of (3.17) is the empty set condition. However the condition $\{z \in \mathbf{R} \mid z < \underline{z},\ P(z, x) = 0\} = \emptyset$ can be read as *the number of real roots of the polynomial P (in z), where $z - \underline{z} < 0$, is* 0. This can be approached using the theory described in Section 2.2.1. Let us denote by $c(x, \underline{z})$ the cardinality of the set $\{z \in \mathbf{R} \mid z < \underline{z},\ P(z, x) = 0\}$. Then according to Section 2.2.1, Theorems 2.2.6 and 2.2.5, we have

$$c(x, \underline{z}) = \frac{1}{2}[V(P, (z - \underline{z})^2 P') - V(P, (z - \underline{z})P')], \tag{3.18}$$

where $V(A, B) = V(A, B, -\infty) - V(A, B, +\infty)$ and $V(A, B, a)$ denotes the number of sign changes in the Sylvester-Habicht sequence. Since Sylvester-Habicht's algorithm, for computing the Sylvester-Habicht sequence of (A, B) evaluated at a, is based on Euclidean division, it can be applied to polynomials with symbolic (non-numeric) coefficients. What we obtain then is a Sylvester-Habicht sequence with coefficients depending on $\underline{z}$, x_i, $i = 1, \ldots, n$, actually rational functions in these variables. Requiring that $c(x, \underline{z}) = 0$ is equivalent to the satisfiability of certain sign conditions for these coefficients, hence with the feasibility of a finite union of semi-algebraic sets (i.e. sets described by polynomial equations and inequalities).

The idea of applying (different variants of) Sturm's algorithm for rewriting the condition above is not new. It has been noted already in [67] and the basic idea is summarized in [67], Note 12: *Apart, however, from technicalities connected with the notion and construction of Sturm chains, the mathematical content of Sturm's theorem essentially consists in the following: given any algebraic equation in one variable z, and with coefficients $a_0, a_1, \ldots, a_n$, there is an elementary criterion for this equation to have exactly k solutions (which may be in addition subjected to the condition that they lie in a given interval): such a criterion is obtained by constructing a certain finite sequence of systems, each consisting of finitely many equations and inequalities which involve the coefficients $a_0, a_1, \ldots, a_n$ of the given equation (and possibly the end points of the interval); it is shown that the equation has exactly k roots if and only if its coefficients satisfy all the equations and inequalities of at least one of these systems.*

In our case the number of solutions must be 0. With the idea above, $\underline{z}$ of problem (3.17) is the unique solution of a semi-algebraic set. We present the procedure for obtaining this semi-algebraic set on a particular example.

Example 3.3.2 *Let us consider the following problem*

$$\min_{y \in \mathbf{R}} (y^4 + y^2 x_1 + y x_2 + x_1). \tag{3.19}$$

Notice that the Assumption 1 is satisfied. By applying the Algorithm 3.3.1 we obtain that (a lower bound for) the solution of (3.19) is given by (3.17) with

$$P(z,x) = z^3 + \left(-3x_1 + \frac{1}{2}{x_1}^2\right) z^2 + \left(-{x_1}^3 + 3{x_1}^2 + \frac{1}{16}{x_1}^4 + \frac{9}{16}{x_2}^2 x_1\right) z \\ -\frac{1}{16}{x_1}^5 - \frac{9}{16}{x_2}^2{x_1}^2 + \frac{1}{2}{x_1}^4 - {x_1}^3 + \frac{27}{256}{x_2}^4 + \frac{1}{64}{x_2}^2{x_1}^3.$$

We know that the condition $\{z \in \mathbf{R} \mid z < \underline{z},\ P(z,x) = 0\} = \emptyset$ can be rewritten as a finite union of semi-algebraic sets using either Sylvester or Sylvester-Habicht sequences. Let us explain the procedure on this particular example.

Due to (3.18), we need to construct the Sylvester sequences associated to the pairs of polynomials $(P, (z - \underline{z})P')$, respectively $(P, (z - \underline{z})^2 P')$. We start with the pair $(P, (z - \underline{z})P')$ and obtain the following sequence $\{r_0(z) = P, r_1(z) = (z - \underline{z})P', r_2(z), r_3(z), r_4(z)\}$ where

$$r_2 = \left(\tfrac{1}{2}{x_1}^2 - 3x_1 + 3\underline{z}\right) z^2 + \left(6{x_1}^2 + \underline{z}{x_1}^2 - 6\underline{z}x_1 + \tfrac{9}{8}{x_2}^2 x_1 + \tfrac{1}{8}{x_1}^4 - 2{x_1}^3\right) z \\ -3{x_1}^3 + \tfrac{1}{16}\underline{z}{x_1}^4 + \tfrac{3}{2}{x_1}^4 + \tfrac{3}{64}{x_2}^2{x_1}^3 + \tfrac{81}{256}{x_2}^4 + 3\underline{z}{x_1}^2 + \tfrac{9}{16}\underline{z}{x_2}^2 x_1 \\ -\tfrac{27}{16}{x_2}^2{x_1}^2 - \underline{z}{x_1}^3 - \tfrac{3}{16}{x_1}^5$$

$$r_3 = -\frac{1}{128}\frac{\left(16{x_1}^7 - 936{x_1}^4{x_2}^2 - 64\underline{z}^2{x_1}^4 - 486\underline{z}{x_2}^4 + 486x_1{x_2}^4 + \ldots\right)}{\left(-6x_1 + {x_1}^2 + 6\underline{z}\right)^2}$$

$$r_4 = \frac{1}{256}\frac{\left({x_2}^2\left(-6x_1 + {x_1}^2 + 6\underline{z}\right)^2 \left(8{x_1}^3 + 27{x_2}^2\right)^3 \left(16{x_1}^5 - \ldots\right)}{\left(16{x_1}^7 - 936{x_1}^4{x_2}^2 - 64\underline{z}^2{x_1}^4 - 486\underline{z}{x_2}^4 + 486x_1{x_2}^4 + \ldots\right)}$$

In order to compute the number of sign changes in the the sequences

$$\{r_0(\infty), r_1(\infty), r_2(\infty), r_3(\infty), r_4(\infty)\},$$

respectively

$$\{r_0(-\infty), r_1(-\infty), r_2(-\infty), r_3(-\infty), r_4(-\infty)\}$$

we need to consider all possible combinations of signs of $r_i(\infty)$ respectively $r_i(-\infty)$, where $r_i(\infty)$, $r_i(-\infty)$ are in general functions of $x_1, \ldots, x_n$ and $\underline{z}$. They are of course determined by the signs of the highest degree coefficients of each polynomial in the sequence. Now, computing the Sylvester sequence associated to the second pair of polynomials $(P, (z - \underline{z})^2 P')$ we obtain another set of *conditions on the parameters $\underline{z}, x_i,\ i = 1, \ldots, n$. Requiring that $c(x, \underline{z}) = 0$ is satisfied, where $c(x, \underline{z})$ given by (3.18), allows us to select the right conditions and obtain the desired semi-algebraic set.*

Notice however that there is a small technical problem related to the Sylvester sequence of a polynomial, known as specialization problem. *The term refers to a situation in which we want to* specialize *the parameters $\underline{z}, x_i,\ i = 1, \ldots, n$ to numerical values. In the example above, when the high order coefficient of r_2, $(1/2{x_1}^2 - 3x_1 + 3\underline{z})$, becomes 0, the value we computed for r_3 is not well defined since 0 appears in the denominator and the computation has to be redone for this particular case.*

Fortunately, there is a way to avoid such technical difficulties by using instead of Sylvester sequences the so-called Sylvester-Habicht sequences (see Section 2.2.1).

We have made the computations of the Sylvester sequence above in order to make clear what the problems are. Also Sylvester sequences are better known and therefore it is easier to explain the theory. However, for practical applications we use always Sylvester-Habicht sequences.

Let us now compute the Sylvester-Habicht sequences associated to the pairs $(P, (z - \underline{z})P')$ and $(P, (z - \underline{z})^2P')$. They are $\{SH_3^1, SH_2^1, SH_1^1, SH_0^1\}$ respectively $\{SH_3^2, SH_2^2, SH_1^2, SH_0^2\}$ computed as in Section 2.2.1, Algorithm 2.2.2, or [21], Algorithms 2.11 and 2.13 of Chapter 6. We have $SH_3^1 = P$ and

$$
\begin{aligned}
SH_2^1(z) &= \left(-\frac{1}{2}{x_1}^2 + 3x_1 - 3\underline{z}\right) z^2 + \left(-6{x_1}^2 - \underline{z}{x_1}^2 + 2{x_1}^3 + 6\underline{z}x_1 - \ldots\right) z \\
SH_1^1(z) &= \left(-\frac{1}{32}{x_1}^7 + \frac{117}{64}{x_1}^4{x_2}^2 + \frac{1}{8}\underline{z}^2{x_1}^4 + \frac{243}{256}\underline{z}{x_2}^4 - \frac{243}{256}x_1{x_2}^4 - \ldots\right) z \\
SH_0^1(z) &= -\frac{1}{16777216}{x_2}^2\left(8{x_1}^3 + 27{x_2}^2\right)^3\left(16{x_1}^5 - 16\underline{z}{x_1}^4 - 128{x_1}^4 + \ldots\right)
\end{aligned}
$$

All elements in the Sylvester-Habicht sequence are polynomials, hence well defined for any values of $\underline{z}, x_1, x_2$. Let us denote for simplicity

$$
\begin{aligned}
SH_3^1(z) &= \alpha_{3,3}(\underline{z}, x_1, x_2)z^3 + \alpha_{3,2}(\underline{z}, x_1, x_2)z^2 + \alpha_{3,1}(\underline{z}, x_1, x_2)z + \alpha_{3,0}(\underline{z}, x_1, x_2) \\
SH_2^1(z) &= \alpha_{2,2}(\underline{z}, x_1, x_2)z^2 + \alpha_{2,1}(\underline{z}, x_1, x_2)z + \alpha_{2,0}(\underline{z}, x_1, x_2) \\
SH_1^1(z) &= \alpha_{1,1}(\underline{z}, x_1, x_2)z + \alpha_{1,0}(\underline{z}, x_1, x_2) \\
SH_0^1(z) &= \alpha_{0,0}(\underline{z}, x_1, x_2)
\end{aligned}
$$

*Now we need to evaluate the sign sequence of $\{SH_i^1(\infty),\ i = 0, \ldots, 3\}$ and of $\{SH_i^1(-\infty),\ i = 0, \ldots, 3\}$.
Let us look at the first element. Since $\alpha_{3,3}(\underline{z}, x_1, x_2) = 1$ we have that*

$sign(SH_3^1)(\pm\infty)$ is well determined. For the next element, SH_2^1 we have

$$sign(SH_2^1(\infty)) = \begin{cases} sign(\alpha_{2,2}) & , \alpha_{2,2} \neq 0 \\ sign(\alpha_{2,1}) & , \alpha_{2,2} = 0 \textit{ and } \alpha_{2,1} \neq 0 \\ sign(\alpha_{2,0}) & , \textit{otherwise} \end{cases}$$

and

$$sign(SH_2^1(-\infty)) = \begin{cases} sign(\alpha_{2,2}) & , \alpha_{2,2} \neq 0 \\ -sign(\alpha_{2,1}) & , \alpha_{2,2} = 0 \textit{ and } \alpha_{2,1} \neq 0 \\ sign(\alpha_{2,0}) & , \textit{otherwise} \end{cases}$$

In the same way we can consider the signs of all the elements in the Sylvester-Habicht sequence, for $-\infty$ as well as for ∞, and evaluate $V(P,(z-\underline{z})P')$. For example, when $\alpha_{2,2} > 0,\ \alpha_{1,1} > 0, \alpha_{0,0} = 0$ the sign sequence at $+\infty$ is $\{+,+,+,0\}$ (no sign changes) while at $-\infty$ is $\{-,+,-,0\}$ (two sign changes), so in this case $V(P,(z-\underline{z})P') = 2$.

Next we evaluate $V(P,(z-\underline{z})^2P')$. We obtain a sequence of polynomials $\{SH_3^2, SH_2^2, SH_1^2, SH_0^2\}$. We notice that $SH_3^2 = SH_3^1$ since they are both equal to P. The expressions for SH_2^2, SH_1^2, SH_0^2 are rather complicated and we do not reproduce them here.

Then we select all cases which satisfy our condition $c(x_1,x_2,\underline{z}) = 0$ (see (3.18)), i.e. which satisfy $V(P,(z-\underline{z})P') = V(P,(z-\underline{z})^2P')$. Their union is exactly defining the semi-algebraic set we were looking for.

The procedure is straight-forward but the symbolic expressions of the coefficients of SH_i^1, SH_i^2 are complicated, therefore we do not write down these conditions. We remark however that $P(\underline{z},x)$ appears as a factor in both $SH_0^1(z)$ and $SH_0^2(z)$. That implies that the only interesting case for us, due to the condition $P(\underline{z},x) = 0$, is when the both Sylvester-Habicht sequences end in 0. Since 0 does not affect the number of sign changes we will simply ignore it in the following. Let us construct now a table containing all possible sign sequences of $\{SH_3^1(-\infty), SH_2^1(-\infty), SH_1^1(-\infty)\}$ and $\{SH_3^1(\infty), SH_2^1(\infty), SH_1^1(\infty)\}$ depending on the signs of their coefficients, and the difference in the number of sign change corresponding to each such pair of sequences. Note that, for brevity, the table below corresponds to the situation where $\alpha_{1,1} \neq 0$. The first group of 4 columns corresponds to the situation $\alpha_{2,2} \neq 0$, the second group to $\alpha_{2,2} = 0 \ \wedge\ \alpha_{2,1} \neq 0$, the third group to $\alpha_{2,2} = 0 \ \wedge\ \alpha_{2,1} = 0 \ \wedge\ \alpha_{2,0} \neq 0$ and the last one, consisting out of 2 columns, to the situation in which $SH_2^1(z) \equiv 0$. The sign of $SH_i^1(\pm\infty)$ depends on the sign of the highest degree (non-zero) coefficient and the parity of the highest degree.

$SH_3^1(-\infty)$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$SH_2^1(-\infty)$	+	+	-	-	+	+	-	-	+	+	-	-	0	0
$SH_1^1(-\infty)$	+	-	+	-	+	-	+	-	+	-	+	-	+	-
$SH_3^1(\infty)$	+	+	+	+	+	+	+	+	+	+	+	+	+	+
$SH_2^1(\infty)$	+	+	-	-	-	-	+	+	+	+	-	-	0	0
$SH_1^1(\infty)$	-	+	-	+	-	+	-	+	-	+	-	+	-	+
V^1	0	2	0	-2	0	0	0	0	0	2	0	-2	0	0

Here V^1 stands for $V(P,(z-\underline{z})P')$. After considering in a similar way the sequence $\{SH_3^2(z), SH_2^2(z), SH_1^2(z)\}$, one has to consider all cases (that is, all sets of conditions on the coefficients of $SH_i^1(z)$, respectively $SH_j^2(z)$) which lead to $V(P,(z-\underline{z})P') = V(P,(z-\underline{z})^2P')$. These conditions will be in the form of polynomial equalities or inequalities and, considered in the problem (3.17).

The method can be applied in principle to the following problem

$$\max_{x \in \mathbf{R}^n} \underline{\sigma}(A(x)), \tag{3.20}$$

where $\underline{\sigma}(A(x))$ denotes the smallest singular value of the matrix $A(x)$, whose entries are real rational functions of $x \in \mathbf{R}^n$. Clearly, problem (3.20) can be rewritten in the form (3.17) with $P(z,x)$ being the numerator of $\det(zI - A^T(x)A(x))$. Using the Sylvester-Habicht procedure, problem (3.20) reduces to solving a polynomial optimization problem, more precisely, a constraint optimization problem with linear criterion function and polynomial equality and inequality constraints. The latter problem can be solved for example using numerical methods of Section 3.1.2.

3.4 Conclusions

We have presented in this chapter both numerical (approximative) methods and algebraic (exact) methods for optimization of polynomial functions. We will try to summarize here the main aspects characterizing each type of method.

It is easy of course to note that from the point of view of the computational complexity, the numerical methods seem to have an advantage over the algebraic methods. This is of course not a surprise. However we should remark here that even the numerical algorithms cannot *easily* solve the problem, their complexity increasing exponentially with the degree of the polynomial or the number of variables.

An important remark is that the numerical methods *do not* solve the polynomial optimization problem (3.1) but convex relaxations of it, returning therefore, in general, not the sought infimum (supremum) but a lower (upper) bound of it. As we have noted already, this is still interesting enough and these methods may be the first to give such bounds. Also, the hope is that for a particular problem at hand, one might be lucky that the bound and the actual optimum coincide.

Another problem that might occur in the numerical algorithm case is related to round-off errors. It may be crucial in some cases to decide whether a given polynomial is nonnegative everywhere or not. We do not enter into details here, however one can find such an example in the next chapter.

Of course, in the case of the algebraic methods, the actual optimum (and not a lower/upper bound of it) is found and one can decide with arbitrary accuracy the value of the optimum.

Besides, we have tried to show in Section 3.3 that the algebraic algorithms can reach much further than the numerical ones and can be employed for symbolic computations. The section *hints* to a new area of applications of these algorithms but it may be, at the moment, still somewhat underdeveloped.

Chapter 4

Global optimization of rational functions

This chapter presents extensions of the algorithms of Chapter 3 to the class of rational functions. The main contribution here is constituted by the different reformulations of the problem based on theoretical results from real algebraic geometry, as well as by the exact and the numerical algorithms proposed for solving them (see Sections 4.1 and 4.2). We also compare our methods with a numerical algorithm based on [42] (see Section 4.2.2). The basic ideas of the methods presented in this chapter are indeed different and that reflects on their performance in different examples. We intend to illustrate these ideas in a few cases.

4.1 Unconstrained optimization of rational functions

In this section we will be concerned with the following problem

$$\inf_{x\in\mathbf{R}^n}\frac{p(x)}{q(x)} \quad \text{with} \quad p(x),\ q(x)\in\mathbf{R}[x] \quad \text{relatively prime.} \tag{4.1}$$

It will be considered, using different methods, throughout the entire Section 4.1. Here we rewrite the rational optimization problem as a certain constrained polynomial optimization problem for which exact or numerical methods can be subsequently used for solving it.

4.1.1 An equivalent formulation

In this subsection, a necessary condition is given for the function to have a finite infimum. In case the condition is satisfied, the problem is shown to be equivalent to a specific constrained polynomial optimization problem.

Let us first give a theoretical result. The result can be found in [9], although stated and proved differently, and constitutes a basic result in real algebraic geometry. However, we chose to give here a direct proof.

If $f\in\mathbf{R}(x)$ is a polynomial or rational function and $\exists x_1,\ x_2\in\mathbf{R}^n$ such that

$f(x_1) > 0$ and $f(x_2) < 0$, we say that f *changes sign on* $\mathbf{R}^n$. Otherwise we say that f does not change sign on $\mathbf{R}^n$.

Theorem 4.1.1 *Let $a(x)/b(x)$ be a real rational multivariate function, with $a(x)$, $b(x)$ relatively prime polynomials. If $a(x)/b(x) \geq 0$, $\forall x \in \mathbf{R}^n$ with $b(x) \neq 0$, then one of the two following statements holds:*

- $a(x) \geq 0, \quad b(x) \geq 0 \quad \forall x \in \mathbf{R}^n,$
- $a(x) \leq 0, \quad b(x) \leq 0 \quad \forall x \in \mathbf{R}^n.$

Proof Note that the condition $a(x)/b(x) \geq 0$, $\forall x \in \mathbf{R}^n$ with $b(x) \neq 0$ is equivalent, by multiplication with $b^2(x)$, to $a(x)b(x) \geq 0$, $\forall x \in \mathbf{R}^n$.

We want to prove that the decomposition of the polynomial $a(x)b(x)$ into irreducible factors has the following form

$$a(x)b(x) = \prod_{i=1}^{K_1} g_i(x)^{2m_i} \prod_{j=K_1+1}^{K_2} g_j(x)^{m_j}, \tag{4.2}$$

where $g_i, i = 1, \ldots, K_1$ are all the factors that change sign on $\mathbf{R}^n$ and $g_j, j = K_1 + 1, \ldots, K_2$ the factors that do not change sign on $\mathbf{R}^n$. In other words, we want prove that if there exists an irreducible divisor of $a(x)b(x)$ that changes sign on $\mathbf{R}^n$, it actually has an even power in the decomposition of $a(x)b(x)$. In this case, using the fact that a, b are relatively prime polynomials, it follows that all irreducible factors of a (respectively b), either do not change sign or, if they do, they appear at an even power in the decomposition of a (respectively b) into irreducible factors. This proves that neither a nor b changes sign on $\mathbf{R}^n$. Since their product is non-negative, it also implies that in fact they are both either non-negative or non-positive.

In order to prove (4.2), let us consider $g_1 \in \mathbf{R}[x]$, an irreducible divisor of $g(x) = a(x)b(x)$ which changes sign. By Theorem 4.5.1 of [6], the ideal generated by g_1, and denoted (g_1), is a real ideal.

Let us denote

$$\frac{g(x)}{g_1(x)} = \tilde{g_1}(x),$$

which can be rewritten equivalently

$$g(x) = \tilde{g_1}(x)g_1(x) \geq 0 \quad \forall x \in \mathbf{R}^n.$$

A nonnegative polynomial can be written as a sum of squares of rational functions (Theorem 2.3.2 or [6], Theorem 6.1.1). Formally, there exist polynomials $r(x)$, $s_i(x)$, $i = 1, \ldots, m$ such that

$$r^2(x)\tilde{g_1}(x)g_1(x) = \sum_{i=1}^{m} s_i^2(x) \quad \forall x \in \mathbf{R}^n. \tag{4.3}$$

Take r minimal with respect to the division, having the property that $r^2(x)\tilde{g_1}(x)g_1(x)$ can be written as a sum of squares of polynomials.

The left hand side obviously belongs to the real ideal (g_1). By the definition of a real ideal ([6], Definition 4.1.3), the relation (4.3) implies that $s_i \in (g_1)$, $i = 1, \ldots, n$. Hence there exist polynomials $t_i(x)$ such that $s_i(x) = t_i(x) g_1(x)$.

By replacing s_i's in (4.3) and dividing both sides of the equality by g_1 we get

$$r^2(x)\tilde{g_1}(x) = g_1(x) \sum_{i=1}^{m} t_i^2(x) \quad \forall x \in \mathbf{R}^n. \tag{4.4}$$

Therefore g_1 must divide $r^2(x)\tilde{g_1}(x)$ and since g_1 is irreducible, g_1 divides $\tilde{g_1}$ or g_1 divides r.

Suppose first that g_1 divides r. Then there exists a polynomial $r_1(x)$ satisfying $r(x) = g_1(x) r_1(x)$. By replacing r into (4.4) and dividing both sides of the equality by $g_1(x)$ we obtain

$$r_1^2(x)\tilde{g_1}(x)g_1(x) = \sum_{i=n}^{m} t_i^2(x), \quad \forall x \in \mathbf{R}^n.$$

However, by comparing with (4.3) we obtain a contradiction with the minimality of r. Hence it must be that g_1 divides $\tilde{g_1}$ which implies g_1^2 divides g.

If the polynomial g/g_1^2 also has irreducible divisors which change sign, the same procedure can be applied to the g/g_1^2. In this way, one can show that any irreducible factor of $g(x) = a(x)b(x)$ which changes sign must have an even power in the decomposition of $g(x)$. □

The theorem above states that when a rational function $a(x)/b(x)$, with a, b relatively prime, does not change sign on its entire domain of definition, then neither the denominator nor the numerator changes sign on $\mathbf{R}^n$. This is quite a strong result.

Remark 4.1.2 *The hypothesis a, b are relatively prime polynomials can be checked by computing the greatest common divisor* $\mathrm{GCD}(a, b)$, *using for example the algorithms of [13] or [12]. If* $\mathrm{GCD}(a, b) \neq 1$, *then one can divide both a and b by* $\mathrm{GCD}(a, b)$. *The equivalent representation of the rational function obtained in this way satisfies the hypothesis of Theorem 4.1.1.*

In the following we use Theorem 4.1.1 in order to obtain some criteria for our problem. An immediate consequence of Theorem 4.1.1 is formulated below:

Theorem 4.1.3 *Let $p(x)/q(x)$ be a rational function with $p(x)$, $q(x)$ relatively prime polynomials. If $q(x)$ changes sign on $\mathbf{R}^n$ then $\inf_{x \in \mathbf{R}^n} p(x)/q(x) = -\infty$.*

Proof We prove this result by reductio ad absurdum, that is, we assume that there exists a finite lower bound on the function $p(x)/q(x)$. Let $\alpha \in \mathbf{R}$ be such that $p(x)/q(x) \geq \alpha \;\; \forall x \in \mathbf{R}^n$ with $q(x) \neq 0$. Then $(p(x) - \alpha q(x))/q(x)$ satisfies the hypothesis of Theorem 4.1.1 hence both $p(x) - \alpha q(x)$ and $q(x)$ do not change sign on $\mathbf{R}^n$. That contradicts the hypothesis that q changes sign. □

Note that the converse is not true, i.e. $\inf_{x \in \mathbf{R}^n} p(x)/q(x)$ may be $-\infty$ even if q does not change sign on $\mathbf{R}^n$. An obvious example is $\inf_{x \in \mathbf{R}} -1/x^2$.

In case q does not change sign on $\mathbf{R}^n$, we can reformulate the problem (4.1) in the following way. Assume, without loss of generality, that $q(x) \geq 0, \ \forall x \in \mathbf{R}^n$. Then using the proof of Theorem 4.1.3, problem (4.1) is equivalent to

$$\begin{array}{ll} \sup & \alpha \\ \text{s.t.} & p(x) - \alpha q(x) \geq 0, \quad \forall x \in \mathbf{R}^n. \end{array} \tag{4.5}$$

Obviously the largest α satisfying the condition is the infimum of $p(x)/q(x)$.

Note that the feasibility domain of (4.5) may be the empty set. That is, there is no $\alpha \in \mathbf{R}$ satisfying the polynomial inequality for every $x \in \mathbf{R}^n$. In this case the supremum will be $-\infty$.

The fact that q does not change sign on $\mathbf{R}^n$ can be checked in the following way:

1. Evaluate q at an arbitrary point and suppose that it is positive (otherwise work with the fraction $-p/(-q)$);
2. Then q is non-negative on $\mathbf{R}^n$ if and only if $\inf_{x \in \mathbf{R}^n} q(x) \geq 0$. Hence we only need to compute the infimum of a polynomial on $\mathbf{R}^n$ and this can be done using for example the algorithms described in Chapter 3.

Note the importance of the condition $\inf_{x \in \mathbf{R}^n} q(x) \geq 0$ at step 2. If the infimum is just slightly smaller than 0, the conclusion may be a completely different one. Therefore, one might have to use *exact* methods, as in Section 3.2, for computing the value $\inf_{x \in \mathbf{R}^n} q(x)$.

To conclude, in this section we have rewritten the rational optimization problem as a constrained polynomial optimization problem. Several options are possible now. We intend to discuss here an exact method as well as a numerical method. Throughout the rest of the Section 4.1, we consider the case $q(x) \geq 0, \ \forall x \in \mathbf{R}^n$.

4.1.2 An exact solution

We deal here with equation (4.5) which is a particular constrained polynomial optimization problem. We claim that we can solve this in an exact way using a variant of Algorithm 3.2.18 of Section 3.2.

The problem (4.5) is nothing else than finding the largest α for which the following holds:

$$\inf_{x \in \mathbf{R}^n} [p(x) - \alpha q(x)] \ \geq \ 0. \tag{4.6}$$

Let us look at (4.6) which is a polynomial optimization problem over $x \in \mathbf{R}^n$ of a family of polynomials, family parameterized by α. However, as discussed in

Section 3.3, an exact algorithm for polynomial optimization, can handle parameters in the expression of the polynomial to be optimized. The optimum would obviously depend on the parameter, in this case α, either implicitly or explicitly. The fact that one can optimize over a class using an exact algorithm constitutes actually a major difference (and advantage) of the exact algebraic algorithm compared to numerical algorithms. We are going to exploit this feature in our approach.

In Section 3.3 we have discussed the optimization of an arbitrary family of polynomials under the Assumption 1. However the problem at hand (4.6) is a particular case in which the family of polynomials depends linearly on a single parameter. We want to exploit this particularity here and give a general theory for this case, without considering the Assumption 1 satisfied.

Let us first study more closely the infima of the family of polynomials. For that, we define

$$M(\alpha) = \inf_{x\in\mathbf{R}^n} \left[p(x) - \alpha q(x)\right].$$

Proposition 4.1.4 *The function $M : \mathbf{R} \to \mathbf{R} \cup \{-\infty\}$ is non-increasing on $\mathbf{R}$.*

Proof Let $\alpha_1, \alpha_2 \in \mathbf{R}$, with $\alpha_1 < \alpha_2$. Using the fact that $q(x) \geq 0,\ \forall x \in \mathbf{R}^n$, it follows immediately that $p(x) - \alpha_1 q(x) \geq p(x) - \alpha_2 q(x),\ \forall x \in \mathbf{R}^n$. Hence $M(\alpha_1) \geq M(\alpha_2)$. □

One can prove immediately the following.

Corollary 4.1.5 *The following hold:*

- *If $\exists \alpha_0 \in \mathbf{R}$ such that $M(\alpha_0) > -\infty$, then $M(\alpha) > -\infty,\ \forall \alpha < \alpha_0$.*
- *If $\exists \alpha_1 \in \mathbf{R}$ such that $M(\alpha_1) = -\infty$, then $M(\alpha) = -\infty,\ \forall \alpha > \alpha_1$.*

Proposition 4.1.6 *The function $M : \mathbf{R} \to \mathbf{R} \cup \{-\infty\}$ is piecewise continuous.*

Proof According to Section 3.3, if $M(\alpha)$ is finite, then $M(\alpha)$ is the smallest (admissible) real root in z of $P(z, \alpha)$, polynomial in both z and α ($P(z, \alpha)$ was obtained by running the Algorithm 3.2.18). The roots z of $P(z, \alpha)$ describe actually an algebraic function and therefore the smallest admissible real root, $M(\alpha)$, will be piecewise continuous. □

Problem (4.5) can be reformulated as

$$\begin{array}{ll} \sup & \alpha \\ \text{s.t.} & M(\alpha) \geq 0. \end{array} \tag{4.7}$$

Various methods, like bisection, can be applied now to problem (4.5), using the fact that $M(\alpha)$ is piecewise continuous and decreasing. Note that the bisection method is an *exact* method in the sense that it returns, at a certain computational cost, an arbitrarily precise approximation of the solution.

We would like however, for computational purposes, to use the polynomial $P(z, \alpha)$, which is known, instead of $M(\alpha)$, which is only implicitly given. We notice that the problem simplifies in the case when the infimum of (4.1) is attained.

Proposition 4.1.7 *If* $\inf_{x\in\mathbf{R}^n} p(x)/q(x)$ *is attained, then the problem* (4.5) *is equivalent to*

$$\begin{array}{ll} \sup & \alpha \\ \text{s.t.} & M(\alpha) = 0. \end{array} \tag{4.8}$$

In general however, (4.8) *is a lower bound of* (4.7).

Proof The value of (4.7) is always larger than or equal to the value of (4.8) since its constraint is less restrictive. Now we prove that if our hypothesis is satisfied, they are actually equal. Let $\alpha_* \in \mathbf{R}$ denote the value of (4.7) and let $x_* \in \mathbf{R}^n$ such that $\alpha_* = p(x_*)/q(x_*) = \inf_{x\in\mathbf{R}^n} p(x)/q(x)$. This implies that

$$p(x) - \alpha_* q(x) \geq 0, \quad \forall x \in \mathbf{R}^n \quad \text{and} \quad p(x_*) - \alpha_* q(x_*) = 0.$$

In other words,

$$M(\alpha_*) = \inf_{x\in\mathbf{R}^n} [\; p(x) - \alpha_* q(x) \;] = 0$$

and therefore the value of (4.7) is smaller than or equal to the value of (4.8). Hence they are equal. □

Recall that, when $M(\alpha)$ is finite, $M(\alpha)$ satisfies $P(M(\alpha), \alpha) = 0$, where $P(z, \alpha)$ is the polynomial obtained by running Algorithm 3.3.1 for the family of polynomials $p(x) - \alpha q(x)$. The solution of (4.8) is therefore obtained as a root of $P(0, \alpha) = 0$. Note that $P(0, \alpha)$ is a univariate polynomial, hence it has a finite number of roots. These will be our candidates for the solution of (4.8). Note that one needs to check the positivity of $p(x) - \alpha q(x)$ for a finite number of values of α, that is for the roots of $P(0, \alpha)$. In this case $M(\alpha)$ decreases *through* 0.

Let us formalize the procedure into an algorithm, based on Algorithm 3.3.1 for families of polynomials depending on a parameter.

Algorithm 4.1.8 *The following procedure computes the solution of* (4.8), *which is the solution, or a lower bound, of* (4.1).

1. *Construct the matrix* $A_{p-\alpha q}(\lambda)$.
2. *Compute the* HOCM$((\bar{A}_{p-\alpha q}(1/\lambda, z))$ *by running the Algorithm 3.2.15.*
3. *Compute* $P(z, \alpha) = \det(\mathrm{HOCM}((\bar{A}_{p-\alpha q}(1/\lambda, z))/\Delta$, *polynomial in* z *and* α.

4. *Compute the roots* $\alpha_1, \ldots, \alpha_r \in \mathbf{R}$ *of* $P(0, \alpha)$, *with* $\alpha_1 \geq \ldots \geq \alpha_r$; $i \leftarrow 1$.
5. *If* $\inf_{x \in \mathbf{R}^n} [p(x) - \alpha_i q(x)]$ *is indeed nonnegative, then go to step 6. If not,* $i \leftarrow i + 1$ *and repeat step 5.*
6. *Output:* α_i.

Remark that at step 5, α_i is a number, hence we have an infimization of a polynomial. It is important to verify whether $\inf_{x \in \mathbf{R}^n} [p(x) - \alpha_i q(x)] = -\infty$ or not, and this can be done as in Section 3.2.4. In certain applications, where optimization of a rational function is involved, it is known apriori that the minimum is attained. In such case Algorithm 4.1.8 turns out to be very useful. For the optimization of an arbitrary rational function (i.e., the minimum is not necessarily attained), bisection can be applied for determining the value of α at which $M(\alpha)$ passes over 0. Such a procedure is computationally more involved. Note also that the Algorithm 4.1.8 returns in general a lower bound for the sought value of α. To illustrate the approach we present here a simple example. All computations were done using a Maple implementation of the Algorithm 3.2.18.

Example 4.1.9

$$\inf_{(x_1, x_2) \in \mathbf{R}^2} \frac{x_1^2 + x_2^2}{(x_1 - 1)^2}.$$

The infimum is obviously 0 *and it is actually attained at* $(x_1, x_2) = (0, 0)$. *Following our approach from the previous section we first check that the numerator and denominator have no common factors and that the denominator does not change sign on* $\mathbf{R}^2$. *Then we rewrite the problem as in* (4.5)

$$\begin{array}{ll} \sup & \alpha \\ s.t. & x_1^2 + x_2^2 - \alpha(x_1 - 1)^2 \geq 0, \quad \forall (x_1, x_2) \in \mathbf{R}^2. \end{array} \tag{4.9}$$

By applying the Algorithm 4.1.8 *to our family of polynomials,* $p(x) - \alpha q(x) = x_1^2 + x_2^2 - \alpha(x_1 - 1)^2$, *we obtain*

$$P(z, \alpha) = (\alpha - 1)^3 (2 - 2\alpha + \alpha^2)^4 (-\alpha + z(\alpha - 1)) \tag{4.10}$$

where the smallest root of $P(0, \alpha)$ *is our candidate for* $M(\alpha)$. *By solving* $P(0, \alpha) = 0$ *we obtain as values for* α, 0 *with multiplicity* 1, 1 *with multiplicity* 4 *and a few complex solutions* ($1 + i$ *and* $1 - i$, *both with multiplicity* 4). *The largest among the real values,* 1, *is our first candidate. However, it turns out that for* $\alpha = 1$ *the infimum is indeed infinite* ($M(1) = -\infty$), *but finite for* $\alpha = 0$. *Hence we conclude that* 0 *is the solution. Note that the CPU-time needed to compute the polynomial* $P(z, \alpha)$ *equals* $0.97s$.

In this simple example it is possible to obtain an explicit expression of the function $M(\alpha)$. *We compute*

$$M(\alpha) = \inf_{x \in \mathbf{R}^n} \left[(1 - \alpha) x_1^2 + 2\alpha x_1 - \alpha + x_2^2 \right]$$

by noticing that it is the sum of two univariate polynomials, depending on different variables. Therefore, the infimum of the sum equals the sum of the infima of the two polynomials. Since each of the two polynomials is quadratic, the two infima can be computed for any value of α. We have

$$\inf_{x\in\mathbf{R}^2} [\ (1-\alpha)x_1^2 + 2\alpha x_1 - \alpha\ ] \ = \ \begin{cases} -\alpha/(1-\alpha) & , \ \alpha < 1, \\ -\infty & , \ \alpha \geq 1, \end{cases}$$

and

$$\inf_{x\in\mathbf{R}^n} x_2^2 \ = \ 0,$$

hence

$$M(\alpha) = \begin{cases} -\alpha/(1-\alpha) & , \ \alpha < 1, \\ -\infty & , \ \alpha \geq 1. \end{cases}$$

With the computed value of $M(\alpha)$, we obtain immediately that $\alpha = 0$ is the solution of problem (4.7). Notice however that the jump to $-\infty$ occurs at $\alpha = 1$.

Notice that in Example 4.1.9, the polynomial $P(z,\alpha)$ could be factored as $P(z,\alpha) = P_0(\alpha)\tilde{P}(z,\alpha)$. It is clear that every root α_0 of P_0 plays a special role in this approach since $P(z,\alpha_0) \equiv 0$ for every $z \in \mathbf{R}$. Notice also that the jump point (i.e. the value of α at which $M(\alpha)$ jumps to $-\infty$) was a root of the polynomial $P_0(\alpha)$. This behavior was observed in all examples we ran. We conjecture that the jump point of $M(\alpha)$ to $-\infty$ is a root of $P_0(\alpha)$ in general.

4.1.3 A numerical solution using LMI's

In this subsection we propose to solve (4.5) using a numerical method. For that, we study the extension to rational functions of the method presented in Section 3.1.1 (see also [59] and [50]), used previously for polynomial functions. As in Section 3.1.1, we want to rewrite the rational optimization problem into a semi-definite programming (SDP) problem, also called a linear matrix inequality (LMI) problem, which is known to have good computational complexity. In general, as in the polynomial case, we obtain an LMI relaxation of the original problem, which gives a lower bound for the solution of the original problem.

Let us study now how to rewrite the problem (4.5) as an LMI. For this we consider the polynomial $F(x) = p(x) - \alpha q(x)$. We use of course the method described in Section 3.1.1 in order to construct a matrix Q corresponding to F. Here F is considered as a polynomial in x, and α is a parameter. Let $2d$ be the total degree of F in x. Define $z^T = [1, x_1, x_2, \ldots, x_n, x_1x_2, \ldots, x_n^d]$. Then using relation (3.3) one can construct the generic symmetric matrix Q which satisfies (3.3).

Note that in Section 3.1.1 (3.4), Q describes an affine subspace, parameterized by a vector λ. We extend this result to the case of $F(x) = p(x) - \alpha q(x)$ (hence depending on α) and show that the generic matrix Q, satisfying (3.3), describe a subspace which is affine in α and in the entries of the vector λ.

Theorem 4.1.10 *Let the symmetric matrix Q satisfy $p(x) - \alpha q(x) = z^T Q z$. Then Q describes an affine subspace, that is*

$$Q = Q_0 + \sum_{i=1}^{\kappa} Q_i \lambda_i + Q_{\kappa+1}\alpha, \qquad (\alpha, \lambda) \in \mathbf{R}^{\kappa+1}. \tag{4.11}$$

Proof It is well known that the linear part of the affine subspace is completely determined by the relations between monomials of z. The fact that Q is affine in α is due to the fact that the polynomial $p(x) - \alpha q(x)$ is linear in α. □

Let us denote the matrix of (4.11) by $Q(\alpha, \lambda)$, with $\lambda \in \mathbf{R}^\kappa$, $\alpha \in \mathbf{R}$.

Let us look at the LMI problem:

$$\begin{array}{ll} \sup & \alpha \\ \text{s.t.} & Q(\alpha, \lambda) \succeq 0. \end{array} \tag{4.12}$$

Indeed, since $Q(\alpha, \lambda)$ is symmetric, the matrix coefficients of α and the elements of the vector λ will be symmetric matrices. Moreover, $Q(\alpha, \lambda)$ is affine in α and λ, hence the problem is a standard LMI problem (or equivalently an SDP).

The relation between the problems (4.12) and (4.5) is studied in the following.

Theorem 4.1.11 *Let us denote by $\alpha_{RAT} \in \mathbf{R} \cup \{-\infty\}$ the solution of the problem* (4.5)*, and consequently of the rational optimization problem* (4.1)*, and by $\alpha_{LMI} \in \mathbf{R} \cup \{-\infty\}$ the solution of* (4.12)*. Then we have*

$$\alpha_{RAT} \geq \alpha_{LMI}.$$

If $p(x) - \alpha_{RAT} q(x)$ can be written as a sum of squares of polynomials, then

$$\alpha_{RAT} = \alpha_{LMI}.$$

Proof Let λ_{LMI} be such that $(\alpha_{LMI}, \lambda_{LMI})$ satisfy (4.12). Since

$$p(x) - \alpha_{LMI} q(x) = z^T Q(\alpha_{LMI}, \lambda_{LMI}) z \quad \text{and} \quad Q(\alpha_{LMI}, \lambda_{LMI}) \succeq 0$$

we have

$$p(x) - \alpha_{LMI} q(x) \geq 0, \quad \forall x \in \mathbf{R}^n.$$

Hence α_{LMI} satisfies the constraints of (4.5) and therefore

$$\alpha_{RAT} \geq \alpha_{LMI}.$$

If $p(x) - \alpha_{RAT} q(x)$ can be written as a sum of squares, then there exists a λ_{RAT} such that

$$p(x) - \alpha_{RAT} q(x) = z^T Q(\alpha_{RAT}, \lambda_{RAT}) z, \qquad Q(\alpha_{RAT}, \lambda_{RAT}) \succeq 0.$$

Hence

$$\alpha_{RAT} \leq \alpha_{LMI}.$$

From the result above, equality holds in fact. □

If the polynomial $F(x) = p(x) - \alpha q(x)$ is in one of the first two cases of Theorem 2.3.1 then, according to Theorem 4.1.11, the algorithm will find the infimum. If not, then there is always a polynomial $G(x)$ such that $F(x)G^2(x)$ can be written as a sum of squares of polynomials (see Theorem 2.3.2). It is not clear however how to choose the polynomial $G(x)$.

From the practical point of view we are more interested in deciding whether for a particular rational function the infimum was found or just a lower bound of it. For that, the checking procedure of Section 3.1.1 can be used.

Although in general it returns just a lower bound of the sought infimum, the numerical approach of this section may be very important in applications. The translation of the rational optimization problem into this setting was immediate. We illustrate the method by an example.

Example 4.1.12

$$\inf_{x \in \mathbf{R}^3} \frac{(x_1 + x_2)^4 + {x_1}^3 x_3}{{x_1}^4 + {x_3}^4}.$$

This translates, using (4.5), into

$$\begin{array}{ll} \sup & \alpha \\ s.t. & (x_1 + x_2)^4 + {x_1}^3 x_3 - \alpha({x_1}^4 + {x_3}^4) \geq 0, \quad \forall x \in \mathbf{R}^3. \end{array}$$

Since the polynomial is homogeneous of even degree, according to [50] it is sufficient to consider in the vector z, all monomials in the variables x having as degree half the degree of the original polynomial. In our case, this will be 2, hence we define $z^T = \begin{bmatrix} {x_1}^2 & {x_2}^2 & {x_3}^2 & x_1x_3 & x_1x_2 & x_2x_3 \end{bmatrix}$.
We compute the symmetric matrix $Q(\alpha, \lambda)$ using the identity of polynomials

$$z^T Q(\alpha, \lambda) z = {x_1}^4 + {x_2}^4 + {x_1}^2{x_2}^2 - \alpha \left({x_1}^2 + {x_3}^2\right)^2. \tag{4.13}$$

We obtain

$$Q(\alpha, \lambda) = \begin{bmatrix} -\alpha + 1 & \lambda_3 & \lambda_4 & 1/2 & 2 & -\lambda_5 \\ \lambda_3 & 1 & \lambda_2 & -\lambda_6 & 2 & 0 \\ \lambda_4 & \lambda_2 & -\alpha & 0 & -\lambda_1 & 0 \\ 1/2 & -\lambda_6 & 0 & -2\lambda_4 & \lambda_5 & \lambda_1 \\ 2 & 2 & -\lambda_1 & \lambda_5 & -2\,\lambda_3 + 6 & \lambda_6 \\ -\lambda_5 & 0 & 0 & \lambda_1 & \lambda_6 & -2\,\lambda_2 \end{bmatrix}. \tag{4.14}$$

$Q(\alpha,\lambda)$ given by (4.14), with (α,λ) varying over $\mathbf{R}\times\mathbf{R}^6$, gives a complete description of the set of symmetric matrices $Q(\alpha,\lambda)$ satisfying (4.13). With this $Q(\alpha,\lambda)$, let us solve the LMI problem (4.12). Suitable algorithms can be employed for solving it. By running SeDuMi 1.03 (see [66]) for the above LMI problem, we obtain the solution of (4.12), -0.5699. Since our problem is one of the special cases mentioned in Theorem 2.3.1, we know that this is the actual infimum.

Let us however, perform the checking procedure as described in Section 3.1.1. We compute the solution $Y^*\in\mathbf{R}^{6\times6}$ of the dual problem of (4.12) and notice, by running Gaussian elimination, that the matrix Y^* has indeed rank 1. Hence, there exists a z^* such that $Y^*=z^*z^{*T}$, where

$$z^{*T}=(0.7514, 0.7529, 0.4338, -0.5709, -0.7521, 0.5715).$$

From z^* and the definition of the vector z we recover the solution point $x^*=(-0.8668, 0.8677, 0.6587)$. The rational function evaluated at x^* is equal to the value we have previously found, -0.5699, as expected. We therefore conclude that the infimum of the function is actually attained and one such point is x^*.

4.2 Constrained optimization of rational functions

The problem studied in this section is an optimization problem where the objective is a rational function and the constraints are polynomial inequalities. To formalize

$$\begin{array}{ll}\inf & p(x)/q(x)\\ \text{s.t.} & r_i(x)\geq 0,\quad i=1,\ldots,l.\end{array}\tag{4.15}$$

with $p(x),\ q(x), r_i(x)\in\mathbf{R}[x],\quad i=1,\ldots,l$ and $p(x),\ q(x)$ relatively prime.

One idea for constrained optimization is, especially when the constraints are equalities, to work modulo the feasibility set. However a bit of care is required when working with equivalent representations of the criterion function, as shown in the following example (Example 6.1.8 of [6])

Example 4.2.1

$$\begin{array}{ll}\inf & [x^2+y^2-z^2]\\ s.t. & x^3=z(x^2+y^2).\end{array}$$

The feasibility set describes the Cartan umbrella. *One can check that the criterion $x^2+y^2-z^2$ is equivalent, modulo $x^3=z(x^2+y^2)$, to $(3x^4y^2+3x^2y^4+y^6)/(x^2+y^2)^2$. They are indeed equal except a* thin *subset $\{(0,0,z)\mid z\in\mathbf{R}\}\subset\mathbf{R}^3$ on which the second rational function is not defined. However, while the first function has no lower bound on the feasible set, that is its infimum is $-\infty$, the other one is nonnegative everywhere. Hence*

$$\begin{array}{ll}\inf & [x^2+y^2-z^2]\\ s.t. & x^3=z(x^2+y^2)\end{array}\quad\neq\quad\inf\ (3x^4y^2+3x^2y^4+y^6)/(x^2+y^2)^2.$$

However

$$\begin{array}{lll} \inf \ [x^2+y^2-z^2] & & = \\ s.t.\ x^3 = z(x^2+y^2) & & \\ & = \min & \{\inf_{(x,y)\neq(0,0)} \frac{3x^4y^2+3x^2y^4+y^6}{(x^2+y^2)^2}, \inf_{(x,y)=(0,0)} -z^2\} = \\ & = \min & \{0, -\infty\} \quad = \quad -\infty. \end{array}$$

In this section, we investigate the possibility of extending the results of Section 4.1 to constrained rational optimization. There are two aspects that need to be considered in such an extension. The first one is the extension of theoretical results. More precisely, we want to know whether Theorem 4.1.1 of Section 4.1 can be extended to the constrained case. It is probably the more important aspect, in the sense that, it would allow us to rewrite rational constrained optimization problems as polynomial constrained optimization problems. The second aspect is more practical, involving methods, exact or numerical, to solve the new constrained polynomial optimization problem.

Let us investigate now the possibility of extending the theoretical result of Theorem 4.1.1 to the constraint case mentioned. We will show some of the difficulties that appear in the constrained case, when the feasible set is *thin*, that is when its dimension drops. For that, we give an extremely simple counterexample, in two variables.

Example 4.2.2

$$\begin{array}{lll} \inf(x-y+z+1)/(x+y+z+1) & = & 1 \\ s.t.\ \ y^2+z^2=0. & & \end{array}$$

Here the numerator and denominator are relatively prime polynomials and the denominator changes sign on $\mathbf{R}^3$*. However, when restricted to the feasible set* $\{(x,0,0) \mid x \in \mathbf{R}\}$ *(which is a thin set), the rational function becomes* $(x+1)/(x+1) = 1$*,* $\forall x \in \mathbf{R}$*. Hence, in general, Theorem 4.1.1 will not hold in the constrained case as well.*

Example 4.2.2 shows that the results of Section 4.1 do not extend immediately to the constrained case. However, as we prove in the following, they can be extended for particular classes of feasible sets. In particular we discuss in Section 4.2.1 open feasible sets, and more specifically the simple case when the feasible domain is a ball $\mathbf{B} = \mathbf{B}(x_0, r)$. In Section 4.2.2 we present a different numerical method for this problem based on [42].

4.2.1 A particular case

We consider here the case when the feasible domain is a ball $\mathbf{B} = \mathbf{B}(x_0, r)$.

Theoretical results

Theorem 4.2.3 *Let* $a(x)$*,* $b(x)$ *be relatively prime polynomials and* $\mathbf{B}$ *an open ball in* $\mathbf{R}^n$*. If* $a(x)b(x) \geq 0$*,* $\forall x \in \mathbf{B}$*, then one of the two following statements* *holds:*

- $a(x) \geq 0, \quad b(x) \geq 0 \quad \forall x \in \mathbf{B}$,
- $a(x) \leq 0, \quad b(x) \leq 0 \quad \forall x \in \mathbf{B}$.

Proof Assume that a changes sign on $\mathbf{B}$, therefore there must exist an irreducible factor of a, denoted a_1, which changes sign on $\mathbf{B}$.

We follow the proof of Lemma 6.14 of [41]. We want to prove that $f = a_1$ divides $g = b$. We know that f changes sign in $\mathbf{B}$, that is there exist two points $\tilde{x}$, $\hat{x} \in \mathbf{B}$ such that $f(\tilde{x}) > 0$ and $f(\hat{x}) < 0$. Let us make a suitable change of coordinates such that $f(y, z_1) < 0 < f(y, z_2)$ where $y \in \mathbf{R}^{n-1}$, $z_1, z_2 \in \mathbf{R}$. This can be achieved by considering a system of coordinates for which one axis passes through $\hat{x}$ and $\tilde{x}$. After the change of coordinates, $\mathbf{B}$ becomes the ball $\tilde{\mathbf{B}}$. Let $G = \mathbf{R}[x_1, \ldots, x_{n-1}]$ and F the quotient ring of G. View f and g as polynomials in x_n in the ring $G[x_n] \subset F[x_n]$. Suppose that f does not divide g in $G[x_n](= \mathbf{R}[x_1, \ldots, x_n])$. We know that f remains irreducible in $F[x_n]$ and f does not divide g also in $F[x_n]$. Since $F[x_n]$ is a principal ideal domain, there exist $\rho, \gamma \in F[x_n]$ such that $f\rho + g\gamma = 1$. Write $\rho = \rho_0/h$ and $\gamma = \gamma_0/h$, where $\rho_0, \gamma_0 \in G[x_n]$ and $0 \neq h \in G$. Then $f\rho_0 + g\gamma_0 = h$. Choose a neighborhood V of y in $\mathbf{R}^{n-1}$ such that $V \times \{z_1\}$, $V \times \{z_2\} \subset \tilde{\mathbf{B}}$ and $f(V, z_1) < 0 < f(V, z_2)$. For any $v \in V$, $f(v, z_1) < 0 < f(v, z_2)$ implies that $f(v, b_v) = 0$ for some b_v between z_1 and z_2. Actually, since $f(x)g(x) \geq 0$ we have $g(V, z_1) \leq 0 \leq g(V, z_2)$ and there exists a b_v where both $f(v, b_v) = 0$ and $g(v, b_v) = 0$. Therefore $f\rho_0 + g\gamma_0 = h$ implies that $h(v) = 0$, $\forall v \in V$ and so $h(x_1, \ldots, x_{n-1})$ vanishes on a non-empty open set in $\mathbf{R}^{n-1}$. This forces $h \equiv 0$, a contradiction. Hence $a_1 = f$ divides $b = g$, but this contradicts the hypothesis that a and b are relatively prime. Hence, a cannot change sign on $\mathbf{B}$. □

Remark 4.2.4 *In Theorem 4.2.3 the condition $a(x)b(x) \geq 0$, $\forall x \in \mathbf{B}$ is equivalent to, and therefore can be replaced by, $a(x)/b(x) \geq 0$, $\forall x \in \mathbf{B}$, with $b(x) \neq 0$.*

As in Section 4.1 we formulate the following result.

Corollary 4.2.5 *Let $p(x)/q(x)$ be a rational function with $p(x)$, $q(x)$ relatively prime polynomials. If $q(x)$ changes sign on $\mathbf{B}$ then $\inf_{x \in \mathbf{B}} p(x)/q(x) = -\infty$.*

Proof The proof is identical to the one of Theorem 4.1.1. Assume $\exists\ \alpha \in \mathbf{R}$ a lower bound of the function. For every $x \in \mathbf{B}$, with $q(x) \neq 0$, we have

$$\frac{p(x)}{q(x)} \geq \alpha \iff \frac{p(x) - \alpha q(x)}{q(x)} \geq 0\ .$$

Applying Theorem 4.2.3, we deduce that both $p(x) - \alpha q(x)$ and $q(x)$ do not change sign on $\mathbf{B}$ which contradicts the hypothesis. □

Remark 4.2.6 *Theorem 4.2.3 and Corollary 4.2.5 remain valid if the open ball $\mathbf{B}$ is replaced by the closed ball $\bar{\mathbf{B}} = \{x \in \mathbf{R}^n \mid \|x - x_0\| \leq r\}$.*

We have shown that, in this simple case, when the feasible set is a ball $\mathbf{B}$, the theoretical result, Theorem 4.1.1, can be extended to the constrained case. In fact, Theorem 4.2.3 and Corollary 4.2.5 hold when the open ball $\mathbf{B}$ is replaced by a set C such that $D \subseteq C \subseteq \bar{D}$, where $D \subseteq \mathbf{R}^n$ is a connected open set. The proof is immediate. For simplicity, we choose to work in the following with the initial case, where the feasibility set is an open ball.

Let us now consider the practical aspects, namely the computations. For simplicity we consider here that $\mathbf{B} = \mathbf{B}(\mathbf{0}, 1)$. The extension to the case $\mathbf{B}(x_0, r)$ above is immediate.

Let us assume now that $q(x) \geq 0$, $\forall x \in \mathbf{B}$. An equivalent formulation of the problem $\inf_{x \in \mathbf{B}} p(x)/q(x)$ is, according to Theorem 4.2.3, the following:

$$\begin{array}{ll} \sup & \alpha \\ \text{s.t.} & p(x) - \alpha q(x) \geq 0, \quad \forall x \in \mathbf{B}. \end{array} \tag{4.16}$$

The two problems remain equivalent if $\mathbf{B}$ is replaced by $\bar{\mathbf{B}}$.

Exact methods-special case

We want to briefly mention here that a reparametrization which maps the whole space $\mathbf{R}^n$ into the closed ball $\bar{\mathbf{B}}$ brings us into the well-known unconstrained problem. For example,

$$m : \mathbf{R}^n \to \bar{\mathbf{B}}, \quad m(y_1, \ldots, y_n) = \left(\frac{2y_1}{1 + y_1^2 + \ldots + y_n^2}, \ldots, \frac{2y_n}{1 + y_1^2 + \ldots + y_n^2} \right)$$

is such a mapping. One can easily check that the mapping m is surjective. Moreover, $m|_{\bar{\mathbf{B}}} : \bar{\mathbf{B}} \to \bar{\mathbf{B}}$ is a bijection (which maps $\partial\mathbf{B}$ into $\partial\mathbf{B}$), and $m|_{\mathbf{R}^n \setminus \mathbf{B}} : \mathbf{R}^n \setminus \mathbf{B} \to \bar{\mathbf{B}}$ is bijective as well.

Using the mapping m, the problem

$$\inf_{x \in \bar{\mathbf{B}}} \frac{p(x)}{q(x)} \tag{4.17}$$

can be reduced to the unconstrained case by considering the problem

$$\inf_{y \in \mathbf{R}^n} \frac{p(m(y))}{q(m(y))}. \tag{4.18}$$

Obviously the two problems, (4.17) and (4.18), are equivalent and for (4.18) the methods of Section 4.1 (exact or numerical) can be applied.

Numerical methods based on LMI's

Obviously, the numerical methods can be applied to the reparametrized problem as in the previous subsection. However, that doubles the total degree of the polynomial. We believe it is worthwhile to investigate here a different approach. The approach may turn out to be more efficient in some cases.

For the new formulation we use Section 6.2 of [6] (see also Section 2.3 of this thesis). Hence, we know that

$$\begin{aligned} & p(x) - \alpha q(x) \geq 0, \quad \forall x \in \bar{\mathbf{B}} \iff \\ & p(x) - \alpha q(x) = \sum_{i=1}^{I} s_i^2(x) + \sum_{j=1}^{J} t_j^2(x)(1 - x_1^2 - \ldots - x_n^2), \end{aligned} \tag{4.19}$$

for some s_i, t_j rational functions. Let us remind here that $1 - x_1^2 - \ldots - x_n^2 \geq 0$ is equivalent to $x \in \bar{\mathbf{B}}$. For the actual computations we will consider s_i, t_j polynomials, not rational functions, solving in this way only a relaxation of the problem.

This is a good moment to notice the first difficulty arising in this setting. Due to the negative signs in the right-hand side of the formula (4.19), we do not have a bound on the degrees of the polynomials s_i and t_j. Hence a sequence of LMI's must be used, corresponding to increasing degrees of some of the s_i and t_j polynomials. One element of the sequence would have the correct degree and the sequence would therefore be convergent (actually constant, once it reaches the correct degree). Let us consider now the case when $\max\{\text{tdeg}(s_i) \mid i = 1, \ldots, I\}$ is $2d$ (see Definition 2.1.7 for $\text{tdeg}(s_i)$). Then, the maximum degree for the t_j polynomials is $2(d-1)$. We want to rewrite the formula (4.19) in terms of positive semi-definite matrices as in the unconstrained case. Let z be the vector containing all the monomials of degree less than or equal to d. We consider the generic matrices $N, L, L_1, \ldots, L_n$ such that

$$\sum_{i=1}^{I} s_i^2(x) = z^T N(\alpha, \lambda) z, \qquad \sum_{j=1}^{J} t_j^2(x) = z^T L(\alpha, \lambda) z$$

$$\sum_{j=1}^{J} t_j^2(x) x_k^2 = z^T L_k(\alpha, \lambda) z, \quad k = 1, \ldots, n.$$

Actually, by considering the relation between the sums above, we can actually describe the matrices $L_k,\ k = 1, \ldots, n$ using only elements of the matrix L, by doing some permutations of the zero rows and columns of the (free coefficient) matrix of L. This observation may be useful in order to save computational time. However, we do not pursue here this idea any further.

We are able now to formulate an LMI relaxation of (4.16)

$$\begin{aligned} \sup\ & \alpha \\ \text{s.t.}\ & N(\alpha, \lambda) + L(\alpha, \lambda) - \textstyle\sum_{k=1}^{n} L_k(\alpha, \lambda) \succeq 0. \end{aligned}$$

which can be subsequently solved using standard algorithms.

4.2.2 Numerical methods for the general case

The method presented here is a very basic extension of the method presented in [42] for solving the constraint polynomial optimization problem

$$\begin{array}{ll} \inf & p(x) \\ \text{s.t.} & r_i(x) \geq 0, \quad i = 1, \ldots, l, \qquad p,\ r_i\ \in \mathbf{R}[x],\ \forall i = 1, \ldots, l. \end{array}$$

Obviously, the problem (4.15) can be easily recast into the above formulation by introducing a new variable, say x_{n+1}, and a new equality (or two inequalities) constraint in the following way

$$\begin{array}{ll} \inf & x_{n+1}p(x) \\ \text{s.t.} & r_i(x) \geq 0, \quad i = 1, \ldots, l. \\ & x_{n+1}q(x) = 1 \end{array}$$

It seems that this trick would solve, at least numerically, the general problem of rational optimization with constraints. However, we have encountered examples where this is not the case. It may be that the approach relies a bit too strongly on the constraints, preferring (and performing well in) the situations where the feasible domain is actually a bounded set.

Example 4.2.7 (Example 4.1.12 revisited)

$$\inf_{(x_1,x_2,x_3)\in\mathbf{R}^3} \frac{(x_1+x_2)^4 + x_1^3x_3}{x_1^4 + x_3^4} = \begin{array}{l} \inf x_4[(x_1+x_2)^4 + x_1^3x_3] \\ s.t.\ \ x_4[x_1^4 + x_3^4] = 1 \end{array}$$

This is an example of an unconstrained rational optimization problem and the (implementation [32] of the) algorithm does not perform well on it. The user is required to enforce a so-called feasibility radius. That changes the problem into a constrained optimization problem, with the same criterion function, and bounded feasible domain. However, for this particular example, choosing a large radius bound does not seem to help while choosing a small one changes the problem radically (one cannot estimate how well the infimum of the modified problem approximates the infimum of the original problem). Without imposing a feasibility radius, the algorithm reports that that problem may be infeasible, i.e. the infimum of the original problem is $-\infty$ which is of course not correct. Note that the method of the previous section found the correct value without any problems.

4.3 Conclusions

In this chapter we extend algorithms, designed for global polynomial minimization, to the larger class of rational functions. The extensions are based on results in real algebraic geometry, which allow us to rewrite a rational optimization problem over $\mathbf{R}^n$ (or $\mathbf{B} \subset \mathbf{R}^n$) as a constrained polynomial optimization problem of a particular type. We develop here both exact and numerical methods for unconstrained and (a particular case of) constrained optimization.

However, as the Example 4.2.7 above intends to illustrate, the algorithms developed in this section *do not* perform equally well on every instance. Hence, there may be no *best* method, even for optimization in the special class of rational function. Example 4.1.12 supports in this sense our effort for describing various approaches.

Chapter 5

The H_2 model reduction problem

Let us introduce first the general model reduction problem. Suppose that a time-invariant, continuous-time or discrete-time linear system of order n is given. In practical applications, very often the order n is very high, making it very difficult to operate with the model. In this situation, one wants to work with an approximant of the original system which has a lower order. The model reduction problem consists in finding an approximant of low order which is close in some sense to the original. There are different ways to estimate how close two systems are. Typically, one uses the distance induced by some well-known norms, say H_2, H_∞ or the Hankel norm (see, for example [72]).

One of the most used model reduction methods is based on truncation of a balanced realization of the original system. This method has been shown to be closely related to the Hankel norm approximation problem. For the H_2 and H_∞ norms, no analytic solution is known. The H_∞ norm is extremely popular in robust control theory. However, we only discuss here the H_2 model reduction problem. The literature on the H_2 problem is quite broad. See for example [71], [2], [61], [4], [26], [17], [35] and the references contained therein.

We approach the H_2 model reduction here in a different manner, as a rational optimization problem, and show how the algorithms of the previous chapters can be applied. Sections 5.1 and 5.2 review certain aspects related to the H_2 norm and H_2 model reduction problem. There, it is also shown that the problem reduces to optimization of rational functions. Section 5.3 discusses H_2 optimal model reduction for continuous-time linear SISO (single-input-single-output) systems and is based on [39], while Section 5.4 discusses the same problem for MIMO (multi-input-multi-output) systems. Section 5.5 is based on [53]. The method of Section 5.5 was proposed by the first two authors, while the author of the thesis has contributed with results in Section 5.5.2 and the computations in the example section.

5.1 Computing the H_2 norm

There are different formulas available in the literature for computing the H_2 norm of a linear system (see [24], [72]) both for continuous-time systems and discrete-time systems. Let us treat these cases separately. Note that we assume here stability of the system. We restrict ourselves to the case of real systems although the formulas are similar for the general case of complex systems.

5.1.1 Continuous-time systems

Using the (matrix) transfer function T of a system $\tilde{\Sigma}$, one defines:

$$\|\tilde{\Sigma}\|_2^2 = \frac{1}{2\pi}\text{trace}(\int_{-\infty}^{\infty}(T(iy)-\tilde{D})^*(T(iy)-\tilde{D})dy + \text{trace}(\tilde{D}^T\tilde{D}).$$

Using a state space representation $(\tilde{A},\tilde{B},\tilde{C},\tilde{D})$ of $\tilde{\Sigma}$

$$\|\tilde{\Sigma}\|_2^2 = \text{trace}(\tilde{D}^T\tilde{D}) + \text{trace}(\tilde{B}^T M_o\tilde{B}) = \text{trace}(\tilde{D}^T\tilde{D}) + \text{trace}(\tilde{C}M_c\tilde{C}^T),$$

with

$$\tilde{A}M_c + M_c\tilde{A}^T = -\tilde{B}\tilde{B}^T,\ \tilde{A}^T M_o + M_o\tilde{A} = -\tilde{C}^T\tilde{C}.$$

A proof of the equivalence between the two formulations can be found in [72].

5.1.2 Discrete-time systems

We have

$$\|\tilde{\Sigma}\|_2^2 = \frac{1}{2\pi}\text{trace}(\int_0^{2\pi} T^*(e^{iy})T(e^{iy})dy).$$

and an equivalent formulation is

$$\|\tilde{\Sigma}\|_2^2 = \text{trace}(\tilde{D}^T\tilde{D}) + \text{trace}(\tilde{B}^T L_o\tilde{B}) = \text{trace}(\tilde{D}^T\tilde{D}) + \text{trace}(\tilde{C}L_c\tilde{C}^T),$$

with

$$L_o - \tilde{A}^T L_o\tilde{A} = \tilde{C}^T\tilde{C},\quad L_c - \tilde{A}L_c\tilde{A}^T = \tilde{B}\tilde{B}^T.$$

In this thesis we work with the formulas for the norm involving state space representations of the system rather than with their (matrix) transfer functions. The advantage for us is that the H_2 norm can be expressed as a rational function in the entries of $\tilde{A},\tilde{B},\tilde{C},\tilde{D}$.

5.2 The H_2 distance and the optimization problem

Let us come back now to the H_2 model reduction problem. We denote by $\Sigma = (A,B,C,D)$ a time-invariant, continuous-time or discrete-time, linear, stable system of order n. By stable, we mean that all eigenvalues of A are in the open left-half plane in the continuous-time case, respectively in the open unit circle in the discrete-time case. The H_2 model reduction problem is finding the closest (in H_2 distance) system $\hat{\Sigma} = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ time-invariant, continuous-time respectively discrete-time, linear, stable system of *given* order $\hat{n}$. Formulated differently, we want to solve:

$$\min_{\hat{\Sigma}-\text{stable, order}(\hat{\Sigma})\leq\hat{n}} \|\Sigma - \hat{\Sigma}\|_2^2.$$

It is well known that the minimum of the above optimization problem is *attained.* Moreover, any best approximant of order at most $\hat{n}$ has exactly order $\hat{n}$. For discrete-time systems the proof can be found in [2]. For continuous-time systems, the same holds due to the existing bijective isometry which maps the transfer function of a discrete-time system of order n to the transfer function of a continuous-time system of order n, and vice versa (see, e.g., [23], [3]).

In order to compute the H_2 distance it is sufficient to apply the formulas of Section 5.1 for the H_2 norm to the difference system $\tilde{\Sigma} = \Sigma - \hat{\Sigma}$ defined by

$$\tilde{A} = \begin{pmatrix} A & 0 \\ 0 & \hat{A} \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} B \\ \hat{B} \end{pmatrix}, \quad \tilde{C} = \begin{pmatrix} C & -\hat{C} \end{pmatrix}, \quad \tilde{D} = D - \hat{D}.$$

Then we obtain the following formulas.

5.2.1 Continuous-time systems

$$\begin{aligned}\|\Sigma - \hat{\Sigma}\|_2^2 &= \text{trace}((D-\hat{D})^T(D-\hat{D})) + \text{trace}(B^T M_1^o B + 2B^T M_2^o \hat{B} + \hat{B}^T M_3^o \hat{B}) \\ &= \text{trace}((D-\hat{D})^T(D-\hat{D})) + \text{trace}(C M_1^c C^T - 2C M_2^c \hat{C}^T + \hat{C} M_3^c \hat{C}^T)\end{aligned}$$

with

$$\begin{aligned}&A^T M_1^o + M_1^o A = -C^T C, \; A^T M_2^o + M_2^o \hat{A} = C^T \hat{C}, \; \hat{A}^T M_3^o + M_3^o \hat{A} = -\hat{C}^T \hat{C}, \\ &A M_1^c + M_1^c A^T = -BB^T, \; A M_2^c + M_2^c \hat{A}^T = -B\hat{B}^T, \; \hat{A} M_3^c + M_3^c \hat{A}^T = -\hat{B}\hat{B}^T.\end{aligned}$$

5.2.2 Discrete-time systems

$$\begin{aligned}\|\Sigma - \hat{\Sigma}\|_2^2 &= \text{trace}((D-\hat{D})^T(D-\hat{D})) + \text{trace}(B^T L_1^o B + 2B^T L_2^o \hat{B} + \hat{B}^T L_3^o \hat{B}) \\ &= \text{trace}((D-\hat{D})^T(D-\hat{D})) + \text{trace}(C L_1^c C^T - 2C L_2^c \hat{C}^T + \hat{C} L_3^c \hat{C}^T)\end{aligned}$$

with

$$\begin{aligned}&L_1^o - A^T L_1^o A = C^T C, \; L_2^o - A^T L_2^o \hat{A} = -C^T \hat{C}, \; L_3^o - \hat{A}^T L_3^o \hat{A} = \hat{C}^T \hat{C} \\ &L_1^c - A L_1^c A^T = BB^T, \; L_2^c - A L_2^c \hat{A}^T = B\hat{B}^T, \; L_3^c - \hat{A} L_3^c \hat{A}^T = \hat{B}\hat{B}^T.\end{aligned}$$

Obviously, in both cases the criterion is minimized for $D = \hat{D}$ and trace($(D - \hat{D})^T(D - \hat{D})$) becomes 0.

Note that, in order to compute the H_2 distance between the systems Σ and

$\hat{\Sigma}$, one needs to solve the Lyapunov/Sylvester equations in M_1^o, M_2^o, M_3^o or M_1^c, M_2^c, M_3^c (respectively L_1^o, L_2^o, L_3^o or L_1^c, L_2^c, L_3^c). Clearly the H_2 distance is invariant with respect to similarity transformations of the triples (A, B, C) or $(\hat{A}, \hat{B}, \hat{C})$. Therefore, when choosing a parameterization, one might consider simplifying the computations of the solutions of the Lyapunov/Sylvester equations. Indeed, some canonical forms of (A, B, C) are more advantageous for this problem, as we will see later. It should be remarked however that M_1^o, M_2^o, M_3^o and M_1^c, M_2^c, M_3^c (respectively L_1^o, L_2^o, L_3^o and L_1^c, L_2^c, L_3^c) will be multivariate rational functions, depending on the entries of $(\hat{A}, \hat{B}, \hat{C})$. Therefore the criterion to be minimized will be a multivariate rational function as well.

Notice also the particular structure of the problem. The role of B, $\hat{B}$ and C, $\hat{C}$ respectively are somehow symmetric. Moreover, the criterion function is quadratic in either $\hat{B}$ or $\hat{C}$. This leads to an optimization in two steps. By first optimizing with respect to $\hat{B}$, using derivatives, one obtains a criterion which is rational in the entries of $\hat{A}$, $\hat{C}$. Let us give also the exact formulas here. The optimal $\hat{B}$ is

$$\hat{B} = -M_3^{o^{-1}} M_2^{o^T} B, \quad \text{respectively} \quad \hat{B} = -L_3^{o^{-1}} L_2^{o^T} B$$

for continuous-time systems, respectively for discrete-time systems. The existence of the inverse matrix $M_3^{o^{-1}}$, respectively $L_3^{o^{-1}}$ follows from the fact that $(\hat{A}, \hat{C})$ is an observable pair. Then, the squared H_2 distance between the two systems (for $\hat{D} = D$) becomes

$$\text{trace}(B^T M_1^o B) - \text{trace}(B^T M_2^o M_3^{o^{-1}} M_2^{o^T} B),$$
$$\text{respectively} \quad \text{trace}(B^T L_1^o B) - \text{trace}(B^T L_2^o L_3^{o^{-1}} L_2^{o^T} B),$$

for continuous-time systems, respectively for discrete-time systems. Remark that $\text{trace}(B^T M_1^o B)$ and $\text{trace}(B^T L_1^o B)$ depend only on the initial system, hence they are known constants. If instead of $\hat{B}$ we choose to optimize with respect to $\hat{C}$, we have

$$\hat{C} = C M_2^c M_3^{c^{-1}}, \quad \text{respectively} \quad \hat{C} = C L_2^c L_3^{c^{-1}}$$

for continuous-time systems, respectively for discrete-time systems. The existence of the inverse matrix $M_3^{c^{-1}}$, respectively $L_3^{c^{-1}}$ follows from the fact that $(\hat{A}, \hat{B})$ is an controllable pair. Then, the squared H_2 distance between the two systems (for $\hat{D} = D$) becomes

$$\text{trace}(C M_1^c C^T) - \text{trace}(C M_2^c M_3^{c^{-1}} M_2^{c^T} C^T),$$
$$\text{respectively} \quad \text{trace}(C L_1^c C^T) - \text{trace}(C L_2^c L_3^{c^{-1}} L_2^{c^T} C^T),$$

for continuous-time systems, respectively for discrete-time systems. Remark that, as before, $\text{trace}(B^T M_1^c B)$ and $\text{trace}(B^T L_1^c B)$ depend only on the initial system, hence they are known constants.

We are therefore left with a nonlinear optimization problem and here we employ the algorithms of Chapter 4. Let us now make a few remarks concerning the complexity of the problem, which is related to the *complexity* of the rational function, and is expressed in terms of its total degree and number of variables. Suppose the original system has n states m inputs and p outputs and assume that $m \geq p$. Then an approximant of order $\hat{n}$ will have m inputs and p outputs as well. The parameterization of the approximant in any canonical form requires $(m+p)\hat{n}$ parameters (the number of parameters equals the dimension of the manifold). Let us consider now that $(\hat{A}, \hat{B}, \hat{C})$ is in a canonical form which leaves either $\hat{B}$ or $\hat{C}$ free. For example, in the observable canonical form the matrix $\hat{B}$ is free, with the only condition that the pair $(\hat{A}, \hat{B})$ is controllable. Similarly, the controller canonical form provides a free matrix $\hat{C}$, with the only condition that the pair $(\hat{A}, \hat{C})$ is observable. Then, as we noted before, it is possible to optimize first with respect to either $\hat{B}$ or $\hat{C}$ reducing in this way the number of free parameters. Since we have assumed $m \geq p$ we optimize with respect to B since this would reduce mostly the number of remaining parameters, in fact it would reduce it to $\hat{n}p$. Notice also that at the optimal value of $\hat{B}$, the pair $(\hat{A}, \hat{B})$ is indeed controllable, due to the minimality of the optimal approximant. Similarly, for the optimal value of $\hat{C}$, the pair $(\hat{A}, \hat{C})$ is observable. So far, we have seen that by optimizing with respect to $\hat{B}$ or $\hat{C}$, we can reduce quite a lot the number of variables in the optimization criterion.

Let us assume now that $(\hat{A}, \hat{C})$ is in an output normal form. That means $M_3^o = I_{\hat{n}}$, respectively $L_3^o = I_{\hat{n}}$, which simplifies the criterion function even further. As one can see from the expression of the criterion, the total degree of the denominator of rational function, which is larger than the total degree of the numerator, is (at most) $(n\hat{n})^2$ (in the entries of $\hat{A}$ and $\hat{C}$).

To resume, the number of parameters required is completely determined by the approximant system, namely, the number of parameters equals $\hat{n}\min\{m,p\}$. However, the complexity of the rational function is also determined by its degree, namely, the degree is at most $(n\hat{n})^2$ and it is influenced by the order of the original system. In practical application, large systems are reduced to systems of small or very small order, therefore the number of variables in the expression of the rational function is in general rather small.

In the following we are going to discuss in full detail a few examples of model reduction in the SISO, respectively MIMO continuous-time case where we apply the numerical algorithm of Section 4.1.3. A MIMO discrete-time example will be treated using exact methods.

5.3 H_2 model reduction: continuous-time SISO case

We treat here a particular case of the H_2 optimal model reduction, namely reduction of SISO continuous-time systems. As we have mentioned before, we still have at this point the choice for a parameterization of (A, b, c). In the following we choose the parameterization trying to satisfy two criteria. The first one is the stability requirement, therefore we use canonical forms for *stable* systems. Secondly we want to simplify our calculations (i.e. solving the Lyapunov/Sylvester equations) as much as possible. One way is to choose a so-called output normal form for (A, c) (respectively $(\hat{A}, \hat{c})$), that is equivalent to saying that M_1, the solution of the Lyapunov equation associated to (A, c), satisfies $M_1 = I_n$ (respectively $M_3 = I_{\hat{n}}$). Both mentioned requirements are satisfied by the so called Schwarz-like canonical forms (see [26]). Notice also that in a Schwarz-like canonical form the b vector is free with the sole condition that (A, b) is a reachable pair. Therefore optimization with respect to b is possible in a first step.

Let us describe now the Schwarz-like canonical form for SISO *stable* systems. A triple $(\hat{A}, \hat{b}, \hat{c})$ is in a Schwarz-like canonical form when

$$\hat{A} = \begin{pmatrix} -\frac{1}{2}x_1^2 & -x_2 & 0 & \ddots & 0 & 0 \\ x_2 & 0 & -x_3 & \ddots & 0 & 0 \\ 0 & x_3 & 0 & \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \ddots & \ddots & -x_{\hat{n}} \\ 0 & 0 & 0 & \ddots & x_{\hat{n}} & 0 \end{pmatrix}, \quad \hat{c} = \begin{pmatrix} x_1 & 0 & 0 & \dots & 0 \end{pmatrix} \tag{5.1}$$

with $x_i > 0$, $i = 1, \dots, \hat{n}$, and $\hat{b} \in \mathbf{R}^{\hat{n} \times 1}$ is a free vector, with the condition that $(\hat{A}, \hat{b})$ is a reachable pair. Notice that the conditions $x_i > 0$, $\forall i = 1, \dots, \hat{n}$ imply that the pair $(\hat{A}, \hat{c})$ is observable. The proof for the fact that this is indeed a canonical form can be found in [28]. It is easy to see that $(\hat{A}, \hat{c})$ is output-normal, i.e. $\hat{A}^T + \hat{A} = -\hat{c}^T\hat{c}$. Since $\hat{A}$ is stable, the equation $\hat{A}^T M_3 + M_3\hat{A} = -\hat{c}^T\hat{c}$ has a unique solution, namely $M_3 = I_{\hat{n}}$.

Let us return to the H_2 model reduction problem and consider (A, b, c), $(\hat{A}, \hat{b}, \hat{c})$, with A and $\hat{A}$ stable, in Schwarz-like canonical form. Then using the fact that (A, c) and $(\hat{A}, \hat{c})$ are output-normal, the equations $A^T M_1 + M_1 A = -c^T c$, $\hat{A}^T M_3 + M_3\hat{A} = -\hat{c}^T\hat{c}$ have each a unique solution, namely $M_1 = I_n$, $M_3 = I_{\hat{n}}$.

The criterion is quadratic in $\hat{b}$. By optimizing first with respect to $\hat{b}$ one obtains $\hat{b} = -M_2^T b$ and $\|\Sigma - \hat{\Sigma}\|_2^2 = b^T b - b^T M_2 M_2^T b$, where M_2 is the solution of the Sylvester equation $A^T M_2 + M_2\hat{A} = c^T\hat{c}$. The optimization problem becomes

$$\min_{x_i > 0, i=1\dots,\hat{n}} b^T b - b^T M_2 M_2^T b = b^T b - \max_{x_i > 0, i=1\dots,\hat{n}} \frac{p(x_1, \dots, x_{\hat{n}})}{q(x_1, \dots, x_{\hat{n}})}. \tag{5.2}$$

Schwarz-like canonical forms have the following property.

Proposition 5.3.1 *If the approximant* $(\hat{A}, \hat{b}, \hat{c})$ *is in Schwarz-like canonical* *form, then the criterion* $p(x)/q(x) = b^T M_2 M_2^T b$ *contains only even powers of* $x_1, \ldots, x_{\hat{n}}$.

Proof It is well-known that the H_2 distance between the two systems, and therefore our criterion $p(x)/q(x)$, does not depend on a particularly chosen $(\hat{A}, \hat{b}, \hat{c})$ representation of the approximant system. It is not difficult to check that in the following two cases:

i) If for a certain $i = 2, \ldots, \hat{n}$ one replaces x_i by $-x_i$ in a given Schwarz-like canonical representation $(\hat{A}, \hat{b}, \hat{c})$,

ii) If one replaces x_1 by $-x_1$ and simultaneously $\hat{b}$ by $-\hat{b}$ in a given Schwarz-like canonical representation $(\hat{A}, \hat{b}, \hat{c})$,

then one obtains an *equivalent* Schwarz-like representation. That implies

$$\frac{p(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{\hat{n}})}{q(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{\hat{n}})} = \frac{p(x_1, \ldots, x_{i-1}, -x_i, x_{i+1}, \ldots, x_{\hat{n}})}{q(x_1, \ldots, x_{i-1}, -x_i, x_{i+1}, \ldots, x_{\hat{n}})}$$

and therefore the criterion p/q contains only even powers of $x_i,\ i = 1, \ldots, \hat{n}$. □

Notice also that the global optimum of $p(x)/q(x)$ can not be attained at a point where $x_i = 0$ for some $i = 1, \ldots, \hat{n}$. That holds because if $x_i = 0$ for some $i = 1, \ldots, \hat{n}$, then the approximant looses its minimality, which contradicts the (well-known) result (see, e.g. [3]) which states that the optimal approximant of order at most $\hat{n}$ *has* order $\hat{n}$. In conclusion, one only needs to solve an unconstrained rational optimization problem. In this case, when minimizing over $(x_1, \ldots, x_{\hat{n}}) \in \mathbf{R}^{\hat{n}}$ we obtain *symmetric* solutions with respect to the axes, i.e. for any solution $(x_1, \ldots, x_{\hat{n}})$, we know that $(\pm|x_1|, \ldots, \ \pm|x_{\hat{n}}|)$ are solutions as well.

For the convenience of the reader, we structure the above discussion into a procedure.

Procedure 5.3.2 *The following procedure computes the optimal approximant of a given order with respect to the* H_2*-norm.*

1. *Compute a state-space representation* (A, b, c) *of the initial system* Σ *(for example, in Schwarz-like canonical form (5.1)).*

2. *Construct a parameterized representation* $(\hat{A}, \hat{b}, \hat{c})$ *of the approximant* $\hat{\Sigma}$ *in a Schwarz-like canonical form.*

3. *Compute symbolically the solution of the Sylvester equation* $A^T M_2 + M_2 \hat{A} = c^T \hat{c}$.

4. *Compute analytically the optimal value for* $\hat{b} = -M_2^T b$.

5. *Compute the global maximum of* $b^T M_2 M_2^T b$ *over* $\mathbf{R}^{\hat{n}}$*, using either one of the algorithms of Chapter* 4 *for optimization of rational functions.*

6. Evaluate the approximant $(\hat{A}, \hat{b}, \hat{c})$ at the global optimum.

Note that at step 5, the user is allowed the choice of an algorithm. In Chapter 4 we have proposed both a numerical algorithm (Section 4.1.3) and an exact algorithm (Section 4.1.2) for the unconstrained optimization of a rational function. In general, we expect the numerical algorithm to be more efficient. Note however that the numerical algorithm is only guaranteed to find a lower bound of the global infimum while by using the exact algorithm with the above procedure, we obtain an *exact* solution to the H_2 optimal model reduction problem, provided we start with an exact representation of (A, b, c). It is important to note that the H_2 model order reduction *can* be solved exactly in the SISO case, using Procedure 5.3.2 with an exact algorithm at step 5.

We consider now a concrete example, namely the *Oscillatory system* of [26]. There, an optimal approximant was constructed using constructive algebra methods. In the end we intend to compare the results.

Example 5.3.3 *Find the* best *H_2-approximant of second order ($\hat{n} = 2$), of the system*

$$T(s) = \frac{s^2 - s + 2}{s^3 + 0.5s^2 + 2s + 0.5} \qquad (n = 3).$$

T corresponds to a Schwarz-like canonical form with parameters $x_1 = x_2 = x_3 = b_1 = b_2 = b_3 = 1,\ d = 0$.

Let us consider a general second order, stable system in Schwarz-like canonical form

$$\hat{A} = \begin{pmatrix} -\frac{1}{2}x_1^2 & -x_2 \\ x_2 & 0 \end{pmatrix}, \quad \hat{c} = \begin{pmatrix} x_1 & 0 \end{pmatrix}, \quad x_1 > 0,\ x_2 > 0$$

We want to find the values of the parameters $x_1,\ x_2$ for which the criterion $b^T b - b^T M_2 M_2^T b$ is minimized. Since b is given, let us now compute the matrix $M_2 \in \mathbf{R}^{3\times 2}$ (as function of $x_1,\ x_2$) from the *linear* system of equations $A^T M_2 + M_2 \hat{A} = c^T \hat{c}$. We obtain

$$M_2 = \frac{2}{\Delta} \begin{pmatrix} x_1(-4 - x_1^4 + 8x_2^2 - 4x_1^2x_2^2 - 4x_2^4) & 2x_1x_2(8 + x_1^4 - 12x_2^2 + 4x_2^4) \\ 2x_1(x_1^2 + 8x_2^2 - 4x_2^4) & 4x_1x_2(1 - x_2^2 - x_1^2x_2^2) \\ -4x_1(1 - x_2^2 - x_1^2x_2^2) & 2x_1x_2(8 + x_1^2 + x_1^4 - 4x_2^2) \end{pmatrix}$$

with $\Delta = 4 + 8x_1^2 + x_1^4 + x_1^6 + 56x_2^2 - 4x_1^2x_2^2 + 8x_1^4x_2^2 - 60x_2^4 + 4x_1^2x_2^4 + 16x_2^6$.

The criterion will be

$$b^T b + \min -b^T M_2 M_2^T b = 3 + \min -\frac{p(x_1, x_2)}{q(x_1, x_2)}$$

where
$p(x_1, x_2) = 4x_1^2(64 - 32x_1^2 + 20x_1^4 - 4x_1^6 + x_1^8 + 848x_2^2 + 256x_1^2x_2^2 + 236x_1^4x_2^2 + 16x_1^6x_2^2 + 16x_1^8x_2^2 - 1616x_2^4 - 480x_1^2x_2^4 - 280x_1^4x_2^4 - 32x_1^6x_2^4 + 1200x_2^6 + 320x_1^2x_2^6 + 80x_1^4x_2^6 - 432x_2^8 - 64x_1^2x_2^8 + 64x_2^{10})$

and
$q(x_1, x_2) = (4 + 8x_1^2 + x_1^4 + x_1^6 + 56x_2^2 - 4x_1^2x_2^2 + 8x_1^4x_2^2 - 60x_2^4 + 4x_1^2x_2^4 + 16x_2^6)^2$.
We apply now the procedure described in Section 4.1.3. Note that the denominator of the rational function is the square of a polynomial, hence it is non-negative on $\mathbf{R}^2$. For solving the problem (4.5), we construct the LMI relaxation (4.12), using the vector of monomials z. In order to reduce the size of our problem, and since the polynomials p, q contain only even powers of the variables, we consider only monomials of even power in the vector z as well. We have $\text{tdeg}(p(x) - \alpha q(x)) = 12$, therefore the vector z will contain monomials of degree less or equal its half, that is $m = 6$ and $z^T = \begin{pmatrix} 1 & x_1^2 & x_1^4 & x_1^6 & x_2^2 & x_2^4 & x_2^6 & x_1^2x_2^2 & x_1^4x_2^2 & x_1^2x_2^4 \end{pmatrix}$. In this case, considering this vector z turns out to be sufficient for finding the global minimum. In general however, restricting the number of monomials in z may lead to a strict lower bound of the global minimum.
Let us now construct an arbitrary, symmetric matrix $Q \in \mathbf{R}[\alpha, \lambda]^{10\times10}$. Its dimension is obviously determined by the length of z. We compute $Q(\alpha, \lambda)$ by equalizing the coefficients of the polynomials $-p(x) - \alpha q(x)$ and z^TQz, as in Section 3.1.1. It turns out from the computations that $\lambda \in \mathbf{R}^{28}$. The only thing left now is to compute the solution for the LMI relaxation. We obtain, as a lower bound of $\min -p(x_1, x_2)/q(x_1, x_2)$ the value -1.7642. At this point we still need to decide whether this is a strict lower bound or not. In this case we have run a standard steepest descent algorithm which finds a (local) minimum at $(x_1, x_2) = (1.1916, 0.4183)$ for which the (numerically) computed value of the criterion equals the (numerically) computed value of the lower bound! This tells us two things, first that the lower bound was sharp, secondly that the point $(1.1916, 0.4183)$ is actually a global minimum. Hence, we have found a best approximant in H_2 norm and this is given by

$$\hat{A} = \begin{pmatrix} -0.7099 & -0.4183 \\ 0.4183 & 0 \end{pmatrix}, \quad \hat{c} = \begin{pmatrix} 1.1916 & 0 \end{pmatrix}, \quad \hat{b} = \begin{pmatrix} 0.2080 \\ -1.3118 \end{pmatrix}.$$

The approximant coincides with the one found in [26] by exact methods. The optimal H_2 distance between the given system and its approximant is obtained by taking the square root of $3 - 1.7642$, that is 1.1117. Under certain conditions, other (more direct) methods than the one presented here can be used to decide whether the obtained lower bound is exact or not. For more details see [50] or Section 3.1.1 of this thesis.

For obtaining the lower bound we have run an algorithm which consists of two parts. The first one, for constructing the LMI relaxation was implemented in Mathematica 4.0 and takes 12 seconds and 16.7 Kb on a Sun Ultra 5 station. Then, for solving the LMI problem we use SeDuMi 1.03, a free software package (see [66]) running under Matlab. This takes another 5 seconds (of which 2 are used to read the data obtained with Mathematica). Unfortunately, we have experienced numerical problems with SeDuMi 1.03. This problems seem to be, at least partly, solved in the next version, SeDuMi 1.05.

The method is not restricted to model order reduction by 1. To make this clear, we consider here reduction from a 6-th order system to a 2-nd order system. The following example was considered in [61] (Example 3).

Example 5.3.4 *Given the stable continuous-time system with the transfer function*

$$T(s) = \frac{0.00001s^2 + 0.011s + 1}{s^6 + 0.222s^5 + 22.1242s^4 + 3.5445s^3 + 122.4433s^2 + 11.3231s + 11.11}$$

find its H_2 optimal approximant of order 2.

We apply the procedure and construct a realization of the original system in the Schwarz-like canonical form (see (5.1)). We obtain the values of the parameters of the 6-th order system, in (5.1) $(0.666, 2.4815, 2.0893, 0.5745, 2.4091, 2.3382)$. The approximant has also a Schwarz-like canonical form with the unknowns (x_1, x_2), $x_1 > 0$, $x_2 > 0$.

Next we compute the H_2 distance between the original system and its approximant, as a rational function in x_1, x_2. More precisely, in the formula (5.2) we have

$$\begin{aligned}
p(x_1, x_2) = {} & 123.432{x_2}^2 - 2590.99{x_1}^2{x_2}^2 + 15372.4{x_1}^4{x_2}^2 - 5236.47{x_1}^6{x_2}^2 + \\
& 667.289{x_1}^8{x_2}^2 - 35.347{x_1}^{10}{x_2}^2 + 0.466075{x_1}^{12}{x_2}^2 + \\
& 0.01208{x_1}^{14}{x_2}^2 + 1. \times 10^{-10}{x_1}^{16}{x_2}^2 + 13.5775{x_2}^4 + 1049.3{x_1}^2{x_2}^4 + \\
& 1206.16{x_1}^4{x_2}^4 - 510.89{x_1}^6{x_2}^4 + 73.3059{x_1}^8{x_2}^4 - 4.62864{x_1}^{10}{x_2}^4 + \\
& 0.109996{x_1}^{12}{x_2}^4 + 2.22 \times 10^{-11}{x_1}^{14}{x_2}^4 + 0.373999{x_2}^6 + \\
& 3640.91{x_1}^2{x_2}^6 - 1299.41{x_1}^4{x_2}^6 + 247.668{x_1}^6{x_2}^6 - 24.3873{x_1}^8{x_2}^6 + \\
& 0.96937{x_1}^{10}{x_2}^6 - 0.00605951{x_1}^{12}{x_2}^6 + 5. \times 10^{-11}{x_1}^{14}{x_2}^6 + \\
& 0.0000339438{x_2}^8 + 156.993{x_1}^2{x_2}^8 + 87.0313{x_1}^4{x_2}^8 - 38.4692{x_1}^6{x_2}^8 + \\
& 4.48862{x_1}^8{x_2}^8 - 0.162334{x_1}^{10}{x_2}^8 + 1.09999 \times 10^{-6}{x_1}^{12}{x_2}^8 + \\
& 7.71451 \times 10^{-10}{x_2}^{10} + 338.262{x_1}^2{x_2}^{10} - 135.118{x_1}^4{x_2}^{10} + \\
& 19.2398{x_1}^6{x_2}^{10} - 0.870429{x_1}^8{x_2}^{10} + 0.00530232{x_1}^{10}{x_2}^{10} - \\
& 1.875 \times 10^{-11}{x_1}^{12}{x_2}^{10} + 7.15503{x_1}^2{x_2}^{12} + 2.94556{x_1}^4{x_2}^{12} - \\
& 1.05916{x_1}^6{x_2}^{12} + 0.0749554{x_1}^8{x_2}^{12} - 2.74999 \times 10^{-7}{x_1}^{10}{x_2}^{12} + \\
& 11.4991{x_1}^2{x_2}^{14} - 3.40262{x_1}^4{x_2}^{14} + 0.250758{x_1}^6{x_2}^{14} - \\
& 0.00094686{x_1}^8{x_2}^{14} + 1.5625 \times 10^{-12}{x_1}^{10}{x_2}^{14} + 0.132126{x_1}^2{x_2}^{16} + \\
& 0.0553241{x_1}^4{x_2}^{16} - 0.0102705{x_1}^6{x_2}^{16} + 1.71875 \times 10^{-8}{x_1}^8{x_2}^{16} + \\
& 0.173038{x_1}^2{x_2}^{18} - 0.0269622{x_1}^4{x_2}^{18} + 0.0000473438{x_1}^6{x_2}^{18} + \\
& 0.000867188{x_1}^2{x_2}^{20} + 0.000429688{x_1}^4{x_2}^{20} + 0.000976563{x_1}^2{x_2}^{22}
\end{aligned}$$

and $q(x_1, x_2)$ is the square of the polynomial

$$\begin{aligned}123.432 - 2592.48{x_1}^2 + 15403.7{x_1}^4 - 5422.55{x_1}^6 +\\ 732.793{x_1}^8 - 44.1991{x_1}^{10} + 1.{x_1}^{12} + 62.8998{x_2}^2 + 634.15{x_1}^2{x_2}^2 -\\ 152.606{x_1}^4{x_2}^2 + 26.7437{x_1}^6{x_2}^2 - 2.86096{x_1}^8{x_2}^2 +\\ 0.111{x_1}^{10}{x_2}^2 + 340.086{x_2}^4 - 235.766{x_1}^2{x_2}^4 + 699.724{x_1}^4{x_2}^4 -\\ 122.247{x_1}^6{x_2}^4 + 5.53105{x_1}^8{x_2}^4 + 4.92242{x_2}^6 + 29.7728{x_1}^2{x_2}^6 -\\ 3.67914{x_1}^4{x_2}^6 + 0.443063{x_1}^6{x_2}^6 + 15.3625{x_2}^8 - 4.00914{x_1}^2{x_2}^8 +\\ 7.65271{x_1}^4{x_2}^8 - 1.49639 \times 10^{-16}{x_1}^6{x_2}^8 + 0.0770756{x_2}^{10} +\\ 0.353847{x_1}^2{x_2}^{10} - 4.67621 \times 10^{-18}{x_1}^4{x_2}^{10} + 0.173594{x_2}^{12}\end{aligned}$$

The vector z has in this case length 36, therefore the matrix Q belongs to $\mathbf{R}[\alpha, \lambda]^{36\times 36}$. It turns out that the dimension of the affine space of the LMI relaxation is 547, that is $(\alpha, \lambda) \in \mathbf{R}^{547}$. By running an SDP algorithm we find a lower bound at 0.0042. By using a local search algorithm we also find a local minimum at $(x_1, x_2) = (0.4327, 0.3049)$ whose value equals the value of our lower bound, therefore we conclude it is actually the global minimum. Hence, the approximant we obtain is given by the transfer function

$$\hat{T} = \frac{-0.0085 - 0.0004s}{0.0930 + 0.0936s + s^2}.$$

The optimal H_2 distance between the given system of order 6 and its 2-nd order approximant is 0.0661.

Again we have used an algorithm which consists of two parts. The first part implemented in Mathematica 4.0 takes 105 seconds (plus 70 seconds to write the data into a file readable with Matlab) and 178 Kb. Then, for solving the LMI problem we use SeDuMi 1.05 (an update of SeDuMi 1.03) running under Matlab. This takes another 70 seconds (of which 13 are used to read the data obtained with Mathematica).

5.4 H_2 model reduction: continuous-time MIMO case

The MIMO continuous-time case can be treated in a similar manner. The main difference here is that the space of MIMO models of specified order $\hat{n}$ is a real analytic manifold (see [31]), but there is no continuous canonical form which covers it entirely. However, the manifold can be covered by a finite number of charts, where each such chart is *generic*, i.e. the chart covers the whole manifold except for a thin subset. Now, given a finite number of charts it is possible in principle to run an H_2 optimization algorithm on each chart and choose afterwards the global optimum. However, since the charts can be chosen to be generic, we claim that it is sufficient to run such an algorithm on a single chart.

That is true even if the minimum is not attained on the chart, but on its boundary. It may happen that for the particularly chosen chart the global optimum is situated on the boundary of the chart. Theoretically this is not a problem since we employ algorithms which compute the infimum of the criterion. However, for computing an optimal approximant, one might have to switch to a different chart. We do not go into details concerning this problem here.

Another question that one may raise in this situation is whether certain parameterizations, or even specific charts of a chosen parameterization, have advantages over other parameterizations, respectively charts, from the computational point of view. We do not give an answer here. Note that the choice of a parameterization is already restricted by the fact that the approximant system must be stable. We use the input-normal canonical form for stable MIMO systems introduced in [29]. The property of being input-normal proved to be very advantageous in the SISO case. As for the choice of a chart, we discuss this later, on a particular example.

Example 5.4.1 *We present here some preliminary results. The example is meant to exhibit the high complexity of the problem, even for initial systems of small order.*

We have applied the procedure to a MIMO model reduction problem. The original system is given by

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 & -2 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 2 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{5.3}$$

We want to approximate this by a lower order system $(\hat{A}, \hat{B}, \hat{C})$, of order 2. For this example we construct the set of all charts necessary for covering the manifold, corresponding to a input-normal canonical form. We follow closely [29] in order to parameterize the pair of matrices $(\hat{A}, \hat{B})$ with $\hat{A}, \hat{B} \in \mathbf{R}^{2\times 2}$. A set of overlapping parameterizations, which covers the entire manifold, can be constructed in the following way. Each nice selection determines in a unique manner a chart. In our case, we have exactly 3 such nice selections of indices in the reachability matrix $\left[\begin{array}{cc} \hat{B} & \hat{A}\hat{B} \end{array}\right]$, namely $(1,2)$, $(1,3)$ and $(2,4)$. Consequently we have exactly 3 charts, as described below

$$\hat{A} = \begin{pmatrix} -\frac{x_1^2+x_2^2}{2} & -x_2x_3 - x_4 \\ x_4 & -\frac{x_3^2}{2} \end{pmatrix}, \quad \hat{B} = \begin{pmatrix} x_1 & x_2 \\ 0 & x_3 \end{pmatrix}, \quad x_1, x_3 > 0, \tag{5.4}$$

or,

$$\hat{A} = \begin{pmatrix} -\frac{x_1^2+x_2^2}{2} & -x_2x_3 - x_4 \\ x_4 & -\frac{x_3^2}{2} \end{pmatrix}, \quad \hat{B} = \begin{pmatrix} x_1 & x_2 \\ 0 & x_3 \end{pmatrix}, \quad x_1, x_4 > 0, \tag{5.5}$$

or,

$$\hat{A} = \begin{pmatrix} -\frac{x_1^2+x_2^2}{2} & -x_2x_3 - x_4 \\ x_4 & -\frac{x_3^2}{2} \end{pmatrix}, \quad \hat{B} = \begin{pmatrix} x_2 & x_1 \\ x_3 & 0 \end{pmatrix}, \quad x_1, x_4 > 0. \tag{5.6}$$

We chose to work with the chart (5.4). In this case, as discussed in Section 5.2.1, the optimal $\hat{C}$ *can be determined as a function of* $(\hat{A}, \hat{B})$ *from the formula* $\hat{C} = CM_2$ *where* M_2 *is the solution of* $AM_2 + M_2\hat{A}^T = -B\hat{B}^T$. *We have computed symbolically the matrix* M_2*, solution of the above equation using Mathematica 4.0. We also compute the criterion to be maximized* $CM_2M_2^TC^T$. *Since its expression is very complicated, we do not reproduce it here. Let us mention though that the criterion is a rational function whose numerator and denominator are polynomials in* 4 *variables having degree 22 and 1091 terms, respectively degree 24 and 726 terms.*

Following the procedure in Chapter 4 we construct a polynomial $p(x) - \alpha q(x)$ *in* 4 *variables and* 1 *parameter, of degree* 24*, having* 1817 *terms.*

At first sight, we cannot detect any special structure in our problem, hence we go on and apply the algorithm directly by constructing the LMI relaxation. Since our vector z*, of monomials in* $x_1, \ldots, x_4$ *of degree less or equal to* 12*, has in this case length* 1820*, we would have to work with (symmetric) square matrices of the same size. That is, each matrix can be represented using* $1820 \times 1820/2 = 1657110$ *elements. The algorithm we run, SOSTOOLS 1.0 reports however the size of 1 731 857 such* elements. *It also reports that the dimension of the affine subspace is 15 015, information which we do not check by other means. Constructing the corresponding LMI took approximately 22 hours using SOSTOOLS 1.0.*

We have mentioned this case here in order to show the increase in computational complexity that the MIMO case brings about. The procedure failed by running out of memory, hence we conclude that, at least at this moment, the method is too complex to be applied even for small examples.

5.5 Analysis on a lower bound for the H_2 model reduction: discrete-time MIMO case

As we have seen in the previous section, although it is in principle possible to compute *exactly* an optimal solution of the H_2 model order reduction problem, the expressions involved (i.e. the H_2 distance expressed as a rational function) become extremely complicated in the MIMO case. Therefore, the exact methods become less attractive even for small sized problems. In this section we discuss a relaxation of the H_2 model reduction problem in the discrete-time MIMO case. The relaxation reduces to optimizing a rational function, having however a much simpler expression than the one of the H_2 distance. Then we compare the *exact* H_2 model reduction with its relaxation on two particular examples. It turns out that the relaxation is *sharp* in the first example. We prove this by using the exact methods of Chapter 3. In the second example, we do not know the exact value of the H_2 optimal approximant. However, numerical computations of the optimal approximant indicate that the relaxation gives a very good approximation of the optimal reduced order model, although it may be slightly different. In general, we do not expect the relaxation to be always sharp.

Let us consider the H_2 model order reduction problem in the MIMO discrete-time case. The method is based on rewriting the H_2 distance in a novel way using Faddeev reachability matrices. Let us describe briefly the method. Let (A, B, C) be a state-space representation of a time-invariant, discrete-time, linear stable system with $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$ and $C \in \mathbf{R}^{p\times n}$. We denote by $(\hat{A}, \hat{B}, \hat{C})$ an approximant of order $\hat{n}$. After optimizing with respect to $\hat{B}$, the criterion becomes:

$$\|\Sigma - \hat{\Sigma}\|_2^2 = \mathrm{trace}(B^T L_1 B) - \mathrm{trace}(B^T L_2 L_3^{-1} L_2^T B).$$

See also the notation of Section 5.2.2 where the matrices L_o^i there are here denoted for simplicity by L_i, $i \in \{1, 2, 3\}$.

Since the term $\mathrm{trace}(B^T L_1 B)$ depends only on the initial system, it is constant, and therefore we concentrate on the criterion to be maximized

$$W_c = \mathrm{trace}(B^T L_2 L_3^{-1} L_2^T B). \tag{5.7}$$

The main idea is to rewrite W_c as

$$\mathrm{trace}\left(ZP(P^TP)^{-1}P^T\right), \tag{5.8}$$

where the matrix Z depends on the original system (A, B, C) as well as on the unknown, parameterized matrix $\hat{A}$ and the matrix P depends on the matrices $\hat{A}$, $\hat{C}$ of the approximant. The exact definition of Z respectively P can be deduced from the construction of the following subsection.

5.5.1 An equivalent formulation of the H_2 criterion

As in Section 5.3, we work with parameterized representation of the approximant $(\hat{A}, \hat{B}, \hat{C})$. Although in the MIMO case there is no continuous parameterization of the whole manifold, we may work with a single generic chart. In order to fix the ideas, let us consider the following parameterization for our approximant

$$\hat{A} = \begin{pmatrix} 0 & 1 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & 1 \\ \star & \star & \cdots & \star \end{pmatrix}, \quad \hat{B} = \begin{pmatrix} \star & \cdots & \star \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \star & \cdots & \star \end{pmatrix}, \quad \hat{C} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \star & \star & \cdots & \star \\ \vdots & \vdots & & \vdots \\ \star & \star & \cdots & \star \end{pmatrix}.$$

Here the starred entries denote parameters to be chosen freely under the sole constraint that stability of $\hat{A}$ holds. This parameter chart involves $\hat{n}(m+p)$ parameters. Since the optimal $\hat{B}$ is computed analytically, after fixing $\hat{B}$ at its optimal value we are left with only $\hat{n}p$ parameters, that is the free entries of $(\hat{A}, \hat{C})$.

Based on [30], we have the following definitions and results.

Definition 5.5.1 *Let X be a matrix of size $s \times s$, and y a vector of size $s \times 1$. Then:*
(i) The characteristic polynomial of X is denoted by

$$\chi_X(z) = \det(zI_s - X) = z^s + \chi_1 z^{s-1} + \ldots + \chi_{s-1} z + \chi_s.$$

(ii) The controller companion form matrix associated with X is denoted by X_c and defined by

$$X_c = \begin{pmatrix} -\chi_1 & \cdots & -\chi_{s-1} & -\chi_s \\ 1 & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & & 1 & 0 \end{pmatrix}.$$

(iii) The Faddeev sequence of X is denoted by $\{X_0, X_1, \ldots, X_{s-1}\}$ and defined recursively by

$$\begin{aligned} X_0 &= I_s, \\ X_k &= XX_{k-1} - \frac{\operatorname{trace}\{XX_{k-1}\}}{k} I_s, \qquad (k = 1, 2, \ldots, s-1). \end{aligned}$$

Equivalently, if we define $\chi_0 = 1$, it holds (for $k = 0, 1, \ldots, s-1$) that

$$X_k = \chi_0 X^k + \chi_1 X^{k-1} + \ldots + \chi_{k-1} X + \chi_k I_s.$$

(iv) The Faddeev reachability matrix of (X, y) is denoted by $F_X(y)$ and defined by

$$F_X(y) = \left(\ X_0 y \mid X_1 y \mid \cdots \mid X_{s-1} y\ \right).$$

It then follows (see [30]) that the matrices L_2 and L_3 of (5.7) are given by

$$\begin{aligned} L_2 &= \sum_{i=1}^{p} F_{A^T}(c_i)\hat{\Delta}(F_{\hat{A}^T}(\hat{c}_i))^T, \\ L_3 &= \sum_{i=1}^{p} F_{\hat{A}^T}(\hat{c}_i)\Delta(F_{\hat{A}^T}(\hat{c}_i))^T, \end{aligned} \tag{5.9}$$

where (for $i = 1, \ldots, p$) the column vectors c_i and $\hat{c}_i$ denote the transposed rows of C and $\hat{C}$, respectively, whence:

$$C = \left(\begin{array}{c} c_1^T \\ \hline \vdots \\ \hline c_p^T \end{array}\right), \qquad \hat{C} = \left(\begin{array}{c} \hat{c}_1^T \\ \hline \vdots \\ \hline \hat{c}_p^T \end{array}\right),$$

and where the matrices $\hat{\Delta}$ (of size $n \times \hat{n}$) and Δ (of size $\hat{n} \times \hat{n}$) are the unique solutions of the associated highly structured Sylvester and Lyapunov equations in controller companion form:

$$\begin{aligned} \hat{\Delta} - A_c \hat{\Delta} \hat{A}_c^T &= e_1 e_1^T, \\ \Delta - \hat{A}_c \Delta \hat{A}_c^T &= e_1 e_1^T. \end{aligned}$$

Here, $e_1^T = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$ and A_c, respectively $\hat{A}_c$, denote the controller companion form of A, respectively $\hat{A}$. It is noted that the expression for L_3 involves a finite number of Faddeev reachability matrices, but also the positive definite symmetric matrix Δ. Let $\Gamma \in \mathbf{R}^{n \times n}$, invertible, such that $\Delta = \Gamma\Gamma^T$. We introduce the following matrices:

$$\begin{aligned} G_{\hat{A}^T}(\hat{c}_i) &= F_{\hat{A}^T}(\hat{c}_i)\Gamma, && \text{(of size } \hat{n} \times \hat{n}), \\ G_{A^T}(c_i) &= F_{A^T}(c_i)\hat{\Delta}(\Gamma^T)^{-1}, && \text{(of size } n \times \hat{n}). \end{aligned}$$

Then:

$$\begin{aligned} L_2 &= \sum_{i=1}^{p} G_{A^T}(c_i)(G_{\hat{A}^T}(\hat{c}_i))^T, \\ L_3 &= \sum_{i=1}^{p} G_{\hat{A}^T}(\hat{c}_i)(G_{\hat{A}^T}(\hat{c}_i))^T. \end{aligned}$$

The matrices $G_{\hat{A}^T}(\hat{c}_i)$ and $G_{A^T}(c_i)$ depend *linearly* on the entries of $\hat{c}_i$ and c_i, respectively. Therefore, these matrices can also be rewritten as:

$$\begin{aligned} G_{\hat{A}^T}(\hat{c}_i) &= \left(\hat{M}_1 \hat{c}_i \,\middle|\, \cdots \,\middle|\, \hat{M}_{\hat{n}} \hat{c}_i \right), \\ G_{A^T}(c_i) &= \left(M_1 c_i \,\middle|\, \cdots \,\middle|\, M_{\hat{n}} c_i \right), \end{aligned}$$

with $\hat{M}_1, \dots, \hat{M}_{\hat{n}}$ of size $\hat{n} \times \hat{n}$ depending only on $\hat{A}$, and $M_1, \dots, M_{\hat{n}}$ of size $n \times \hat{n}$ depending only on $\hat{A}$ and A. For later use we introduce the matrices

$$\begin{aligned} G_{\hat{A}^T} &= \left(G_{\hat{A}^T}(\hat{c}_1) \,\middle|\, \cdots \,\middle|\, G_{\hat{A}^T}(\hat{c}_p) \right), && \text{(of size } \hat{n} \times \hat{n}p), \\ G_{A^T} &= \left(G_{A^T}(c_1) \,\middle|\, \cdots \,\middle|\, G_{A^T}(c_p) \right), && \text{(of size } n \times \hat{n}p). \end{aligned}$$

We have:

$$\begin{aligned} L_2 &= \sum_{i=1}^{p} \sum_{j=1}^{\hat{n}} M_j c_i \hat{c}_i^T \hat{M}_j^T = \sum_{j=1}^{\hat{n}} M_j C^T \hat{C} \hat{M}_j^T, \\ L_3 &= \sum_{i=1}^{p} \sum_{j=1}^{\hat{n}} \hat{M}_j \hat{c}_i \hat{c}_i^T \hat{M}_j^T = \sum_{j=1}^{\hat{n}} \hat{M}_j \hat{C}^T \hat{C} \hat{M}_j^T. \end{aligned} \tag{5.10}$$

Finally we therefore may write:

$$\begin{aligned} L_2 &= M(C)(\hat{M}(\hat{C}))^T, \\ L_3 &= \hat{M}(\hat{C})(\hat{M}(\hat{C}))^T, \end{aligned}$$

where

$$\begin{aligned} M(C) &= \left(\; M_1 C^T \; \middle| \; \cdots \; \middle| \; M_{\hat{n}} C^T \; \right), \qquad \text{(of size } n \times \hat{n}p\text{)}, \\ \hat{M}(\hat{C}) &= \left(\; \hat{M}_1 \hat{C}^T \; \middle| \; \cdots \; \middle| \; \hat{M}_{\hat{n}} \hat{C}^T \; \right), \qquad \text{(of size } \hat{n} \times \hat{n}p\text{)}. \end{aligned}$$

Notice that $M(C)$ is in fact obtained from the matrix G_{A^T} by reordering its columns in a particular way. The same holds true for $\hat{M}(\hat{C})$ and $G_{\hat{A}^T}$. Substitution of the expressions (5.10) into the expression for W_c yields:

$$W_c = \text{trace}\left(\left[M(C)^T B B^T M(C) \right] (\hat{M}(\hat{C}))^T (\hat{M}(\hat{C})(\hat{M}(\hat{C}))^T)^{-1} (\hat{M}(\hat{C})) \right). \tag{5.11}$$

5.5.2 A relaxation of the H_2 criterion

Note that the criterion W_c of (5.11) is indeed of the form $\text{trace}(ZP(P^TP)^{-1}P^T)$ with

$$Z = M(C)^T B B^T M(C) \quad \text{and} \quad P = \hat{M}(\hat{C})^T.$$

The criterion W_c needs to be maximized now. We use the following known result (see [53] for a proof):

Theorem 5.5.2 *Let Z be a fixed $n \times n$ hermitian matrix. Consider the expression*

$$W(P) = \text{trace}(ZP(P^TP)^{-1}P^T)$$

where P ranges over the set of $n \times \hat{n}$ matrices of full column rank $\hat{n} \leq n$. Then:

a. The (globally) maximal value of $W(P)$ is equal to the sum of the $\hat{n}$ largest eigenvalues of Z (multiplicities included).

b. This maximal value is attained for any matrix P of which the column space is spanned by $\hat{n}$ independent eigenvectors of Z corresponding to these $\hat{n}$ largest eigenvalues.

The idea now is that the optimal matrix P of (5.8) is determined by Z, according to Theorem 5.5.2 b. Hence, using now Theorem 5.5.2 a, we can concentrate on maximizing the sum of the $\hat{n}$ largest eigenvalues of $Z = M(C)^T B B^T M(C)$. By construction, Z is a positive semidefinite matrix in $\mathbf{R}^{\hat{n}p \times \hat{n}p}$ of rank at most $\min\{n, \hat{n}p, m\}$.

Lemma 5.5.3 *The following holds*

$$\text{trace}(M(C)^T B B^T M(C)) = \text{trace}(B^T (\sum_{s=1}^{p} F_{A^T}(c_s) \hat{\Delta} \Delta^{-1} \hat{\Delta}^T F_{A^T}(c_s)^T) B).$$

Proof Let $Z = M(C)^T B B^T M(C)$. We have

$$\begin{aligned} \text{trace}(Z) \quad &= \quad \text{trace}(M(C)^T B B^T M(C)) \quad = \text{trace}(B^T M(C) M(C)^T B) = \\ &= (\textstyle\sum_{l,j} (B^T M(C))(l,j)^2) \end{aligned}$$

Here we have used the fact that trace$(SS^T) = \sum_{l,j} S(l,j)^2$, where S is an arbitrary matrix and $S(l,j)$ denote the (l,j) entry of S. Next using the fact that $M(C)$ was obtained from G_{A^T} by rearranging its columns, we obtain

$$\begin{aligned}\text{trace}(Z) &= (\textstyle\sum_{l,j}(B^T G_{A^T})(l,j)^2) = \text{trace}(B^T G_{A^T} G_{A^T}^T B) = \\ &= \text{trace}(\textstyle\sum_{s=1}^{p} B^T G_{A^T}(c_s) G_{A^T}(c_s)^T B)\end{aligned}$$

By using now the definition of $G_{A^T}(c_s)$ we have

$$\text{trace}(Z) = \text{trace}(\sum_{s=1}^{p} B^T F_{A^T}(c_s)\hat{\Delta}\Delta^{-1}\hat{\Delta}^T F_{A^T}(c_s)^T B)$$

which had to be proved. □

Theorem 5.5.4 *Let $\hat{n} \geq m$. Then*

1. *The sum of the $\hat{n}$ largest eigenvalues of $M(C)^T BB^T M(C)$ equals the sum of* all *eigenvalues, that is* trace$(M(C)^T BB^T M(C))$.
2. trace$(M(C)^T BB^T M(C))$ *is a rational function in the entries of $\hat{A}$.*

Proof *1.* This is obvious since the rank of $M(C)^T BB^T M(C)$ is at most m and therefore $\hat{n} - m$ eigenvalues (multiplicities included) of $M(C)^T BB^T M(C)$ are equal to 0.
2. Use the formula in the Lemma 5.5.3. Since both matrices Δ, $\hat{\Delta}$ are rational, hence Δ^{-1} is rational, trace$(M(C)^T BB^T M(C))$ is also a rational function. □

In the following we concentrate on the case $\hat{n} \geq m$, that is, the order of the approximant is larger than or equal to the number of inputs. Hence our problem becomes now maximizing trace(Z) which is a rational function, but whose expression is much more compact than the one corresponding to the original H_2 criterion. Also the number of variables in the rational function has decreased from $\hat{n}p$ to $\hat{n}$, due to the parameterization chosen in Section 5.5.1. It is interesting to compare the formulas of the H_2 criterion and the constructed relaxation.

Remark 5.5.5 *The following hold:*

- *Using the formulas (5.7) and (5.9) we obtain the following expression for the H_2 criterion:*

$$\text{trace}\left(B^T(\sum_{i=1}^{p} F_{A^T}(c_i)\hat{\Delta}(F_{\hat{A}^T}(\hat{c}_i))^T)(\sum_{i=1}^{p} F_{\hat{A}^T}(\hat{c}_i)\Delta(F_{\hat{A}^T}(\hat{c}_i))^T)^{-1}\right.$$
$$\left.(\sum_{i=1}^{p} F_{A^T}(c_i)\hat{\Delta}(F_{\hat{A}^T}(\hat{c}_i))^T)^T B\right).$$

- *Lemma 5.5.3 gives the following expression for the relaxation of the H_2 criterion*

$$\text{trace}(\sum_{s=1}^{p} B^T F_{A^T}(c_s)\hat{\Delta}\Delta^{-1}\hat{\Delta}^T F_{A^T}(c_s)^T B).$$

Notice the resemblance between the two criteria. The formulas above also explain why the second criterion has always a much more compact expression that the first one. Notice also that the unknown (parameterized) vectors $\hat{c}_i,\ i = 1, \dots, s$ do not appear in the second criterion.

5.5.3 A comparison of the exact H_2 criterion with its relaxation

The following remark is rather important for this approach, based on Theorem 5.5.2. Our P in (5.8) equals $\hat{M}(\hat{C})$ (see (5.11)), and hence, has a certain structure. On the other hand, the optimum P returned by Theorem 5.5.2 may not have the required structure. Therefore the above procedure returns in general an upper bound of the sought maximum, and therefore a lower bound on the H_2 minimization problem. The lower bound is proved to be sharp for reduction to order 1 models (see [53]). Naturally, one would like to know how good or bad such relaxation is in the general case, that is reduction to $\hat{n} \geq 1$ order models. Let W_c be the criterion of (5.11) and W its relaxation (where the *structured* matrix $\hat{M}(\hat{C})$ is replaced by an arbitrary, *unstructured* matrix P). In fact, although the two criteria (W_c and W) *do not* coincide as functions, one might hope that they still coincide *at their global maximum.* And this is truly the question we are interested in. In order to answer this question two examples are considered. In each example a 4-th order model with 2 inputs and 2 outputs is considered together with an H_2 (locally) optimal approximant of order 2. Then we want to know whether W has a global maximum *equal to the global maximum of W_c at the given approximant.* If that were true, then it would mean both that the approximation is sharp at the optimum and that the (local) optimum for the H_2 problem is actually a global optimum. For this we used the methods developed in Chapters 3 and 4 in order to establish whether, in each example, the H_2 optimal approximant was also an optimal approximant for our relaxation designed. Exact methods were employed. In the first case, it turned out that the H_2 optimal approximant *coincided* with an optimal approximant of our relaxation. However, in the second example we were less lucky since the locally optimal approximant at hand, of the H_2 criterion, proved to be indeed a local, not *global*, optimum. But other, numerical methods indicated that the global optimum of our relaxation was, if not equal, at least extremely close to the *numerical* value of the H_2 optimum. We present here the first example.

Example 5.5.6 *This example was constructed by Ralf Peeters in [53], such that both the original system and its approximant are known exactly, i.e. their* *entries are rational numbers. The system Σ is given by:*

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{1}{8} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}, \quad B = \begin{pmatrix} \frac{1}{2} & -\frac{3}{4} \\ \frac{383}{2080} & \frac{279}{1040} \\ \frac{1839}{8320} & -\frac{1317}{4160} \\ \frac{1419}{33280} & \frac{99}{1280} \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

The following second order approximation $\hat{\Sigma}$ is known to be a *local* maximum of W_c.

$$\hat{A}_0 = \begin{pmatrix} 0 & 1 \\ \frac{4}{9} & 0 \end{pmatrix}, \quad \hat{B}_0 = \begin{pmatrix} \frac{1}{2} & -\frac{3}{4} \\ \frac{1}{6} & \frac{1}{4} \end{pmatrix}, \quad \hat{C}_0 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

The exact expression for W_c was computed. It is a rational function involving 4 variables, the entries of

$$\hat{A} = \begin{pmatrix} 0 & 1 \\ -x_2 & -x_1 \end{pmatrix}, \quad \hat{C} = \begin{pmatrix} 1 & 0 \\ y_1 & y_2 \end{pmatrix}.$$

The numerator polynomial has total degree 14 and consists of 705 terms, while the denominator has total degree 12 and consists of 277 terms. The exact criterion turns out to be too complicated to be handled by the exact methods of Chapter 4. Next we compute the relaxation of the problem and obtain $\mathrm{trace}(Z) = p(x_1, x_2)/q(x_1, x_2)$ where p, q are the polynomials

$$\begin{aligned} p(x_1, x_2) = {} & -(-1 + x_2)(7907854144 + 3296829824x_1 - 6927169920{x_1}^2 - \\ & 2897270720{x_1}^3 + 818184528{x_1}^4 + 438089920{x_1}^5 - 18864576{x_1}^6 + \\ & 18209891616x_2 + 6894389696x_1x_2 - 5524025360{x_1}^2x_2 - 2356001232{x_1}^3x_2 - \\ & 65091328{x_1}^4x_2 - 56513808{x_1}^5x_2 + 60932736{x_1}^6x_2 + 14986490756{x_2}^2 + \\ & 5732635144x_1{x_2}^2 - 1324024308{x_1}^2{x_2}^2 - 583912552{x_1}^3{x_2}^2 - \\ & 270351000{x_1}^4{x_2}^2 + 30466368{x_1}^5{x_2}^2 + 5601025568{x_2}^3 + \\ & 2101676108x_1{x_2}^3 + 244751432{x_1}^2{x_2}^3 - 85974588{x_1}^3{x_2}^3 - \\ & 57124440{x_1}^4{x_2}^3 + 1003427217{x_2}^4 + 357191240x_1{x_2}^4 + 128010888{x_1}^2{x_2}^4 - \\ & 30466368{x_1}^3{x_2}^4 + 91843237{x_2}^5 + 57115746x_1{x_2}^5 + 11424888{x_1}^2{x_2}^5 + \\ & 5940378{x_2}^6 + 7616592x_1{x_2}^6 + 952074{x_2}^7) \\ q(x_1, x_2) = {} & 1081600(16 + 4x_1 + x_2)^2(-4 + 2{x_1}^2 - 4x_2 - {x_2}^2)^2. \end{aligned}$$

Notice how simple this expression is compared to the exact expression of the H_2 criterion. We optimize now the relaxation with respect to x_1, x_2 in the stability region. It turns out that $(x_1, x_2) = (0, -4/9)$, corresponding to $\hat{\Sigma}$ defined above, is an *exact* global optimum of the rational function $\mathrm{trace}(Z)$. For that we have employed the exact methods developed in Section 3.2 of this thesis.

Let us describe now briefly our calculations. We know that $(\hat{A}, \hat{B}, \hat{C})$ defined above correspond to a local maximum of W_c and that trace(Z) above is an upper bound on W_c. In order to show that $(\hat{A}, \hat{B}, \hat{C})$ is a global maximum of trace$(Z) = p/q$ it is sufficient to show that $(p/q)(0, -4/9) \geq (p/q)(x_1, x_2)$, for all (x_1, x_2) in the stability region. That is, the (x_1, x_2) for which $\hat{A}$ is stable, which is $\mathcal{S} = \{(x_1, x_2) \mid -1 < x_2 < 1,\ 1 + x_1 + x_2 > 0,\ 1 - x_1 + x_2 > 0\}$. Since $(p/q)(0, -4/9) = 35/16$ and $q(x_1, x_2)$, the denominator of p/q, is always nonnegative, we need to show that the polynomial $F = (35/16q - p)(x_1, x_2) \geq 0$ for all (x_1, x_2) in the stability region. We prove this by computing the minimum of F using the method of Section 3.2 . We have computed all the critical values of F and they were all nonnegative, with a single critical value equal to 0 attained in the stability region $\mathcal{S}$ at $(0, -4/9)$. Next, we have evaluated the polynomial F on the boundary of the stability region. F restricted to every edge of the boundary of $\mathcal{S}$ is a univariate polynomial, and it is positive. Therefore we concluded that F was nonnegative on the entire $\mathcal{S}$.

It is a completely open question what are necessary conditions for sharpness of the relaxation. Also we do not know why the bound was sharp in the Example 5.5.6.

Remark that in this example we do not employ our optimization methods for *computing* the optimal H_2 approximant. Both the original system and a local optimal approximant are given. Our purpose is to *perform analysis* on a proposed approximation method. We have opted here for the exact methods rather than their numerical counterparts (based on LMI's) for two reasons. Firstly, deciding whether the two optimal values are equal requires their exact computation of the real numbers. Secondly, for the first example, we found that the LMI method returned a *strict* lower bound (in fact it returned $-\infty$) and that was not sufficient for our purpose.

Let us give the second example (designed by Ralf Peeters in [53])

Example 5.5.7 *The original system is given by*

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{1}{8} & -\frac{1}{2} & \frac{3}{4} \end{pmatrix}, \quad B = \begin{pmatrix} -\frac{519}{8} & -\frac{829}{64} \\ \frac{2573}{16} & \frac{569}{8} \\ -\frac{951}{8} & -\frac{25673}{384} \\ -\frac{3767}{96} & \frac{2707}{384} \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

while the local optimal approximant (which turns out not *to be a global optimal approximant) is*

$$\hat{A}_0 = \begin{pmatrix} 0 & 1 \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad \hat{B}_0 = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}, \quad \hat{C}_0 = \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}.$$

We do not reproduce the calculation here.

5.6 Conclusions

We have approached the optimal H_2 model reduction as an optimization problem with a rational criterion function. We argued that although in principle it is possible to compute the criterion and its global minimum, the computational complexity increases tremendously with the order of the system and the order of the desired approximant. Even if small problems are manageable in the SISO case, more effort has to be put into solving the MIMO case. There the H_2 criterion function obtained has too high complexity. Since the complexity of the criterion is strongly related to the parameterization of the approximant, the hope is, based on the experience with the SISO case, that serious simplifications can be obtained using different parameterizations. This idea was not pursued in this thesis.

Section 5.5 explores a different possible approach to the problem in the MIMO discrete-time case, by looking at a considerably simpler lower bound. The lower bound was compared to the exact solution in two examples and proved to be either tight, in one example, or at least very close to it, in the second example.

Chapter 6

Other applications of optimization algorithms

In this chapter we present further applications of the algorithms developed in Chapters 3, 4 of the thesis. Chapter 5 presented an application to optimal model reduction in H_2 norm. We show here that other problems, like optimal model reduction with respect to the divergence rate criterion (Section 6.1) and estimation of the worst case H_2 norm of a system with uncertainties (Section 6.2) reduce to finding the global optimum of a rational function. Moreover, the exact algorithm for global optimization of a polynomial function (Algorithm 3.2.18), which computes a point in every connected component of the set of minimizers, finds application to the global identifiability question, as discussed in Section 6.3.

Section 6.1 is based on [40] while Section 6.3 is based on [38]. Section 6.2 is based on an idea of A. Stoorvogel and has not been published previously.

6.1 Optimal Model Reduction of Stationary Gaussian Systems with respect to the Divergence Rate Criterion

System identification for a particular approach (see Procedure 6.1.1) involves model reduction, that is determining for a model with a high state-space dimension a model of low state-space dimension. For Gaussian systems the problem of model reduction is considered with the divergence rate criterion. The divergence or Kullback-Leibler pseudo-distance corresponds to the expected value of the negative natural logarithm of the likelihood function. An algebraic method is proposed for model reduction. The results are a procedure for infimization of the criterion and a theorem that this problem reduces to an infimization problem for a rational function. As illustration, two examples of model reduction are presented which show that in general one can expect many local minima of the criterion. This section is based on [40] and is a sequel to the papers [65, 64].

6.1.1 Introduction

The aim of Section 6.1 is to show that model reduction for Gaussian systems by the divergence rate criterion reduces to optimization of a rational function, for which the methods of Chapter 4 can be subsequently applied.

The motivation is system identification of Gaussian systems. A finite-dimensional Gaussian system is a linear system driven by a Gaussian white noise process. Here, attention is limited to discrete-time systems. As is well known, a stationary Gaussian system is a mathematical model for an observed stationary Gaussian process. The system identification problem is to construct from observed data and from assumptions a mathematical model, here a Gaussian system, such that the observed processes of the model approximate the observed data as well as possible according to an approximation criterion.

Methods of system identification for Gaussian systems often used include the maximization of the likelihood function, the subspace identification algorithm, and the least-squares method. The divergence between two probability measures is a well known pseudo-distance. It equals the expectation of the negative of the natural logarithm of the likelihood function.

The approximation problem of system identification is one of the major problems of this area. The main questions of parameter estimation include: How to find the global infimum? How to derive the first-order conditions? How to compute the local minima? How many local minima are there? Is the global infimum unique?

The aim here is: (1) To present an algebraic approach and an algorithm for the infimization of the divergence rate criterion of Gaussian systems. (2) To show for several low order Gaussian systems that model reduction leads for the divergence rate criterion to two or more local minima. for system identification of Gaussian systems by the maximum likelihood method. Though it is known from theoretical investigations and from numerical experiments with examples of system identification problems that two or more local minima exist, the consequences of this for system identification practice seem not to be widely known.

The results of Section 6.1 include a procedure to determine the global infimum by an algebraic method. Determining the global infimum is proven to be equivalent to infimization of a rational function for which the methods of Chapter 4 can be applied. The approach is illustrated in Example 6.1.7 with the reduction from a third order Gaussian system to a second order one. The set of local minima is not completely determined in this case although an upper bound on its cardinality is provided. Example 6.1.6 treats model reduction for a Gaussian system of state-space dimension 2 to one of state-space dimension 1. In this case there are two potential minima, one is the global minimum and the other a local one. The values of the criterion estimated at these two points are quite close.

The novelty here, compared to previous publications (see [65, 64]), is in the algebraic approach to model reduction and to maximum likelihood parameter estimation of Gaussian systems. A description of the contents follows. The next subsection contains a short problem formulation. This subsection is best read in combination with the appendix. Section 6.1.3 presents the procedure for model reduction via divergence rate infimization. The algebraic method is presented in Section 6.1.4. Examples are provided in Section 6.1.5. Conclusions are stated in Section 6.1.6. Appendix 6.1.7 contains notation and terminology on linear systems, on Gaussian systems, and the formulas for the divergence rate of stationary Gaussian systems.

6.1.2 Problem formulation

The motivating engineering problem is to determine a simple mathematical model for a time series. One speaks of the system identification problem or of the approximate realization problem. Examples of such a problem are the modeling of a signal in a noisy communication channel, of messages in a digital communication network, and of the traffic flow on a motor-way.

Mathematical notation for the problem is summarized below. See the appendix for further details. Let (Ω, F) be a measurable space and $T = \mathbf{Z}$ denote the time index set. Let P_1 be a probability measure on (Ω, F) induced by a stationary Gaussian process $y : \Omega \times T \to \mathbf{R}^p$ with zero mean value function and covariance function $W : T \to \mathbf{R}^{p \times p}$.

A time-invariant finite-dimensional Gaussian system on a probability space (Ω, F, P) is a stochastic system with representation

$$\begin{aligned} x(t+1) &= Ax(t) + Bv(t), \\ y(t) &= Cx(t) + Dv(t), \\ &\quad (A, B, C, D) \in GSP(p, n, p), \\ &\quad x : \Omega \times T \to \mathbf{R}^n, \quad y : \Omega \times T \to \mathbf{R}^p, \end{aligned}$$

see Appendix 6.1.7 for the full specification of the system. If the parameters of the model are in the set $SGSP_{min}(p, n, p)$ then the output process is a stationary Gaussian process. The probability measure induced by this system on the output process y is denoted by $P(q)$ where $q \in QD$ represents the parameter of a selected parameterization and $QD \subseteq \mathbf{R}^k$ is the domain of parameterization.

In the following attention is restricted to the approximation problem of the system identification procedure.

Procedure 6.1.1 *1. Determine from a finite time series a high-order Gaussian system.*

2. *Model reduction: Determine from a high-order Gaussian system a low-order Gaussian system.*

In Section 6.1 attention for the approximation problem is restricted to the divergence rate criterion. The concept of divergence of two probability measures is used in information theory. In probability theory divergence corresponds to the Kullback-Leibler measure, see [10]. For a stationary stochastic process the concept of divergence rate of two probability measures has been defined. In Appendix 6.1.7 an expression is provided for the divergence rate of two measures induced by stationary Gaussian processes which are outputs of two time-invariant finite-dimensional Gaussian systems. Denote this divergence rate by $D_r(P_1 \| P_2)$.

Let $\bar{n} \in \mathbf{N}$ denote an upper bound on the dimension of the Gaussian system to be determined.

Problem 1 *Solve*

$$\inf_{n \leq \bar{n},\ f_p(q) \in SGSP_{min}(p,n,p)} D_r(P_1 \| P(q)). \tag{6.1}$$

The problem involves establishing whether or not a minimum exists, if a minimum exists to characterize the set of minima, and to construct a procedure to compute a minimum or to approximate an infimum.

6.1.3 Procedure for infimization of divergence rate

Algorithm 6.1.2 Infimization of the divergence rate of stationary Gaussian processes.
Data: Below System 1 represents the given probability measure and System 2 represents the probability measure associated to the parameterized approximant.

$$\begin{array}{lll} \textit{System 1} & n_1 \in \mathbf{N}, & (A_1, B_1, C_1, D_1) \in SGSP_{min}(p, n_1, p), \\ \textit{System 2} & n_2 \in \mathbf{N}, & (A_2, B_2, C_2, D_2) \in SGSP_{min}(p, n_2, p). \end{array}$$

1. *Compute the parameters of System 3, the inverse system of the approximant model System 2, by $n_3 = n_2$ and*

$$\begin{aligned} (A_3, B_3, C_3, D_3) = \ & (A_2 - B_2 D_2^{-1} C_2, B_2 D_2^{-1}, -D_2^{-1} C_2, D_2^{-1}) \\ & \in SGSP_{min}(p, n_3, p). \end{aligned}$$

2. *Determine by an algebraic method described in Section 6.1.4, if there exists, a parameter value $\hat{q}_3 \in QD$ such that*

$$\begin{aligned} \hat{q}_3 &= \text{argmin}_{q \in QD} f_c(q), & (6.2) \\ f_c(q) &:= D_r(P_1 \| P_2(q)), & (6.3) \end{aligned}$$

The expression for the divergence rate is expressed in terms of a realization, (A_4, B_4, C_4, D_4), of the series interconnection of System 3 and

System 1 (see Appendix 6.1.7).

$$\begin{aligned} f_c(q) &= D_r(P_1 \| P_2) \\ &= \frac{1}{2}\text{trace}(C_4 Q_4 C_4^T + D_4 D_4^T - I) - \frac{1}{2}\ln\det(D_4 D_4^T). \end{aligned} \tag{6.4}$$

where $Q_4 \in \mathbf{R}^{n_4 \times n_4}$ is the solution of the discrete-time Lyapunov equation,

$$Q_4 = A_4 Q_4 A_4^T + B_4 B_4^T. \tag{6.5}$$

3. *Set $(\hat{A}_3, \hat{B}_3, \hat{C}_3, \hat{D}_3) = f_p(\hat{q}_3)$ according to the parameterization map.*

4. *Compute the approximant System 2 according to*

$$(\hat{A}_2, \hat{B}_2, \hat{C}_2, \hat{D}_2) = \left(\hat{A}_3 - \hat{B}_3\hat{D}_3^{-1}\hat{C}_3, \hat{B}_3\hat{D}_3^{-1},\ -\hat{D}_3^{-1}\hat{C}_3, \hat{D}_3^{-1}\right). \tag{6.6}$$

6.1.4 Algebraic method

For the divergence infimization an algebraic method will be used. The *algebraic method* refers to the use of abstract algebra, computer algebra, and the use of the computer programs like Maple and Mathematica. The difficulties to be overcome in the algebraic methods are to organize the calculations and to find an approach that is of low complexity.

Procedure 6.1.3 *1. Select a parameterization for the matrices of System 3, A_3, B_3, C_3, and D_3, see Algorithm 6.1.2. In view of Theorem 6.1.4, we choose a parameterization which leaves matrices C_3, D_3 free. The control canonical form gives such a parameterization. Alternatively, one can consider canonical forms for stable systems as in [29]. Then the matrix C_3 does and the matrix B_3 does not explicitly depend on the parameter vector q. Consequently, see Algorithm 6.1.8, step 2, the matrix $C_4(q)$ does and matrix B_4 does not explicitly depend on the parameter $q \in QD$.*

2. *Solve by computer algebra the discrete-time Lyapunov equation (6.5) for the symbolic matrix $Q_4(q)$.*

3. *Calculate the value of the criterion according to formula (6.4). The criterion f_c is the sum of a rational function and of a natural logarithm of the parameters of the model matrices A_3, C_3, and D_3.*

4. *Apply the reduction technique formulated in Theorem 6.1.4 to solve analytically for the matrices C_3 and D_3 and to derive the simplified formula for the criterion see (6.4). There remains then an infimization problem for a rational function.*

5. *Determine the value of the infimum. If, moreover, the infimum is attained, i.e. the global minimum exists, then determine its location as well. For this use the approach of Chapter 4.*

6. *If this is of interest then information on the local minima can be obtained. Derive the first order conditions of the simplified criterion with respect to the elements of the parameter vector $q \in QD$. Computer algebra provides programs for this.*

7. *Determine all real solutions of the equation obtained by setting to zero the first derivative of the criterion with respect to the parameter vector. This is the most difficult and demanding part of the procedure.*

8. *Calculate for each solution the second derivative of the criterion. Discard all points for which the second derivative is not positive semi-definite.*

9. *For each of the remaining points calculate the value of the criterion $f_c(q)$. By comparing the different values numerically determine the global minimum or the set of global minima if there exist two or more parameter vectors which attain exactly the same value.*

Remark that steps *6-9* are optional. They should be executed only if there is interest in local minima.

Theorem 6.1.4 *Consider the infimization problem of step 2 of Algorithm 6.1.2,*

$$\inf_{q \in QD} f_c(q),$$

where the matrices (A_3, B_3, C_3, D_3) depend on the parameter vector $q \in QD$ except for B_3.

(a) *The minimization of the criterion with respect to the matrix C_3, for fixed A_3, D_3, is reached at the matrix*

$$\begin{aligned} C_3 &= -D_3 C_1 Q_2 Q_3^{-1}, \textit{ where,} && (6.7)\\ Q_4 &= \begin{pmatrix} Q_1 & Q_2 \\ Q_2^T & Q_3 \end{pmatrix} \in \mathbf{R}^{n_4 \times n_4} \textit{ is the solution of (6.5),} && (6.8)\\ Q_2 &\in \mathbf{R}^{n_1 \times n_2},\ Q_3 \in \mathbf{R}^{n_2 \times n_2}. && (6.9) \end{aligned}$$

Hence the criterion depends on the matrices A_3 and D_3 only.

(b) *The minimization with respect to $D_3 \in \mathbf{R}^{p \times p}$, for fixed A_3, is reached for D_3 satisfying*

$$\begin{aligned} D_3^T D_3 &= M^{-1}, \\ & \quad \textit{and the criterion simplifies to,} \\ f_c(q) &= -\frac{1}{2} \ln \det(D_1^T M^{-1} D_1) \ \ \textit{, where} && (6.10) \end{aligned}$$

$$M = C_1 \left(Q_1 - Q_2 Q_3^{-1} Q_2^T\right) C_1^T + D_1 D_1^T \in \mathbf{R}^{p \times p}.$$

The simplified criterion is a natural logarithm of a function which is a rational function with respect to the entries of the matrix A_3. Thus the infimization problem is reduced to an infimization problem for a rational function.

Proof Use in the proof is made of formulas of differentials of functions with respect to matrices, see e.g. [1]. More precisely, we use

$$\frac{\partial}{\partial X}\text{trace}[XS_1X^T] = XS_1 + XS_1^T \text{ and } \frac{\partial}{\partial X}\ln\det[XS_2X^T] = 2(X^{-1})^T.$$

From the formulas of Algorithm 6.1.8, we have

$$f_c(q) = \frac{1}{2}\text{trace}\left(D_3C_1Q_1C_1^TD_3^T + C_3Q_2^TC_1^TD_3^T + D_3C_1Q_2C_3^T + C_3Q_3C_3^T + \right.$$
$$\left. + D_3D_1D_1^TD_3^T - I\right) - \frac{1}{2}\ln\det\left(D_3D_1D_1^TD_3^T\right).$$

(a) Differentiation with respect to C_3 leads us to the optimal value for C_3, as in (6.7), for which value of C_3, the criterion becomes

$$f_c(q) = \frac{1}{2}\text{trace}\left(D_3MD_3^T - I\right) - \frac{1}{2}\ln\det\left(D_3D_1D_1^TD_3^T\right).$$

where

$$M = C_1Q_1C_1^T - C_1Q_2Q_3^{-1}Q_2^TC_1^T + D_1D_1^T.$$

(b) Differentiation with respect to D_3 in the formula above leads us, by equating to 0, to $D_3^TD_3M = I$. Hence, whenever M is invertible, D_3 must satisfy $D_3^TD_3 = M^{-1}$. Notice that, with our assumption on the invertibility of D_1, M is the sum between a positive definite matrix and a positive semi-definite matrix, hence it is invertible.

Using the fact that trace(XY) = trace(YX) and det(XY) = det(YX) for any two matrices X, Y for which the above products are defined, and the fact that trace is linear, we obtain

$$\begin{aligned} f_c(q) &= \frac{1}{2}\text{trace}\left(D_3^TD_3M - I\right) - \frac{1}{2}\ln\det\left(D_3^TD_3D_1D_1^T\right) \\ &= -\frac{1}{2}\ln\det\left(D_1^TM^{-1}D_1\right). \end{aligned}$$

□

Remark 6.1.5 *Relation (6.10) allows different formulations. The following might be useful*

$$f_c(q) = \frac{1}{2}\ln\det(M) - \ln\det(D_1),$$

where $\ln\det(D_1)$ *is a constant.*

6.1.5 Examples

Example 6.1.6 *Consider a Gaussian system of order* 2 *with representation in the control canonical form as*

$$A_1 = \begin{pmatrix} -0.4 & -0.32 \\ 1 & 0 \end{pmatrix}, \quad B_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$
$$C_1 = \begin{pmatrix} 0 & -0.28 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 1 \end{pmatrix}.$$

An approximant will be determined in the form of a Gaussian system of order 1, according to the divergence rate criterion. The class of Gaussian systems in which an approximant is to be sought is taken to be $SGSP_{min}(1,1,1)$. *This class is parameterized by the control canonical form, hence*

$$(A_2, B_2, C_2, D_2) = (a_2, 1, c_2, d_2).$$

If $d_2 > 0$, $|a_2| < 1$, $|a_2 - c_2 d_2^{-1}| < 1$, $c_2 \neq 0$, *then* $(A_2, B_2, C_2, D_2) \in SGSP_{min}(1,1,1)$.

We construct the quadruple, in control canonical form

$$(a_3, b_3, c_3, d_3) = (a_2 - c_2 d_2^{-1}, 1, -c_2 d_2^{-2}, d_2^{-1})$$

and compute the criterion to be minimized. As remarked, the optimum with respect to c_3 *and* d_3 *can be computed analytically. The criterion becomes*

$$f_c(q) = -\frac{1}{2}\ln\left(\frac{-34\,(25 + 10\,a_3 + 8\,a_3{}^2)\,(56907\,a_3{}^2 - 230375 - 79900\,a_3)}{(731\,a_3{}^2 + 1801\,a_3 + 19500)\,(391\,a_3{}^2 + 7039\,a_3 + 11000)}\right)$$

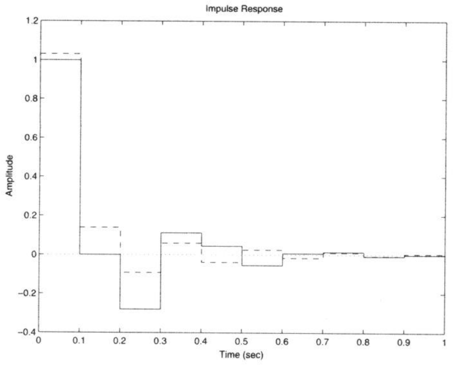
The critical points equation with respect to a_3 *is a univariate polynomial in* a_3 *whose roots are computed by numerical approximation. It turns out that in the stability region we find two points of minimum of the criterion* $f_c((\hat{a}_3, \hat{b}_3, \hat{c}_3, \hat{d}_3))$ *such that*

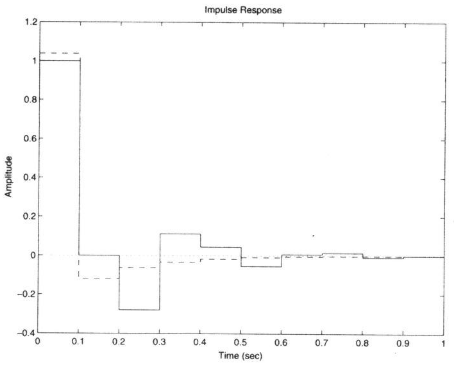
$$\begin{aligned} f_c((0.6353, 1, 0.1059, 0.9631)) &= 0.0376, \\ f_c((-0.7835, 1, -0.1269, 0.9693)) &= 0.0312. \end{aligned}$$

We conclude that the second point is a global minimum, while the first returns a local minimum, although their values are close. As in Step (4) of Algorithm 6.1.2, we compute the approximants

$$\begin{aligned} (\hat{a}_2, \hat{b}_2, \hat{c}_2, \hat{d}_2) &= (0.5253, 1.0383, -0.1142, 1.0383), \\ \textit{respectively} & \;\; (-0.6525, 1.0317, 0.1351, 1.0317). \end{aligned}$$

However the two approximant systems have a very different behavior. Below we have plotted the impulse response of the true system against the impulse response of the global approximant (left), respectively the impulse response of the true system against the impulse response of the local approximant (right). In both figures, the impulse response of the approximant is represented by a dashed line.





Example 6.1.7 *We have also considered a model reduction from order* 3 *to order* 2*, for the system*

$$A_1 = \begin{pmatrix} -1/4 & 1/2 & 1/3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \qquad B_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$
$$C_1 = \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}, \qquad D_1 = \begin{pmatrix} 2 \end{pmatrix}.$$

The approximant is taken in the control canonical form, parameterized by $\alpha_1, \alpha_2, \gamma_1, \gamma_2, \delta$*. After optimizing analytically with respect to* $\gamma_1, \gamma_2, \delta$ *we are left with optimization of a logarithm of a rational function*

$$-\frac{1}{2}\ln\left(\frac{(5640{\alpha_1}^3 + 85896{\alpha_1}^2 + 201240\alpha_1 + 181548 + 64746{\alpha_1}^2\alpha_2 + \ldots)}{(376{\alpha_2}^3 - 618{\alpha_2}^2 + 150\alpha_2 + 5139 - 564\alpha_1{\alpha_2}^2 - 444\alpha_1\alpha_2 + \ldots)}\right),$$

which reduces, due to the monotonicity of the logarithm function, to optimization of a rational function. By computing a Gröbner basis with respect to a total degree ordering for the first order conditions, we are able to establish that the function above has at most 100 *complex critical points, including multiplicities. Note that we were not able to compute a Gröbner basis with respect to a lexicographical ordering. Hence in order to solve the polynomial system, we have applied the Stetter-Möller method (Section 2.1.3) to the already computed total degree Gröbner basis. This method (i.e., its Maple 7.00 implementation) failed as well. In principle, the methods of Chapter 4 for constrained optimization of rational functions can be employed for computing global optimum. However, we did not perform the calculations.*

6.1.6 Conclusions

The main result of Section 6.1 is Procedure 6.1.3 with an algebraic method for infimization of the divergence rate between a Gaussian system and a class of such systems of lower state-space dimension. Theorem 6.1.4 establishes that the infimization problem reduces to an infimization problem for a rational function. Two examples illustrate the approach. In general a model reduction problem with this criterion and, by analogy, the parameter estimation with the likelihood function, will have many local minima. Further research is required to make the algebraic method more efficient and to streamline the computer algebra.

6.1.7 Appendix

Linear systems

In the body of this section concepts and results for time-invariant finite-dimensional linear systems are needed.

A *discrete-time time-invariant finite-dimensional linear system* is a dynamical system with the representation

$$\begin{aligned} x(t+1) &= Ax(t)+Bu(t), \quad x(t_0)=x_0, \\ y(t) &= Cx(t)+Du(t), \end{aligned}$$

where $T=\{t_0,t_0+1,\ldots\}$ is called the time axis, $x_0 \in \mathbf{R}^n$ for some $n \in \mathbf{N}$ is called the initial state, $u : T \to \mathbf{R}^m$ is called the input function, $x : T \to \mathbf{R}^n$ is called the state function, $y : T \to \mathbf{R}^p$ is called the output function, and $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{p\times n}$, $D \in \mathbf{R}^{p\times m}$. The parameters of this system will be denoted by

$$(A,B,C,D) \in LSP(p,n,m).$$

Denote the reachability matrix and the observability matrix of this model respectively by

$$\begin{aligned} \mathcal{R}(A,B) &= \begin{pmatrix} B & AB & \ldots & A^{n-1}B \end{pmatrix} \in \mathbf{R}^{n\times mn}, \\ \mathcal{O}(A,C) &= \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} \in \mathbf{R}^{np\times n}. \end{aligned}$$

It is said that (A,B) is a *reachable pair* if $\text{rank}(\mathcal{R}(A,B))=n$ and that (A,C) is an *observable pair* if $\text{rank}(\mathcal{O}(A,C))=n$. Denote the spectrum of the matrix $A \in \mathbf{R}^{n\times n}$ by $\text{spec}(A)$ and let $\mathbf{C}^- = \{\lambda \in \mathbf{C} | \ |\lambda|<1\}$ denote the inside of the unit disc in the complex plane. Define the subclasses of linear systems

$$\begin{aligned} LSP_{min}(p,n,m) &= \left\{ \begin{array}{l} (A,B,C,D) \in LSP(p,n,m) \mid \\ (A,B) \text{ reachable pair}, (A,C) \text{ observable pair} \end{array} \right\}, \\ SLSP(p,n,p) &= \left\{ \begin{array}{l} (A,B,C,D) \in LSP(p,n,p) \mid \text{rank}(D)=p, \\ \text{spec}(A) \subset \mathbf{C}^-, \text{spec}(A-BD^{-1}C) \subset \mathbf{C}^- \end{array} \right\}, \\ SLSP_{min} &= SLSP(p,n,p) \cap LSP_{min}(p,n,p). \end{aligned}$$

Gaussian systems

A time-invariant finite-dimensional *Gaussian system* (without inputs) is a stochastic system with representation

$$\begin{aligned} x(t+1) &= Ax(t) + Bv(t), & (6.11)\\ y(t) &= Cx(t) + Dv(t), & (6.12) \end{aligned}$$

where $r, n, p \in \mathbf{N}$, $p \geq 1$, $v : \Omega \times T \to \mathbf{R}^r$ is a Gaussian white noise process, thus an independent sequence of random variables with for each $t \in T$, $v(t) \in G(0, V)$ ($v(t)$ has a Gaussian probability distribution function with parameters 0 and V), $V \in \mathbf{R}^{r\times r}$, $V = V^T \succ 0$ (positive definite); $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times r}$, $C \in \mathbf{R}^{p\times n}$, $D \in \mathbf{R}^{p\times r}$; $x : \Omega \times T \to \mathbf{R}^n$, $y : \Omega \times T \to \mathbf{R}^p$ are stochastic processes satisfying the recursions (6.11,6.12).

Below a canonical form is used for Gaussian systems with respect to the covariance function of the output of the Gaussian system. For this purpose the reader is reminded of the theorem that a Gaussian system is a minimal stochastic realization of its output process iff it is stochastically observable and stochastically reconstructible, see [44, 45]. Consider a Gaussian system that is stable, with $\text{spec}(A) \subset \mathbf{C}^-$. Let $Q \in \mathbf{R}^{n\times n}$ be the solution of the discrete Lyapunov equation $Q = AQA^T + BB^T$, and let $G = AQC^T + BD^T \in \mathbf{R}^{n\times p}$. Then the Gaussian system is a minimal stochastic realization of its output process iff (A, C) is an observable pair and (A, G) is an observable pair. A time-invariant finite-dimensional Gaussian system is said to be a *Kalman realization* if in addition to being of minimal state-space dimension it satisfies $r = p$, $\text{rank}(D) = p$, $\text{spec}(A) \subset \mathbf{C}^-$, and $\text{spec}(A - BD^{-1}C) \subset \mathbf{C}^-$.

Define the set of parameters of Gaussian systems with $p, n, r, \in \mathbf{N}$ by

$$GSP(p,n,r) = \{\ (A,B,C,D) \in \mathbf{R}^{n\times n} \times \mathbf{R}^{n\times r} \times \mathbf{R}^{p\times n} \times \mathbf{R}^{p\times r}\ \},$$

$$SGSP_{min}(p,n,p) = \left\{ \begin{array}{l} (A,B,C,D) \in SGSP(p,n,p) \mid \text{rank}(D) = p,\ V = I, \\ (A,B) \text{ reachable pair,} (A,C),\ (A,G) \text{ observable pairs,} \\ \text{spec}(A) \subset \mathbf{C}^-, \text{spec}(A - BD^{-1}C) \subset \mathbf{C}^- \end{array} \right\}.$$

Divergence rate

The *divergence* or the *Kullback-Leibler pseudo-distance* on the set of probability measures of a measurable space (Ω, F) is defined by the formula

$$\begin{aligned} D(P_1 \| P_2) &= E_Q[r_1 \ln(\frac{r_1}{r_2}) I_{(r_2>0)}] \\ &= \int_\Omega r_1(\omega) \ln\left(\frac{r_1(\omega)}{r_2(\omega)}\right) I_{(r_2(\omega)>0)} Q(d\omega), \end{aligned}$$

where Q is a σ-finite measure on (Ω, F) such that

$$P_1 \ll Q,\ \frac{dP_1}{dQ} = r_1,\ \ P_2 \ll Q,\ \frac{dP_2}{dQ} = r_2.$$

Here $P \ll Q$ stands for P is absolutely continuous with respect to Q, i.e. $Q(A) = 0$ implies $P(A) = 0$ (see [10, Ch. 16], [36], and [65, Def. C.7]).

Let $y_1 : \Omega \times T \to \mathbf{R}^p$ be a stationary stochastic process on $T = \mathbf{Z}$. Denote by P_1, P_2 two measures for process y_1 on $(\mathbf{R}^p)^T$. The *divergence rate* between P_1, P_2 is defined by the formula

$$D_r(P_1 \| P_2) = \lim_{n\to\infty} \frac{1}{2n+1} D(P_1|_{[-n,n]} \| P_2|_{[-n,n]}), \tag{6.13}$$

if the limit exists, where $P_1|_{[-n,n]}, P_2|_{[-n,n]}$ denote the restrictions of P_1, P_2 respectively to probability measures of processes defined on the time index set $\{-n, \ldots, -1, 0, 1, \ldots, n\}$, see [36, 2.1.6] or [65, Def. E.4]. The following algorithm is based on a theorem of [65] and modified in [64].

Algorithm 6.1.8 *Computation of the divergence rate of two probability measures induced by the output processes of two time-invariant finite-dimensional Gaussian systems.*
Data. Let $p, n_1 \in \mathbf{N}^*$, $n_2 \in \mathbf{N}$,

$$(A_1, B_1, C_1, D_1) \in SGSP_{min}(p, n_1, p), \quad (A_2, B_2, C_2, D_2) \in SGSP_{min}(p, n_2, p).$$

1. *Construct the parameters of the inverse system of System 2 by the formulas* $n_3 = n_2$,

$$\begin{aligned}(A_3, B_3, C_3, D_3) = \; & (A_2 - B_2 D_2^{-1} C_2, B_2 D_2^{-1}, -D_2^{-1} C_2, D_2^{-1}) \\ & \in SLSP(p, n_3, p).\end{aligned}$$

2. *Construct the parameters of the series connection of System 3 and of System 1 according to the formulas* $n_4 = n_1 + n_3$,

$$\begin{aligned}(A_4, B_4, C_4, D_4) = & \left(\begin{pmatrix} A_1 & 0 \\ B_3 C_1 & A_3 \end{pmatrix}, \begin{pmatrix} B_1 \\ B_3 D_1 \end{pmatrix}, \right. \\ & \left. \begin{pmatrix} D_3 C_1 & C_3 \end{pmatrix}, D_3 D_1 \right) \in SLSP(p, n_4, p).\end{aligned}$$

3. *Solve the following discrete-time Lyapunov equation for the matrix* $Q_4 \in \mathbf{R}^{n_4 \times n_4}$,

$$Q_4 = A_4 Q_4 A_4^T + B_4 B_4^T. \tag{6.14}$$

4. *Compute the expression for the divergence rate,*

$$D_r(P_1 \| P_2) = \frac{1}{2}\text{trace}(C_4 Q_4 C_4^T + D_4 D_4^T - I) - \frac{1}{2} \ln \det(D_4 D_4^T), \tag{6.15}$$

where P_1 *and* P_2 *are the probability measures associated with the output processes of the Gaussian systems 1 and 2 respectively.*

6.2 Systems with uncertainties: the H_2 norm

In this section we discuss aspects related to systems with uncertainties. Since we work with a model (*nominal* system), which is an approximation of the *true* system and study its properties, we would like to know whether these properties extend to the true system as well. Typically, the nominal system is simple, say linear, time-invariant, finite-dimensional. In robustness analysis, one actually defines a class of systems by adding uncertainty to the nominal system. A very important assumptions is that the true system belongs to this class. This class can consist, for example, of the linear systems situated in the neighborhood of the given nominal model. This can be useful when one is confident that the true system is linear but unsure of its actual representation. More generally, the uncertainty can include unmodelled dynamics of the plant such as nonlinearities, time-variance, etc. Therefore the uncertainty can be represented as a (dynamical or static) system, connected to the nominal system. The literature (see, e.g. [72]) treats the following two cases, where the transfer matrix $\Delta(s)$ representing the uncertainty is: (1) *unstructured* (i.e. Δ is a full matrix); (2) has a block-diagonal structure.

Here we allow *arbitrarily* structured perturbations and we are interested in answering two questions about our class of systems. One question relates to stability, i.e. we want to know if all the systems in the class remain stable, knowing that the nominal system is stable. In this case we say that the nominal system is *robustly stable* against the perturbations Δ. Another question relates to the performance of the system, namely we are interested in the worst case H_2 norm of the systems in this class.

Notice that the H_2 norm was defined for *linear, time-invariant* systems. If one wants to include time-variance and/or nonlinearities in the uncertainty, then one needs to extend the notion of H_2 to these classes of systems. Several ways to do that are developed in [63]. Notice that for each class, there are few possible definitions for the H_2 norm but they are not necessarily equivalent, nor do they define a norm (only a pseudo-norm, i.e. it can be 0 for nonzero systems). We do not discuss further this issue since we deal primarily with linear systems.

The robust stability problem has been previously addressed for general (not necessarily linear stationary) uncertainty, see for example [63], [15]. For unstructured dynamic uncertainty, the so called *small gain theorem* gives a necessary and sufficient condition for robust stability ([72], Theorem 9.1). For structured uncertainty the problem is more complicated. The so-called *structured singular value* is used, see e.g. [72], Section 11.2, to derive necessary and sufficient conditions for robust stability in the case of block-diagonal structured uncertainty. The structured singular value is difficult to compute in practice and in general only upper and lower bounds of its value are known.

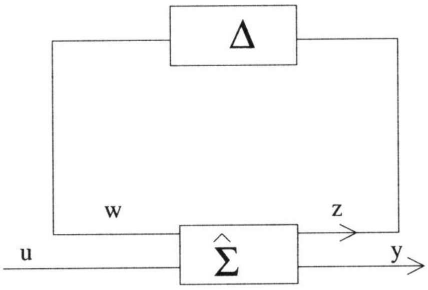
6.2.1 Problem formulation

Let us introduce our problem formulation. We consider a particular case where the uncertainty is in the state matrix, as follows

$$\Sigma : \quad \begin{cases} \dot{x} = (A + M\Delta N)x + Bu \\ y = Cx \end{cases} \tag{6.16}$$

with $A \in \mathbf{R}^{n\times n}$ *stable*, $B \in \mathbf{R}^{n\times m}$, $M \in \mathbf{R}^{n\times k_1}$, $C \in \mathbf{R}^{p\times n}$, $N \in \mathbf{R}^{k_2\times n}$ given and $\Delta \in \mathbf{F}^{k_1\times k_2}$ is an unknown matrix representing the uncertainty ($\mathbf{F}$ is a field, equal either to $\mathbf{R}$ or $\mathbf{C}$). Notice that the uncertainty is in the state transition matrix, the input and output matrices being considered known. This can be extended to the general case in a straightforward manner.

The following situation reduces naturally to a formulation of the type (6.16). *Consider the interconnection:*



Here $\hat{\Sigma}$ is a system described by

$$\hat{\Sigma} : \quad \begin{cases} \dot{x} = Ax + Bu + Mw \\ z = N \\ y = Cx \end{cases}$$

We have $w = \Delta z$ and a simple calculation shows that the closed loop system perturbed with the static matrix Δ, has the form of system Σ.

As we already mentioned, we are interested here in computing the H_2 norm of the system Σ. But since the system is not completely known, we can only compute the worst case H_2 norm, corresponding to the system with the *largest* H_2 norm in our class of interest.

Let us consider the matrix Δ affinely parameterized by $\delta \in \mathbf{R}^k$, where k is the number of parameters. We consider the case of a bounded uncertainty, since in general a large enough perturbation can destabilize the system. We consider here two cases for the condition Δ bounded: (a) $\|\Delta\|_2 \leq 1$ (boundness in the 2-norm); (b) $\max_{i,j} |\delta_{i,j}| \leq 1$. The bound 1 on the right-hand side of each inequality can be replaced by an arbitrary number. Let us define the parameter domain $\Omega \subseteq \mathbf{R}^k$ as the set of all parameters satisfying the boundness condition on Δ. It is not difficult to see that in both cases (a) and (b) above, the corresponding Ω domain is convex, hence it is also connected. Moreover, the interior of Ω is an open set (as the counter-image of an open set through a continuous function). These properties of the domain will be used later in proving Proposition 6.2.2.

To describe the perturbation matrix Δ and the class of systems Σ given by (6.16), we denote them by $\Delta(\delta)$, respectively $\Sigma(\delta)$, with $\delta \in \Omega$. We may now formulate the main problem of Section 6.2, namely we want to compute:

$$\begin{array}{l} \sup \|\Sigma(\delta)\|_2^2 \\ \text{s.t.} \quad \delta \in \Omega \end{array} \tag{6.17}$$

where $\|.\|_2$ denotes the H_2 norm of a system. Notice that if $\Sigma(\delta) : \ (A+M\Delta(\delta)N, B, C)$ is stable for all $\delta \in \Omega$, its H_2 norm can be computed using the formulas of Section 5.1.1. Therefore we have:

Proposition 6.2.1 *Let $\Sigma(\delta)$ given by (6.16) describe a class of uncertain systems with A stable and with $\Delta(\delta)$, the bounded uncertainty as in cases (a), (b) above, such that $A + M\Delta(\delta)N$ is stable for all $\delta \in \Omega$. Then the worst case H_2 norm in the class is given by the square root of*

$$\begin{array}{ll} \sup & \text{trace}(CL(\delta)C^T) \\ \text{s.t.} & \qquad \delta \in \Omega \end{array} \tag{6.18}$$

where $L(\delta)$ is the solution of $(A + M\Delta(\delta)N)L + L(A + M\Delta(\delta)N)^* = -BB^T$.

Proof Obvious, since all matrices $A+M\Delta(\delta)N$, for $\delta \in \Omega \subseteq \mathbf{R}^k$, are stable. □

One may wonder however what happens to the H_2-norm of a system $\Sigma(\delta) : \ (A+M\Delta(\delta)N, B, C)$ when $A + M\Delta(\delta)N$ becomes unstable. Does the norm become unbounded? The answer is negative in general, but positive in the following case.

Proposition 6.2.2 *If any $\Sigma(\delta)$ in (6.16) is minimal for all $\delta \in \Omega$, then the nominal system is robustly stable against the perturbations $\Delta(\delta)$ if and only if the supremum of (6.18) is finite.*

Proof We show that with this hypothesis, the system is not robustly stable if and only if its norm is ∞.

Suppose that there is a $\delta_0 \in \Omega$, such that $A + M\Delta(\delta_0)N$ is unstable. Since all systems Σ parameterized by $\delta \in \Omega$ are minimal, there is no pole-zero cancellation. Using the fact that A has all poles in the left-half plane, $A + M\Delta(\delta_0)N$ has at least one pole in the right-half plane and the domain Ω is connected, there must exist a $\delta_1 \in \Omega$, such that $A + M\Delta(\delta_1)N$ has a pole on the imaginary axis (due to the fact that Ω is connected). Therefore, the norm of $\Sigma_1 : \ (A + M\Delta(\delta_1)N, B, C)$ is ∞. □

Remark 6.2.3 *The minimality hypothesis in Proposition 6.2.2 is quite important for robust stability, without it the theorem is not true as the following counter-example shows. We prove there that in a non-minimality situation, the pole on the imaginary axis may cancel against one of the zeros, while the criterion remains bounded.*

Counter-example 6.2.4 *Let us consider the nominal system given by the triple*

$$A = \begin{pmatrix} -3 & -2 \\ 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & -1 \end{pmatrix}, \tag{6.19}$$

stable and minimal. Let

$$M = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad N = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Delta = \begin{pmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{pmatrix}, \tag{6.20}$$

where Δ is the disturbance matrix. Hence,

$$A + M\Delta N = \begin{pmatrix} -3 & -2+\delta_2 \\ \delta_1 + 1 & 0 \end{pmatrix}.$$

The transfer function of the system Σ (see (6.16)) is

$$Tf(s) = \frac{s - (\delta_1 + 1)}{s^2 + 3s - (\delta_1 + 1)(\delta_2 - 2)}.$$

The example was designed such that, when $\delta_1 = -1$, the system has a pole on the imaginary axis which is however unobservable or uncontrollable, since the system looses minimality. We want to show that in this example the H_2 norm of Σ remains finite, even when the system becomes unstable for a disturbance which is large enough ($\delta_1 = -1$ is in the Ω domain).

Let us compute now the criterion (6.18). We compute the solution of the Lyapunov equation

$$L = \frac{1}{6}\begin{pmatrix} 1 & 0 \\ 0 & -\frac{\delta_1+1}{\delta_2-2} \end{pmatrix}.$$

The final expression for L was obtained by simplifying out the factor $1 + \delta_1$ in the entries of L. The criterion (6.18) equals

$$\frac{\delta_2 - \delta_1 - 3}{6(\delta_2 - 2)}.$$

Notice that its denominator does not depend on δ_1, hence we expect that by varying δ_1, the expression of the criterion will remain bounded. Indeed, let us consider Δ such that $\|\Delta\|_2 \leq 3/2$, that is $\Omega = \{(\delta_1, \delta_2) \in \mathbf{R}^2 \mid |\delta_i| \leq 3/2, i = 1, 2\}$. Since the denominator of the criterion is (strictly) positive on Ω, then the criterion is bounded on the domain. Notice however that for $(-5/4, 1) \in \Omega$, the matrix $A + M\Delta(-5/4, 1)N$ has an unstable eigenvalue, namely $(\sqrt{10} - 3)/2$.

As we have seen, the loss of minimality is quite essential. In this example, the system Σ becomes non-minimal for $1 + \delta_1 = 0$ (loss of reachability) or $\delta_2 - \delta_1 - 6 = 0$ (loss of observability).

Remark 6.2.5 *The value (6.18) can serve as a criterion for testing the* lack *of robust stability in the following way. If (6.18) is ∞, then the system is* not *robustly stable. If (6.18) is finite we cannot draw immediately a conclusion.*

The following example shows a somewhat peculiar behavior of the H_2-norm of a system.

Example 6.2.6 *Let*

$$A = \begin{pmatrix} -3 & -2 \\ 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & \delta - 1 \end{pmatrix},$$

$$M = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad N = \begin{pmatrix} 1 & 2 \end{pmatrix}, \quad \Delta = \begin{pmatrix} \delta \end{pmatrix},$$

with the transfer function

$$Tf(s) = \frac{s + \delta - 1}{(s+2)(s - \delta + 1)}.$$

The system $\Sigma : (A + M\Delta N, B, C)$ is stable and minimal for $\delta = 0$. It is interesting to see that the H_2-norm of Σ is constant, namely it equals $1/2$, on the entire stability region. This can also be seen immediately by using the formulas of Section 5.1.1.

Since the H_2-norm of the system Σ can be expressed as a rational function in the parameters of Δ, we can treat the problem (6.17) as a rational optimization problem. We investigate here the applicability of the rational optimization algorithm algorithms developed in Chapter 4.

6.2.2 Computing the worst case H_2 norm

The system Σ is given in a state-space representation as in (6.16). Suppose we choose to compute the norm using the formula

$$||\Sigma||_2^2 = \text{trace}(CLC^T),$$

where L is symmetric in the real case and hermitian in the complex case and satisfies

$$(A + M\Delta N)L + L(A^T + N^T\Delta^* M^T) + BB^T = 0.$$

The latter equation is linear with respect to the entries of L, therefore the solution L can in principle be computed symbolically.

Since this is a constrained rational optimization problem, we know that there are several ways of solving it. We discuss here one possibility.

Reparametrization method

It is possible to rewrite our constrained rational optimization problem as an unconstrained one using a reparametrization of Δ, as follows. For a constraint of type (a) $\|\Delta\|_2 \leq 1$ we have:

Proposition 6.2.7 *Let $\Delta \in \mathbf{F}^{k_1 \times k_2}$. Then $\|\Delta\|_2 \leq 1$ if and only if there exist a matrix $\Gamma \in \mathbf{F}^{k_1 \times k_2}$ such that $\Delta = 2\Gamma(I + \Gamma^*\Gamma)^{-1}$.*

Proof Assume first that a matrix Γ such that $\Delta = 2\Gamma(I + \Gamma^*\Gamma)^{-1}$ exists. We make use of the fact that $\|\Delta\|_2 \leq 1 \iff \Delta^*\Delta - I \preceq 0$. Then

$$\begin{aligned}\Delta^*\Delta - I &= 4(I + \Gamma^*\Gamma)^{-1}\Gamma^*\Gamma(I + \Gamma^*\Gamma)^{-1} - I \\ &= -[4(I + \Gamma^*\Gamma)^{-2} - 4(I + \Gamma^*\Gamma)^{-1} + I] \\ &= -[2(I + \Gamma^*\Gamma)^{-1} - I]^*[2(I + \Gamma^*\Gamma)^{-1} - I] \preceq 0.\end{aligned}$$

Conversely, assume that $\|\Delta\|_2 \leq 1$ and consider the singular value decomposition of $\Delta = UDV^*$ with $U \in \mathbf{C}^{k_1 \times k_1}, V \in \mathbf{C}^{k_2 \times k_2}$ orthogonal matrices and $D = \operatorname{diag}(d_1, \ldots, d_k)$ with $1 \geq d_1 \geq \ldots \geq d_k \geq 0$ and $k = \min\{k_1, k_2\}$. For every $i = 1, \ldots, k$, there exists (at least one) t_i such that $d_i = 2t_i/(1 + t_i^2)$. Define $T = \operatorname{diag}(t_1, \ldots, t_k)$ and $\Gamma = UTV^*$. Then it is fairly easy to see that Γ constructed in this way satisfies $\Delta = 2\Gamma(I + \Gamma^*\Gamma)^{-1}$. □

Remark that if Δ is a block diagonal matrix, the above parameterization can be applied to each block individually and in this way we obtain a parameterization of the whole matrix.

For both cases of real and complex perturbation matrices Δ (Γ), the objective function of the optimization problem is a rational function, with real coefficients, in the entries of Γ, respectively in the real and imaginary parts of the entries of Γ. Let us notice here that the reparametrization increases a lot the computational complexity, especially in the case of full uncertainty matrix.

A constraint of type (b) $\max_{i,j} |\delta_{i,j}| \leq 1$ is parameterized using the same idea by $\delta_{i,j} = 2t_{i,j}/(1 + t_{i,j}^2)$.

The choice of a canonical form: computational aspects

Obviously, no matter which of the equivalent representations we consider, the objective, i.e. the H_2 norm of the system, is the same in the end. However, our choice might affect the complexity of the objective function computations. Notice that the the solution L of the Lyapunov equation (linear equation in the entries of L) can be in principle solved using Cramer's rule. That is, computing symbolically some determinants.

Let us remark that we need not compute a full solution of the Lyapunov equation, just a few elements of the matrix $L = (l_{s,r})$ may suffice. That is due to the particular form of the criterion:

$$\text{trace}(CLC^T) = \sum_{i=1}^{p} c_i L c_i^T,$$

where c_i denotes the $i-$th row of C. It is clear that if C is sparse, then not all elements of L need to appear in the criterion, hence we need not compute them. In particular, if for $i = 1, \ldots, p$ the i-th row of C is $c_i = \begin{pmatrix} 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \end{pmatrix}$ with a 1 on the j_i-th position, the criterion becomes

$$\text{trace}(CLC^T) = \sum_{i=1}^{p} l_{j_i,j_i}.$$

Hence, out of $n(n+1)/2$ different elements of L we only need to compute p elements, as described above. That helps, since with Cramer's rule we compute the entries of L individually, as the fraction of two determinants. Notice that choosing a representation in which C has the required form is possible in the MIMO case, as soon as we assume that C has full rank.

On the other hand, we have to take into account that computing determinants symbolically may not be an easy task. It normally helps if the determinant is sparse. That reduces in our case to having a sparse matrix $A + M\Delta N$. The question is now whether we could combine the two requirements, having C as a submatrix of I_n and having a sparse $A + M\Delta N$ in the same parameterization. Or, if not, what would be the right balance, between a sparse C and a sparse $A + M\Delta N$? In other words, should we put more effort in computing just a few determinants or should we go for computing many determinants, provided that the computation of each of them is somewhat simpler? We do not expect to give a general answer to this question, that may depend on the example at hand. Our intention is merely to make the user aware of these issues and give, say, some guidelines.

One main problem here is that it is difficult to estimate apriori how difficult the symbolic computation of the determinant(s) would be. We claim here that this is related to the sparsity of $A + M\Delta N$ (and this is influenced by both the sparsity of A and $M\Delta N$ as well as by the overlapping of their zeros). Also, for an intermediate complexity of the computations, we could choose $A + M\Delta N$ with as many numerical entries (not necessarily 0) as possible. This task seems to be somewhat easier since all the parameters are contained in Δ and M, N have both rank smaller than or equal to n.

Procedure

We propose the following procedure for computing the worst case H_2 norm.

Procedure 6.2.8 *The following procedure computes the worst case H_2 norm of an uncertain (robustly stable) system.*

1. *Compute the criterion (6.18) as a rational (constrained) optimization function.*
2. *Rewrite the optimization problem of step 1 as an unconstrained optimization problem, using for example the reparametrization method discussed earlier.*
3. *Compute the supremum, using either one of the methods of Sections* 4.1.2, 4.1.3.

Remark 6.2.9 *In case the algorithm of Section 4.1.2 is employed, an exact value is obtained while the procedure of Section 4.1.3 may return in general an upper bound on the actual supremum.*

6.2.3 Example

We treat here a small example using the reparametrization method.

Example 6.2.10 *Let us consider a system $\hat{\Sigma}$ and the diagonal perturbation Δ given by*

$$A = \begin{pmatrix} 1 & 2 & -3 \\ 0 & -3 & 1 \\ 1 & 0 & -2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 2 & 0 \\ -1 & 0 & 3 \end{pmatrix},$$

$$M = \begin{pmatrix} -1 & -2 \\ 0 & 3 \\ 0 & 1 \end{pmatrix}, \quad N = \begin{pmatrix} 1 & -4 & 0 \\ 0 & 1 & -2 \end{pmatrix}, \quad \Delta = \begin{pmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{pmatrix}.$$

We approach this example slightly differently, namely, instead of computing the worst case H_2-norm in a domain where the system is robustly stable, we determine an area Ω where the system $\Sigma : (A + M\Delta N, B, C)$ is minimal. Then, in Ω, we know that the boundness of our criterion (6.18) is equivalent to robust stability. Straight-forward calculations show that:

(i) the system $\Sigma : (A + M\Delta N, B, C)$ looses observability for $(\delta_1, \delta_2) = (16/81, 4/9)$;

(ii) the system $\Sigma : (A + M\Delta N, B, C)$ losses reachability when $\delta_2 = 1/6$ (δ_1 is free);

We are also interested to know whether the system has hidden unstable modes. By studying the eigenvalues of $A + M\Delta N$ in these two cases, we obtain

(i) all eigenvalues are stable;

(ii) all eigenvalues are stable if and only if $\delta_1 \in (2(5-\sqrt{21})/3, 2(5+\sqrt{21})/3)$.

The above computation shows that in $\Omega = \{(\delta_1, \delta_2) \mid \delta_i \leq 1/10,\ i = 1, 2\}$, the system Σ is minimal.

The criterion (6.18) is a rational function, denoted here $p(\delta_1, \delta_2)/q(\delta_1, \delta_2)$, and was computed using Mathematica

$$\begin{aligned} p(\delta_1, \delta_2) &= -319 - 137\delta_1 - 20{\delta_1}^2 + 1271\delta_2 - 279\delta_1\delta_2 - 226{\delta_1}^2\delta_2 - \\ &\quad -1262{\delta_2}^2 + 27\delta_1{\delta_2}^2 + 46{\delta_1}^2{\delta_2}^2 + 245{\delta_2}^3 - 46\delta_1{\delta_2}^3 \\ q(\delta_1, \delta_2) &= 2(1 + 2\delta_1 - 6\delta_2 + 23\delta_1\delta_2) \\ &\quad (-15 - 22\delta_1 - 5{\delta_1}^2 + 14\delta_2 + 36\delta_1\delta_2 + {\delta_1}^2\delta_2 - 4{\delta_2}^2 - \delta_1{\delta_2}^2) \end{aligned}$$

Notice that unlike the H_2 model reduction where the denominator was always a positive polynomial, we are here in a different situation. The denominator may become 0 if the system becomes unstable. If the system is minimal then the denominator will become 0 whenever the system becomes unstable.

It turns out that for $\|\Delta\|_2 \leq 1/10$ the system is not robustly stable. That follows from Theorem 4.1.3 and the fact that the denominator changes sign in the domain defined by $\|\Delta\|_2 \leq 1/10$. Hence, according to Proposition 6.2.2, the system is not robustly stable for a disturbance Δ with $\|\Delta\|_2 \leq 1/10$.

Let us consider a smaller disturbance, for example 1/20. In order to satisfy the condition $\|\Delta\|_2 \leq 1/20$ we make the substitution $\delta_1 = \gamma_1/10(1+\gamma_1^2)$, $\delta_2 = \gamma_2/10(1+\gamma_2^2)$ and apply the optimization algorithm.

We know that if the denominator changes sign, the criterion (6.18) is ∞. Hence we first check the denominator. Since the denominator is positive at a certain point, we compute its minimum by constructing an LMI relaxation. The LMI relaxation has a linear space of dimension 316 and uses square symmetric matrices of size 28. The lower bound on the minimum of the denominator returned by the algorithm is $1.42\ 10^6$. Since this is positive, we conclude that the denominator does not change sign, moreover $q(\gamma) > 0,\ \forall\gamma \in \mathbf{R}^2$.

Let us return to our problem. We want

$$\begin{aligned} \max\frac{p(\gamma)}{q(\gamma)} &= -\min\frac{-p(\gamma)}{q(\gamma)} = -\max\{\alpha \mid \frac{-p(\gamma)}{q(\gamma)} \geq \alpha \ \ \forall\gamma \in \mathbf{R}^2\} = \\ &= -\max\{\alpha \mid -p(\gamma) - \alpha q(\gamma) \geq 0 \ \ \forall\gamma \in \mathbf{R}^2\}. \quad (6.21) \end{aligned}$$

We consider now problem (6.21). The LMI relaxation for this problem has a linear space of dimension 316 and uses square symmetric matrices of size 28 as well. The upper bound on the supremum, returned by the algorithm is 17.3852. By local search we find this value as a local maximum, and therefore conclude the the bound is attained.

We draw two conclusions from here. Firstly, all the systems in the class are stable for a disturbance Δ, with $\|\Delta\|_2 \leq 1/20$. Secondly, the worst-case H_2 norm is 4.1696, while the norm of the nominal system is 3.2609. For these calculations we took into account that our criterion returns the square of the H_2 norm.

6.2.4 Conclusions

We have shown here how the algorithms for rational optimization developed in Chapter 4 can be applied to compute the worst case H_2 norm of a robustly stable system with (structured) uncertainty. It should be noted that we have treated here only linear uncertainties since in this case the H_2 norm is indeed a rational function in the uncertainty parameters.

6.3 Global identifiability analysis using algorithms for detecting connected semi-algebraic components

In the process of modeling of phenomena, one proposes a model defined up to a set of parameters. Determining the values of these parameters completely identifies the model. It may happen that in the proposed class of models, the value of the parameter (vector) cannot be uniquely determined. Determining whether there exists a unique value of the parameter vector in this model class is called the identifiability problem. A system is considered globally identifiable if there is exactly one model corresponding to the system in the given class of models. The system is called locally identifiable if there exists a neighborhood in which it is identifiable (equivalent to the concept of global and local injectivity of a multidimensional, multivariable function). The global identifiability problem is considered hard.

Formally, one proposes a model $\mathcal{M}(\theta)$, $\theta \in \Omega \subseteq \mathbf{R}^n$ to be fit to the data. The model may contain little information. It may just specify that the system is linear, time-invariant and finite dimensional of a specified order n. The model may also be represented in various ways. In the case of a linear, time-invariant, finite dimensional system, the model could be described either using the transfer function, that is a matrix rational function of degree n, or using a state-space description, that is a tuple of matrices (A, B, C, D) of appropriate sizes. In the first case, the parameter vector θ may represent the unknown coefficients appearing in the transfer function. In the later case, θ may represent the unknown entries in the matrices A, B, C, D, taken in a certain canonical form. In this case, the model is called *unstructured.* If knowledge exists about 'realistic' values that the parameters θ can take, this may be included in Ω.

On the other hand, there may exist apriori knowledge about the system which needs to be included in the model. One may think for example of a system composed of smaller subsystems which are interconnected in a specified manner. In this case we speak about *structured systems.* Linear structured systems are typically represented by tuples $(A(\theta), B(\theta), C(\theta), D(\theta))$ with (highly) structured matrices. Very often, the values of the parameter vector θ have a certain interpretation. Especially in this case, the identifiability question, that is the existence of a unique corresponding value for θ, becomes extremely important.

Identifying an unstructured system means fitting the data to the chosen model while minimizing a certain criterion, for example minimizing the prediction error criterion. The algorithms used for optimizing the criterion find in general a local minimum. In general, there are no guarantees that this is a global minimum as well, or if it is, there may still exist several values of θ, leading to the same optimum but corresponding to different models in the same class $\mathcal{M}(\theta)$. It is desirable to have efficient algorithms which can locate multiple global minima or local minima with a close value of the criterion.

In the case of structured models, the problem becomes even more complicated. There, one needs to make sure that the mapping $\theta \in \Omega \mapsto \mathcal{M}(\theta)$ is injective, that is the *identifiability of the structure*. This is also our main object of study in this section.

The question concerning the identifiability of the structure can be approached in the following way (see, e.g., [70]). In case the model has a finite complete set of invariants, the problem related to global identifiability of the structure can be reduced to solving a system of equations. The existence of a unique solution is equivalent to global identifiability. For specific classes of models such equations are in fact polynomial, hence specific tools can be employed. The Buchberger algorithm used for solving a system of polynomial equations has, in the worst case, doubly exponential computational complexity. Moreover, as we shall argue later, the algorithm cannot be easily adapted for handling inequality constraints on the parameters. In practice, we are often interested in identifiability on restricted real domains.

We propose here to split the problem of global identifiability into two separate steps (problems): local identifiability plus a check on the existence of 'remote' indistinguishable values of the parameter. The second step aims to complete the analysis on the global identifiability.

We work under the assumption that the system allows a finite complete m-tuple of invariants and that the invariants are multivariable polynomial or rational functions in terms of the parameters. Moreover, when the feasible region in which we want to study identifiability is a strict subset of $\mathbf{R}^n$, we assume that it is in fact a semi-algebraic set, i.e. a set defined by polynomial equations and inequalities. Note that for linear time-invariant state-space models there exist finite complete sets of invariants (Markov parameters, coefficients of the transfer function) and that in many cases they are polynomial or rational functions.

Section 6.3.1 reviews results on the problem of local identifiability which plays an important role in our approach. Section 6.3.2 discusses briefly the existent approach to global identifiability based on Gröbner bases and Buchberger algorithm. Our approach is discussed in Section 6.3.3 and the section is concluded with a small example.

6.3.1 Local identifiability

Let $\theta = (\theta_1, \ldots, \theta_n)$ denote the parameter and $f = (f_1, \ldots, f_m)$ be a complete set of invariants. Assuming a nominal point $\hat{\theta}$ is given, one would like to know whether the function f is locally injective at $\hat{\theta}$. The basic idea is that the system is not locally identifiable whenever slight modifications on the parameters exist that leave the invariants of the system unchanged. A natural approach would therefore be to consider the Jacobian of the transformation which maps the unknown parameters θ to the invariants of the system. The following holds (see [19]).

Let the parameter feasibility domain Ω be an open set in $\mathbf{R}^n$ and $f : \Omega \to \mathbf{R}^m$ be a $\mathbf{C}^k$ map with $k \geq 1$. Then if the Jacobian $(\partial f(\theta)/\partial\theta)$ has constant rank r in a neighborhood of $\hat{\theta}$, f is locally injective at $\hat{\theta}$ if and only if $r = n$.

In a stochastic framework for linear systems, a different approach based on the Fisher information matrix can be used. Local unidentifiability implies the singularity of the Fisher information matrix. Conversely, for some parameterized classes of systems it has been shown that the asymptotic Fisher information matrix becomes singular if and only if the system is non-identifiable due to over-parameterization ([52], [51]).

In both cases, the analysis of the local identifiability reduces to the check on the singularity of a certain matrix. A nonsingular matrix corresponds to local identifiability. The converse is in general not true, the singularity of the matrix is however a signal that there may be local non-identifiability.

6.3.2 Global identifiability

Gröbner bases are powerful tools for solving systems of polynomial equations and obtaining *complete* solutions, i.e all solutions over the complex field. For identifiability they are used in the following way:

Given a finite complete set of invariants of a system $f_1, \ldots, f_m$ depending on $\theta = (\theta_1, \ldots, \theta_n)$, solve the system for $\theta \in \Omega (\subseteq \mathbf{R}^n)$

$$\begin{cases} f_1(\theta_1, \ldots, \theta_n) = f_1(\phi_1, \ldots, \phi_n) \\ \vdots \\ f_m(\theta_1, \ldots, \theta_n) = f_m(\phi_1, \ldots, \phi_n), \end{cases} \tag{6.22}$$

in the variables $\theta_1, \ldots, \theta_n$, where $\phi_1, \ldots, \phi_n$ are parameters. Polynomial systems are solved by the Buchberger algorithm which is known to return a complete set of solutions over $\mathbf{C}^n$, in case the system has a *finite* number of solutions in $\mathbf{C}^n$. This is the case we hope for. If the algorithm returns a unique solution $\theta_i = \phi_i$, $i = 1, \ldots, n$, then the function $(f_1, \ldots, f_s)$ is (globally) injective for generic values of the parameters. That is, except maybe for a thin subset, the parameterization is (globally) identifiable. The Buchberger algorithm is suited for symbolic calculations, hence it can handle the parameters ϕ.

There are two drawbacks of such a method. One is the theoretical complexity, the second is that the set of solutions is given over the whole $\mathbf{C}^n$. It is not clear how one would restrict the domain to (subsets) of $\mathbf{R}^n$. Even if a system is not identifiable on $\mathbf{C}^n$, it may be (globally) identifiable on the feasible set in $\mathbf{R}^n$, and this is not clear from the approach presented above. In the following section we look at algorithms that work over the reals and include constraints.

6.3.3 Algorithms for obtaining partial solutions of systems of polynomial equations and inequalities

In the particular problem of global injectivity we do not need to find all solutions of $f(\theta) = f(\phi)$ to decide on the injectivity. In fact we only need to know whether there are at least two distinct solutions. We cannot answer this question very easily. Let us fix for the moment $\hat{\theta} \in \Omega$. Our approach will be to characterize the set of solutions $S = \{\theta \in \Omega |\ f(\theta) = f(\hat{\theta})\}$. The solution set of a polynomial system of equations and inequalities consists of a finite number of connected components, also called *cells* (see Theorem 2.2.9). Such components may have dimension zero (one point) or may have higher dimension. Obviously, the global injectivity corresponds to the case when S consists of a single, zero-dimensional component. There exists local injectivity at a point $\hat{\theta}$ if the connected component which contains the point is zero-dimensional (i.e. the component is equal to $\{\hat{\theta}\}$).

Therefore the number of components and their dimension will give the answer to problems of global identifiability. First we discuss a way to detect all connected components. In fact we shall discuss below an algorithm that, given a system of polynomial equations (and inequalities), returns at least one point in every connected component. Hence, if there is more than one component, the algorithm will return more than one point. The dimension of a component will be studied by performing local analysis.

Although we do not find all the points in the components, we obtain sufficiently rich information to give an answer to the global injectivity problem (at a given point). To be more precise, let us consider the polynomials $f_1, \ldots, f_m, f_{m+1}, \ldots, f_s$ in $(\theta_1, \ldots, \theta_n)$, $n \leq m$, and the system

$$\begin{cases} f_i(\theta_1, \ldots, \theta_n) - f_i(\hat{\theta}_1, \ldots, \hat{\theta}_n) = 0, & i = 1, \ldots m \\ f_j(\theta_1, \ldots, \theta_n, \hat{\theta}_1, \ldots, \hat{\theta}_n) \geq 0, & j = m+1, \ldots s \end{cases} \tag{6.23}$$

where $\hat{\theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_n)$. Let S denote the solution set of (6.23) and $C_l, l = 1, \ldots, L$ denote the connected components of S, $S = \cup_{l=1}^{L} C_l$. In the following we indicate a method for computing at least one point in every connected component C_l of the solution set.

Suppose for the moment that no inequality constraints are imposed on θ, i.e. $m = s$. We form the polynomial $f(\theta) = \sum_{i=1}^{m} (f_i(\theta) - f_i(\hat{\theta}))^2$. Obviously $f(\theta) \geq 0$ for every value of θ and moreover $f(\theta) = 0$ if and only if θ is a (real) solution of (6.23). Hence the solution set of (6.23) is equal to the set $\{\theta \in \mathbf{R}^n | f(\theta) = 0\}$. An algorithm for finding a point in each connected component of the latter set is described in Section 3.2, Algorithm 3.2.18. There, the problem of finding the global minimum of a polynomial is discussed. The problem is reformulated as a generalized eigenvalue problem for a certain matrix of size at most $(d+1)^n$, where d is the total degree of f. The algorithm returns together with the minimal value, (at least) one point in each connected component of the set of minimal points, that is a point in each C_l. It is shown that at least one of the points found in each C_l has minimal Minkowski norm within C_l.

Remark that we can introduce inequalities of the form $f_j(\theta) \geq 0$ in our setup, introducing in the same time a new variable x_j, with $x_j^2 = f_j(\theta)$. The system (6.23) where the inequalities have been relaxed becomes

$$\begin{cases} f_i(\theta_1, \ldots, \theta_n) - f_i(\hat{\theta}_1, \ldots, \hat{\theta}_n) = 0, & i = 1, \ldots m \\ x_j^2 - f_j(\theta_1, \ldots, \theta_n, \hat{\theta}_1, \ldots, \hat{\theta}_n) = 0, & j = m+1, \ldots s \end{cases} \tag{6.24}$$

in variables (x, θ). Since the algorithm finds a *real* solution in every connected component of the solution set, the problem that we encountered with the Buchberger algorithm is avoided.

Notice that the problem of finding at least one point in every connected component of the solution set of a system of the type (6.23) is extensively studied in the literature. Consequently, various algorithms can be employed for solving it, see for example [57], [8]. An important criterion for comparison of algorithms should be their computational complexity.

We call a system *globally distinguishable at* $\hat{\theta}$ if it is locally identifiable at $\hat{\theta}$ and there are no remote values (in the feasible parameter domain) corresponding to the same invariants of the system. A system is globally identifiable if it is globally distinguishable at every point in the feasible set.

Remark 6.3.1 *In our algorithm we are making use of the nominal value $\hat{\theta}$. That is, the value that has been accepted for the parameter, after applying an identification procedure. The nominal value is than used as input in the algorithm in the search for other values of θ such that $f(\theta) = f(\hat{\theta})$, i.e. the models corresponding to $\hat{\theta}$ and θ are indistinguishable. Remark that such an approach does not answer completely the question on the injectivity of the function. In fact we would rather have the injectivity question answered on a neighborhood of $f(\hat{\theta})$.*

In the identification process, the nominal value is mostly found as a result of a minimization procedure. In principle we could apply the Algorithm 3.2.18 or Algorithm 4.1.8 directly for estimating the parameter and simultaneously finding different indistinguishable values of the parameter, in case they exist.

Remark 6.3.2 *1) Assume that the nominal value is locally identifiable at the nominal point $\hat{\theta}$ and we are interested in the global identifiability. We argue in the following that the method based on Algorithm3.2.18 gives some important extra information.*

Suppose the system is not globally identifiable. Therefore there will be more than one component. As we already mentioned, Algorithm 3.2.18 computes points (in the cells) of minimal Minkowski norm. Suppose now that we apply the algorithm

to the polynomial translated to the nominal point, i.e. $f(\theta_1, \ldots, \theta_n)$ *becomes* $f(\theta_1 - \hat{\theta}_1, \ldots, \theta_n - \hat{\theta}_n)$. *Then the algorithm, applied to* $f(\theta_1 - \hat{\theta}_1, \ldots, \theta_n - \hat{\theta}_n)$, *will return points (in different cells) situated at minimal Minkowski distance from the nominal value* $\hat{\theta}$. *In other words we know the Minkowski distance between* $\hat{\theta}$ *and all other cells. This will insure us that inside the Minkowski ball centered at the nominal value and radius equal to the distance returned by the algorithm (the distance to the closest cell), the system is identifiable. This may be useful if one can decide for example that the parameter feasible region is completely contained in the 'identifiable' region.*
2) Suppose now that the system is globally distinguishable at $\hat{\theta}$. *With some supplementary computational effort Algorithm 3.2.18 will return other local minima of the criterion function. Suppose we know the value of the smallest local minimum, say* α, *different from the global minimum which is 0. Hence* $\alpha > 0$. *Our claim is that* α *measures how close we are to having a remote indistinguishable parameter.*

The biggest problem with our approach at this point is the computational complexity of the algorithms involved. We work with matrices of size at most $(d+1)^n$ where n is the number of variables and d is the degree of the constructed polynomial f. Further improvements are required in order to make our approach more efficient in practice.

Example 6.3.3 *Let us consider the system described by:*

$$A = \begin{bmatrix} \theta_1^2 + \theta_2^2 & -\theta_1^2 - \theta_2^2 - 1 \\ \theta_1^2 + \theta_2^2 - 1 & -\theta_1^2 - \theta_2^2 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$c = \left[\; (\theta_1 - 1/2)\left(\theta_1^2 + \theta_2^2 - 1\right) \quad \theta_2 - 2 \; \right]$$

The transfer function is

$$\frac{1}{2}\,\frac{\left(\theta_1^2+\theta_2^2-1\right)\left((-1+2\,\theta_1)s+2\,\theta_1^3+2\,\theta_1\theta_2^2+2\,\theta_2-\theta_1^2-\theta_2^2-4\right)}{s^2-1} \tag{6.25}$$

The example was designed in such a way that for different values of the parameter vector (θ_1, θ_2), *we encounter different situations corresponding to global identifiability and local but not global identifiability. In the latter case, the components are either finite or infinite dimensional.*
Since a linear system is completely described by the coefficients of its transfer function, we obtain the following system of equations:

$$\left\{ \begin{array}{l} \left(\theta_1^2 + \theta_2^2 - 1\right)\left(2\,\theta_1 - 1\right) = \left(\hat{\theta}_1^2 + \hat{\theta}_2^2 - 1\right)\left(2\,\hat{\theta}_1 - 1\right) \\ \left(\theta_1^2 + \theta_2^2 - 1\right)\left(2\,\theta_1^3 + 2\,\theta_1\theta_2^2 - \theta_1^2 - \theta_2^2 + 2\,\theta_2 - 4\right) = \\ \left(\hat{\theta}_1^2 + \hat{\theta}_2^2 - 1\right)\left(2\,\hat{\theta}_1^3 + 2\,\hat{\theta}_1\hat{\theta}_2^2 - \hat{\theta}_1^2 - \hat{\theta}_2^2 + 2\,\hat{\theta}_2 - 4\right) \end{array} \right. \tag{6.26}$$

Remark 6.3.4 *For different values of the parameter vector* $(\hat{\theta}_1, \hat{\theta}_2)$, *we have the following situations:*

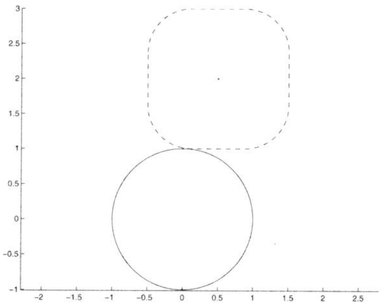
1. *For $(\hat{\theta}_1, \hat{\theta}_2) = (1,2)$ there is a single zero-dimensional component $\Longrightarrow$ the system is globally identifiable on $\mathbf{R}^n$ at $(1,2)$.*

2. *For $(\hat{\theta}_1, \hat{\theta}_2) = (-1,2)$ there are three zero-dimensional components: $(-1,2)$, $(0.3938, 7.5755)$, $(-1.7174, -0.8696)$ $\Longrightarrow$ the system is locally identifiable on $\mathbf{R}^n$ at $(-1,2)$.*

3. *For $(\hat{\theta}_1, \hat{\theta}_2) = (1/2,2)$ we have 1 zero-dimensional component $(1/2,2)$ and 1 component of higher dimension $\{(\theta_1,\theta_2) \in \mathbf{R}^2 \mid \theta_1^2 + \theta_2^2 - 1 = 0\}$ $\Longrightarrow$ the system is locally identifiable at $(1/2,2)$.*

We apply the algorithm to the 3-rd case: $(\hat{\theta}_1, \hat{\theta}_2) = (1/2,2)$ and obtain the points $(1/2,2)$, $(0.0004831158, 0.999999883)$, $(0,-1)$. The following hold:

- *The Jacobian is nonsingular at $(1/2,2)$, hence we conclude that the system is locally (but not globally) identifiable on $\mathbf{R}^n$.*
- *The Minkowski distance from $(1/2,2)$ to the other component is 1.000242729, hence the point $(1/2,2)$ is globally identifiable inside the Minkowski ball $B((1/2,2), 1.000242729)$.*



Note also that the system is minimal (controllable and observable) in the first two cases mentioned above, but it looses observability at $(1/2,2)$.

In the structured identifiability case, the loss of minimality does not imply the loss of identifiability from the coefficients of the transfer function. See for example

$$A = \begin{pmatrix} 1 & \theta \\ 1+\theta & 1+\theta \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

with the transfer function

$$Tf(s) = \frac{s - (1+\theta)}{s^2 - s(\theta+2) + (1-\theta^2)}.$$

Here, the system is not observable at $\theta = 0$ and not reachable at $\theta = -1$ but it is globally identifiable on $\mathbf{R}$.

6.3.4 Conclusions

We have proposed a new way of analyzing the global identifiability. In this approach, the global identifiability at the nominal value is established by studying the local identifiability at the given point, plus the existence of indistinguishable remote values. The existence of remote indistinguishable values requires the application of an algorithm. Its computational complexity is rather high and this restricts, at the moment, its applicability.

However, the approach allows the study of global identifiability on a restricted domain of $\mathbf{R}^n$ which is the interesting case in most examples.

The textbooks recommend to study the structural identifiability of a parameterization, before performing the actual identification. However, as it is well-known and also illustrated in the Example 6.3.3, global identifiability is not a property of the model class. It may differ for specific values of the parameter. Since structural identifiability is rather hard to establish and somewhat stronger than what one needs, an alternative approach is as follows. By applying an identification procedure, find a value $\theta_0 \in \Omega$, and then establish the existence or nonexistence of other values $\theta \in \Omega$ which are indistinguishable from θ_0. In case there are no indistinguishable θ's, we say that the system is globally identifiable at θ_0. The drawback of this approach stems from the uncertainty present in the data. Remark also that in most applications, the global identifiability issue is most of the time ignored.

Chapter 7

Conclusions and directions for further research

A fundamental question for the development of this thesis was the well-known optimal H_2 model reduction problem. Based on the paper [26], there was hope that methods of constructive algebra can be employed for answering such a question. The approach for the H_2 model reduction problem presented in this thesis is based on the well-known fact that the problem reduces to (global) constrained optimization of rational functions.

In this way, we started research in a different area of mathematics, that is global optimization for particular classes of functions. The layout of the thesis follows, relatively well and mostly by coincidence, the chronologic development of the thesis. We started with the study of optimization of polynomial functions. A first reason was that studying polynomials seemed to be a somewhat easier task than studying rational functions since polynomials have in some sense a better behavior. For example, polynomials in n variables are well-defined on the entire $\mathbf{R}^n$ and that is not true for rationals. Hence, in the case of a rational function, one needs to investigate the behavior of the rational function in a neighborhood of such a point where the function is not defined and what the limits are, if they exists, and what happens when both numerator and denominator cancel at the same point, etcetera. A second reason for starting with the study of polynomials was the hope that the extension to rationals could be realized relatively easy. And indeed, we managed with our approach to avoid dealing with the different situations and to give a unitary treatment for global optimization of rational functions.

In Chapter 3 we present an algorithm which *guarantees* finding the infimum of a polynomial. In case the infimum is attained the algorithm finds also at least one point where the global minimum is attained. The examples we studied suggested that, in case the set of global minimizers consists of several connected components, then the algorithm finds at least one point in every connected component. This hypothesis turned out to be true. Moreover, it led the investigations towards related questions and problems in real algebraic geometry. The 'introduction' to real algebraic geometry proved to be also very useful for extending the results obtained for optimization of polynomials to optimization of rational functions (namely Theorem 4.1.1) in Chapter 4. The results of the two above mentioned chapters were further developed in certain directions, to deal with symbolic instances of polynomials (i.e. families of polynomial functions), respectively with constrained optimization of rational functions.

Finally, the methods were applied to the optimal H_2 model reduction in Chapter 5. The application of the methods for global optimization is not straight forward, and that can be noticed for H_2 model reduction as well as for the problems discussed in Chapter 6. One important step in all cases is *choosing a parameterization* for the class of systems on which we want to optimize. Such choice can improve a lot the actual performance of the algorithms and there is still work to be done in this direction. In fact, we did not contribute to the theoretical development of this issue in the present thesis. In one particular case of the optimal H_2 model reduction problem, namely the SISO continuous-time case, a very 'suitable' parameterization was already known in the literature. We refer to the Schwarz-like canonical form, used here to parameterize minimal stable continuous-time systems, of a specific order. The canonical form is moreover output-normal. It turns out that with this parameterization, the criterion to be optimized (i.e. the rational functions) presents some interesting features which can be exploited in order to decrease the running time and the memory requirements of the algorithm. In the MIMO continuous-time case, although there is no theoretical problem in applying the method, the computations are extremely demanding even for small size problems. This may be related to the chosen parameterization, input-normal, designed for MIMO stable continuous-time systems of a given order. It remains an open question whether different parameterizations of this class of systems would lead to better computational time and memory requirements.

As expected, global (constrained) optimization of polynomial and rational functions finds more than a single application in system identification and system and control theory. Our purpose in Chapter 6 is to show this clearly by rewriting various problems in system identification and system and control theory as rational optimization problems. In Chapter 6 we put less emphasis on the actual calculations. In Sections 6.1 and 6.2, we encounter again the parameterization problem. For example, in Section 6.1, the approximant is a stable system of a given order with a stable inverse system. We did not investigate parameterizations for this particular class of systems. Also the parameterization problem did not constitute an important issue in our treatment of systems with uncertainties. We were more interested in proving that the approach is theoretically sound. Section 6.3 discusses parameterizations as well but from a different point of view. There we show how the well-known problem of global identifiability of a given structure can be approached with one of the algorithms developed in a previous chapter.

Algebraïsche optimalisatie met toepassingen in de systeemtheorie

Samenvatting

Een fundamentele vraagstelling voor de ontwikkeling van dit proefschrift was het bekende optimale H_2 modelreductieprobleem. Op grond van het artikel [26] was er de hoop dat methoden van de constructieve algebra toegepast kunnen worden ter beantwoording van een dergelijke vraagstelling. De aanpak van het H_2 modelreductieprobleem zoals die gepresenteerd wordt in dit proefschrift is gebaseerd op het bekende feit dat het probleem teruggebracht kan worden tot (globale) optimalisering van een rationale functie onder nevenvoorwaarden.

Globale optimalisatie van polynomiale en rationale functies wordt hier onderzocht. In Hoofdstuk 3 presenteren we een algoritme dat *garandeert* dat het infimum van het polynoom gevonden wordt. In geval het infimum aangenomen wordt vindt het algoritme tevens een punt waar het globale minimum bereikt wordt. In feite geeft het algoritme tenminste één punt in iedere samenhangende component van de verzameling van punten waarin het globale minimum wordt aangenomen. Deze resultaten worden verder ontwikkeld om te kunnen werken met polynomen met onbekende parameters (families van polynoom functies). Hoofdstuk 4 behandelt globale optimalisatie van rationale functies met of zonder nevenvoorwaarden.

De methoden ontwikkeld in Hoofdstuk 4 worden toegepast op het optimale H_2 modelreductieprobleem in Hoofdstuk 5. De toepassing van de methoden voor globale optimalisatie is niet triviaal, en dat kan opgemerkt worden voor zowel H_2 modelreductie als voor de problemen die behandeld worden in Hoofdstuk 6. Een belangrijke stap in al deze gevallen is het *kiezen van een parametrisatie* voor de klasse van systemen waarover we willen optimaliseren. Deze keuze kan de praktische werking van de algoritmen veel doen verbeteren.

Globale optimalisatie van polynomen en rationale functies onder nevenvoorwaarden kent vele toepassingen in systeemidentificatie en systeem- en regeltheorie. Het doel van Hoofdstuk 6 is om dit duidelijk aan te tonen door problemen in verschillende deelgebieden van systeemidentificatie en systeem- en regeltheorie te herschrijven als rationale of polynomiale optimalisatieproblemen.

References

[1] M. Athans. The matrix minimum principle. *Information and Control*, 11:592 – 606, 1968.

[2] L. Baratchart. Existence and generic properties of L_2 approximants for linear systems . *IMA Journal of Mathematical Control and Information*, 3:89– 101, 1986.

[3] L. Baratchart and M. Olivi. Critical points and error rank in best H_2 matrix rational approximation of fixed McMillan degree. *Constructive Approximation*, 14:273–300, 1998.

[4] L. Baratchart and F. Wielonski. Rational approximation in the real Hardy space H_2 and Stieltjes integrals: a uniqueness theorem. *Constructive Approximation*, 9:1–21, 1993.

[5] G.A. Bliss. *Algebraic functions*, volume XVI of *Colloquium Publications*. American Mathematical Society, 1933.

[6] J. Bochnak, M. Coste, and M.-F. Roy. *Real algebraic geometry.* Springer-Verlag, 2nd edition, 1998.

[7] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM Studies in Applied Mathematics. SIAM, 1994.

[8] B.F. Caviness and J.R. Johnson, editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Springer-Verlag, Wien, 1998.

[9] M.D. Choi, M. Knebusch, T.-Y. Lam, and B. Reznick. Transversal zeros and positive semidefinite forms. In *Real algebraic geometry and quadratic forms*, number 959 in Lecture Notes in Mathematics, pages 273–298, 1982.

[10] T.M. Cover and J.A. Thomas. *Elements of information theory.* John Wiley & Sons, New York, 1991.

[11] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties and algorithms.* Springer, New York, 2nd edition, 1997.

[12] D. Cox, J. Little, and D. O'Shea. *Using algebraic geometry.* Springer-Verlag, New York, 1998.

[13] J.H. Davenport, D. Siret, and E. Tournier. *Computer algebra.* Springer-Verlag, New York, 2000.

[14] L.M. Druzkowsi. *The Jacobian conjecture: survey of some results*, volume 31, pages 163–171. Banach Center Publications, 1995.

[15] G.E. Dullerud and F. Paganini. *A course in robust control theory: a convex approach.* Academic Press, New York, 1988.

[16] G.D. Forney. Minimal bases of rational vector spaces, with applications to multivariable linear systems. *SIAM Journal on Control*, 13(3):493–520, 1975.

[17] P. Fulcheri and M. Olivi. Matrix rational H_2 approximation: a gradient algorithm based on Schur analysis. *SIAM Journal on Control and Optimization*, 36(6):2103–2127, 1998.

[18] K.O. Geddes, S.R. Czapor, and G. Labahn. *Algorithms for computer algebra.* Kluwer Academic Publishers, 1992.

[19] K. Glover and J.C. Willems. Parametrizations of linear dynamical systems: canonical forms and identifiability . *IEEE Transactions on Automatic Control*, AC-19(6):640–646, 1974.

[20] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix polynomials.* Academic Press, 1982.

[21] L. Gonzalez-Vega, F. Rouillier, M.-F. Roy, and G. Trujillo. Symbolic recipes for real solutions. In A.M. Cohen, H. Cuypers, and H. Sterk, editors, *Some tapas of computer algebra*, pages 121–167, 1999.

[22] K. Hägglöf, P.O. Lindberg, and L. Stevenson. Computing global minima to polynomial optimization problems using Gröbner bases. *Journal of Global Optimization*, 7(2):115–125, 1995.

[23] B. Hanzon. Riemannian geometry on families of linear systems, the deterministic case. Technical Report 88-62 (ISSN 0922-5641), Faculty of Technical Mathematics and Informatics, Delft University of Technology, 1988.

[24] B. Hanzon. *Identifiability, recursive identification, and spaces of linear dynamical systems.* Number 63 and 64 in CWI Tracts. Centre for Mathematics and Computer Science, Amsterdam, 1989.

[25] B. Hanzon and D. Jibetean. Global minimization of a multivariate polynomial using matrix methods. Technical Report PNA-R0109, CWI, Amsterdam, 2001. To appear in the Journal of Global Optimization.

[26] B. Hanzon and J.M. Maciejowski. Constructive algebra methods for the L_2-problem for stable linear systems. *Automatica*, 32(12):1645–1657, 1996.

[27] B. Hanzon, J.M. Maciejowski, and C.T. Chou. Model reduction in H_2 using matrix solutions of polynomial equations. Technical Report CUED/F-INFENG/TR314, Cambridge University Engineering Department, 1998.

[28] B. Hanzon and R.J. Ober. Overlapping block-balanced canonical forms and parametrizations: the stable SISO case. *SIAM Journal on Control and Optimization*, 35(1):228–242, 1997.

[29] B. Hanzon and R.J. Ober. Overlapping block-balanced canonical forms for various classes of linear systems. *Linear Algebra and its Applications*, 281:171–225, 1998.

[30] B. Hanzon and R.L.M. Peeters. A Faddeev sequence method for solving Lyapunov and Sylvester equations . *Linear Algebra and its Applications*, 241-243:401–430, 1996.

[31] M. Hazewinkel and R.E. Kalman. On invariants, canonical forms and moduli for linear, constant, finite dimensional, dynamical systems. In G. Marchesini and S.K. Mitter, editors, *Proc. of the International Symposium on Mathematical System Theory*, pages 48–60, 1976.

[32] D. Henrion and J.B. Lasserre. GloptiPoly. Available at `http://www.laas.fr/~henrion/software/gloptipoly`.

[33] J.-B. Hiriart-Urruty. Conditions for global optimality. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*, pages 1–26. Kluwer Academic Publishers, Dordrecht, 1995.

[34] R. Horst and P.M. Pardalos, editors. *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht, 1995.

[35] X.-X. Huang, W.-Y. Yan, and K.L. Teo. H_2 near-optimal model reduction. *IEEE Transactions on Automatic Control*, 46(8):1279–1284, 2001.

[36] S. Ihara. *Information theory for continuous systems*. World Scientific, Singapore, 1993.

[37] D. Jibetean. Global optimization of rational multivariate functions. Technical Report PNA-R0120, CWI, Amsterdam, 2001.

[38] D. Jibetean and B. Hanzon. Global identifiability analysis using algorithms for detecting connected semi-algebraic components. In *Proc. of the 40th IEEE Conference on Decision and Control*, pages 3114–3115, 2001.

[39] D. Jibetean and B. Hanzon. Linear matrix inequalities for global optimization of rational functions and H_2 optimal model reduction. In D.S. Gilliam and J. Rosenthal, editors, *Proc. of the 15th International Symposium on MTNS*, 2002.

[40] D. Jibetean and J.H. van Schuppen. An algebraic method for system reduction of stationary Gaussian systems . To appear in *Proc. of the 13th IFAC Symposium on System Identification*, 2003.

[41] T.Y. Lam. An introduction to real algebra. *The Rocky Mountain Journal of Mathematics*, 14(4):767–814, 1984.

[42] J.B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[43] Y. Lecourtier and A. Raksanyi. The testing of structural properties through symbolic computation. In E. Walter, editor, *Identifiability of parametric models*, pages 75–84. Pergamon Press, Oxford, 1987.

[44] A. Lindquist and G. Picci. A geometric approach to modelling and estimation of linear stochastic systems. *Journal of Mathematical Systems, Estimation, and Control*, 1:241–333, 1991.

[45] A. Lindquist and G. Picci. Geometric methods for state space identification. In *Identification, adaption, learning*, pages 1–69. Springer, London, 1996.

[46] L. Ljung. *System identification. Theory for the user*. Prentice Hall International and System Sciences Series, 2-nd edition, 1999.

[47] L. Meier and D. Luenberger. Approximation of linear constant systems. *IEEE Transactions on Automatic Control*, AC-12(5):585–588, 1967.

[48] H.M. Möller and H.J. Stetter. Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems. *Numerische Mathematik*, 70:311–329, 1995.

[49] Y. Nesterov. Squared functional systems and optimization problems. In H. Frenk, K. Roos, and T. Terlaky, editors, *High Performance Optimization*, pages 405–439, Dordrecht, 2000. Kluwer Academic Publishers.

[50] P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. Available at `http://www.cds.caltech.edu/~pablo/pubs.htm`, 2001.

[51] R.L.M. Peeters. *System identification based on Riemannian geometry. Theory and algorithms.* Number 64 in Tinbergen Institute reasearch series. Tinbergen Institute, Amsterdam, 1993.

[52] R.L.M. Peeters and B. Hanzon. Symbolic computation of Fisher information matrices for parametrized state-space systems. *Automatica*, 35:1059–1071, 1999.

[53] R.L.M. Peeters, B. Hanzon, and D. Jibetean. Optimal H_2 model reduction in state-space: a case study, 2002. Submitted.

[54] S. Pinchuk. A counterexample to the strong real Jacobian conjecture. *Mathematische Zeitschrift*, 217(1):1–4, 1994.

[55] B. Reznick. Some concrete aspects of Hilbert's 17th problem, 1999. Preprint, available at `http://www.math.uiuc.edu/Reports/reznick/98-002.html`.

[56] F. Rouillier, M.-F. Roy, and M. Safey El Din. Finding at Least One Point in Each Connected Component of a Real Algebraic Set Defined by a Single Equation. *Journal of Complexity*, 16(4):716–750, 2000.

[57] M.F. Roy. Basic algorithms in real algebraic geometry and their complexity: from Sturm theorem to the existential theory of reals. In *Lectures on Real Geometry*, volume 23 of *Expositions in Mathematics*, pages 1–67. de Gruyter, Berlin, 1996.

[58] S. Schaible. Fractional programming. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*, pages 495–608. Kluwer Academic Publishers, Dordrecht, 1995.

[59] N. Shor. Class of global minimum bounds of polynomial functions. *Translated from Kibernetica*, 6:9–11, 1987.

[60] T. Söderström and P. Stoica. *System Identification.* Series in Systems and Control Engineering. Prentice Hall International, 1989.

[61] J.T. Spanos, M.H. Milman, and D.L. Mingori. A new algorithm for L_2 optimal model reduction. *Automatica*, 28(5):897–909, 1992.

[62] H.J. Stetter. Matrix eigenproblems are at the heart of polynomials systems solving. *SIGSAM Bulletin*, 30(4):22–25, 1996.

[63] A. Stoorvogel. The robust H_2 control problem: a worst-case design. *IEEE Transactions on Automatic Control*, 38(9):1358–1370, 1993.

[64] A.A. Stoorvogel and J.H. van Schuppen. Divergence rate approximation of a stationary Gaussian process by the output of a Gaussian system. In A. Beghi, L. Finesso, and G. Picci, editors, *Mathematical Theory of Networks and Systems*, pages 879–882, 1998.

[65] A.A. Stoorvogel and J.H. van Schuppen. System identification with information theoretic criteria. In S. Bittanti and G. Picci, editors, *Identification, adaptation, learning*, pages 289–338. Springer, Berlin, 1996.

[66] J. Sturm. SeDuMi 1.05. Available at `http://fewcal.kub.nl/sturm/software/sedumi.html`.

[67] A. Tarski. A decision method for elementary algebra and geometry. In B.F. Caviness and J.R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, pages 24–84, 1998.

[68] A.Y. Uteshev and T.M. Cherkasov. Polynomial optimization problem. *Doklady Mathematics*, 58:46–48, 1998.

[69] S.A. Vavasis. Complexity issues. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*, pages 27–41. Kluwer Academic Publishers, Dordrecht, 1995.

[70] E. Walter. *Identifiability of State-Space Models*. Springer-Verlag, 1982.

[71] D.A. Wilson. Model reduction for multivariable systems. *International Journal of Control*, 20:57–64, 1974.

[72] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.